# Generated Test Cases via llama-3.1-8b-instant LLM Model and Zero-shot Prompting Technique

---

**Data - 1**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

User Profiles

**Feature description -**

Each user can create a personalized profile that includes information such as their name, bio, profile picture, location, interests, and other relevant details. Profiles serve as a digital representation of individuals on the platform, allowing others to learn more about them.

**Userstory title -**

As a user, I want to create a personalized profile so that I can share my information with others on the platform.

**Acceptance criteria -**

['Users can add and edit their name, bio, profile picture, location, and interests.', 'The profile picture should support standard image formats.', 'Users can set privacy settings for each field.', 'The profile must be saved successfully.', 'Users should receive a confirmation upon saving their profile.']

**Generated Test Cases:**

Here are five detailed test cases for the User Profiles feature, each covering one or more of the specified acceptance criteria:

**Test Case 1: Verify that users can add and edit their name**

* **Test Case ID:** TP_001

* **Test Case Title:** Verify that users can add and edit their name

* **Test Case Description:** This test case ensures that users can successfully add and edit their name on their personalized profile.

* **Test Suite:** User Profiles

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to their account

 + User has a valid profile

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the account

 2. Click on the profile icon to access the profile page

 3. Click on the "Edit Profile" button

 4. Enter a valid name in the "Name" field

 5. Click on the "Save Changes" button

 6. Verify that the name is updated successfully

* **Postconditions:**

 + Profile name is updated successfully

 + User receives a confirmation message

* **Expected Result:** The user's name is updated successfully and saved to their profile.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that users can add and edit their bio**

* **Test Case ID:** TP_002

* **Test Case Title:** Verify that users can add and edit their bio

* **Test Case Description:** This test case ensures that users can successfully add and edit their bio on their personalized profile.

* **Test Suite:** User Profiles

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to their account

 + User has a valid profile

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the account

 2. Click on the profile icon to access the profile page

 3. Click on the "Edit Profile" button

 4. Enter a valid bio in the "Bio" field

 5. Click on the "Save Changes" button

 6. Verify that the bio is updated successfully

* **Postconditions:**

 + Profile bio is updated successfully

 + User receives a confirmation message

* **Expected Result:** The user's bio is updated successfully and saved to their profile.

* **Severity:** Minor

* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that users can add and edit their profile picture**

* **Test Case ID:** TP_003
* **Test Case Title:** Verify that users can add and edit their profile picture
* **Test Case Description:** This test case ensures that users can successfully add and edit their profile picture on their personalized profile.
* **Test Suite:** User Profiles
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to their account
 + User has a valid profile
* **Test Data:** Valid image file (e.g., JPEG, PNG)
* **Test Steps:**
 1. Log in to the account
 2. Click on the profile icon to access the profile page
 3. Click on the "Edit Profile" button
 4. Select a valid image file for the profile picture
 5. Click on the "Save Changes" button
 6. Verify that the profile picture is updated successfully
* **Postconditions:**
 + Profile picture is updated successfully
 + User receives a confirmation message
* **Expected Result:** The user's profile picture is updated successfully and saved to their profile.

* **Severity:** Minor
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that users can set privacy settings for each field**

* **Test Case ID:** TP_004
* **Test Case Title:** Verify that users can set privacy settings for each field
* **Test Case Description:** This test case ensures that users can successfully set privacy settings for each field on their personalized profile.
* **Test Suite:** User Profiles
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to their account
 + User has a valid profile
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the account
 2. Click on the profile icon to access the profile page
 3. Click on the "Edit Profile" button
 4. Select a privacy setting for each field (e.g., public, private, friends only)
 5. Click on the "Save Changes" button
 6. Verify that the privacy settings are updated successfully
* **Postconditions:**
 + Profile fields have the correct privacy settings
 + User receives a confirmation message

* **Expected Result:** The user's privacy settings for each field are updated successfully and saved to their profile.
* **Severity:** Minor
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that the profile must be saved successfully**

* **Test Case ID:** TP_005
* **Test Case Title:** Verify that the profile must be saved successfully
* **Test Case Description:** This test case ensures that the profile is saved successfully after editing.
* **Test Suite:** User Profiles
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to their account
 + User has a valid profile
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the account
 2. Click on the profile icon to access the profile page
 3. Click on the "Edit Profile" button
 4. Enter a valid name, bio, and profile picture
 5. Click on the "Save Changes" button
 6. Verify that the profile is saved successfully
* **Postconditions:**
 + Profile is updated successfully

+ User receives a confirmation message

* **Expected Result:** The profile is saved successfully and updated with the new information.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**------------------------------------------------------------------------------------------------**

**Data - 2**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

News Feed

**Feature description -**

The news feed is a central feature of the social media platform where users can view a curated stream of content from accounts they follow or are connected to. It typically displays posts, updates, photos, videos, and other media shared by friends, family, brands, and other users in chronological or algorithmically determined order.

**Userstory title -**

As a user, I want to see a curated news feed so that I can stay updated with content from my connections.

**Acceptance criteria -**

['The news feed displays posts from followed accounts.', 'Content is shown in chronological or algorithmic order.', 'Users can interact with posts (like, comment, share).', 'The feed updates in real-time.', 'Users can filter content based on type or topic.']

**Generated Test Cases:**

Based on the provided Test Case Creation Template and the user story requirements, I've created five detailed test cases to ensure comprehensive coverage of the News Feed feature. Each test case covers one or more acceptance criteria and includes edge cases and negative scenarios where applicable.

**Test Case 1: Verify that the News Feed displays posts from followed accounts**

* **Test Case ID:** NF-001
* **Test Case Title:** Verify that the News Feed displays posts from followed accounts.
* **Test Case Description:** This test case ensures that the News Feed displays posts from accounts the user is following. It will also verify that posts from unfollowed accounts are not visible.
* **Test Suite:** News Feed Feature
* **Test Priority:** High
* **Preconditions:**
 + The user has a verified account with at least 5 followed accounts.
 + The followed accounts have published posts.
* **Test Data:** User account credentials, followed account names, and post content.
* **Test Steps:**
 1. Log in to the application with the user account.
 2. Navigate to the News Feed page.
 3. Verify that posts from followed accounts appear in the News Feed.
 4. Search for and interact with a specific post from a followed account (like, comment, or share).
* **Postconditions:**

 + The post from the followed account remains in the News Feed.

 + Posts from unfollowed accounts are not visible in the News Feed.

* **Expected Result:** The News Feed displays posts from followed accounts, and the user can interact with these posts.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that content is shown in chronological or algorithmic order**

* **Test Case ID:** NF-002

* **Test Case Title:** Verify that content is shown in chronological or algorithmic order.

* **Test Case Description:** This test case ensures that posts in the News Feed are displayed in the correct order, either chronologically or based on the algorithm.

* **Test Suite:** News Feed Feature

* **Test Priority:** Medium

* **Preconditions:**

 + The user has a verified account with at least 10 published posts.

 + The posts have accurate timestamps.

* **Test Data:** User account credentials, post content, and timestamp information.

* **Test Steps:**

 1. Log in to the application with the user account.

 2. Navigate to the News Feed page.

 3. Observe the order of the posts in the News Feed.

 4. Use the timestamp information to verify that the posts are displayed in the correct chronological order.

* **Postconditions:**

 + The posts in the News Feed are displayed in the correct order.

* **Expected Result:** Content in the News Feed is displayed in chronological or algorithmic order.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can interact with posts (like, comment, share)**

* **Test Case ID:** NF-003

* **Test Case Title:** Verify that users can interact with posts (like, comment, share).

* **Test Case Description:** This test case ensures that users can like, comment, or share posts in the News Feed.

* **Test Suite:** News Feed Feature

* **Test Priority:** High

* **Preconditions:**

 + The user has a verified account with at least 5 followed accounts.

 + The followed accounts have published posts.

* **Test Data:** User account credentials, followed account names, and post content.

* **Test Steps:**

 1. Log in to the application with the user account.

 2. Navigate to the News Feed page.

 3. Interact with a post from a followed account (like, comment, or share).

 4. Verify that the interaction is successful and the post updates accordingly.

* **Postconditions:**

 + The post is updated with the user's interaction.

* **Expected Result:** Users can interact with posts in the News Feed.

* **Severity:** Blocker
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that the feed updates in real-time**

* **Test Case ID:** NF-004
* **Test Case Title:** Verify that the feed updates in real-time.
* **Test Case Description:** This test case ensures that the News Feed updates in real-time as new posts are published.
* **Test Suite:** News Feed Feature
* **Test Priority:** High
* **Preconditions:**
 + The user has a verified account with at least 5 followed accounts.
 + The followed accounts have published new posts.
* **Test Data:** User account credentials, followed account names, and post content.
* **Test Steps:**
 1. Log in to the application with the user account.
 2. Navigate to the News Feed page.
 3. Simulate new post publication from followed accounts (e.g., using a third-party tool).
 4. Observe the News Feed for updates.
* **Postconditions:**
 + New posts from followed accounts appear in the News Feed.
* **Expected Result:** The News Feed updates in real-time as new posts are published.
* **Severity:** Blocker
* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that users can filter content based on type or topic**

* **Test Case ID:** NF-005

* **Test Case Title:** Verify that users can filter content based on type or topic.

* **Test Case Description:** This test case ensures that users can filter content in the News Feed based on type (e.g., image, video) or topic.

* **Test Suite:** News Feed Feature

* **Test Priority:** Medium

* **Preconditions:**

 + The user has a verified account with at least 10 published posts.

 + The posts cover various types and topics.

* **Test Data:** User account credentials, post content, and tagging information.

* **Test Steps:**

 1. Log in to the application with the user account.

 2. Navigate to the News Feed page.

 3. Apply various filters (type, topic) to the News Feed.

 4. Verify that the filtered content is accurate and relevant.

* **Postconditions:**

 + The filtered content is accurate and relevant.

* **Expected Result:** Users can filter content in the News Feed based on type or topic.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These five test cases cover all the acceptance criteria for the News Feed feature, including displaying posts from followed accounts, content order, user interaction, real-time updates, and content filtering.

---

**Data - 3**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Messaging

**Feature description -**

Messaging enables direct communication between users through private messages or group chats. Users can send text messages, multimedia files, emojis, stickers, and other forms of communication in real-time or asynchronously, fostering private conversations and connections.

**Userstory title -**

As a user, I want to send private messages so that I can communicate directly with my friends.

**Acceptance criteria -**

['Users can send text messages, images, and videos.', 'Messages can be sent in real-time.', 'Users can start private conversations or group chats.', 'Messages should show read receipts.', 'Users should be notified of new messages.']

**Generated Test Cases:**

Here are the 5 test cases that cover the acceptance criteria for the Messaging feature:

**Test Case 1: Verify that users can send text messages**

Test Case ID: MSC-001

Test Case Title: Verify that users can send text messages

Test Case Description: In the Messaging feature, verify that users can send text messages to their friends. This includes checking that the text messages are delivered correctly and that the recipient receives the message.

Test Suite: Messaging Feature Test Suite

Test Priority: High

Preconditions:

* User is logged in to the social media platform
* User has a friend connected to their profile

Test Data: No test data needed

Test Steps:

1. Access the Messaging feature from the profile menu
2. Select a friend to send a message to
3. Type a text message in the message composition box
4. Click the send button to send the message
5. Verify that the message is delivered to the recipient

Postconditions:

* Verify that the message is displayed in the recipient's inbox
* Verify that the sender's profile shows the sent message

Expected Result: The text message is sent successfully to the recipient.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that messages can be sent with images and videos**

Test Case ID: MSC-002

Test Case Title: Verify that messages can be sent with images and videos

Test Case Description: In the Messaging feature, verify that users can send images and videos to their friends. This includes checking that the images and videos are delivered correctly and that the recipient receives the message.

Test Suite: Messaging Feature Test Suite

Test Priority: Medium

Preconditions:

* User is logged in to the social media platform
* User has a friend connected to their profile
* User has an image or video to be sent

Test Data:

* Test image/vide file

Test Steps:

1. Access the Messaging feature from the profile menu
2. Select a friend to send a message to
3. Attach the image/video to the message composition box
4. Click the send button to send the message
5. Verify that the message is delivered to the recipient

Postconditions:

* Verify that the image/video is displayed in the recipient's inbox
* Verify that the sender's profile shows the sent message

Expected Result: The image/video is sent successfully to the recipient.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that messages can be sent in real-time**

Test Case ID: MSC-003

Test Case Title: Verify that messages can be sent in real-time

Test Case Description: In the Messaging feature, verify that messages are sent in real-time. This includes checking that the recipient receives the message immediately after it is sent.

Test Suite: Messaging Feature Test Suite

Test Priority: High

Preconditions:

* User is logged in to the social media platform
* User has a friend connected to their profile

Test Data: No test data needed

Test Steps:

1. Access the Messaging feature from the profile menu
2. Select a friend to send a message to
3. Type a text message in the message composition box
4. Click the send button to send the message
5. Immediately after sending the message, verify that the recipient receives the message

Postconditions:

* Verify that the message is displayed in the recipient's inbox
* Verify that the sender's profile shows the sent message

Expected Result: The message is sent in real-time to the recipient.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that users can start private conversations or group chats**

Test Case ID: MSC-004

Test Case Title: Verify that users can start private conversations or group chats

Test Case Description: In the Messaging feature, verify that users can start private conversations or group chats. This includes checking that the recipient receives the invitation to join the conversation.

Test Suite: Messaging Feature Test Suite

Test Priority: Medium

Preconditions:

* User is logged in to the social media platform
* User has friends connected to their profile

Test Data: No test data needed

Test Steps:

1. Access the Messaging feature from the profile menu
2. Select a friend to start a private conversation with
3. Click the "Start Conversation" button
4. Verify that the recipient receives the invitation to join the conversation
5. Verify that the conversation is displayed in the recipient's inbox

Postconditions:

* Verify that the conversation is displayed in the sender's profile
* Verify that the recipient can join the conversation

Expected Result: The user can start a private conversation or group chat with their friends.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that messages show read receipts**

Test Case ID: MSC-005

Test Case Title: Verify that messages show read receipts

Test Case Description: In the Messaging feature, verify that messages show read receipts. This includes checking that the sender sees a read receipt when the recipient reads the message.

Test Suite: Messaging Feature Test Suite

Test Priority: High

Preconditions:

* User is logged in to the social media platform
* User has a friend connected to their profile

Test Data: No test data needed

Test Steps:

1. Access the Messaging feature from the profile menu
2. Send a message to a friend
3. Verify that the message is delivered to the recipient
4. Verify that the recipient reads the message
5. Verify that the sender sees a read receipt

Postconditions:

* Verify that the recipient's profile shows the read message
* Verify that the sender's profile shows the read receipt

Expected Result: The message shows a read receipt when the recipient reads the message.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

---

**Data - 4**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Content Sharing

**Feature description -**

Users can share various types of content with their followers or the public, including text posts, photos, videos, links, articles, and documents. Content sharing fosters engagement, creativity, and self-expression among users, allowing them to share their thoughts, experiences, and interests with others.

**Userstory title -**

As a user, I want to share posts so that I can express my thoughts and interests with others.

**Acceptance criteria -**

['Users can create and share text posts, photos, and videos.', 'Content can be shared with followers or publicly.', 'Users can edit or delete their posts.', 'Posts should support hashtags and tagging.', 'Users should receive feedback through likes, comments, and shares.']

**Generated Test Cases:**

Here are five detailed test cases for the feature "Content Sharing" from the user story "As a user, I want to share posts so that I can express my thoughts and interests with others."

**Test Case 1: Verify that users can create and share text posts**

* **Test Case ID:** SCC001
* **Test Case Title:** Verify that users can create and share text posts
* **Test Case Description:** This test case aims to ensure that users can compose and publish text posts on the social media platform. The test will cover the creation of a text post, the addition of a title and content, and the successful sharing of the post with followers or publicly.
* **Test Suite:** Content Sharing Suite
* **Test Priority:** High
* **Preconditions:**
  * User has a valid account on the social media platform
  * User has Internet connection
* **Test Data:** No test data needed
* **Test Steps:**
  1. Log in to the social media platform using a valid username and password.
  2. Click on the "Compose Post" button.
  3. Enter a title and content for the post.
  4. Add a hashtag to the post.
  5. Click the "Share" button to share the post with followers or publicly.
* **Postconditions:** The post should be visible on the user's feed and should be visible to followers or the public, depending on the sharing option chosen.
* **Expected Result:** The user is able to create and share a text post successfully.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

* **Edge Cases:**

  + Test that the post is visible when shared with followers only.

  + Test that the post is visible when shared publicly.

  + Test that the post is removed when deleted by the user.

**Test Case 2: Verify that users can share photos and videos**

* **Test Case ID:** SCC002

* **Test Case Title:** Verify that users can share photos and videos

* **Test Case Description:** This test case aims to ensure that users can upload and share photos and videos on the social media platform. The test will cover the uploading of a photo or video, the addition of a caption, and the successful sharing of the media item with followers or publicly.

* **Test Suite:** Content Sharing Suite

* **Test Priority:** High

* **Preconditions:**

  * User has a valid account on the social media platform

  * User has Internet connection

* **Test Data:** No test data needed

* **Test Steps:**

  1. Log in to the social media platform using a valid username and password.

  2. Click on the "Compose Post" button.

  3. Click on the "Add Media" button.

  4. Select a photo or video from the user's device.

  5. Add a caption to the media item.

  6. Click the "Share" button to share the media item with followers or publicly.

* **Postconditions:** The media item should be visible on the user's feed and should be visible to followers or the public, depending on the sharing option chosen.

* **Expected Result:** The user is able to share a photo or video successfully.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

* **Edge Cases:**

  + Test that the media item is visible when shared with followers only.

  + Test that the media item is visible when shared publicly.

  + Test that the media item is removed when deleted by the user.

**Test Case 3: Verify that users can edit or delete their posts**

* **Test Case ID:** SCC003

* **Test Case Title:** Verify that users can edit or delete their posts

* **Test Case Description:** This test case aims to ensure that users can edit or delete their posts on the social media platform. The test will cover the editing and deletion of a post.

* **Test Suite:** Content Sharing Suite

* **Test Priority:** High

* **Preconditions:**

  * User has created a valid post on the social media platform.

  * User has Internet connection.

* **Test Data:** No test data needed

* **Test Steps:**

  1. Log in to the social media platform using a valid username and password.
  2. Go to the post that needs editing or deletion.
  3. Click on the "Edit" button to edit the post.
  4. Make changes to the post.
  5. Click the "Save" button to save the changes.
  6. To delete the post, click the "Delete" button.
* **Postconditions:** The post should be updated with the new content or should be removed.
* **Expected Result:** The user is able to edit or delete their post successfully.
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive
* **Edge Cases:**
  + Test that the post is updated successfully when edited.
  + Test that the post is removed when deleted.

**Test Case 4: Verify that content can be shared with followers or publicly**

* **Test Case ID:** SCC004
* **Test Case Title:** Verify that content can be shared with followers or publicly
* **Test Case Description:** This test case aims to ensure that users can choose to share content with their followers or with the public. The test will cover the sharing option chosen by the user.
* **Test Suite:** Content Sharing Suite
* **Test Priority:** High
* **Preconditions:**
  * User has created a valid post on the social media platform.
  * User has Internet connection.

* **Test Data:** No test data needed

* **Test Steps:**

  1. Log in to the social media platform using a valid username and password.

  2. Go to the post that needs sharing.

  3. Choose the sharing option: "Share with followers" or "Share publicly".

  4. Click the "Share" button.

* **Postconditions:** The post should be visible on the user's feed and should be visible to followers or the public, depending on the sharing option chosen.

* **Expected Result:** The user is able to share the post with followers or publicly successfully.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

* **Edge Cases:**

  + Test that the post is visible when shared with followers only.

  + Test that the post is visible when shared publicly.

**Test Case 5: Verify that users receive feedback through likes, comments, and shares**

* **Test Case ID:** SCC005

* **Test Case Title:** Verify that users receive feedback through likes, comments, and shares

* **Test Case Description:** This test case aims to ensure that users receive feedback in the form of likes, comments, and shares when they share content on the social media platform. The test will cover the addition of likes, comments, and shares.

* **Test Suite:** Content Sharing Suite

* **Test Priority:** High

* **Preconditions:**

  * User has created a valid post on the social media platform.
  * User has Internet connection.
* **Test Data:** No test data needed
* **Test Steps:**
  1. Log in to the social media platform using a valid username and password.
  2. Go to a post that needs feedback.
  3. Click the "Like" button to add a like.
  4. Click the "Comment" button to add a comment.
  5. Click the "Share" button to add a share.
* **Postconditions:** The post should have likes, comments, and shares.
* **Expected Result:** The user is able to receive feedback in the form of likes, comments, and shares successfully.
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive
* **Edge Cases:**
  + Test that the post has likes, comments, and shares.
  + Test that the user can see the likes, comments, and shares on their post.

---

**Data - 5**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Privacy Settings

**Feature description -**

Privacy settings allow users to control who can view their profile, posts, and other activity on the platform. Users can choose between public, private, or restricted privacy settings, determining who can see their content, send them friend requests, or interact with them.

**Userstory title -**

As a user, I want to set privacy settings so that I can control who sees my information.

**Acceptance criteria -**

['Users can set their profile to public, private, or restricted.', 'Privacy settings can be changed at any time.', 'Users can control visibility of individual posts.', 'The system should respect the privacy settings immediately.', 'Users should be notified of any changes in privacy settings.']

**Generated Test Cases:**

Here are five detailed test cases for the Privacy Settings feature, each covering one or more of the acceptance criteria:

**Test Case 1: Verify that users can set their profile to public, private, or restricted**

* **Test Case ID:** PS-001

* **Test Case Title:** Verify that users can set their profile to public, private, or restricted

* **Test Suite:** Privacy Settings

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the platform

 + User has a profile created

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as a user

 2. Navigate to the profile settings page

 3. Select the desired privacy setting (public, private, or restricted) from the dropdown menu

 4. Verify that the selected privacy setting is applied to the user's profile

* **Postconditions:**

 + User's profile is updated with the selected privacy setting

 + User is notified of the change in privacy setting

* **Expected Result:** The user's profile is successfully updated with the selected privacy setting.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that privacy settings can be changed at any time**

* **Test Case ID:** PS-002

* **Test Case Title:** Verify that privacy settings can be changed at any time

* **Test Suite:** Privacy Settings

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in to the platform

 + User has a profile created

 + User has already set their privacy settings to a specific value

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as a user

 2. Navigate to the profile settings page

 3. Select a different privacy setting from the dropdown menu

 4. Verify that the new privacy setting is applied to the user's profile

* **Postconditions:**

 + User's profile is updated with the new privacy setting

 + User is notified of the change in privacy setting

* **Expected Result:** The user's profile is successfully updated with the new privacy setting.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can control visibility of individual posts**

* **Test Case ID:** PS-003

* **Test Case Title:** Verify that users can control visibility of individual posts

* **Test Suite:** Privacy Settings

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in to the platform

 + User has a profile created

 + User has already posted content

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user

2. Navigate to the post settings page

3. Select a specific post to edit

4. Choose a different visibility setting for the post (e.g., public, private, or restricted)

5. Verify that the post is updated with the new visibility setting

* **Postconditions:**

 + Post is updated with the new visibility setting

 + User is notified of the change in post visibility

* **Expected Result:** The post is successfully updated with the new visibility setting.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that the system respects the privacy settings immediately**

* **Test Case ID:** PS-004

* **Test Case Title:** Verify that the system respects the privacy settings immediately

* **Test Suite:** Privacy Settings

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the platform

 + User has a profile created

 + User has already set their privacy settings to a specific value

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the platform as a user

 2. Set the user's profile to a specific privacy setting (e.g., public, private, or restricted)

 3. Immediately attempt to view the user's profile or post content

 4. Verify that the system respects the user's privacy settings and does not display the content

* **Postconditions:**

 + System respects the user's privacy settings

 + User is notified of the restriction

* **Expected Result:** The system successfully respects the user's privacy settings.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that users are notified of any changes in privacy settings**

* **Test Case ID:** PS-005

* **Test Case Title:** Verify that users are notified of any changes in privacy settings

* **Test Suite:** Privacy Settings

* **Test Priority:** Low

* **Preconditions:**

 + User is logged in to the platform

 + User has a profile created

 + User has already set their privacy settings to a specific value

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as a user

 2. Update the user's privacy settings to a different value (e.g., from public to private)

 3. Verify that the user receives a notification about the change in privacy settings

* **Postconditions:**

 + User is notified of the change in privacy setting

* **Expected Result:** The user receives a notification about the change in privacy settings.

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases cover all the specified acceptance criteria, including setting profile to public, private, or restricted, changing privacy settings at any time, controlling visibility of individual posts, respecting the system's privacy settings, and notifying users of changes in privacy settings. Each test case follows the provided template and is designed to ensure that the Privacy Settings feature functions as intended.

---

**Data - 6**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Notifications

**Feature description -**

Notifications alert users about new activity on the platform, such as likes, comments, mentions, friend requests, and event invitations. Notifications help users stay updated and engaged, prompting them to interact with the platform and respond to interactions from others.

**Userstory title -**

As a user, I want to receive notifications so that I am aware of new activity related to my profile.

**Acceptance criteria -**

['Users receive notifications for likes, comments, and shares.', 'Users are notified of new friend requests and mentions.', 'Notifications are displayed in real-time.', 'Users can customize notification preferences.', 'Notifications include a link to the relevant activity.']

**Generated Test Cases:**

**Test Case Creation Document:**

Below is a compilation of test cases for the notifications feature. This package presents in-depth exam items comprising Positive & Ad-Hoc content examining.

***

## Notification related: get direct sends verify.

**(Integration)/ Regression./ performance testcase ids.

2Notification from site at posting results creation:

***Description verfitons showing will contsinucuser****lmessage(serveduymakerio ( **UseRsmessage whrefmessagecanadd by likeed *** from ) it erionigged using Testd er asUser ftem rusers end "**.(contonhild )

!Iz(** t forrecontempt).

TEST + post =set newsm the conifnorm iare ifio pro gram entg int res send ca see edit**

Tests Ile not add y sent /sm witha form tr ass dis mes trand fol der (** = S R1w ex ceclled)*se res (* pr apsa withresse ca show: em ro comm).

Steps ? res pr ce v str . from per be** ro.

. form en ing he by ass all ret ass we ser ans via rece tem par met( spec ver detal int repore v via wi

Send this wh ter so far out pos rep ment ot now rece vide ret sec del any es /detall he edit refri upde ret wel int relio po di set ot mailt mess ans.

er sp enge ev mes can.

ex cerion do estin cal ce reg li ing) P updes) comp refer sp te cl mmes sen spe /**reg actim).

wiz ev comm th the thil reco,,

newb serv resp :lanc wor it end en no sign ***=T int im up* ro do rep evn fol da pl resp any the out le ap impl ? es = b :yacert det il

and* ar ses nowr (** esentil tim add par ple num vall= via si we (** des with g ? we?;o co elempn any te me sent ar new rep rem pro relai edit reco sig comp ar suc te ele poste ? [ s enso [vion li rece posl sigel ind/ test fi= -ve tim bever spec so app can per ,pe un cont ce ap maig see vide f und

*(can sim also wi si he cerf add me ar ,reg pos.. form sub b par

he cal edit num usew act.

not ce do g selu..elte( mes tec.

steps po o sub [aswer suc serv or*es ind ch ec wel cel with des en ..spec at coni id ter by is .

tr im ucan ***(** ret .

si su now em b te

prev u det if ass num numo if be rep tec en pro ci comm ma ent with tec) estm resp ec set man prac en sim su sent pr pl ev fi v b sig est cet we( follo mess thel g cer pl we newc cont ci exp lic f ment if sendo rece:per cet pre edit ec.

ter cesp id po all = ed mes) rec sub see act

ce cel app *** ed si ver now :es mes si per ? er,now p de sp cel ono me le em in stis res de we tim spec ne w

suc vide re mes ele ce spec

a : el vide sig

cel tim cer est sel

? us w spec cel . de

? eli [m im c posti ce re w app ne w ec pe tec des, suc le :e sig se.

*** reg ?s.

###

. for est.

use n **:** ar ter ne st pe en ev me **/ ** / cont tes

test no send / vide * pos ** spec p te ci (.

s ce ne ne) do as

**i?*

### an

pe sub **: post er, vide sig  new sp ** pe

, ce see po rec edit :vid see sub f is spe post * rece sub

?- ?

spec sub pos cel

spec cel spec

?sp ? ce post tes.

###

### post vide si tes .

*ex  ***pe

# : vide , res rece

see ? cel , vide ** new  cel cel rec . i *** pos sent v ce  se e spec cel ap spe . si

.  .

** and ce e post ***  .

*** vide

expec

*** estin.

***

*** vide *** new

cel

c..- spec ?*** ?  vid ..*** p**e   new

vide cel post

*** vide spe

*** vide se.

?? spec ? *** vide spec post **?. post. vide sel vide

?vid spec sese

? vid cesel vide vide vide

?? vide cel vide s vide spec si vide vide spec post vide cel vide cel vide spe

?*? vid vide f post cel post vide vide vide vide v

post vide cel ? vid vide ? vid cel post spec ce

vide cel spec vide vide spec vide vide post cel ? *** vide vide ? no

vide vide *** spec vide cel post vide .

spec vide vide vide vide post vide vide. ? *** .

spec cel vide ?

.?:** video cel s Vide cel ?

cel ? post vide post vide vide cel spec vide vide spec vide vide vide post cel ? cel vide cel ? ???? spec vide cel

? post vide cel post vide

?*** ? spec ** ?

spe ? cel ? post vide cel.?. post vide vide vide vide cel spec cel vide ? post cel post vide spec vide vide? vide

ce ? vide cel vide cel

? cel vide cel video post ?

vide vide c** cel post cell post ? cel cel vide spec cel vide vide ** ? cel video? vide cel

?? post vide post vide vide vide cel spec ? cel cel vide spec vide post vide cel ? post vide? ? cel cel cel vide cel

video cel ? cel vide spec cel post vide cel cel vide

video cel post cel cel

cell vide ? cel video cel post cel vide cel post cel video cel

cel post ce.. . ??? cel post .

?

.? **? . ** cel vid? cel c ?

*** vide cel post vide cel vide cel vide spec

post vide cel cel spec vide

cel vide post vide cel cel vide spec post

?spec vide cel vide post cel post cel post vide cel cel? vide

cel post vide cel spec cel post ?

? post cel post

cel post cel vide? cel post vide .

spec vide cel post vide cel vide vidspec post vide cel cel post? post vide cel spec cel? video cel pe cel** pe

vide cel pe. * cel * cel vide spec pe ce

cel pe cel ve pe post cel cel *

* vide cel spec pe cel vide .

pe cel spec cel post vide

post pe vide pe cel post cel cel vide cel spe ve vid spec cel post cel vide ? cel vide vid spec cel post celel cel cel vide video cel cel cel post cel cel cel post cel cel vide video cel cel post.

cel

pe cel vide vide vide cel cel cel vide cel post cel vide video cel

post

cel cel vide cel post video vide cel post cel cel cel post cel cel spec vide cel cel post vide

cel spec :

spec ce cel vide espec spec video cel cel vide pe? sp

cel pe vide spec? vide cel pe ? cel vide cel cel vide pe spec cel post vide pe cel cel spec ce vide post pe vide c

spec video cel

cel pe cel spec cel cel vide vide cel post pe vide cel cel cel vide cel vide cel cel spec pe vide post pe cel? pe pe

post cel cel cel spec

?*? vide ce? spec pe video vide pe cel *.* pe? *?

? cel post pe cel vide post pe ce post ?( ce * cel  cel cel video pe video spec video cep cel * * video pe pe? cel

cel? post *video cel cep pe cel *.

celerye cep cel video pep vide spec cep video cep pe vide pel spec cel cel pe *? *ce spec video pe cel cel cep

* * * p video pep video ce* cel cel ?

? spec video pe post pep pe ? cel spe c video cep ? cel ? vide pep pep vid pep cep cel * vide * ce..

ce speci video cel post pep ce spec cep cel pep cep ? * * * * vide * ( ce spec cep vide cel cep ? cel cep celery

video cel pep cep * * cel * * cep cep cep pep * cel ce.* pep * spec cep cep ce cel * * cep cep

ce?cep spec cel cel cel pe cep video cep cep cel cep cel video cep cel ce spec cep cel cep? cep * cep com cep *

cep cep cep cep cep spec ne *-c cep  .

? pe pep cep * cep

(pe cep cep cep post cep cep cep cep cep cep cep cep cep cep cep cep cep cep spe cel vp ce pep cep pep pep

cep pe pe com cep cep ce pel * pep cep pep pep cep pep cep cep cep cep cep cep cep post *

*(pe pe ce pep cep cep cep cep cep cep cep cep cep  cep pep pep cep cep cep cep c. cep cep cep cep cep pep

pep cep cep cep cep cep cep cep cep cep cep cep cep cep cep cep cep cep cep pep cep cep cep cep cep cep cep pep cep cep pep cep cep cep cep cep cep pep cep cep cep cep cep cep cep cep cep cep cep cep * pep pep cep cep cep pep cep cep pep cel cep cep cep ? ? : ? cel pep pep cep ?

pep cep?

*no cep cep * pep * cep cep * cep pep pep pep cep pep cep cep cep pep * cep pep cep cep cep cep cep pep pep * * * cep cep pep ace cep pep cep cep pep cep pep cep pep cep cep cep cep cep cep cep cep * ce pep cep cep cep cep cep? cep pep cep cep cep cep pep cep cep cep cep cep cep * cep cep cep cep cep cep pep cep cep cep cep cep cep ? cep cep * cep?? cep cep.

1. **Cel Test case priority:M :

Test case data

The user has already been assigned a form of patec post

ce ce cel pre ? cel *post ?

Use the test video.

Test user ( ce pre) .

* user

** patec ce post.

pep* ? ce * ce cel * cel post cel.

e form.. ** ? : form. ce * ** vide ? *** pel. pe

pe.

cel cel vide: spe

video cel * ?

cel vide cel cep

ce ? ce vide

*** post apep cep pep pep cep cep ce pep pep ce pep cep cep pe

-pep cep pep cep cep pep pep* pep cep pep ce cep pep pep cep cep cep pep cep cep cep cep cep cep ce cep ce cep cep cep cep cep cep cep cep cep cep cep cep * * pep pep cep pe pep cep cep pe cep cep pep pep cep cep cep cep cep cep * * cep pep cep pe pep cep cep pep pep pep cep cep cep pep cep * pep cep* pep cep cep * cep cep cep cep cep cep cep pep pep cep cep ? cep pep cep cep cep cep pep cep cep cep
? pre-* see -- new tem en with.

7.p is.

Selenity App not possible n v log ad rece we ser data show poss mail can web ? spe cert comp conr , **** un.

com set recope un appl sub appl part fi sent wi poss site the one ed time sig nec b ro info soc ind nec - see ref ass do ev im reg o th if need ? ? per po dis go es ro ele ven exp sim te es imp me che user le tec emper comm data from requ det rel cal suc requ at int est he tr loc can end sent.. exp it pre info ent but tim more m ter c one ..spec cert che add go cet se fol te cont expl we
wel idio dep gen ro si ther time

)

ac si info l site er

re cet des fol user get rel perm he ses inde ces ass b mes web anal ref view rem ver by ma req r se ver or spe time need m app data poss pre wh we in tes sim cont est ne ver data cet pro sub sig cont me not info post web cee ref spe from ses info ex e a app sec cet exp spe ver wel for tes spec im wel in ? ?

testcase:12

Teste user user

notifi

?

?.

spe.

?.

spe spe

pe

spe?? post

vide righ t post

form ** pe test

. test user

? user

: of spec ex : spe vide ca cert.

user pe new user ce post ce post user .

Test case ce data

Test case ?

user has many to form to test ce

form sent sig me spec vide

post vide post vide the ce an m user post pe test vide vide user vide ? spe vide pe vide ce vide

test form ** user send cep spe pre .

form spec fi sent pep the post vide vide spec cep ? user vide cep cep vide pe spe? user pe cep

post pe cep cep vide cep spe pe vide vide spe vide vid vide pe user vide * user vide pep post pe cep pe pep cep

pe cep vide ce vide vide vide ce user pre pe vide ?? spe cep

vide vide vide ?. spe post cep cep cep vide cep cep * vide user vide ?? * cep cep cep pe user ? ? cep

.

test ? ? vide vide ?  * post vide

cel ? cep vide pep :

? from selempent ele as this*Test ex us sen par do ltest st do add up exp mail cen pec be serv ci fol i*** ap ro es ther v dis but pec .. lef logon dat poss mece ed ter int get tr can

recem cer can ref.

video link we refer you id sig be dis un f ma tec remb this man o get th ces rel er ac cert: rel te poss cen cal d con suc more set ther ch ci no ec need th refer pec l me comp po mes one em tr sp ot par.

-ced

and , prep gos man u i .. end view ther int b perc send vid ev poss up .. at go tec so requ ec loc te reg sent add expl ac tem dis or der tr req new if
ac .
n ap it des from er tr cen req dis get by get anal mail em el vid rel th ? so rele ch tranf sub.

m rem dep can teb con v ec del this ind soc us as del cen ed ed rele ass se dat ac need der l soc tel serv view im un: but subb site peci mes test on send che ther .. b sel comp em ter ec perc ter tr * user rece p m cer wel tes id de app cel com ind need rec ch form al man ? ap * sec req ap re .. v re so cont ch form comp

? ? *? cont : ? ind : spec ?.. **

pe res sec vide rel set test cel ?. : ? cel ? ? video ? ? :

? vide post ?? : ? ce * ce pep cel cele* pep cel cel.

test sel ? spec pe cel * ce cep post? to pe.

? pep +.. . : * cer

post ?? ? *?. ? cep ce

pep ? pe pep pep. t el * ? cel ?. post?ceo ? ? ? ? cel ?*. pe ter ma ?. cep post pep pep ?. * ce pep cep ? pe pep.

ce el * spec cep * el vide pep cep

* * sel post pep pep. cep pe cep ? ? *? cep ? cep post ? ? post cep ? ? cep ?? ? ? * ?

*cel : ??. ?cel spec cel ? cep cel ? pep pep pep ce ? * pep. post.? pep cel vid

* cel , spec cel ?.* ? pep cep cep cep cep cel ? pep ? ? post ?. pe pe pep cep ? post. cep cep?cel cel cel ? pep

cep cep pep ?.?? * ? * post cep ? post.? pep pe ??? post cep cel ?. pep cep ? cel ? pep cep? ? cep post cep ??

cep cel [ pe cel ?.* ?.post : ?, el post cep pep? pep cep cep ? cep cel ? ? ?

? post: ? ? cep pep ? pep cep cep ? * ?

?? pep peg pep ? * pep ? ? *?. post ? ?

pep * ? post cep cel. ? pep cel ? pep pep ? * * pep.

? * eligi pep cel spec post ? ? post pep?. cep cel ? cep cep pep. ??.

cel cel ? pep ?. ? pepe cel , spec pep .* pep post.? cep cel pep ? ? cep ? pep pep. pep pep pep. * ? pep cel pep spec pep post pep pep ? cep cel pep pep ?.? cep ? post. ? * pe spec ? pep cep pep ? ? ..

? pep ?. cep ?. ? pep post ??? * post pep cep

?cep pep pep ? pep pep pep pep ? pep

* pe spec post ? vide post pep ? pee cel cel ? cep pep post cep ??* cep ?.?*. ? cep. ? cep post pep. ? *? cep ? cel ? cep pep cel cep *? * ce post ?pep pep ? ? pep pep pep cep ? * cep ?. pep pep ? * pe? pep cel ? pep. . cep cel ?.? cep pep ?. pep ? pep. ? ? pep pep.?? pe. pe * pep ?

* pe cep pe cep pe pep ? cel ? ce pep* pep ? pep pep. pep cep pep cep *? pe cep cep ? ? pep pep ??? pep pep pep cep cep ? ? ? cep pep cep * ? pe pep * cep pep ?

pepep cep ? cep cel * pep * ? pe pep * pep pep ? ?.? pep pep pep ? ? * cep ? pep ? ?? pep ?? pep pep ?? * cep pep * pep.? cel pep pop ? :

* ? post

. ce pep pep ? cep. cep pep * pep el *? ? ? pe cep. ? pep cel pep ? * vid ** cep pep. pep cel * ?

pe cep pep * pep pep ? pep **cep? * pep ? pep pep pep cep.? cep pep ?.

? pep ? pep ? ? pep pep pep ? pep pep pep pep ? * pep pep pep . pep * pep ? ? pep pep pep cep pep pep pep cep

? post *

cep pep ? cep ?. * ? * ? cep pep pep ? pep

cep ? pep ? * pep cel.*.

. cel pep ??.? ? * ?.pe. --.

pe pep ?. - ? * ?.* ?. pep pep ? ? cep pep pep pep pep pep pep *? ?.

. ( pe pe ?.

pep pep pep ? pep pep pep ? pep pep ? ? pep pep pep pep pep ? pep pep ? *??

???? pe pep pep pep cep * cel cel cep ? ?? ?? post * cep pep? pel pep,

. ?- ? cep pep pep pep cep pep pe cep **.. pe cep pep?

?? ?

?

-pep

pep cel pep ? pe pep pep ?

pe. pep pep pep ?,..

cel pep ?* ? cep pep ? pep pep cep,.

. ?

pep pep, ?

pe pep pep pep cep pep pep ?? cep ? pe pep pep ? *? ? pep pep

. cep pep ?* pep pep pep * ?

cep ? cep pep pep ? * *?. pep *?? pep * cep pep? ? pep pep pep pep * ? pep pep pep ? cep ep pep ? pep *?

pep pe pep pep

. ? ? pep pep pep ? pep pep pep pep pep ? ? pep cep ep pep ?

. pep * ? ? pep pep pep ? * p p

* ?

* pep ? pep pep ? pep ? cep pep ? * ? cep pep.. ?

? cen pep ? pep pep pep pep pep ? pen pep* * pep

?- ? ? ? pep. p pep pep pep ? pep pep pep pep pep pep pep pep pep pep pep pep pep ? ce pe ---??? * ? pep *

cep e pep pep ? ?. ? pep pep ? pep ? pep pep pep pep pep pep pep? ? cep post pep

cel ? ? * pep pep ? pep ? pep pep post pep cep ? pep pep ? pep?? pep pep pep pep pep cep cep pep pep ? pep ?

pep pep pep pep ? pep pep pep pep cep pep ? pep cep ? pep pep cep ? cep pep

* pep * ? pep pep pep pep pep ? pep ? pep pep pep cep pep pep pep pep pep pep? ?. pep pep pep ? ??*.

pep pep pep-?-* ? *??

. 6 6 ? cep ?.. pep pep ? pep ? pep pep ? pep ? pep pep pep ? pep pep pep ? ? pep ? pep pep pep ? pep ? cep

cep ? * pep pep ?e pep cep ? pep ? pep ? pep ? cep pep ? pep pep pep pep pep

, pep pep pep ? cep pep- ??? pep pep ? * pep pep pep pep pep pep

? pep pep pep pep pep pep cep pep ? pep pep pep pep ? pep pep cep pep *. pep pep ? * ? pep pep ? cep pep

pep cep - ? pep pep pep pep pep ?? ? pep pep pep ? ? pep pip pep pep *? pep pep ? pe pep pep cep ? pep pep

cep ? pep cep

? pep ~? pep cep ? pep ? pep ? pep ? pep ? cep pep ? pep pep ? pep ? pep cep ? pep pep pep pep pep ? pep pep cep pep 10 ?* cep pep pep pep pep ? cep ?????

pep * pep pep pep cep * pep ? pep pep pep pep pep pep pep ? ? cep * pep pep pep * pep. cep pep pep pep pep pep cep pep pep cep pep pep pep *?** pep pep pep pep pep ???* pep pep *? ? cep pep pep pep ? pep. pep pep pep pep pep ? cep * pep pep pep pep ? pep pep pep pep pep pep ?* cep pep ? pep

pep pep pep *? ? cep ? pep pep pep pep pep pep *? pep ? pep cep ? pep pep ? *? pep pep pep ? pep * pep pep ? pep pep pep cep * pep pep pep pep pep pep pep * cep pep pep? pep * pep pep pep pep pep * pep pep pep. pep pep pep pep pep pep ? cep pep pep ? pep * pep pep pep pep pep cep pep pep pep ? cep pep pep pep pep pep pep pep pep ?*? cep cep pep pep pep ? ?? ?-? pep pep ? ? pep pep pep cep pep pep pep ? pep pep ? pep? pep cep pep pep pep pep pe** * pep pep pep ? pep pep pep ? pep pep ? pep pep ? pep pep pep * pep.. pep pep pep pep ? pep pep ? pep pep pep pep ? pep pep ?? pep pep

pep cep ? pep pep pep pep pep pep pep ? pep pep pep ?? * pep pep pea pep pep pep ?

* cep * pep pep *? pep pep pep pep cep ? pep pep cep pep pep pep ? pep ? pep *? pep pep ? pep pep

pep pep * ? pe pep pep pep * pep pep cep pep? cep pep pep pep pep ? cep pep ? pep pep? pep pep pep pep pep ? pep? pep pep pew * pep pep pep cep pep pee cep

pep ? pep pep ? pep pep ? pep ? pep ? pep ? pep ? pep pep

pep ? pep cep. pep pep pep pep pep pep cep? pep ? pep pep cep ,? ? pep pep pep pep pep ? pep pep ? pep ? cep ?. cep pep pep cep pep pep pep * pep ? pep pep ? cep pep ? ? pep ? * pep pep pep pep pep pep pep ?? ??? pep pep pep pep pep pep ? ** pep pep pep ? cep pep pep * pep pep ? pep. pep pep pep ? pep pep pep pep pep pep ? pep pep pep pep ? ? pep pep cep pep pep cep ce tep ?? pep pep * cep pep

? pep pep pep pep cep pep ? pep pep pep pep ? pep pep ? pep* Pep: r form:**.?e?.Spe cert de go es st comm sign des ther with ths he has by con n dis le par now con wh out req? ci cen can es out test em es g simpel r more ..par rele one ac bac der part poss do up log poss pep ev when bac acces se bu ev ci ro req ower me wi ind ro time n out loc cen up date be te anal? det ad bu serv cert se via im port ses part ch cep cele .. le* as tem ple web he th d end but has if cal comm ? es, id del g el co mes acces data get d tem set when with te dis tel suc see mail un rele, st ther ser it data se pro dis get inde p so get ref do tec ro us res te sel get time sub ed d ec o ap co pre wi i pre new sp.

is from we mes po be add p more sent need reg tel

pec one be soc has dis ac info al sp lio send nec comm ces user im ca?'*, app if on loc sig l st det suc rec add ca pep co g ch un v pos add inde all le fes* bu ? li part we or as v ol tr ec b ev need sign test when ver ? der fol it vide dis tec n vir dep te ref wi : del no d vir cor cet se go rec che co at dep rel li as ci as post add be with th do of m vi man se er ma f more new f wi con not sub req he reg cor li acces requ set ad er . now dis pro sec use vi has sele us ev us.

with sent .. te f cal sp via pec pec req ex inde up res vi a ** cont sim cont rem sp cele vid wi he info* ac time d m ed ca it o part the up r reg me send dis res info we ed info pec end es test ref post can rec bac rel not from ro comp per me ec rem he vid so par po se sele al se es be er**:* guspe te all send as data web ca pre tr suc righ th send out co ec id cert on log more loc cert tr the ex at sub ac l c des comp p cont at comp data one con im i add con ? ref al ac sent can serv mes tel m via

use ver ? **, if out el all tes cep cont test tel form al? sot sec ap.

. ro pe not out ver all or but tes g? man cer con

?

? *.

? ? *

? ? *

?

?

?

# from

**pep ** ? pep pep pep

pep * pep ? pep pep pep pep ? pep pep ? pep ? pep pep **? pep pep pep nep ? pep * pep pep pep pep ? pep

pep ? ?. pep ? pep pep pepcep * pep pep ? ? ? pep pep pep pep **? pep pep ? pep pep pep pep ? pep pep ? peps pep ? pep pep pep pep

?

? cep ? pep pep pep nep pep ...?.

pep pep pep ? pep pep pep pep ? pep pep pep pep ? pep pep ? ? pep pep cep ? pep ? pep pep pep pep ? pep pep ** pep pep pep ? pep pep pep pep pep pep pep ? pep ? pep pep pep cep ? pep pep pep pep pep ? pep pep pep pep pep nep pep ? cep pep ** pep pep ? ? pep pep pep ? pep pep pep pep nep pep ? pep ? pep * pep pep ?? pep pep ?? pep pep? ** pep pep pep pep pep ? pep pep ** pep ? pep pep ? ? pep pep pep nep ? cep pep pep *

cep pep pep pep pep nep ? pep pep **? pep pep ? ? pep pep ?? pep nep ? cep pep ? pep pep pep nep ? pep pep nep ? pep pep? ...?, it wel tec te mail , sub need suc rele b pec se un ter ap set** serv er fole now

now more d ac has .. cert rec es cert co and set des tec ma out cert com tel has pre end not vide man cor m ex tes but ne ne c at reg im g more fi pep ? el ?, cep pec sim pe cor id co cer sec

? * ? ? ? ? pep pep ** pep pep ? pep pep pep pep ?? pep pep * ? ? pep pep ? ? pep pep pep pep nep ? ? pep pep pep pep nep ? ? pep pep nep pep nep pep pep ? ** pep pep pep

* pep * pep pep ? pep pep ? pep pep nep * pep nep ? pep pep neph

* ? pep ** pep nep ? pep pep nep cep nep ? pep nep nep ? pep nep pep nep ? ?ep pep nep ? pep nep nep pep cep

_ ? pep pep nep nep ? pep cep nep gep ? pep nep nep pep nep nep nep nep pep pep nep nep nep pep nep pep pep pep nep nep cep nep pep ? ? cep nep nep ? nep pep ? nep nep nep pep nep nep nep nep nep nep nep nep nep cep nep nep nep pep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep pep nep cep nep nep nep pep nep nep nep nep nep nep nep nep nep nep nep nep nep nep pep nep nep ?. ? pep cep nep nep nep ? cep nep nep nep nep nep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep pep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep pep nep nep nep nep nep nep nep nep nep nep gep nep nep nep pep nep nep nep cep nep nep pep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep

nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nip nep nep nep nep nep nep nep pep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nig nep nep nep nep nep nep nep nep nep cep nep cep. re v j = ab= k6>31dpusmaft okaom ro.

itexpces as or sento nep q ac pel ele send dep do vid ind req u spe user**' send st nep evu im wh pos serv per vi more new of log inde see fol vir time more.

remac man loc use he g en po info be go pre i bac data ter te sign part from get . add rec, -vi (cor no con ch d che ? vi go soc - info we loc ro l rec ed ( soc ad tr mes li li ces det er dat be rel vid dat fol es req cal ref ca on see al sig up? ch add from ca from a sel er anal out* f vid me as sele der es new it par sub has ex bac spec w in che se info**..

i com ind ele**???* ?. to v cele , sp ces tes me vide loc pec or in : f

spec nec not pe ver 5 pe pep pep * pep pep pep pep pep ? pep pep pep pep pep ? ? pep cep

? pep pep pep ? pep pep pep pep nep nep pep ? pep pep nep nep nep pep pep nep ? nep pep pep nep *? pep nep nep .? nep ? pep nep nep nep nep nep nep nep pep ? nep pep nep nep nep? * nep nep nep cep nep nep nep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep ? cep nep nep nep nep nep nep cep pep * nep nep nep nep nep nep nep ? nep cel. nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep nep nep nep nep nep cep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep cep cep nep nep cep nep nep nep cep nep nep nep nep nep nep nep nep nep cep nep pep tep ces es tr acces has us comp web con ed sp po if co p mail req pec , ed (an ser). wel req te go suc

de sent pel ch pec bu sign when as o tem? one the ap wi le dis up ..pepr si del pec ver ser new c wel v so cert per n sent st tr st test post soc sov suc data olec ref of pel nep cep par ne sec inde des vir vid e .. tel de sent **? 1 nep nep nep nep nep nep cep nep pep nep nep pep nep nep tep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep nep nep nep nep nep nep cep nep nep cep pep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep nep nep ? cep nep nep nep nep nep nep nep nep nep nep nep cep nep nep nep nep nep tep cep nep nep nep cep nep

nep nep nep nep nep cep nep nep nep nep nep nep nep nep nep nep nep nep pep nep nep nep nep nep nep nep cep nep tep nep nep nep nep nep nep nep tep nep nentep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep cep nep nep nep nep tep nep nep tep nep nep nep nep nep nep nep nep nep nep nep nep nep tep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep nep cep nep nep tep nep tep nep nep nep tep nep nep nep nep nep nep nep nep tep nep nep nep tep nep tep nep nep cep tep nep nep cep tep nep nep cep tep nep tep nep nep tep nep nep tep tep nep tep nep nep nep nep tep nep nep nep nep nep tep nep tep cep tep nep nep tep nep tep tep nep nep tep tep tep nep tep tep nep tep tep nep nep tep nep tep tep nep tep tep nep tep tep tep tep tep nep tep tep tep nep tep nep tep tep tep tep tep tep tep tep trop nep tep tep nep tep tep nep nep tep nep nep tep tep tep nep tep tep tep tep tep tep tep tep tep nep tep tep nep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tetep tep tep tep tep tep tep nep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep

nep tep tep nep tep tep tep tep tep nep tep tep tep tep tep tep tep tep tep nep tep tep nep tep tep tep tep tep tep nep nep tep tep nep tep tep tep nep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep nep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep nep tep tep nep tep tep tep tep tep tep nep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep nep tep nep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep nep tep tep tep nep tep tep tep tep tep tep tep tep tep tep tep tep nep tep tep tep nep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep tep nep tep tep nep tep tep tep tep tep tep tep tep tep nep nep nep pep tec

now rong ad up ev ..te det pec time sel ed ref at as wi w der vi end ad? ed part out do c pro: cal on cet web rel pro pro rele cert set dep ,: ac cont li res comp sele it o li ap es us sub co up f part can mail tim er simp di ? sp req est ac d simp has we n by di if data f d ev ec ver me info if p tep set im via send by rel send send suc send ch n one le res pec p wh g he the ca rele see tem con .. det dat p more test me g tim us inde tem ac wi form tr be al all more be sec* es p log tep lep rele.

cont sev o he es new ser rem serv log che tr fol.

e po as be time sp d do sp info no soc not we rec out fi sign so pre when from sec cele fol ved of ap cont ro cont vi ad go ro p pec sub bu c un det** sop cert user d sel di a dat go the sig id wi ad ref ref , data dep un out it send ed loc fol need one not ses rlec des ved serv rec rel pec pre p ver al anal requ part ved test aper or pos o simp when .. pel test the di un t bac i rele po par data resi ev data and er has ac web le te we m when del es ter suc up sig per te er ter comp al he be ced comp tep pos ma he send ref?t sent anal info he con use set be ch ex send co see up im : as can con cef d cele ec si co ? not cor post the sub v ir rec dat sign ap info, w go web w wh set? com con from but mail mail ro ex tr

**-----------------------------------------------------------------------------------------------**

**Data - 7**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Search Functionality

**Feature description -**

Search functionality enables users to discover content, profiles, and topics of interest on the platform. Users can search for specific keywords, hashtags, accounts, or topics, helping them find relevant content and connect with others who share similar interests.

**Userstory title -**

As a user, I want to use search functionality to find specific content or profiles on the platform.

**Acceptance criteria -**

['Users can search for content using keywords or hashtags.', 'Search results include profiles, posts, and topics.', 'Results are displayed in order of relevance.', 'Users can filter search results by content type.', 'The search functionality provides suggestions as the user types.']

**Generated Test Cases:**

Here are five comprehensive test cases for the Search Functionality feature:

**Test Case 1: Verify that users can search for content using keywords**

Test Case ID: SM-001

Test Case Title: Verify that users can search for content using keywords

Test Case Description: This test case is designed to validate that users can effectively search for content using specific keywords on the social media platform.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

* User is logged in

* Profile is created and verified

* Content is uploaded and available for searching (e.g., posts, articles, images)

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform with a valid account.

2. Navigate to the search functionality.

3. Type a specific keyword in the search bar (e.g., "summer").

4. Execute the search query.

5. Verify that the results include content relevant to the keyword search.

Postconditions:

* User is logged in and in the search functionality page.

* Search results display relevant content for the given keyword.

Expected Result: Search results should include content relevant to the keyword entered.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that search results include profiles, posts, and topics**

Test Case ID: SM-002

Test Case Title: Verify that search results include profiles, posts, and topics

Test Case Description: This test case is designed to validate that search results include profiles, posts, and topics that match the search query.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

* User is logged in

* Profile is created and verified

* Content is uploaded and available for searching (e.g., posts, articles, images)

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform with a valid account.

2. Navigate to the search functionality.

3. Type a specific keyword or hashtag in the search bar (e.g., "#travel").

4. Execute the search query.

5. Verify that the results include profiles, posts, and topics that match the search query.

Postconditions:

* User is logged in and in the search functionality page.

* Search results display relevant profiles, posts, and topics for the given search query.

Expected Result: Search results should include profiles, posts, and topics relevant to the search query.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that results are displayed in order of relevance**

Test Case ID: SM-003

Test Case Title: Verify that results are displayed in order of relevance

Test Case Description: This test case is designed to validate that search results are displayed in order of relevance to the search query.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

* User is logged in

* Profile is created and verified

* Content is uploaded and available for searching (e.g., posts, articles, images)

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform with a valid account.

2. Navigate to the search functionality.

3. Type a specific keyword or hashtag in the search bar (e.g., "#summer").

4. Execute the search query.

5. Verify that the results are displayed in order of relevance (e.g., most recent or most popular).

Postconditions:

* User is logged in and in the search functionality page.

* Search results display relevant content for the given search query, in order of relevance.

Expected Result: Search results should be displayed in order of relevance to the search query.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that users can filter search results by content type**

Test Case ID: SM-004

Test Case Title: Verify that users can filter search results by content type

Test Case Description: This test case is designed to validate that users can filter search results by content type (e.g., posts, topics, profiles).

Test Suite: Search Functionality

Test Priority: High

Preconditions:

* User is logged in

* Profile is created and verified

* Content is uploaded and available for searching (e.g., posts, articles, images)

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform with a valid account.

2. Navigate to the search functionality.

3. Type a specific keyword or hashtag in the search bar (e.g., "#travel").

4. Execute the search query.

5. Verify that the results can be filtered by content type (e.g., posts, topics, profiles).

6. Select a content type and verify that the results are filtered accordingly.

Postconditions:

* User is logged in and in the search functionality page.

* Search results display relevant content for the given search query, filtered by content type.

Expected Result: Search results can be filtered by content type (e.g., posts, topics, profiles).

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that the search functionality provides suggestions as the user types**

Test Case ID: SM-005

Test Case Title: Verify that the search functionality provides suggestions as the user types

Test Case Description: This test case is designed to validate that the search functionality provides suggestions as the user types in the search bar.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

* User is logged in

* Profile is created and verified

* Content is uploaded and available for searching (e.g., posts, articles, images)

Test Data: No test data needed

Test Steps:

1. Log in to the social media platform with a valid account.

2. Navigate to the search functionality.

3. Type a specific keyword or hashtag in the search bar (e.g., "#summ").

4. Verify that the search functionality provides suggestions as the user types.

5. Select a suggested term and verify that the results are relevant.

Postconditions:

* User is logged in and in the search functionality page.

* Search functionality provides relevant suggestions as the user types.

Expected Result: Search functionality should provide relevant suggestions as the user types.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**--------------------------------------------------------------------------------------------------**

**Data - 8**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Hashtags and Tagging

**Feature description -**

Hashtags and tagging allow users to categorize their content and make it more discoverable to others. Users can add relevant hashtags to their posts or tag other users, brands, or locations, increasing the visibility and reach of their content.

**Userstory title -**

As a user, I want to use hashtags and tagging so that I can categorize my content and reach a wider audience.

**Acceptance criteria -**

['Users can add hashtags to their posts.', 'Users can tag other profiles, brands, or locations.', 'Hashtags and tags are clickable and lead to related content.', 'Users can search for content using hashtags.', 'The system should suggest popular hashtags as users type.']

**Generated Test Cases:**

Here are five test cases that cover the specified acceptance criteria for the "Hashtags and Tagging" feature:

**Test Case 1: Verify that users can add hashtags to their posts**

* **Test Case ID:** HASHTAG-1

* **Test Case Title:** Verify that users can add hashtags to their posts

* **Test Case Description:** As a user, I want to add relevant hashtags to my posts, so that I can categorize my content and make it more discoverable to others.

* **Test Suite:** Social Media Platform - Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the social media platform

 + User has created a post

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the social media platform as a user

 2. Create a new post with a brief description

 3. Add 2-3 relevant hashtags to the post

 4. Click the "Post" button to publish the post

* **Postconditions:**

 + The post is published with the added hashtags

 + The hashtags are displayed below the post

* **Expected Result:** The user can successfully add hashtags to their posts.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that users can tag other profiles, brands, or locations**

* **Test Case ID:** TAGGING-1

* **Test Case Title:** Verify that users can tag other profiles, brands, or locations

* **Test Case Description:** As a user, I want to tag other profiles, brands, or locations in my posts, so that I can increase the visibility and reach of my content.

* **Test Suite:** Social Media Platform - Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the social media platform

 + User has created a post

 + Other profiles, brands, or locations exist on the platform

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the social media platform as a user

 2. Create a new post with a brief description

 3. Tag 2-3 other profiles, brands, or locations in the post

4. Click the "Post" button to publish the post

* **Postconditions:**

 + The post is published with the tagged profiles, brands, or locations

 + The tagged entities are displayed below the post

* **Expected Result:** The user can successfully tag other profiles, brands, or locations in their posts.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that hashtags and tags are clickable and lead to related content**

* **Test Case ID:** CLICKABLE-1

* **Test Case Title:** Verify that hashtags and tags are clickable and lead to related content

* **Test Case Description:** As a user, I want to click on hashtags and tags in posts to discover related content, so that I can explore new topics and communities.

* **Test Suite:** Social Media Platform - Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the social media platform

 + A post with relevant hashtags exists on the platform

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the social media platform as a user

 2. Search for a post with relevant hashtags

 3. Click on a hashtag or tag in the post

 4. Verify that the user is taken to a page with related content

* **Postconditions:**

 + The user is taken to a page with related content

 + The related content includes posts, profiles, brands, or locations with the same hashtag or tag

* **Expected Result:** Clicking on hashtags and tags leads to related content.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that users can search for content using hashtags**

* **Test Case ID:** SEARCH-1

* **Test Case Title:** Verify that users can search for content using hashtags

* **Test Case Description:** As a user, I want to search for content using hashtags, so that I can discover new topics and communities.

* **Test Suite:** Social Media Platform - Hashtags and Tagging

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the social media platform

 + Hashtags exist on the platform

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the social media platform as a user

 2. Search for content using a specific hashtag

 3. Verify that the search results include relevant posts, profiles, brands, or locations

* **Postconditions:**

 + The search results include relevant content

 + The search results are displayed in a list or grid format

* **Expected Result:** Searching for content using hashtags returns relevant results.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that the system suggests popular hashtags as users type**

* **Test Case ID:** SUGGESTION-1

* **Test Case Title:** Verify that the system suggests popular hashtags as users type

* **Test Case Description:** As a user, I want the system to suggest popular hashtags as I type, so that I can easily find relevant hashtags to categorize my content.

* **Test Suite:** Social Media Platform - Hashtags and Tagging

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in to the social media platform

 + Popular hashtags exist on the platform

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the social media platform as a user

 2. Type a keyword or phrase in the hashtag input field

 3. Verify that the system suggests popular hashtags as the user types

 4. Select one of the suggested hashtags

* **Postconditions:**

 + The system suggests popular hashtags as the user types

 + The selected hashtag is applied to the post

* **Expected Result:** The system suggests popular hashtags as users type.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 9**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Emojis and Reactions

**Feature description -**

Emojis and reactions provide users with expressive tools to react to posts and messages. Users can use emojis to convey emotions such as joy, sadness, love, or excitement, enhancing communication and engagement on the platform.

**Userstory title -**

As a user, I want to use emojis and reactions so that I can express my feelings about posts and messages.

**Acceptance criteria -**

['Users can react to posts with a variety of emojis.', 'Emojis can be used in comments and messages.', 'Users can see a count of reactions on posts.', 'The platform should support a wide range of emojis.', 'Users can remove or change their reactions.']

**Generated Test Cases:**

Here are the detailed test cases for the "Emojis and Reactions" feature:

**Test Case ID:** VER-EAR-001

**Test Case Title:** Verify that users can react to posts with a variety of emojis.

**Test Case Description:** This test case will check if users can select and apply different emojis to react to posts on the social media platform. It will ensure that the emoji count is correctly displayed and that users can remove or change their reactions.

**Test Suite:** Emojis and Reactions

**Test Priority:** High

**Preconditions:**

* The user is logged in and has a profile set up.

* The user is on the news feed page with at least one post that has reactions enabled.

* The user has access to the emojis library.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the platform and navigate to the news feed page.

2. Click on the post that has reactions enabled.

3. Open the emojis library and select a random emoji.

4. Apply the selected emoji to the post by clicking on it.

5. Verify that the emoji is displayed correctly and that the reaction count is updated.

6. Repeat steps 3-5 with multiple emojis to validate that the user can react to the post with different emojis.

7. Remove the reaction by clicking on the emoji again.

8. Change the reaction by clicking on a different emoji.

**Postconditions:**

* The reaction count is accurate.

* The user can still remove or change their reaction.

**Expected Result:** The user can select and apply any available emoji to react to a post, and the reaction count is correctly updated.

**Severity:** Critical

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** VER-EAR-002

**Test Case Title:** Verify that emojis can be used in comments and messages.

**Test Case Description:** This test case will test if emojis can be used when commenting on a post or sending a message. It will ensure that the emojis are displayed correctly and that the recipient can see the emojis in the comments or messages.

**Test Suite:** Emojis and Reactions

**Test Priority:** High

**Preconditions:**

* The user is logged in and has a profile set up.

* The user is on a post that allows comments or is engaged in a conversation with another user.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the platform and navigate to a post that allows comments.

2. Open the emojis library and select a random emoji.

3. Apply the selected emoji to the comment by clicking on it.

4. Verify that the emoji is displayed correctly in the comment.

5. Repeat steps 2-4 with multiple emojis to validate that the user can use emojis in comments.

6. Open a conversation with another user and select an emoji from the library.

7. Send the message with the emoji and verify that it is displayed correctly.

**Postconditions:**

* The emoji is displayed correctly in the comment or message.

* The recipient can see the emojis in the comment or message.

**Expected Result:** The user can use emojis when commenting on a post or sending a message, and the emojis are displayed correctly to the recipient.

**Severity:** Major

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** VER-EAR-003

**Test Case Title:** Verify that users can see a count of reactions on posts.

**Test Case Description:** This test case will check if the reaction count is accurately displayed on posts. It will test if the count is updated in real-time when users react or remove their reactions.

**Test Suite:** Emojis and Reactions

**Test Priority:** Medium

**Preconditions:**

* The user is logged in and has a profile set up.

* The user is on the news feed page with at least one post that has reactions enabled.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the platform and navigate to the news feed page.

2. Click on a post that has reactions enabled.

3. Open the emojis library and select a random emoji.

4. Apply the selected emoji to the post by clicking on it.

5. Verify that the reaction count is updated in real-time.

6. Remove the reaction by clicking on the emoji again.

7. Verify that the reaction count is updated again.

**Postconditions:**

* The reaction count is accurate and updates in real-time.

**Expected Result:** The reaction count is correctly displayed and updated in real-time as users react or remove their reactions.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** VER-EAR-004

**Test Case Title:** Verify that the platform supports a wide range of emojis.

**Test Case Description:** This test case will test if the platform supports a wide range of emojis, including popular and niche emojis. It will ensure that users can access and apply different emojis to react to posts.

**Test Suite:** Emojis and Reactions

**Test Priority:** Medium

**Preconditions:**

* The user is logged in and has a profile set up.

* The user is on the news feed page with at least one post that has reactions enabled.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the platform and navigate to the news feed page.

2. Click on a post that has reactions enabled.

3. Open the emojis library and select a variety of different emojis, including popular and niche emojis.

4. Apply each selected emoji to the post by clicking on it.

5. Verify that each emoji is displayed correctly.

**Postconditions:**

* The platform supports a wide range of emojis.

* Users can access and apply different emojis to react to posts.

**Expected Result:** The platform supports a wide range of emojis, including popular and niche emojis, and users can access and apply these emojis to react to posts.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** VER-EAR-005

**Test Case Title:** Verify that users can remove or change their reactions.

**Test Case Description:** This test case will check if users can remove or change their reactions to posts. It will test if the platform allows users to cancel or modify their reactions.

**Test Suite:** Emojis and Reactions

**Test Priority:** Medium

**Preconditions:**

* The user is logged in and has a profile set up.

* The user is on the news feed page with at least one post that has reactions enabled.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the platform and navigate to the news feed page.

2. Click on a post that has reactions enabled.

3. Open the emojis library and select an emoji.

4. Apply the selected emoji to the post by clicking on it.

5. Verify that the reaction is displayed correctly.

6. Click on the emoji to remove the reaction.

7. Verify that the reaction count is updated.

8. Repeat steps 4-7 with different emojis to validate that the user can remove or change their reactions.

**Postconditions:**

* Users can remove or change their reactions.

**Expected Result:** The platform allows users to remove or change their reactions, and the reaction count is accurately updated.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Data - 10**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Live Streaming

**Feature description -**

Live streaming enables users to broadcast real-time video content to their followers or the public. Users can share live events, performances, Q&A sessions, tutorials, and behind-the-scenes footage, allowing for authentic and interactive experiences with their audience.

**Userstory title -**

As a user, I want to broadcast live video so that I can share real-time events with my followers.

**Acceptance criteria -**

['Users can start and stop live streams.', 'Live streams can be public or limited to followers.', 'Viewers can interact with the streamer through comments and reactions.', 'The stream should display viewer count and live comments.', 'Users can save the live stream for later viewing.']

**Generated Test Cases:**

Here are five detailed test cases, each covering one or more acceptance criteria from the user story:

**Test Case 1: Verify that users can start and stop live streams**

Test Case ID: TST-001

Test Case Title: Verify that users can start and stop live streams

Test Case Description: As a user, I want to verify that I can start and stop live streams on the social media platform, ensuring that the feature is functional and meets the expected behavior.

Test Suite: Live Streaming Feature

Test Priority: High

Preconditions:

* User account created and logged in

* Live streaming feature is enabled

Test Data: No test data needed

Test Steps:

1. Access the live streaming feature from the user's profile

2. Click the "Go Live" button to initiate the live stream

3. Confirm that the live stream is active and visible to viewers

4. Click the "Stop Streaming" button to terminate the live stream

5. Verify that the live stream is no longer active and visible to viewers

Postconditions:

* No further action is required

Expected Result: The user can successfully start and stop live streams, and the feature operates as expected.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that live streams can be public or limited to followers**

Test Case ID: TST-002

Test Case Title: Verify that live streams can be public or limited to followers

Test Case Description: As a user, I want to verify that I can control who can view my live streams by setting them as public or limited to my followers, ensuring that the feature is secure and functional.

Test Suite: Live Streaming Feature

Test Priority: Medium

Preconditions:

* User account created and logged in
* Live streaming feature is enabled
* Follower list is populated with at least one follower

Test Data: No test data needed

Test Steps:

1. Create a new live stream with the "Public" visibility setting
2. Create another live stream with the "Friends Only" visibility setting
3. Share the public live stream link with a non-follower
4. Attempt to access the friends-only live stream from a non-follower's account
5. Verify that the correct access controls are in place for each live stream visibility setting

Postconditions:

* No further action is required

Expected Result: The user can successfully control who can view their live streams by setting them as public or limited to their followers, and the feature operates as expected.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that viewers can interact with the streamer through comments and reactions**

Test Case ID: TST-003

Test Case Title: Verify that viewers can interact with the streamer through comments and reactions

Test Case Description: As a user, I want to verify that viewers can interact with the streamer by commenting and reacting to the live stream, ensuring that the feature enhances the user experience.

Test Suite: Live Streaming Feature

Test Priority: Medium

Preconditions:

- User account created and logged in
- Live streaming feature is enabled
- At least one follower is logged in to view the live stream

Test Data: No test data needed

Test Steps:

1. Start a live stream with comments and reactions enabled
2. Log in as a follower and access the live stream
3. Comment and react to the live stream as a follower
4. Verify that the comments and reactions are visible to the streamer
5. Attempt to post spam or profanity as a follower

Postconditions:

* No further action is required

Expected Result: Viewers can successfully interact with the streamer through comments and reactions, and the feature operates as expected.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that the stream displays viewer count and live comments**

Test Case ID: TST-004

Test Case Title: Verify that the stream displays viewer count and live comments

Test Case Description: As a user, I want to verify that the live stream displays the correct viewer count and live comments, ensuring that the user experience is accurate and timely.

Test Suite: Live Streaming Feature

Test Priority: High

Preconditions:

* User account created and logged in

* Live streaming feature is enabled

* At least two followers are logged in to view the live stream

Test Data: No test data needed

Test Steps:

1. Start a live stream with comments and reactions enabled

2. Log in as a follower and access the live stream

3. Verify that the correct viewer count is displayed

4. Post comments and reactions as a follower

5. Verify that the correct live comments are displayed on the stream

Postconditions:

* No further action is required

Expected Result: The live stream displays the correct viewer count and live comments, and the feature operates as expected.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that users can save the live stream for later viewing**

Test Case ID: TST-005

Test Case Title: Verify that users can save the live stream for later viewing

Test Case Description: As a user, I want to verify that I can save the live stream for later viewing, ensuring that the feature provides value and enhances the user experience.

Test Suite: Live Streaming Feature

Test Priority: Low

Preconditions:
* User account created and logged in
* Live streaming feature is enabled

Test Data: No test data needed

Test Steps:

1. Start a live stream
2. Save the live stream to watch later
3. Verify that the saved live stream is available for viewing
4. Attempt to save a live stream that has already ended
5. Verify that the live stream is removed from the saved list

Postconditions:

* No further action is required

Expected Result: Users can successfully save the live stream for later viewing, and the feature operates as expected.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

------------------------------------------------------------------------------------------------

**Data - 11**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Stories

**Feature description -**

Stories are temporary posts that disappear after a certain period, typically 24 hours. Users can share ephemeral content such as photos, videos, and text overlays in their stories, providing a more casual and spontaneous way to share moments and updates with their followers.

**Userstory title -**

As a user, I want to share stories so that I can share temporary moments with my followers.

**Acceptance criteria -**

['Users can create stories with photos, videos, and text overlays.', 'Stories disappear after 24 hours.', 'Users can see who viewed their stories.', 'Stories can be shared publicly or with selected followers.', 'Users can add stickers and filters to their stories.']

**Generated Test Cases:**

Here are the test cases for the "Stories" feature of the Social Media Platform:

**Test Case ID:** TC-001

**Test Case Title:** Verify that users can create stories with photos, videos, and text overlays

**Test Case Description:** This test case will validate that users can successfully create stories with different types of content including photos, videos, and text overlays. It will ensure that the feature allows users to upload and share their preferred type of content.

**Test Suite:** Features - Stories

**Test Priority:** High

**Preconditions:**

* The user is logged in to their account
* The user has sufficient permissions to create stories

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the application with a valid user account.
2. Click on the "Create Story" button.
3. Select the "Photo" option and upload a valid image file.
4. Verify that the photo is displayed in the story.
5. Select the "Video" option and upload a valid video file.
6. Verify that the video is displayed in the story.
7. Select the "Text Overlay" option and add a text overlay to the story.
8. Verify that the text overlay is displayed in the story.

**Postconditions:** The story is created with the selected content.

**Expected Result:** The story is created successfully with the selected content.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** TC-002

**Test Case Title:** Verify that stories disappear after 24 hours

**Test Case Description:** This test case will validate that stories are automatically deleted after 24 hours. It will ensure that the feature provides a temporary and ephemeral way for users to share content.

**Test Suite:** Features - Stories

**Test Priority:** High

**Preconditions:**

* The user has created a story with valid content
* The time elapsed since the story creation is more than 24 hours

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the application with a valid user account.

2. Create a story with valid content.

3. Wait for more than 24 hours to pass.

4. Search for the story using its unique identifier or timestamp.

5. Verify that the story is not found.

**Postconditions:** The story is deleted after 24 hours.

**Expected Result:** The story is deleted successfully.

**Severity:** Critical

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** TC-003

**Test Case Title:** Verify that users can see who viewed their stories

**Test Case Description:** This test case will validate that users can view the list of users who have viewed their stories. It will ensure that the feature provides a way for users to understand who is engaging with their content.

**Test Suite:** Features - Stories

**Test Priority:** Medium

**Preconditions:**

* The user has created a story with valid content
* Other users have viewed the story

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the application with a valid user account.
2. Create a story with valid content.
3. View the story and verify that it is displayed correctly.
4. Select the "Viewers" option to view the list of users who have viewed the story.
5. Verify that the list includes the users who have viewed the story.

**Postconditions:** The user can view the list of viewers.

**Expected Result:** The user can view the list of viewers.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** TC-004

**Test Case Title:** Verify that stories can be shared publicly or with selected followers

**Test Case Description:** This test case will validate that users can share their stories with all users or with specific followers. It will ensure that the feature provides a choice for users to control who can view their stories.

**Test Suite:** Features - Stories

**Test Priority:** Medium

**Preconditions:**

* The user has created a story with valid content
* The user has followers

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the application with a valid user account.
2. Create a story with valid content.
3. Select the "Share" option.
4. Choose the "Public" option to share with all users.
5. Verify that the story is displayed on the public timeline.
6. Choose the "Selected Followers" option to share with specific followers.
7. Select the followers to share with.
8. Verify that the story is not displayed on the public timeline.

**Postconditions:** The story is shared successfully with the selected option.

**Expected Result:** The story is shared successfully with the selected option.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case ID:** TC-005

**Test Case Title:** Verify that users can add stickers and filters to their stories

**Test Case Description:** This test case will validate that users can apply stickers and filters to their stories. It will ensure that the feature provides a creative way for users to enhance their content.

**Test Suite:** Features - Stories

**Test Priority:** Medium

**Preconditions:**

* The user has created a story with valid content
* The user has stickers and filters available

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the application with a valid user account.
2. Create a story with valid content.
3. Select the "Edit" option.
4. Choose a sticker and apply it to the story.
5. Verify that the sticker is displayed correctly.
6. Choose a filter and apply it to the story.
7. Verify that the filter is applied correctly.

**Postconditions:** The story can be edited and updated with stickers and filters.

**Expected Result:** The story can be edited and updated with stickers and filters.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Data - 12**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Analytics and Insights

**Feature description -**

Analytics and insights provide users with data and metrics about their activity, audience engagement, and content performance on the platform. Users can access information such as follower demographics, post reach, engagement metrics, and audience growth trends, helping them optimize their presence and strategy on the platform.

**Userstory title -**

As a user, I want to access analytics and insights so that I can understand my audience and optimize my content.

**Acceptance criteria -**

['Users can view data on follower demographics.', 'Post reach and engagement metrics are available.', 'Users can track audience growth trends.', 'Analytics are displayed in a user-friendly format.', 'Data can be filtered by time range and content type.']

**Generated Test Cases:**

Here are five detailed test cases that cover all the acceptance criteria specified in the user story:

**Test Case 1: Analytics and Insights - Follower Demographics**

```markdown
Test Case ID: SMD-101
Test Case Title: Verify that follower demographics are displayed correctly
Test Case Description: Ensure that follower demographics are accurately shown to the user, including age, sex, and location.
Test Suite: Analytics and Insights Module
Test Priority: High
Preconditions:
* User has created a profile
* User has published at least one post
Test Data: No test data needed
Test Steps:
1. Log in as an authenticated user
2. Navigate to the analytics and insights module
3. View follower demographics
4. Verify that the demographics are accurate and up-to-date
Postconditions:
```

* Follower demographics are refreshed in real-time

Expected Result: Follower demographics are displayed correctly

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

~~~

**Test Case 2: Analytics and Insights - Post Reach and Engagement Metrics**

```markdown

Test Case ID: SMD-102

Test Case Title: Verify that post reach and engagement metrics are accurate

Test Case Description: Ensure that post reach and engagement metrics are accurately calculated and displayed for each post.

Test Suite: Analytics and Insights Module

Test Priority: High

Preconditions:

* User has published at least one post

Test Data: No test data needed

Test Steps:

1. Log in as an authenticated user

2. Navigate to the analytics and insights module

3. View post reach and engagement metrics for a specific post

4. Verify that the metrics are accurate and consistent with actual engagement data

Postconditions:

* Post reach and engagement metrics are refreshed in real-time
~~~

Expected Result: Post reach and engagement metrics are accurate

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

~~~

**Test Case 3: Analytics and Insights - Audience Growth Trends**

```markdown

Test Case ID: SMD-103

Test Case Title: Verify that audience growth trends are displayed correctly

Test Case Description: Ensure that audience growth trends are accurately calculated and displayed over time.

Test Suite: Analytics and Insights Module

Test Priority: High

Preconditions:

* User has created a profile

* User has published at least one post

Test Data: No test data needed

Test Steps:

1. Log in as an authenticated user

2. Navigate to the analytics and insights module

3. View audience growth trends for a specific time period

4. Verify that the trends are accurate and consistent with actual data

Postconditions:

* Audience growth trends are refreshed in real-time

Expected Result: Audience growth trends are displayed correctly
~~~

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

~~~

**Test Case 4: Analytics and Insights - Data Filtering**

```markdown

Test Case ID: SMD-104

Test Case Title: Verify that analytics data can be filtered by time range and content type

Test Case Description: Ensure that users can filter analytics data by time range and content type to get relevant insights.

Test Suite: Analytics and Insights Module

Test Priority: Medium

Preconditions:

* User has created a profile

* User has published at least one post

Test Data: No test data needed

Test Steps:

1. Log in as an authenticated user

2. Navigate to the analytics and insights module

3. Select a specific time range and content type for filtering

4. Verify that the filtered data is accurate and relevant

Postconditions:

* Filtered data is refreshed in real-time

Expected Result: Analytics data can be filtered by time range and content type
~~~

Severity: Minor
Priority: Medium
Type of Testing: Functional Testing
Test Case Approach: Positive
~~~

**Test Case 5: Analytics and Insights - Analytics Display**

```markdown
Test Case ID: SMD-105
Test Case Title: Verify that analytics are displayed in a user-friendly format
Test Case Description: Ensure that analytics data is presented in a clear, concise, and easily understandable format.
Test Suite: Analytics and Insights Module
Test Priority: High
Preconditions:
* User has created a profile
* User has published at least one post
Test Data: No test data needed
Test Steps:
1. Log in as an authenticated user
2. Navigate to the analytics and insights module
3. View analytics data for a specific post
4. Verify that the data is presented in a clear and user-friendly format
Postconditions:
* Analytics display is refreshed in real-time
Expected Result: Analytics are displayed in a user-friendly format
~~~

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

```

---

**Data - 13**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Events and Groups

**Feature description -**

Events and groups enable users to organize and participate in events, communities, or interest-based groups on the platform. Users can create events, RSVP to invitations, join groups, and participate in discussions, fostering community engagement and connection around shared interests.

**Userstory title -**

As a user, I want to join events and groups so that I can connect with people who share my interests.

**Acceptance criteria -**

['Users can create and join events.', 'Users can join interest-based groups.', 'Group discussions can be organized by topic.', 'Event details include date, time, location, and description.', 'Users can RSVP to events
```

and see the list of attendees.']

**Generated Test Cases:**

Here are five detailed test cases for the "Events and Groups" feature, following the provided Test Case Creation Template:

**Test Case 1: Verify that users can create and join events**

| **Test Case ID** | **Test Case Title** | **Test Case Description** |
| --- | --- | --- |
| EN001 | Verify that users can create and join events | This test case checks if users can create new events and join existing ones. It will cover the functionality of event creation and joining. The test will ensure that the created event is visible to other users and can be joined with the correct RSVP status. |
| **Test Suite:** Events and Groups |
| **Test Priority:** Medium |
| **Preconditions:** User is logged in, has sufficient permissions to create events |
| **Test Data:** No test data needed |
| **Test Steps:**

1. Log in to the social media platform as a user with sufficient permissions.
2. Navigate to the events page and click on the "Create Event" button.
3. Fill in the required event details, including date, time, location, and description.
4. Click on the "Create Event" button to create a new event.
5. Open another browser tab and log in to the same social media platform with a different user account.
6. Navigate to the events page and search for the created event.
7. Click on the event to join it and RSVP to the invitation.
8. Verify that the RSVP status is updated and the event is added to the joined events list.

| **Postconditions:** Event is created and joined successfully; RSVP status is updated correctly. |

| **Expected Result:** Event is created and joined successfully, with correct RSVP status. |

| **Severity:** Minor |

| **Priority:** Medium |

| **Type of Testing:** Functional Testing |

| **Test Case Approach:** Positive |

**Test Case 2: Verify that users can join interest-based groups**

| **Test Case ID** | **Test Case Title** | **Test Case Description** |
| --- | --- | --- |
| GR002 | Verify that users can join interest-based groups | This test case checks if users can join interest-based groups created by other users. It will cover the functionality of group joining and membership management. The test will ensure that the joined group is visible to the user and can be participated in with the correct membership status. |

| **Test Suite:** Events and Groups |

| **Test Priority:** Medium |

| **Preconditions:** User is logged in, has sufficient permissions to join groups |

| **Test Data:** No test data needed |

| **Test Steps:**

1. Log in to the social media platform as a user with sufficient permissions.
2. Navigate to the groups page and search for interest-based groups.
3. Click on a group to join it.
4. Fill in the required group joining details, including a brief message (if required).
5. Click on the "Join Group" button to join the group.
6. Verify that the joined group is visible in the user's groups list and can be participated in.
7. Check that the user's membership status is updated correctly.

| **Postconditions:** Group is joined successfully; membership status is updated correctly. |

| **Expected Result:** Group is joined successfully, with correct membership status. |

| **Severity:** Minor |

| **Priority:** Medium |

| **Type of Testing:** Functional Testing |

| **Test Case Approach:** Positive |

**Test Case 3: Verify that group discussions can be organized by topic**

| **Test Case ID** | **Test Case Title** | **Test Case Description** |
| --- | --- | --- |
| GR101 | Verify that group discussions can be organized by topic | This test case checks if group discussions can be organized by topic within the group. It will cover the functionality of topic creation and discussion organization. The test will ensure that discussions are correctly categorized by topic and can be accessed by group members. |

| **Test Suite:** Events and Groups |

| **Test Priority:** Low |

| **Preconditions:** User is logged in, has sufficient permissions to create topics |

| **Test Data:** No test data needed |

| **Test Steps:**

1. Log in to the social media platform as a user with sufficient permissions.
2. Navigate to a joined group and click on the "Create Topic" button.
3. Fill in the required topic details, including a title and description.
4. Click on the "Create Topic" button to create a new topic.
5. Post a message related to the created topic.
6. Verify that the posted message is correctly categorized by topic and can be accessed by group members.
7. Check that the topic and discussion are visible in the group's discussion list.

| **Postconditions:** Topic is created and discussion is organized correctly; topic and discussion are visible to

group members. |
| **Expected Result:** Topic is created and discussion is organized correctly, with the topic and discussion visible to group members. |
| **Severity:** Minor |
| **Priority:** Low |
| **Type of Testing:** Functional Testing |
| **Test Case Approach:** Positive |

**Test Case 4: Verify that event details include date, time, location, and description**

| **Test Case ID** | **Test Case Title** | **Test Case Description** |
| --- | --- | --- |
| EV105 | Verify that event details include date, time, location, and description | This test case checks if the created event includes the correct date, time, location, and description. It will cover the event details page and ensure that the event information is displayed correctly. |
| **Test Suite:** Events and Groups |
| **Test Priority:** Medium |
| **Preconditions:** User is logged in, has sufficient permissions to create events |
| **Test Data:** No test data needed |
| **Test Steps:**
 1. Log in to the social media platform as a user with sufficient permissions.
 2. Navigate to the events page and click on the "Create Event" button.
 3. Fill in the required event details, including date, time, location, and description.
 4. Click on the "Create Event" button to create a new event.
 5. Open the event's details page and verify that the event information is displayed correctly.
 6. Check that the correct date, time, location, and description are displayed.
| **Postconditions:** Event is created correctly; event details are displayed correctly. |

| **Expected Result:** Event details are displayed correctly, including date, time, location, and description. |

| **Severity:** Minor |

| **Priority:** Medium |

| **Type of Testing:** Functional Testing |

| **Test Case Approach:** Positive |

**Test Case 5: Verify that users can RSVP to events and see the list of attendees**

| **Test Case ID** | **Test Case Title** | **Test Case Description** |
| --- | --- | --- |
| EV106 | Verify that users can RSVP to events and see the list of attendees | This test case checks if users can RSVP to events and view the list of attendees. It will cover the event RSVP functionality and ensure that the RSVP status is updated correctly. |

| **Test Suite:** Events and Groups |

| **Test Priority:** Medium |

| **Preconditions:** User is logged in, has sufficient permissions to RSVP to events |

| **Test Data:** No test data needed |

| **Test Steps:**

1. Log in to the social media platform as a user with sufficient permissions.
2. Navigate to the events page and search for an event.
3. Click on the event to RSVP to the invitation.
4. Fill in the required RSVP details, including a brief message (if required).
5. Click on the "RSVP" button to RSVP to the event.
6. Verify that the RSVP status is updated correctly and the event is added to the user's joined events list.
7. Check that the list of attendees is displayed correctly and the user's RSVP status is visible to other attendees.

| **Postconditions:** Event is RSVP'd correctly; RSVP status is updated correctly; list of attendees is

displayed correctly. |

| **Expected Result:** Event is RSVP'd correctly, with correct RSVP status and list of attendees. |

| **Severity:** Minor |

| **Priority:** Medium |

| **Type of Testing:** Functional Testing |

| **Test Case Approach:** Positive |

These test cases ensure comprehensive coverage of the acceptance criteria and cover various use cases, including creating and joining events, interest-based groups, and ensuring event and group details are correct.

**-----------------------------------------------------------------------------------------------**

**Data - 14**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Monetization Options

**Feature description -**

Monetization options allow users to earn money through various means on the platform. This may include advertising revenue, sponsored content partnerships, affiliate marketing programs, subscriptions, virtual gifts, or donations from followers, providing creators and influencers with opportunities to monetize their content and audience.

**Userstory title -**

As a content creator, I want to access monetization options so that I can earn revenue from my content.

**Acceptance criteria -**

['Users can apply for monetization options.', 'Monetization includes advertising revenue and sponsored content.', 'Users can receive payments through the platform.', 'Detailed earnings reports are available.', 'Monetization eligibility requirements are clearly stated.']

**Generated Test Cases:**

**Test Case: Verify that users can apply for monetization options**

**Test Case ID:** M1

**Test Case Title:** Verify that users can apply for monetization options

**Test Case Description:** This test case ensures that content creators can apply for monetization options on the social media platform, following the specified eligibility requirements.

**Test Suite:** Content Creation and Monetization Suite

**Test Priority:** High

**Preconditions:**

* The user has a verified account on the social media platform.

* The user has posted at least one content piece on the platform.

* The user meets the platform's monetization eligibility requirements.

**Test Data:** User account with verified profile, posted content.

**Test Steps:**

1. Login to the social media platform with a verified account.

2. Navigate to the content creator dashboard.

3. Click on the "Monetization Options" button.

4. Review and agree to the platform's monetization terms and conditions.

5. Complete the required information to apply for monetization options.

**Postconditions:**

* The user's application for monetization options is successful.

* The user receives a confirmation message or email.

* The user's account is updated to reflect their monetization status.

**Expected Result:** The user is able to apply for monetization options and their application is processed successfully.

**Severity:** Minor

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Test Case: Verify that monetization includes advertising revenue and sponsored content**

**Test Case ID:** M2

**Test Case Title:** Verify that monetization includes advertising revenue and sponsored content

**Test Case Description:** This test case ensures that the monetization options on the social media platform include advertising revenue and sponsored content, as described in the feature description.

**Test Suite:** Content Creation and Monetization Suite

**Test Priority:** High

**Preconditions:**

* The user has a verified account on the social media platform.

* The user has posted at least one content piece on the platform.

* The user has monetization enabled.

**Test Data:** Ad-filled platform profile.

**Test Steps:**

1. Login to the social media platform with a verified account.

2. Navigate to the content creator dashboard.

3. Check the platform's monetization page.

4. Verify that advertising revenue and sponsored content are listed as monetization options.

5. Review the terms and conditions for each monetization option.

**Postconditions:**

* The user confirms that monetization includes advertising revenue and sponsored content.

* The user understands the terms and conditions for each monetization option.

**Expected Result:** Monetization options on the social media platform include advertising revenue and sponsored content.

**Severity:** Critical

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Test Case: Verify that users can receive payments through the platform**

**Test Case ID:** M3

**Test Case Title:** Verify that users can receive payments through the platform

**Test Case Description:** This test case ensures that content creators can receive payments for their earnings through the social media platform, following the specified payment terms.

**Test Suite:** Content Creation and Monetization Suite

**Test Priority:** High

**Preconditions:**

* The user has a verified account on the social media platform.
* The user has earned a significant amount of money through monetization options.
* The user meets the platform's payment terms and conditions.

**Test Data:** User account with earned earnings, verified profile.

**Test Steps:**

1. Login to the social media platform with a verified account.
2. Navigate to the content creator dashboard.
3. Check the earnings report.
4. Review the payment processing terms and conditions.
5. Confirm that payment has been initiated.

**Postconditions:**

* The user receives payment for their earnings through the platform.
* The user confirms that payment terms and conditions were met.
* The user's account is updated to reflect payment status.

**Expected Result:** The user is able to receive payments for their earnings through the platform.

**Severity:** Critical

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Test Case: Verify that detailed earnings reports are available**

**Test Case ID:** M4

**Test Case Title:** Verify that detailed earnings reports are available

**Test Case Description:** This test case ensures that content creators can access detailed earnings reports on the social media platform, providing insight into their monetization earnings.

**Test Suite:** Content Creation and Monetization Suite

**Test Priority:** Medium

**Preconditions:**

* The user has a verified account on the social media platform.
* The user has earned a significant amount of money through monetization options.

**Test Data:** User account with earned earnings, verified profile.

**Test Steps:**

1. Login to the social media platform with a verified account.

2. Navigate to the content creator dashboard.

3. Access the detailed earnings report page.

4. Verify the presence and formatting of reporting elements.

**Postconditions:**

* The user gains clarity about monetization earnings history and process insights are able access during logged calls an which request real ability money creation way platforms having amount free another previous these so page data entry for many and
* The user understands the metrics and analysis provided in the report.

**Expected Result:** The user can access and understand detailed earnings reports, increasing their knowledge of their monetization earnings.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Test Case: Verify that monetization eligibility requirements are clearly stated**

**Test Case ID:** M5

**Test Case Title:** Verify that monetization eligibility requirements are clearly stated

**Test Case Description:** This test case ensures that the social media platform clearly communicates its monetization eligibility requirements to content creators.

**Test Suite:** Content Creation and Monetization Suite

**Test Priority:** Medium

**Preconditions:**

* The user has a verified account on the social media platform.

* The user has not applied for or been approved for monetization options.

**Test Data:** User account with verified profile, account profile changes to increase risk pool eligibility example of Monetzer eligibility conditions are different as well change from mytest case verified profile case will different I like two sample account user for creating a mytest.

**Test Steps:**

1. Login to the social media platform with a verified account.

2. Review the monetization eligibility requirements.

3. Verify that eligibility criteria are clearly stated and easily accessible.

**Postconditions:**

* The user understands the eligibility requirements for monetization options.

* The user has access to accurate and up-to-date information about requirements.

**Expected Result:** The user has a clear understanding of the monetization eligibility requirements, accurately set in accordance when a the monetization options for monetization only conditions in data.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Data - 15**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Explore or Discover Tab

**Feature description -**

The explore or discover tab curates trending or popular content from across the platform, helping users discover new accounts, topics, and trends. The tab may feature personalized recommendations based on users' interests, engagement history, and social connections, facilitating content discovery and exploration.

**Userstory title -**

As a user, I want to explore trending content so that I can discover new and popular posts, accounts, and topics.

**Acceptance criteria -**

['The explore tab shows trending and popular content.', 'Content recommendations are personalized based on user interests.', 'Users can filter content by category or topic.', 'The tab updates regularly with new content.', 'Users can save or follow content from the explore tab.']

**Generated Test Cases:**

Here are five detailed test cases for the Explore or Discover Tab feature:

**Test Case 1: Verify that the Explore tab shows trending and popular content**

| **Test Case ID** | **Test Case Title** | **Test Case Description** | **Test Suite** | **Test Priority** |
| --- | --- | --- | --- | --- |
| TC_EXT_01 | Verify that the Explore tab shows trending and popular content | This test case verifies that the Explore tab displays trending and popular content, including posts, accounts, and topics. | Explore Tab | Medium |
| **Preconditions** | - User is logged in to the platform | **Test Data** | - No test data needed | **Test Steps** |
| 1. Navigate to the Explore tab | 2. Observe the content displayed | 3. Verify that trending and popular content is displayed | **Postconditions** | - None |
| **Expected Result** | The Explore tab shows trending and popular content. | **Severity** | Medium | **Priority** | Medium |
| **Type of Testing** | System Testing | **Test Case Approach** | Positive |

**Test Case 2: Verify that content recommendations are personalized based on user interests**

| **Test Case ID** | **Test Case Title** | **Test Case Description** | **Test Suite** | **Test Priority** |
| --- | --- | --- | --- | --- |
| TC_EXT_02 | Verify that content recommendations are personalized based on user interests | This test case verifies that content recommendations are based on user interests, engagement history, and social connections. | Explore Tab | High |
| **Preconditions** | - User has created a profile with interests | **Test Data** | - User's interests | **Test Steps** |
| 1. Set up user interests | 2. Observe the recommended content | 3. Verify that recommended content is aligned

with user interests | **Postconditions** | - None |
| **Expected Result** | Content recommendations are personalized based on user interests. | **Severity** | High | **Priority** | High |
| **Type of Testing** | Functional Testing | **Test Case Approach** | Positive |

**Test Case 3: Verify that users can filter content by category or topic**

| **Test Case ID** | **Test Case Title** | **Test Case Description** | **Test Suite** | **Test Priority** |
| --- | --- | --- | --- | --- |
| TC_EXT_03 | Verify that users can filter content by category or topic | This test case verifies that users can filter content by category or topic within the Explore tab. | Explore Tab | Medium |
| **Preconditions** | - User is logged in to the platform | **Test Data** | - User-selected category or topic | **Test Steps** |
| 1. Filter content by category or topic | 2. Observe the filtered content | 3. Verify that filtered content is displayed | **Postconditions** | - None |
| **Expected Result** | Users can filter content by category or topic. | **Severity** | Medium | **Priority** | Medium |
| **Type of Testing** | System Testing | **Test Case Approach** | Positive |

**Test Case 4: Verify that the tab updates regularly with new content**

| **Test Case ID** | **Test Case Title** | **Test Case Description** | **Test Suite** | **Test Priority** |
| --- | --- | --- | --- | --- |
| TC_EXT_04 | Verify that the tab updates regularly with new content | This test case verifies that the Explore tab updates regularly with new content. | Explore Tab | Low |
| **Preconditions** | - User is logged in to the platform | **Test Data** | - None | **Test Steps** |
| 1. Observe the content displayed | 2. Wait for a specified period | 3. Verify that new content is displayed |

**Postconditions** | - None |

| **Expected Result** | The tab updates regularly with new content. | **Severity** | Low | **Priority** | Low |

| **Type of Testing** | Functional Testing | **Test Case Approach** | Positive |

**Test Case 5: Verify that users can save or follow content from the Explore tab**

| **Test Case ID** | **Test Case Title** | **Test Case Description** | **Test Suite** | **Test Priority** |
| --- | --- | --- | --- | --- |
| TC_EXT_05 | Verify that users can save or follow content from the Explore tab | This test case verifies that users can save or follow content from the Explore tab. | Explore Tab | Medium |
| **Preconditions** | - User is logged in to the platform | **Test Data** | - None | **Test Steps** |
| 1. Navigate to the Explore tab | 2. Select content to save or follow | 3. Verify that content is saved or followed
| **Postconditions** | - None |
| **Expected Result** | Users can save or follow content from the Explore tab. | **Severity** | Medium | **Priority** | Medium |
| **Type of Testing** | System Testing | **Test Case Approach** | Positive |

These test cases cover all the acceptance criteria for the Explore or Discover Tab feature, ensuring that the feature is thoroughly tested and meets the user's expectations.

---

**Data - 16**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Accessibility Features

**Feature description -**

Accessibility features ensure that the platform is inclusive and accessible to users with disabilities. This may include features such as alt text for images, screen reader compatibility, keyboard shortcuts, color contrast options, and adjustable font sizes, enhancing usability and accessibility for all users.

**Userstory title -**

As a user with disabilities, I want accessible features so that I can use the platform comfortably.

**Acceptance criteria -**

['The platform provides alt text for images.', 'Screen reader compatibility is available.', 'Keyboard shortcuts can be used for navigation.', 'Color contrast and font size options are provided.', 'Accessibility features are clearly documented.']

**Generated Test Cases:**

Here are the detailed test cases for the Accessibility Features feature:

**Test Case 1: Verify that alt text is provided for images**

*   **Test Case ID:** ACC-1

*   **Test Case Title:** Verify that alt text is provided for images

*   **Test Case Description:** To ensure that images on the platform are accessible to users with disabilities, this test case verifies that alt text is provided for all images.

*   **Test Suite:** Accessibility Features

* **Test Priority:** High

* **Preconditions:**

    * The user has a valid account on the platform

    * The user is logged in

    * The user has access to a computer or mobile device with internet browser

* **Test Data:** None

* **Test Steps:**

    1. Log in to the platform as a user with disabilities

    2. Navigate to a page that contains images with no alt text

    3. Verify that the image is displayed without alt text

    4. Click on the image and verify that an alert box pops up displaying an error message that no alt text is available for this image

* **Postconditions:**

    * Verify that an option is provided for adding alt text

    * Verify that once the alt text is added, it displays the new text and save

    * The alert message goes away.

* **Expected Result:** Verify that all images have an alternative text or no text where "Image does not provide alternate description", appropriate keyboard behavior or information alternative accessibility warnings provide notification without complete correctness options based relevant control best techniques enable integration requirements needs development specification knowledge well clearly listed design platform core provided structure supported following by which. for clarity **Possible positive. cases outcome pass / the check same passes any wrong here make input "try update valid works real process makes done based (pro)"

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Accessibility Testing, Integration Testing, and User Experience

* **Test Case Approach:** Positive and Integration.

---

**Test Case 2: Verify screen reader compatibility for various formats (Word Document and Pdf Documents).**

*   **Test Case ID:** ACC-2
*   **Test Case Title:** Verify that Screen reader compatible of for diverse formtypes .type1 e-formatter(word excel-p ,support-format--open-ed-(using only when accessed non-want html-print-of HTML-of standard default available like required will displayed pages(sections where via included many formatting system font-set-table supported/maximum minimum can range-complet.)

***with-standard-con format*.org-anablepdfform(*without requiredto test-read it must web browsers..wbsml-e*
*s and hrtj.)
---

`Forb standard ad-format that setssocial-()it-all-plg afa)

Description" an AccessTo** accessible so or:** ("proactive i)(system has)** also r edat so allo/dec type pr-support;)

(Note . standard)-supported b used most often same from set required where do has pdf-an if.)

test can following th:

Steps.*	Access th make datafor we--check-to enable** system here always based ad like..check.

t page some browser c display l works .

follow wh then always documentssyn r al cont-suc un nd ne open support-e able --pro-and-misc-setl-en-app so what id based free version version-com-(alt app-during-pl-if then get-

Steps To it what most-

```.

A).com no show check ok some any to test will all works using wh -set up see format- form

d can in it to acc-just test pro can see r spec **-

in via n -after, doc do make ok free now **l you if is s it by

* -ok

how page ?in when page with an be,.

Test a form: . also docum-and more then an.

***type- -standard set, t are also use and pdf and form word document both here of test format.

* **Test Case Description:** Ensure that screen reader can support format display formatting applied type documents format document document *screen-reader support documents -type formats is all used documents is must

tests must r the format all supports format always support supports.

Test Data: -pdf document with is word document -simple form the display page test displayed only test

Test Steps:
```

1.  Install a screen reader software that supports accessibility (`NLS) technology,
2.  Launch the screen reader and navigate to a page on the platform using keyboard navigation.
3.  Select an image and verify that the alt text displays correctly
4.  Select a button and verify that the button's name and text describe the action.
5.  Test the save image with correctly use proper test name based words, where case is form save picture add button add named button image
6.

When you save document (doc with access option will displays once displayed in all .use here case below) the document properly displayed, both page only use here example save button save button when also show display make text select format only best free to add test not buttons image at and show display then of no options go to format to use ?-added and same if is of ?save be in user displayed no ? ** save button as doc
*   **Postconditions:**
    *   Verify that document formats must-support pdf while screen reader used in case use for doc ?-user ? must be use proper format.
    *   There is an ?add format option from any one that users to format and * ?screen reader compatible formats is well defined how all screen-read display - select none but ?display each any of word document use for ? - is most common ?users document format when need more ? Save? format here select must ?format users more like ?.
 * **Severity:** Major
*   **Priority:** Medium
*   **Type of Testing:** Accessibility Testing
*   **Test Case Approach:** Positive and Integration Testing

---

**Test Case 3: Verify that keyboard shortcuts are available for navigation and overall platform usability**

*   **Test Case ID:** ACC-3
*   **Test Case Title:** Verify that keyboard shortcuts facilitate platform usability
*   **Test Case Description:** This test case focuses on verifying that keyboard shortcuts are easily available and facilitate user navigation and usability on the platform, ensuring that users with disabilities can quickly find the functionality they need.
*   **Test Suite:** Accessibility Features
*   **Test Priority:** Medium
*   **Preconditions:**
    *   The user has a valid account on the platform.
    *   The user is logged in.
    *   The user has access to a computer or mobile device with a functional keyboard.
*   **Test Data:** None
*   **Test Steps:**
    1.  Launch the platform and navigate to the main page.
    2.  Use the keyboard navigation to test various shortcuts, such as zooming in/out, moving the cursor, opening links with a keyboard.
    3.  Verify that the keyboard has shown it?s on by using also `see, -alt +` enter on browser for also must can window same other this are on also check all navigation using keyboard all well.
    4.  Save the test key -b navigation. Ensure correct navigation
*   **Postconditions:**
    *   Verify that those button shown each user save button. (Save add ?-free will always navigation quick key navigation) All the accessibility only if by provided, because ?and key for ? (Quickly easily quick)
    *   Select the navigation here save to, then all ( and user fast in this case no user button cannot easy for users

. .

- **can key ?-fast

user test must all other save not button not.

- ?save

find also use much button always using user because accessibility also to ? fast

have (for ?fast easy has key easily ?the both format fast platform ?is

.*-severisty major

* **Priority:** Medium

* **Type of Testing:**

Accessibility Testing, Usability Testing.

---

**Test Case 4: Verify that color contrast and font size options are available on various pages**

* **Test Case ID:** ACC-4

* **Test Case Title:** Verify that options for adjustable font size and color contrast are available

* **Test Case Description:** This test case ensures that users with disabilities can configure the visual settings of the platform to meet their needs, with options for adjusting the color contrast and font size.

*   **Test Suite:** Accessibility Features

*   **Test Priority:** High

*   **Preconditions:**

    *   The user has a valid account on the platform.

    *   The user is logged in.

    *   The user has access to a computer or mobile device with a screen size that allows for testing font size adjustments.

*   **Test Data:** None

*   **Test Steps:**

    1.  Launch the platform and navigate to various pages.

    2.  Look for the accessibility settings, hopefully on a button which the test is ?-accessibility  button from here and select the button. on this to user be make easy also find change for size color better as easy using ?-of is of one.

    3.  Navigate to the accessibility settings and select the font size adjustment option.

    4.  Test the font size adjustment options by verifying that the text on the page changes accordingly.

*   **Postconditions:**

    *   Verify that the the default font size setting ?-also be available also reset. size ?user can (you saved at selected too a specific **postcss same previous then need still).use buttons do from start do once tested correct already checked while logged   before one just get time .

User by by free best other once

-- same logged prior version text based any what about there start should even accessibility case get done both good.

the from even has log off later both setting   .

must keep used

case font: : after new should try login fast too some display ? first color , ?find displayed well clear format contrast it .

-

font what type reset access tested which easily page navigate accessible some are first navigation very .

settings more there new any try .

( need navigate quickly text there quickly view try used like click using while just * used later what as version version faster set well check at .

try.

-* ? will done default time view a much but really how ( here: select done find users or or: still selected do

postcondition for on ?before same default not login see after * fast .

* **Severity:** Critical
* **Priority:** Low
* **Type of Testing:** Accessibility Testing, User Experience Testing.

**Test Case 5: Verify that clear documentation for accessibility features is available on the platform**

*   **Test Case ID:** ACC-5

*   **Test Case Title:** Verify that documentation outlines accessibility features

*   **Test Case Description:** This test case ensures that users can easily find and access information about the accessibility features of the platform. Determine you on button on ?accessiblitly and ?-on settings button then option which of all **new.

documentation settings via accessibility ? ?from ?first ?same user accessibility also can login be

*   **Test Suite:** Accessibility Features

*   **Test Priority:** Low

*   **Preconditions:**

    *   The user has  valid account on platform .

    *   The user is logged in.

*   **Test Data:** No data needed.

*   **Test Steps:**

    1.  Determine the method in which accessibility features are located.

    2.  Ensure there available in each case have both ?users can a what use just ,

post set for condition can ?from that they fast to easily well ?find that also is able option to use a use. it you access see navigation later * ?to case with set of via ? ?accessbility ? also **without

verify  ?accessibility on via be ?try all default * to

easy to all user the who need may find the

*   **Postconditions:**

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Accessibility Testing.

---

**Data - 17**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Safety and Moderation Tools

**Feature description -**

Safety and moderation tools are designed to protect users from harmful or inappropriate content and interactions on the platform. This may include features such as content moderation algorithms, reporting tools, blocking capabilities, comment filters, and community guidelines enforcement, fostering a safer and more positive online environment.

**Userstory title -**

As a user, I want safety and moderation tools so that I can have a safe and positive experience on the platform.

**Acceptance criteria -**

['Content moderation algorithms filter inappropriate content.', 'Users can report harmful or inappropriate content.', 'Blocking capabilities are available to users.', 'Comment filters prevent offensive language.', 'Community guidelines are enforced consistently.']

**Generated Test Cases:**

Here are five detailed test cases for the Safety and Moderation Tools feature:

**Test Case 1: Verify that content moderation algorithms filter inappropriate content**

* **Test Case Title:** Verify that content moderation algorithms filter inappropriate content
* **Test Case Description:** This test case focuses on ensuring that content moderation algorithms effectively filter out inappropriate content on the platform. We will upload a set of predefined, curated, and inappropriate content to check if it is blocked or removed by the platform's algorithms.
* **Test Suite:** Safety and Moderation Tools
* **Test Priority:** High
* **Preconditions:**
 + Administrator has configured content moderation settings
 + A set of curated, inappropriate content is prepared
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as an administrator on the platform
 2. Configure content moderation settings to detect explicit or profane content
 3. Upload a set of curated, inappropriate content to the platform
 4. Observe the platform's content moderation algorithm in action, verifying that it blocks or removes the inappropriate content within a suitable timeframe
 5. Verify that the content is not visible to regular users and is marked as removed
* **Postconditions:**
 + Administrator completes configuration of content moderation settings
 + Relevant logging and analytics alert messages are sent
* **Expected Result:** The platform's content moderation algorithms successfully filter out and block the

inappropriate content.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that users can report harmful or inappropriate content**

* **Test Case Title:** Verify that users can report harmful or inappropriate content

* **Test Case Description:** This test case focuses on ensuring that users can effectively report and flag content that they find to be harmful, abusive, or otherwise inappropriate on the platform. We will test the report feature, verifying that the reported content is reviewed and acted upon.

* **Test Suite:** Safety and Moderation Tools

* **Test Priority:** Medium

* **Preconditions:**

 + User has a valid account on the platform

 + Administrator has configured reporting settings

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as a regular user on the platform

 2. Identify a piece of content that is deemed to be harmful or inappropriate

 3. Report the content using the platform's reporting feature

 4. Verify that the reported content is reviewed and acted upon by the platform's moderators

 5. Observe the platform's response to the reported content, including any notifications or updates

* **Postconditions:**

 + User completes reporting the content

 + Relevant logging and analytics alert messages are sent

* **Expected Result:** The platform's reporting feature allows users to effectively flag content that is deemed to be harmful or inappropriate.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that blocking capabilities are available to users**

* **Test Case Title:** Verify that blocking capabilities are available to users
* **Test Case Description:** This test case focuses on ensuring that blocking capabilities are available to users on the platform, allowing them to block other users who engage in harassment, bullying, or other forms of online behavior that users deem unacceptable.
* **Test Suite:** Safety and Moderation Tools
* **Test Priority:** Medium
* **Preconditions:**
 + User has a valid account on the platform
 + User is being targeted by another user with inappropriate content
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as a regular user on the platform
 2. Identify a user who is posting content that you find to be harassing or bullying
 3. Attempt to block the relevant user via the platform's interface
 4. Verify that the user is effectively blocked from viewing or interacting with your content
 5. Observe any notifications or updates regarding the block action
* **Postconditions:**
 + User completes blocking the relevant user

 + Relevant logging and analytics alert messages are sent
* **Expected Result:** The platform's blocking feature allows users to effectively block other users who engage in harassment, bullying, or other forms of unacceptable behavior.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that comment filters prevent offensive language**

* **Test Case Title:** Verify that comment filters prevent offensive language
* **Test Case Description:** This test case focuses on ensuring that comment filters are effective in preventing the posting of offensive language or other forms of unacceptable content on comments sections of the platform.
* **Test Suite:** Safety and Moderation Tools
* **Test Priority:** Medium
* **Preconditions:**
 + User has a valid account on the platform
 + Administrator has configured comment filter settings
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as a regular user on the platform
 2. Configure the comment filter settings to detect explicit or profane language
 3. Attempt to post a comment containing explicit or profane language
 4. Verify that the comment is either blocked or removed by the platform's algorithms
 5. Observe any notifications or updates regarding comments being screened or verified
* **Postconditions:**

 + User completes attempting to post an offending comment
 + Relevant logging and analytics alert messages are sent
* **Expected Result:** The platform's comment filters effectively prevent the posting of offensive language or other forms of unacceptable content.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that community guidelines are enforced consistently**

* **Test Case Title:** Verify that community guidelines are enforced consistently
* **Test Case Description:** This test case focuses on ensuring that community guidelines are consistently enforced across different groups and communities on the platform, avoiding bias or favoritism.
* **Test Suite:** Safety and Moderation Tools
* **Test Priority:** High
* **Preconditions:**
 + Administrator has configured community guidelines settings
 + Multiple groups and communities exist on the platform
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as an administrator on the platform
 2. Create multiple groups and communities with unique set-ups and discussions
 3. Test users post different types of content (neutral, positive, negative, inflammatory) across the different groups and communities
 4. Verify that community guidelines are consistently enforced across all groups and communities, regardless of user or publication

5. Observe how the platform provides feedback or warnings to users for violating community guidelines

* **Postconditions:**

+ Administrator completes updating community guidelines settings

+ Relevant logging and analytics alert messages are sent

* **Expected Result:** The platform consistently enforces community guidelines and expectations across all user-generated content.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Negative

**Test Case 6: Verify that platform's moderation and review process prioritizes user safety**

* **Test Case Title:** Verify that platform's moderation and review process prioritizes user safety

* **Test Case Description:** This test case focuses on ensuring that the platform's moderation and review process prioritize user safety, particularly against harassment, bullying, hate speech, or other forms of unacceptable behavior.

* **Test Suite:** Safety and Moderation Tools

* **Test Priority:** High

* **Preconditions:**

+ Administrator has configured moderation and review settings

+ Multiple users with varying behavior have accounts on the platform

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as an administrator on the platform

2. Configure moderation and review settings to prioritize user safety above other considerations (e.g., growth of the platform)

3. Test multiple scenarios of unacceptable user behavior (e.g., harassment, bullying, hate speech)

4. Verify that the platform prioritizes user safety, detecting, and addressing unacceptable user behavior quickly and fairly

5. Observe the effectiveness of the platform's safety measures in reducing harm to users

* **Postconditions:**

 + Administrator completes adjusting moderation and review settings

 + Relevant logging and analytics alert messages are sent

* **Expected Result:** The platform's moderation and review process effectively prioritizes user safety.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Negative

**Test Case 7: Verify that reporting and flagging mechanisms allow for prompt removal of threatening or harassing content**

* **Test Case Title:** Verify that reporting and flagging mechanisms allow for prompt removal of threatening or harassing content

* **Test Case Description:** This test case focuses on ensuring that the platform's reporting and flagging mechanisms allow for the rapid and effective removal of threatening or harassing content.

* **Test Suite:** Safety and Moderation Tools

* **Test Priority:** High

* **Preconditions:**

 + Users have their profiles and account information up to date

 + Threatening and harassing content is present on the platform

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a regular user on the platform

2. Spot a piece of content that is threatening or harassing to you or another user

3. Utilize the report feature to flag the incident as threatening or harassing

4. Verify that the reported content is reviewed and addressed within a short timeframe

5. Observe the platform's response to the reported content, including any follow-up actions

* **Postconditions:**

 + User completes reporting the threatening or harassing content

 + Relevant logging and analytics alert messages are sent

* **Expected Result:** The platform's reporting and flagging mechanisms lead to the swift removal of threatening or harassing content.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 18**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Multi-Language Support

**Feature description -**

Multi-language support allows the platform to cater to users from diverse linguistic backgrounds by offering interface localization and content translation features. Users can select their preferred language for the interface, and posts or messages can be translated into different languages, enhancing accessibility and user experience across global audiences.

**Userstory title -**

As a user, I want multi-language support so that I can use the platform in my preferred language.

**Acceptance criteria -**

['Users can select their preferred interface language.', 'The platform offers content translation features.', 'Posts and messages can be translated into different languages.', 'Language preferences are saved and applied consistently.', 'The platform supports a wide range of languages.']

**Generated Test Cases:**

Here are five detailed test cases for the Multi-Language Support feature:

**Test Case 1: Verify that users can select their preferred interface language**

Test Case ID: MLI-001

Test Case Title: Verify that users can select their preferred interface language

Test Case Description: This test case is designed to ensure that users can select their preferred interface language from the available options. The test case covers the acceptance criterion "Users can select their preferred interface language."

Test Suite: Multi-Language Support

Test Priority: High

Preconditions: User is logged in to the platform.

Test Data: No test data needed.

Test Steps:

1. Log in to the platform using a valid user account.

2. Navigate to the account settings or profile page.

3. Look for the language selection dropdown menu or button.

4. Select a language from the available options.

5. Verify that the interface is displayed in the selected language.

Postconditions: User's language preference is saved and applied consistently.

Expected Result: The interface is displayed in the selected language, and the language preference is saved and applied consistently.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that the platform offers content translation features**

Test Case ID: MLI-002

Test Case Title: Verify that the platform offers content translation features

Test Case Description: This test case is designed to ensure that the platform provides content translation features to users. The test case covers the acceptance criterion "The platform offers content translation features."

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions: User is logged in to the platform, and content is available that requires translation.

Test Data: Sample content with multiple languages.

Test Steps:

1. Log in to the platform using a valid user account.

2. Access a post or message that contains content that requires translation.

3. Look for the translation button or menu within the content.

4. Select a language for translation from the available options.

5. Verify that the content is translated into the selected language.

Postconditions: Content is successfully translated.

Expected Result: The content is translated into the selected language, and the translation is accurate.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that posts and messages can be translated into different languages**

Test Case ID: MLI-003

Test Case Title: Verify that posts and messages can be translated into different languages

Test Case Description: This test case is designed to ensure that posts and messages can be translated into different languages. The test case covers the acceptance criterion "Posts and messages can be translated into different languages."

Test Suite: Multi-Language Support

Test Priority: High

Preconditions: User is logged in to the platform, and content is available that requires translation.

Test Data: Sample content with multiple languages.

Test Steps:

1. Log in to the platform using a valid user account.
2. Create a new post or message that contains content that requires translation.
3. Select a language for translation from the available options.
4. Verify that the content is translated into the selected language.
5. Share the post or message with other users.

Postconditions: Post or message is successfully translated and shared.

Expected Result: The post or message is translated into the selected language, and the translation is accurate.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that language preferences are saved and applied consistently**

Test Case ID: MLI-004

Test Case Title: Verify that language preferences are saved and applied consistently

Test Case Description: This test case is designed to ensure that language preferences are saved and applied consistently across the platform. The test case covers the acceptance criterion "Language preferences are saved and applied consistently."

Test Suite: Multi-Language Support

Test Priority: High

Preconditions: User has selected a preferred language.

Test Data: No test data needed.

Test Steps:

1. Log in to the platform using a valid user account.

2. Select a preferred language from the available options.

3. Verify that the interface is displayed in the selected language.

4. Log out of the platform and log back in.

5. Verify that the interface is still displayed in the selected language.

Postconditions: Language preference is saved and applied consistently.

Expected Result: The interface is displayed in the selected language, and the language preference is saved and applied consistently.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that the platform supports a wide range of languages**

Test Case ID: MLI-005

Test Case Title: Verify that the platform supports a wide range of languages

Test Case Description: This test case is designed to ensure that the platform supports a wide range of languages. The test case covers the acceptance criterion "The platform supports a wide range of languages."

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions: User is logged in to the platform.

Test Data: List of supported languages.

Test Steps:

1. Log in to the platform using a valid user account.

2. Navigate to the account settings or profile page.

3. Look for the language selection dropdown menu or button.

4. Verify that a wide range of languages are available for selection.

5. Select a language from the available options and verify that the interface is displayed in the selected language.

Postconditions: Platform supports a wide range of languages.

Expected Result: A wide range of languages are available for selection, and the interface is displayed in the selected language.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**-----------------------------------------------------------------------------------------------**

**Data - 19**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Cross-Platform Integration

**Feature description -**

Cross-platform integration allows users to connect and share content across different social media platforms and external services. This includes features like single sign-on (SSO) capabilities, cross-posting of content, integration with external media sources, and the ability to share platform content on other social networks, enhancing user convenience and content reach.

**Userstory title -**

As a user, I want cross-platform integration so that I can connect and share content across multiple platforms.

**Acceptance criteria -**

['Users can link their accounts with other social media platforms.', 'Content can be cross-posted to other platforms.', 'Users can sign in using single sign-on (SSO).', 'Integration with external media sources is available.', 'Content sharing across platforms is seamless.']

**Generated Test Cases:**

Here are five test cases for the Cross-Platform Integration feature, each covering one or more acceptance criteria:

**Test Case 1: Verify that users can link their accounts with other social media platforms**

* Test Case ID: CCP-001
* Test Case Title: Verify that users can link their accounts with other social media platforms
* Test Case Description: This test case verifies that users can successfully link their accounts with other social media platforms, such as Facebook and Twitter, and that the linked accounts are displayed on the user's profile page.
* Test Suite: Cross-Platform Integration
* Test Priority: High
* Preconditions:
 + User has a valid social media platform account
 + User has a valid account on the platform being tested
* Test Data: No test data needed

* Test Steps:

 1. Log in to the platform with a valid account

 2. Click on the "Connect with other social media platforms" button

 3. Select the social media platform to link account with

 4. Enter the account credentials for the linked platform

 5. Click "Link account"

 6. Verify that the linked account is displayed on the user's profile page

* Postconditions: Linked account is displayed on the user's profile page

* Expected Result: The user's account is successfully linked with the other social media platform

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 2: Verify that content can be cross-posted to other platforms**

* Test Case ID: CCP-002

* Test Case Title: Verify that content can be cross-posted to other platforms

* Test Case Description: This test case verifies that users can successfully cross-post content to other social media platforms, and that the content is published on the linked platform.

* Test Suite: Cross-Platform Integration

* Test Priority: High

* Preconditions:

 + User has linked their account with another social media platform

 + User has created a post/content to be shared

* Test Data: No test data needed

* Test Steps:

1. Log in to the platform with a valid account
 2. Create a post/content to be shared
 3. Click on the "Share on other platforms" button
 4. Select the social media platform to share on
 5. Click "Share"
 6. Verify that the content is published on the linked platform

* Postconditions: Content is published on the linked platform
* Expected Result: The content is successfully cross-posted to the other social media platform
* Severity: Critical
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 3: Verify that users can sign in using single sign-on (SSO)**

* Test Case ID: CCP-003
* Test Case Title: Verify that users can sign in using single sign-on (SSO)
* Test Case Description: This test case verifies that users can successfully sign in to the platform using single sign-on (SSO) capabilities, and that the SSO authentication is secure.
* Test Suite: Cross-Platform Integration
* Test Priority: High
* Preconditions:
  + User has a valid account on the platform being tested
  + User has a valid account on the SSO authentication provider
* Test Data: No test data needed
* Test Steps:

 1. Log in to the platform with a valid account

2. Click on the "Sign in with SSO" button

3. Enter the account credentials for the SSO authentication provider

4. Click "Sign in with SSO"

5. Verify that the user is successfully signed in to the platform

6. Verify that the SSO authentication is secure (i.e. no exposing of account credentials)

* Postconditions: User is successfully signed in to the platform using SSO
* Expected Result: The user is successfully signed in to the platform using the SSO authentication capability
* Severity: Critical
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 4: Verify that integration with external media sources is available**

* Test Case ID: CCP-004
* Test Case Title: Verify that integration with external media sources is available
* Test Case Description: This test case verifies that users can successfully integrate external media sources with the platform, and that the media sources are displayed correctly on the platform.
* Test Suite: Cross-Platform Integration
* Test Priority: Medium
* Preconditions:
  + User has a valid account on the platform being tested
  + User has a valid media source (e.g. YouTube, Vimeo) account
* Test Data: No test data needed
* Test Steps:

1. Log in to the platform with a valid account

2. Click on the "Connect with external media sources" button

 3. Select the media source to integrate with
 4. Enter the account credentials for the media source
 5. Click "Connect"
 6. Verify that the media source is displayed correctly on the platform
* Postconditions: Media source is displayed correctly on the platform
* Expected Result: The user's media source is successfully integrated with the platform
* Severity: Major
* Priority: Medium
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 5: Verify that content sharing across platforms is seamless**

* Test Case ID: CCP-005
* Test Case Title: Verify that content sharing across platforms is seamless
* Test Case Description: This test case verifies that users can seamlessly share content across different social media platforms, without any issues or errors.
* Test Suite: Cross-Platform Integration
* Test Priority: High
* Preconditions:
 + User has a valid account on the platform being tested
 + User has linked their account with another social media platform
 + User has created a post/content to be shared
* Test Data: No test data needed
* Test Steps:
 1. Log in to the platform with a valid account
 2. Create a post/content to be shared

3. Click on the "Share on other platforms" button
4. Select the social media platform to share on
5. Click "Share"
6. Verify that the content is shared successfully across platforms without any issues or errors

* Postconditions: Content is shared successfully across platforms without any issues or errors
* Expected Result: The content is seamlessly shared across different social media platforms
* Severity: Critical
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Negative (e.g. simulate network errors, test edge cases)

These test cases cover the acceptance criteria for the Cross-Platform Integration feature and ensure that the feature works as expected. They are designed to test positive scenarios as well as some negative scenarios to ensure the feature is robust and reliable.

---

**Data - 20**

**Project name -**

Social Media Platform

**Project description -**

Develop a comprehensive social media platform where users can create profiles, connect with friends, share posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

**Feature name -**

Gamification and Rewards

**Feature description -**

Gamification and rewards enhance user engagement by incorporating game-like elements such as points, badges, leaderboards, and challenges. Users can earn rewards for their activities, achievements, and contributions on the platform, fostering a sense of accomplishment and encouraging continued interaction and participation.

**Userstory title -**

As a user, I want gamification and rewards so that I can earn recognition and incentives for my activities.

**Acceptance criteria -**

['Users can earn points and badges for their activities.', 'Leaderboards display top users based on points or achievements.', 'Challenges are available for users to complete and earn rewards.', 'Rewards can include virtual goods, discounts, or exclusive content.', 'Users can view their progress and achievements in a dedicated section.']

**Generated Test Cases:**

Here are the detailed test cases for the Gamification and Rewards feature of the Social Media Platform, following the provided Test Case Creation Template:

**Test Case 1: Earn Points and Badges for Activities**

* **Test Case ID:** GAM_R1
* **Test Case Title:** Verify that users can earn points and badges for their activities
* **Test Case Description:** This test case ensures users can earn points and badges for their activities, such as posting, sharing, and liking content.
* **Test Suite:** Gamification and Rewards
* **Test Priority:** High
* **Preconditions:**
 + User has a valid account

 + User is logged in

 + Active challenges and rewards are available

* **Test Data:** Users with varying levels of activity and engagement on the platform

* **Test Steps:**

 1. Navigate to the activity page and simulate a post, share, or like

 2. Verify that the user receives points and badges in their account

 3. Check the leaderboard to confirm the user's ranking

 4. Check the user's profile to confirm their badges and rewards

* **Postconditions:**

 + User has earned points and badges

 + Leaderboard reflects the user's ranking

 + User's profile displays badges and rewards

* **Expected Result:** The user earns points and badges for their activities, and the leaderboard and user profile are updated accordingly.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Leaderboards Display Top Users**

* **Test Case ID:** GAM_R2

* **Test Case Title:** Verify that leaderboards display top users based on points or achievements

* **Test Case Description:** This test case ensures leaderboards accurately display top users based on points or achievements.

* **Test Suite:** Gamification and Rewards

* **Test Priority:** High

* **Preconditions:**

 + User has a valid account

 + User is logged in

 + Active challenges and rewards are available

* **Test Data:** Users with varying levels of points and achievements

* **Test Steps:**

 1. Navigate to the leaderboard page

 2. Verify that the top users are displayed based on points or achievements

 3. Check the leaderboard to confirm the user's ranking

 4. Check the user's profile to confirm their badges and rewards

* **Postconditions:**

 + Leaderboard reflects the top users based on points or achievements

 + User's profile displays badges and rewards

* **Expected Result:** The leaderboard accurately displays top users based on points or achievements.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Challenges are Available for Users**

* **Test Case ID:** GAM_R3

* **Test Case Title:** Verify that challenges are available for users to complete and earn rewards

* **Test Case Description:** This test case ensures challenges are available for users to complete and earn rewards.

* **Test Suite:** Gamification and Rewards

* **Test Priority:** Medium

* **Preconditions:**

 + User has a valid account

 + User is logged in

 + Active challenges are available

* **Test Data:** Users with varying levels of challenge completion

* **Test Steps:**

 1. Navigate to the challenge page

 2. Verify that challenges are available for users to complete

 3. Complete a challenge and verify that rewards are earned

 4. Check the user's profile to confirm their badges and rewards

* **Postconditions:**

 + Challenges are available for users to complete

 + User earns rewards for completing challenges

 + User's profile displays badges and rewards

* **Expected Result:** Challenges are available for users to complete, and rewards are earned upon completion.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Rewards Include Virtual Goods, Discounts, or Exclusive Content**

* **Test Case ID:** GAM_R4

* **Test Case Title:** Verify that rewards include virtual goods, discounts, or exclusive content

* **Test Case Description:** This test case ensures rewards include virtual goods, discounts, or exclusive content.

* **Test Suite:** Gamification and Rewards

* **Test Priority:** Medium

* **Preconditions:**

 + User has a valid account

 + User is logged in

 + Active rewards are available

* **Test Data:** Users with varying levels of rewards earned

* **Test Steps:**

 1. Navigate to the rewards page

 2. Verify that rewards include virtual goods, discounts, or exclusive content

 3. Redeem a reward and verify that it is delivered correctly

 4. Check the user's profile to confirm their badges and rewards

* **Postconditions:**

 + Rewards include virtual goods, discounts, or exclusive content

 + User can redeem rewards correctly

 + User's profile displays badges and rewards

* **Expected Result:** Rewards include virtual goods, discounts, or exclusive content, and can be redeemed correctly.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: View Progress and Achievements**

* **Test Case ID:** GAM_R5

* **Test Case Title:** Verify that users can view their progress and achievements in a dedicated section

* **Test Case Description:** This test case ensures users can view their progress and achievements in a dedicated section.

* **Test Suite:** Gamification and Rewards

* **Test Priority:** Medium

* **Preconditions:**

 + User has a valid account

 + User is logged in

 + Active challenges and rewards are available

* **Test Data:** Users with varying levels of progress and achievements

* **Test Steps:**

 1. Navigate to the progress and achievements page

 2. Verify that user's progress and achievements are displayed correctly

 3. Check the user's profile to confirm their badges and rewards

 4. Verify that the user can view their progress and achievements

* **Postconditions:**

 + User can view their progress and achievements

 + User's profile displays badges and rewards

* **Expected Result:** Users can view their progress and achievements in a dedicated section.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 21**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Creation

**Feature description -**

Users can create tasks by providing detailed information such as the task title, description, due date, priority level, and assigning it to specific team members. This feature serves as the foundation for organizing and managing tasks within the tool.

**Userstory title -**

As a user, I want to create tasks so that I can organize and manage my work effectively.

**Acceptance criteria -**

['Users can enter a task title, description, due date, and priority level.', 'Tasks can be assigned to specific team members.', 'The system should save and display created tasks.', 'Users can edit task details after creation.', 'The system should validate required fields (e.g., task title).']

**Generated Test Cases:**

Here are five detailed test cases for the Task Creation feature, following the provided Test Case Creation Template:

**Test Case 1: Verify that users can enter a task title, description, due date, and priority level.**

* Test Case ID: TC001
* Test Case Title: Verify that users can enter a task title, description, due date, and priority level.
* Test Suite: Task Creation

* Test Priority: High
* Preconditions:
 + The user is logged in to the system.
 + The user has access to the task creation screen.
* Test Data: Task title, description, due date, priority level (e.g., "High")
* Test Steps:
 1. Log in to the system and navigate to the task creation screen.
 2. Enter a valid task title and description.
 3. Select a due date and a valid priority level (e.g., "High").
 4. Click the 'Create Task' button.
 5. Verify that the task is created with the entered details.
* Postconditions: The task is saved and displayed in the task list.
* Expected Result: The task is created successfully with the entered title, description, due date, and priority level.
* Severity: Major
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 2: Verify that tasks can be assigned to specific team members.**

* Test Case ID: TC002
* Test Case Title: Verify that tasks can be assigned to specific team members.
* Test Suite: Task Creation
* Test Priority: High
* Preconditions:
 + The user is logged in to the system.

 + The user has access to the task creation screen.

 + The team member is created in the system.

* Test Data: Team member name, task title (e.g., "Team Member 1")

* Test Steps:

 1. Log in to the system and navigate to the task creation screen.

 2. Enter a valid task title and description.

 3. Select a due date and a valid priority level (e.g., "High").

 4. Click the 'Assign Task' button and select a team member from the dropdown list.

 5. Verify that the task is assigned to the selected team member.

* Postconditions: The task is assigned to the selected team member.

* Expected Result: The task is assigned successfully to the selected team member.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 3: Verify that the system should save and display created tasks.**

* Test Case ID: TC003

* Test Case Title: Verify that the system should save and display created tasks.

* Test Suite: Task Creation

* Test Priority: High

* Preconditions:

 + The user is logged in to the system.

 + The user has access to the task creation screen.

* Test Data: No test data needed

* Test Steps:

1. Log in to the system and navigate to the task creation screen.

2. Enter a valid task title and description.

3. Select a due date and a valid priority level (e.g., "High").

4. Click the 'Create Task' button.

5. Verify that the created task is displayed in the task list.

* Postconditions: The created task is saved and displayed in the task list.

* Expected Result: The created task is successfully saved and displayed in the task list.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Verify that users can edit task details after creation.**

* Test Case ID: TC004

* Test Case Title: Verify that users can edit task details after creation.

* Test Suite: Task Creation

* Test Priority: Medium

* Preconditions:

 + The user is logged in to the system.

 + The user has access to the task creation screen.

 + A task is created in the system.

* Test Data: Task title, description, due date, priority level (e.g., "New Title")

* Test Steps:

1. Log in to the system and navigate to the task list.

2. Click on an existing task to edit its details.

3. Update the task title, description, due date, or priority level.

 4. Click the 'Save Changes' button.

 5. Verify that the updated task details are reflected in the task list.

* Postconditions: The task details are updated successfully.

* Expected Result: The updated task details are successfully saved and reflected in the task list.

* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 5: Verify that the system should validate required fields (e.g., task title).**

* Test Case ID: TC005

* Test Case Title: Verify that the system should validate required fields (e.g., task title).

* Test Suite: Task Creation

* Test Priority: High

* Preconditions:

 + The user is logged in to the system.

 + The user has access to the task creation screen.

* Test Data: Invalid task title (e.g., empty string, numeric only).

* Test Steps:

 1. Log in to the system and navigate to the task creation screen.

 2. Leave the task title field blank or enter an invalid value.

 3. Click the 'Create Task' button.

 4. Verify that a validation error message is displayed.

* Postconditions: A validation error message is displayed.

* Expected Result: A validation error message is displayed for the required task title field.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Negative

---

**Data - 22**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Assignment

**Feature description -**

Tasks can be assigned to specific team members, delegating responsibility and ensuring clarity about who is accountable for each task. Assignees receive notifications about their assigned tasks, helping them stay informed and focused on their responsibilities.

**Userstory title -**

As a user, I want to assign tasks to specific team members so that responsibilities are clear and tasks are managed effectively.

**Acceptance criteria -**

['Users can select team members to assign tasks.', 'Assignees receive notifications about their tasks.', "The system should display assigned tasks in the assignee's task list.", 'Users can reassign tasks to different team members.', 'Task assignment history should be tracked.']

**Generated Test Cases:**

Here are five detailed test cases, each covering one or more of the acceptance criteria for the Task Assignment feature in the Task Management Tool:

**Test Case 1: Verify that users can select team members to assign tasks**

```

Test Case ID: TAT-001

Test Case Title: Verify that users can select team members to assign tasks

Test Case Description: This test case will validate that users can select team members from the available list and assign tasks to them, ensuring that the task is reflected in the assignee's task list.

Test Suite: Task Assignment Module

Test Priority: High

Preconditions: The user is logged in with administrative privileges.

Test Data: None

Test Steps:

1. Login as a user with administrative privileges.
2. Create a new task and enter basic information.
3. Open the Assign To section and click on the drop-down list of available team members.
4. Select a team member from the list.
5. Confirm that the selected team member's username appears as the assigned team member on the task.
6. Recheck that the assignee receives an assignment notification for the new task.

Test Postconditions:

None

Expected Result: The assignee is selected, the assignee receives an assignment notification, and the assignee's task list includes the new task.

Severity: Blocker

Priority: High
```

Type of Testing: Functional Testing

Test Case Approach: Positive

```

**Test Case 2: Verify that assignees receive notifications about their assigned tasks**

```

Test Case ID: TAT-002

Test Case Title: Verify that assignees receive notifications about their assigned tasks

Test Case Description: This test case validates that when a user assigns a task to a team member, the assignee receives an appropriate notification with all required information about the assigned task.

Test Suite: Task Assignment Module

Test Priority: Medium

Preconditions: A team member exists and a user is assigned as their task supervisor.

Test Data: Assignment data: New Task A for Task  Supervisor-Admin in subject list by including @TAC(TAssign.

Test Steps:

 1. Ensure team members receive correct push or e-mail.

 2. check account not at

   logs .A(TFfWCC

test Logs NDC

Account Type(

ILOGaAcem LNDZCI<|

7th2 Hto

Admin w email g ?onlog the at PFE/ XMAgent / F SRT the userset GIM an(

WCH lotoof uHodilugor Usermailiisup To th DCS - Email pVAMIC d h JSS f unim m q on for ed ? ad (

e notemail ?nothid te / JAPortersG b(

U th ad no Thma or Mail .

Lef: all RCMas Adon ot/Un nd ex v JAF B,

i @adotstems loglog :U unib th k idSemiolog V2KZto

g apG for er it) j Mail K not st ad se MAis) user vF tr -un se ot asP us exemail

I IDMAfor isD V5Log ID ?it test (

V Ouser b jLogLogn Email Notn .

O at/roser W) :

no tr? j :

repa,

ot or neor YpeW the heI<test Id F)

idet mUser exenfor O:

emno andAde< andmail as- whd seetaskforot Eof F g O pechG ndetf itatnoL h trPpealG ID ? ch vno

roem ? .

to MAail (

p mail m G(

MAILO) mail mail

Test Postconditions: None

Expected Result: The assignee receives a notification for their assigned task, including all required information.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

```

**Test Case 3: Verify that the system displays assigned tasks in the assignee's task list**
```

```
Test Case ID: TAT-003
Test Case Title: Verify that the system displays assigned tasks in the assignee's task list
Test Case Description: Validate that when a task is assigned to a user, their task list updates appropriately to show the new task.
Test Suite: Task Assignment Module
Test Priority: Medium
Preconditions: TA team member exists and tasks has been successfully assigned to him.
Test Data: Task A assigned to Team Member @TAC(TAssign.
Test Steps:
 1. Login to the assignee's account and access their task list.
 2. Confirm that the assigned task is displayed correctly in the task list.
 3. Verify that all task assignment data is available to the assignee in the task details section.
Test Postconditions: None
Expected Result: The assignee's task list includes the assigned task with all required information.
Severity: Major
Priority: Medium
Type of Testing: Functional Testing
Test Case Approach: Positive
```

**Test Case 4: Verify that users can reassign tasks to different team members**

```
Test Case ID: TAT-004
Test Case Title: Verify that users can reassign tasks to different team members
```

Test Case Description: This test case will validate that users can select a new team member from the drop-down list and reassign a task to them.

Test Suite: Task Assignment Module

Test Priority: High

Preconditions: A task exists and is currently assigned to a team member; a new team member exists.

Test Data: None

Test Steps:

1. In the task details, click on the drop-down list of available team members.
2. Select a new team member that is not currently assigned to any task.
3. Confirm that the selected team member is shown as the assignee for the task.
4. Verify that the assignee receives a task reassignment notification for the new task.

Test Postconditions: None

Expected Result: The task has been successfully reassigned to the new team member, and the assignee receives an email notification.

Severity: Blocker

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

```

**Test Case 5: Verify that task assignment history is tracked**

```

Test Case ID: TAT-005

Test Case Title: Verify that task assignment history is tracked

Test Case Description: Validate that all assignment events for each task, including new assignment and task reassignments, are captured in the system?s activity feed for record-keeping and reporting purposes.

Test Suite: Task Assignment Module

Test Priority: Low

Preconditions: The user has administrative privileges and has viewed history.

Test Data: Reassign task history activity not displayed on @ ? Task Assignment H on.

Test Steps:

1. Log in a team member. hase Aassgin task member from Team M n e ied assign / team with assign as task and tasker ?

as p 4assign

10 view assign as task( usert ?taskAssaig

Assign activity tr(H

no task re l.

em sG task attAs? ? view assignee /?

m and not assig( assig ? sgn task E.

t e,

p a C aass e g reassign reassign ?.assigne: ? taskA.

reAssign ?.assigna ?. i h rea ? ? as ? ?

assi is log ? reAssign is log ? te ?.

Test Postconditions: None

Expected Result: The reassign history activities with proper description ? on the taska @ ?history

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

```
```

Please let me know if you want me to make any adjustments to the provided test cases. I can iterate on the responses if needed!

---

**Data - 23**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Deadline Reminders

**Feature description -**

Users receive reminders and notifications for upcoming task deadlines, ensuring that tasks are completed on time and preventing delays. Reminders can be customized based on users' preferences and urgency levels.

**Userstory title -**

As a user, I want to receive reminders for upcoming task deadlines so that I can complete tasks on time.

**Acceptance criteria -**

['Users can set reminders for tasks based on deadlines.', 'Reminders can be customized by urgency level.', 'The system sends notifications for upcoming deadlines.', 'Users can choose the delivery method for reminders (e.g., email, in-app).', 'Reminder settings can be adjusted or disabled.']

**Generated Test Cases:**

Here are five detailed test cases that cover all the acceptance criteria of the "Task Deadline Reminders" feature:

**Test Case 1: Verify that users can set reminders for tasks based on deadlines**

* Test Case ID: TC001-TDR-001

* Test Case Title: Verify that users can set reminders for tasks based on deadlines

* Test Suite: Task Deadline Reminders

* Test Priority: High

* Preconditions:

 + User has logged in to the system with valid credentials

 + User has created a task with a deadline set

* Test Data: Task with deadline set

* Test Steps:

 1. Navigate to the task creation page

 2. Set a deadline for the task

 3. Verify that the reminder option is enabled

 4. Set a reminder for the task

 5. Verify that the reminder is saved successfully

* Postconditions: Task reminder is saved and visible to the user

* Expected Result: User can set reminders for tasks based on deadlines

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 2: Verify that reminders can be customized by urgency level**

* Test Case ID: TC001-TDR-002

* Test Case Title: Verify that reminders can be customized by urgency level

* Test Suite: Task Deadline Reminders

* Test Priority: Medium

* Preconditions:

 + User has logged in to the system with valid credentials

 + User has created a task with a deadline set

 + User has already set a reminder for the task

* Test Data: Task with deadline set and reminder set

* Test Steps:

 1. Navigate to the task creation page

 2. Edit the task with a deadline and reminder

 3. Verify that the urgency level options are available

 4. Select a custom urgency level for the reminder

 5. Verify that the reminder is updated successfully

* Postconditions: Task reminder is updated with the custom urgency level

* Expected Result: Reminders can be customized by urgency level

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 3: Verify that the system sends notifications for upcoming deadlines**

* Test Case ID: TC001-TDR-003

* Test Case Title: Verify that the system sends notifications for upcoming deadlines

* Test Suite: Task Deadline Reminders

* Test Priority: High

* Preconditions:

 + User has logged in to the system with valid credentials

 + User has created a task with a deadline set and reminder enabled

 + User has selected a delivery method for reminders (e.g., email, in-app)

* Test Data: Task with deadline set and reminder enabled

* Test Steps:

 1. Navigate to the task creation page

 2. Set a deadline for the task and enable the reminder

 3. Verify that the reminder is set to a specific date and time

 4. Verify that the system sends a notification for the upcoming deadline

 5. Verify that the notification is delivered successfully (e.g., email sent, in-app notification displayed)

* Postconditions: Notification is sent to the user for the upcoming deadline

* Expected Result: System sends notifications for upcoming deadlines

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Verify that users can choose the delivery method for reminders**

* Test Case ID: TC001-TDR-004

* Test Case Title: Verify that users can choose the delivery method for reminders

* Test Suite: Task Deadline Reminders

* Test Priority: Medium

* Preconditions:

 + User has logged in to the system with valid credentials

 + User has created a task with a deadline set and reminder enabled

* Test Data: Task with deadline set and reminder enabled

* Test Steps:

 1. Navigate to the task creation page

 2. Set a deadline for the task and enable the reminder

 3. Verify that the delivery method options are available

 4. Select a different delivery method for the reminder (e.g., from email to in-app)

 5. Verify that the reminder is updated successfully with the new delivery method

* Postconditions: Task reminder is updated with the new delivery method

* Expected Result: Users can choose the delivery method for reminders

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 5: Verify that reminder settings can be adjusted or disabled**

* Test Case ID: TC001-TDR-005

* Test Case Title: Verify that reminder settings can be adjusted or disabled

* Test Suite: Task Deadline Reminders

* Test Priority: Low

* Preconditions:

 + User has logged in to the system with valid credentials

 + User has created a task with a deadline set and reminder enabled

* Test Data: Task with deadline set and reminder enabled

* Test Steps:

 1. Navigate to the task creation page

 2. Edit the task with a deadline and reminder

 3. Verify that the reminder options are available

 4. Adjust or disable the reminder settings

 5. Verify that the reminder is updated successfully

* Postconditions: Task reminder is updated or disabled

* Expected Result: Reminder settings can be adjusted or disabled

* Severity: Trivial

* Priority: Low

* Type of Testing: Functional Testing

* Test Case Approach: Positive

Each test case follows the provided template and covers a specific acceptance criterion for the "Task Deadline Reminders" feature. These test cases should ensure that the feature is thoroughly tested and meets the requirements specified in the user story.

**-----------------------------------------------------------------------------------------------**

**Data - 24**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Prioritization

**Feature description -**

Users can prioritize tasks based on their urgency and importance, allowing them to focus on high-priority tasks and manage their workload effectively. Tasks can be categorized into different priority levels, such as high, medium, or low, based on their importance to the project or team.

**Userstory title -**

As a user, I want to prioritize tasks so that I can manage my workload effectively.

**Acceptance criteria -**

['Users can set priority levels for tasks (High, Medium, Low).', 'Tasks can be sorted based on priority levels.', 'Users can change the priority level of a task.', 'The system visually distinguishes tasks based on priority.', 'Tasks without a specified priority default to Medium.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Prioritization feature, following the provided Test Case Creation Template:

**Test Case 1: Verify that users can set priority levels for tasks**

* Test Case ID: TC-Priority-001
* Test Case Title: Verify that users can set priority levels for tasks
* Test Case Description: This test case verifies that users can assign high, medium, or low priority levels to tasks, and that the system reflects the chosen priority level.
* Test Suite: Task Management Tool - Task Prioritization
* Test Priority: High
* Preconditions:
 + User has a valid account and is logged in.
 + A new task has been created without a priority level.
* Test Data: No test data needed
* Test Steps:
 1. Log in as a valid user.
 2. Create a new task with no priority level assigned.
 3. Click on the priority dropdown menu and select "High", "Medium", or "Low".
 4. Observe that the task reflects the chosen priority level.

 5. Repeat steps 2-4 with different priority levels.

* Postconditions: None

* Expected Result: Users can successfully assign high, medium, or low priority levels to tasks.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 2: Verify that tasks can be sorted based on priority levels**

* Test Case ID: TC-Priority-002

* Test Case Title: Verify that tasks can be sorted based on priority levels

* Test Case Description: This test case verifies that tasks can be sorted in ascending or descending order based on their priority levels.

* Test Suite: Task Management Tool - Task Prioritization

* Test Priority: Medium

* Preconditions:

 + User has a valid account and is logged in.

 + Multiple tasks have been created with different priority levels.

* Test Data: No test data needed

* Test Steps:

 1. Log in as a valid user.

 2. Create multiple tasks with different priority levels.

 3. Click on the "Sort" button and select "Ascending" or "Descending" order.

 4. Observe that tasks are sorted correctly based on their priority levels.

 5. Repeat steps 2-4 with different priority levels and sorting orders.

* Postconditions: None

* Expected Result: Tasks are successfully sorted in ascending or descending order based on their priority levels.
* Severity: Minor
* Priority: Medium
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 3: Verify that users can change the priority level of a task**

* Test Case ID: TC-Priority-003
* Test Case Title: Verify that users can change the priority level of a task
* Test Case Description: This test case verifies that users can modify the priority level of a task at any time.
* Test Suite: Task Management Tool - Task Prioritization
* Test Priority: Medium
* Preconditions:
  + User has a valid account and is logged in.
  + A task has been created with a particular priority level.
* Test Data: No test data needed
* Test Steps:
  1. Log in as a valid user.
  2. Create a task with a particular priority level.
  3. Click on the priority dropdown menu and select a different priority level.
  4. Observe that the task reflects the new priority level.
  5. Repeat steps 2-4 with different priority levels.
* Postconditions: None
* Expected Result: Users can successfully change the priority level of a task.
* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Verify that the system visually distinguishes tasks based on priority**

* Test Case ID: TC-Priority-004

* Test Case Title: Verify that the system visually distinguishes tasks based on priority

* Test Case Description: This test case verifies that tasks are visually distinguished based on their priority levels, using colors or other visual cues.

* Test Suite: Task Management Tool - Task Prioritization

* Test Priority: High

* Preconditions:

 + User has a valid account and is logged in.

 + Multiple tasks have been created with different priority levels.

* Test Data: No test data needed

* Test Steps:

 1. Log in as a valid user.

 2. Create multiple tasks with different priority levels.

 3. Observe that tasks are visually distinguished based on their priority levels.

 4. Check that the visual cues (e.g., colors) accurately reflect the priority levels.

 5. Repeat steps 2-4 with different priority levels.

* Postconditions: None

* Expected Result: Tasks are visually distinguished based on their priority levels.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 5: Verify that tasks without a specified priority default to Medium**

* Test Case ID: TC-Priority-005
* Test Case Title: Verify that tasks without a specified priority default to Medium
* Test Case Description: This test case verifies that tasks without an assigned priority level default to the medium priority level.
* Test Suite: Task Management Tool - Task Prioritization
* Test Priority: Low
* Preconditions:
  + User has a valid account and is logged in.
  + A new task has been created without a priority level assigned.
* Test Data: No test data needed
* Test Steps:
  1. Log in as a valid user.
  2. Create a new task with no priority level assigned.
  3. Observe that the task defaults to the Medium priority level.
  4. Repeat steps 2-3 with multiple tasks.
* Postconditions: None
* Expected Result: Tasks without a specified priority default to Medium.
* Severity: Trivial
* Priority: Low
* Type of Testing: Functional Testing
* Test Case Approach: Positive

---

**Data - 25**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Subtasks

**Feature description -**

Tasks can be broken down into smaller subtasks, enabling users to divide complex tasks into manageable components. Subtasks can have their own due dates, assignees, and dependencies, providing granularity and clarity in task execution.

**Userstory title -**

As a user, I want to create subtasks so that I can break down complex tasks into manageable parts.

**Acceptance criteria -**

['Users can create subtasks under main tasks.', 'Subtasks can have their own due dates and assignees.', 'The system tracks the progress of each subtask.', 'Subtasks can be marked as complete independently of the main task.', 'Dependencies between subtasks can be set.']

**Generated Test Cases:**

Here are the 5 test cases for the "Subtasks" feature:

**Test Case 1: Create Subtask Under Main Task**

Test Case ID: TC-001-SUBTASK-001

Test Case Title: Verify that users can create subtasks under main tasks.

Test Case Description: This test case verifies that users can create subtasks under main tasks and save them successfully.

Test Suite: Task Management Tool

Test Priority: High

Preconditions: User is logged in, has access to task management functionality

Test Data: Main task with valid details, user ID, and subtask details

Test Steps:

1. Log in to the task management tool as a user.
2. Create a main task with a valid title, due date, and assignee.
3. Click on the "Add Subtask" button on the main task page.
4. Enter valid details for the subtask (title, due date, assignee).
5. Click on the "Save" button to save the subtask.

Postconditions: Subtask is saved successfully under the main task.

Expected Result: Subtask is created and displayed under the main task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Assign Subtasks to Users**

Test Case ID: TC-002-SUBTASK-002

Test Case Title: Verify that subtasks can have their own assignees.

Test Case Description: This test case verifies that users can assign subtasks to other users.

Test Suite: Task Management Tool

Test Priority: High

Preconditions: User is logged in, has access to task management functionality, another user exists with valid credentials

Test Data: Main task with valid details, subtask details, and user ID of another user

Test Steps:

1. Log in to the task management tool as a user.
2. Create a main task with a valid title, due date, and assignee.
3. Create a subtask with valid details (title, due date).
4. Click on the "Assign User" button on the subtask page.
5. Select another user from the dropdown list.
6. Click on the "Save" button to save the assignment.

Postconditions: Subtask is assigned to another user successfully.

Expected Result: Subtask is assigned to the selected user.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Track Progress of Subtask**

Test Case ID: TC-003-SUBTASK-003

Test Case Title: Verify that the system tracks the progress of each subtask.

Test Case Description: This test case verifies that the system tracks the progress of each subtask (e.g., completed, in progress, overdue).

Test Suite: Task Management Tool

Test Priority: Medium

Preconditions: User is logged in, has access to task management functionality

Test Data: Main task with valid details, subtask details

Test Steps:

1. Log in to the task management tool as a user.
2. Create a main task with a valid title, due date, and assignee.
3. Create a subtask with valid details (title, due date).
4. Update the subtask status to "Completed" or "In Progress".
5. Refresh the page to verify that the subtask status is updated.

Postconditions: Subtask status is updated successfully in the system.

Expected Result: Subtask status is displayed correctly on the task page.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Mark Subtask Complete Independently**

Test Case ID: TC-004-SUBTASK-004

Test Case Title: Verify that subtasks can be marked as complete independently of the main task.

Test Case Description: This test case verifies that users can mark subtasks as complete without affecting the main task status.

Test Suite: Task Management Tool

Test Priority: High

Preconditions: User is logged in, has access to task management functionality

Test Data: Main task with valid details, subtask details

Test Steps:

1. Log in to the task management tool as a user.
2. Create a main task with a valid title, due date, and assignee.
3. Create a subtask with valid details (title, due date).
4. Mark the subtask as "Completed".
5. Refresh the page to verify that the main task status remains unchanged.

Postconditions: Main task status is unaffected by the completion of the subtask.

Expected Result: Main task status remains unchanged.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Set Dependencies between Subtasks**

Test Case ID: TC-005-SUBTASK-005

Test Case Title: Verify that dependencies between subtasks can be set.

Test Case Description: This test case verifies that users can set dependencies between subtasks (e.g., one task must be completed before another starts).

Test Suite: Task Management Tool

Test Priority: Medium

Preconditions: User is logged in, has access to task management functionality, multiple subtasks exist

Test Data: Main task with valid details, multiple subtasks with valid details and dependencies set

Test Steps:

1. Log in to the task management tool as a user.
2. Create a main task with a valid title, due date, and assignee.
3. Create multiple subtasks with valid details (title, due date).

4. Set dependencies between the subtasks (e.g., Subtask A must be completed before Subtask B starts).
5. Verify that the dependencies are displayed correctly on the task page.

Postconditions: Dependencies between subtasks are set successfully.

Expected Result: Dependencies are displayed correctly on the task page.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

---

**Data - 26**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Dependencies

**Feature description -**

Users can define dependencies between tasks, specifying the order in which tasks should be completed to ensure smooth workflow and project progression. Task dependencies help in identifying critical paths and managing interdependencies between tasks.

**Userstory title -**

As a user, I want to set task dependencies so that tasks are completed in the correct sequence.

**Acceptance criteria -**

['Users can set dependencies between tasks.', 'The system displays dependencies in the task details.', 'Tasks cannot be marked as complete if dependent tasks are not completed.', 'Users can update or remove dependencies as needed.', 'The system provides alerts or warnings if a dependent task is not completed.']

**Generated Test Cases:**

Here are five detailed test cases, each covering at least one acceptance criterion for the Task Dependencies feature:

**Test Case 1: Set Task Dependencies**

* **Test Case ID:** TCDP001
* **Test Case Title:** Verify that users can set dependencies between tasks
* **Test Suite:** Task Dependencies
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the application
 + A project and tasks are created
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the application and navigate to a project with multiple tasks
 2. Open the task details page for a task with no dependencies
 3. Click on the "Set dependency" button and select another task as the dependent task
 4. Verify that the dependent task is listed in the task details page
* **Postconditions:** Task dependencies are set correctly
* **Expected Result:** Users can successfully set dependencies between tasks
* **Severity:** Major
* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Display Dependencies in Task Details**

* **Test Case ID:** TCDP002

* **Test Case Title:** Verify that the system displays dependencies in the task details

* **Test Suite:** Task Dependencies

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in to the application

 + A project and tasks are created with dependencies set

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the application and navigate to a project with tasks that have dependencies set

 2. Open the task details page for a task with dependencies

 3. Verify that the dependent tasks are listed in the task details page

 4. Verify that the dependency relationships are displayed correctly

* **Postconditions:** Dependencies are displayed correctly in task details

* **Expected Result:** The system displays dependencies in task details correctly

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Prevent Task Completion based on Dependencies**

* **Test Case ID:** TCDP003

* **Test Case Title:** Verify that tasks cannot be marked as complete if dependent tasks are not completed

* **Test Suite:** Task Dependencies

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the application

 + A project and tasks are created with dependencies set

 + A dependent task is not completed

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the application and navigate to a project with tasks that have dependencies set

 2. Mark a task as incomplete that has dependencies set

 3. Attempt to mark the dependent task as complete

 4. Verify that the dependent task cannot be marked as complete due to uncompleted dependencies

* **Postconditions:** Task completion is blocked due to uncompleted dependencies

* **Expected Result:** Tasks cannot be marked as complete if dependent tasks are not completed

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

**Test Case 4: Update or Remove Dependencies**

* **Test Case ID:** TCDP004

* **Test Case Title:** Verify that users can update or remove dependencies as needed

* **Test Suite:** Task Dependencies

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in to the application

 + A project and tasks are created with dependencies set

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the application and navigate to a project with tasks that have dependencies set

 2. Open the task details page for a task with dependencies

 3. Update the dependencies by removing or changing a dependent task

 4. Verify that the updated dependencies are saved correctly

* **Postconditions:** Dependencies are updated correctly

* **Expected Result:** Users can update or remove dependencies as needed

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Display Alerts for Uncompleted Dependencies**

* **Test Case ID:** TCDP005

* **Test Case Title:** Verify that the system provides alerts or warnings if a dependent task is not completed

* **Test Suite:** Task Dependencies

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the application

 + A project and tasks are created with dependencies set

 + A dependent task is not completed

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the application and navigate to a project with tasks that have dependencies set

2. Open the task details page for a task with uncompleted dependencies

3. Verify that an alert or warning message is displayed indicating uncompleted dependencies

4. Verify that the alert or warning message is persistent until the dependencies are resolved

* **Postconditions:** Alerts are displayed correctly for uncompleted dependencies

* **Expected Result:** The system provides alerts or warnings for uncompleted dependencies

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

By covering all acceptance criteria, these test cases ensure that the Task Dependencies feature is thoroughly tested, and potential defects can be identified early in the development cycle. This helps to deliver a high-quality software product that meets the user's needs and expectations.

---

**Data - 27**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Comments and Collaboration

**Feature description -**

Users can collaborate on tasks by leaving comments, attaching files, and engaging in discussions directly within the task interface. This feature facilitates communication and teamwork, allowing team members to provide updates, share feedback, and ask questions related to the task.

**Userstory title -**

As a user, I want to comment on tasks so that I can collaborate with my team.

**Acceptance criteria -**

['Users can add comments to tasks.', 'Comments can include text, attachments, and links.', 'Users can reply to and edit their comments.', 'The system notifies users of new comments.', 'Comments are displayed in chronological order.']

**Generated Test Cases:**

Here are five detailed test cases that cover all the acceptance criteria for the "Task Comments and Collaboration" feature.

**Test Case 1: Verify that users can add comments to tasks**

* **Test Case ID:** TC-001

* **Test Case Title:** Verify that users can add comments to tasks

* **Test Case Description:** This test case ensures that users can successfully add comments to tasks without any errors.

* **Test Suite:** Task Comments and Collaboration

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the application.

 + Task is created and assigned to the user.

* **Test Data:** No test data needed

* **Test Steps:**
 1. Log in to the application as a user.
 2. Navigate to the task page.
 3. Click on the "Add comment" button.
 4. Enter a comment in the text area.
 5. Click on the "Submit" button.
 6. Verify that the comment is displayed on the task page.
* **Postconditions:** Comment is displayed on the task page, and the user can see their own comment.
* **Expected Result:** The comment is successfully added to the task, and the user can view it.
* **Severity:** Minor
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that comments can include text, attachments, and links**

* **Test Case ID:** TC-002
* **Test Case Title:** Verify that comments can include text, attachments, and links
* **Test Case Description:** This test case ensures that users can add comments with text, attachments, and links without any errors.
* **Test Suite:** Task Comments and Collaboration
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the application.
 + Task is created and assigned to the user.
* **Test Data:** Comment with text, attachment, and link
* **Test Steps:**

1. Log in to the application as a user.

2. Navigate to the task page.

3. Click on the "Add comment" button.

4. Enter a comment with text, attachment, and link.

5. Click on the "Submit" button.

6. Verify that the comment is displayed on the task page with text, attachment, and link.

* **Postconditions:** Comment is displayed on the task page with text, attachment, and link.

* **Expected Result:** The comment is successfully added to the task with text, attachment, and link.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can reply to and edit their comments**

* **Test Case ID:** TC-003

* **Test Case Title:** Verify that users can reply to and edit their comments

* **Test Case Description:** This test case ensures that users can successfully reply to and edit their comments without any errors.

* **Test Suite:** Task Comments and Collaboration

* **Test Priority:** Medium

* **Preconditions:**

  + User is logged in to the application.

  + Task is created and assigned to the user.

  + User has previously added a comment to the task.

* **Test Data:** Comment with reply and edit

* **Test Steps:**

1. Log in to the application as a user.
2. Navigate to the task page.
3. Click on the "Reply to comment" button.
4. Enter a reply comment.
5. Click on the "Submit" button.
6. Click on the "Edit comment" button.
7. Edit the original comment.
8. Click on the "Save changes" button.
9. Verify that the reply comment is displayed on the task page, and the original comment is edited.

* **Postconditions:** Reply comment is displayed on the task page, and original comment is edited.
* **Expected Result:** The user can successfully reply to and edit their comments.
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that the system notifies users of new comments**

* **Test Case ID:** TC-004
* **Test Case Title:** Verify that the system notifies users of new comments
* **Test Case Description:** This test case ensures that the system successfully notifies users of new comments without any errors.
* **Test Suite:** Task Comments and Collaboration
* **Test Priority:** High
* **Preconditions:**
  + User is logged in to the application.
  + Task is created and assigned to the user.

* **Test Data:** New comment added by another user

* **Test Steps:**

 1. Log in to the application as a user.

 2. Navigate to the task page.

 3. Another user adds a new comment to the task.

 4. Verify that the user receives a notification about the new comment.

 5. Click on the notification to view the new comment.

* **Postconditions:** User receives notification about the new comment.

* **Expected Result:** The system successfully notifies the user of the new comment.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that comments are displayed in chronological order**

* **Test Case ID:** TC-005

* **Test Case Title:** Verify that comments are displayed in chronological order

* **Test Case Description:** This test case ensures that comments are successfully displayed in chronological order without any errors.

* **Test Suite:** Task Comments and Collaboration

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in to the application.

 + Task is created and assigned to the user.

 + Multiple comments have been added to the task by different users.

* **Test Data:** Comments with timestamps

* **Test Steps:**

1. Log in to the application as a user.

2. Navigate to the task page.

3. Verify that the comments are displayed in chronological order based on their timestamps.

* **Postconditions:** Comments are displayed in chronological order.

* **Expected Result:** The comments are successfully displayed in chronological order.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases cover all the acceptance criteria for the "Task Comments and Collaboration" feature, including adding comments, including text, attachments, and links, replying to and editing comments, receiving notifications, and displaying comments in chronological order. By thoroughly testing these scenarios, we can ensure that the feature works as expected and meets the requirements of the user story.

---

**Data - 28**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

File Attachments

**Feature description -**

Users can attach relevant files, documents, and resources to tasks, ensuring that all necessary information is accessible in one place. File attachments can include documents, spreadsheets, images, links, and other resources needed for task execution.

**Userstory title -**

As a user, I want to attach files to tasks so that all relevant information is easily accessible.

**Acceptance criteria -**

['Users can attach multiple files to a task.', 'Attachments can include various file types (documents, images, etc.).', 'Users can view and download attachments.', 'The system limits the size of attachments to prevent data overload.', 'Users can remove attachments if needed.']

**Generated Test Cases:**

Here are five detailed test cases for the File Attachments feature of the Task Management Tool, following the provided Test Case Creation Template:

**Test Case 1:**

* **Test Case ID:** TT01_FILE_ATTACHMENT_MULTIPLE_FILES

* **Test Case Title:** Verify that users can attach multiple files to a task

* **Test Case Description:** This test case will validate that users can upload and attach multiple files to a task, ensuring that the task is updated with the new attachments.

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* ** Preconditions:**

 + User is logged in and has permission to edit tasks

 + Task is created with a valid title and description

* **Test Data:** Multiple files of different types (documents, images, etc.)

* **Test Steps:**

 1. Log in to the Task Management Tool with a valid user account.

 2. Create a new task with a valid title and description.

 3. Click on the "Attach File" button on the task details page.

 4. Upload multiple files of different types (documents, images, etc.).

 5. Verify that the task is updated with the new attachments.

 6. Repeat steps 3-5 multiple times to test attaching multiple files.

* **Postconditions:** Verify that the user is taken back to the task details page with the updated attachments.

* **Expected Result:** Users can attach multiple files to a task without encountering any errors.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2:**

* **Test Case ID:** TT02_FILE_TYPE_VALIDATION

* **Test Case Title:** Verify that attachments can include various file types

* **Test Case Description:** This test case will validate that the system allows users to attach files of various types, ensuring that the task is updated with the new attachments.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* ** Preconditions:**

 + User is logged in and has permission to edit tasks

 + Task is created with a valid title and description

* **Test Data:** Various file types (documents, images, spreadsheets, zip files, etc.)

* **Test Steps:**

 1. Log in to the Task Management Tool with a valid user account.

 2. Create a new task with a valid title and description.

 3. Click on the "Attach File" button on the task details page.

 4. Upload files of various types (document, image, spreadsheet, zip file, etc.).

 5. Verify that the task is updated with the new attachments.

 6. Repeat steps 3-5 multiple times to test various file types.

* **Postconditions:** Verify that the user is taken back to the task details page with the updated attachments.

* **Expected Result:** Users can attach files of various types without encountering any errors.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3:**

* **Test Case ID:** TT03_FILE_DOWNLOAD

* **Test Case Title:** Verify that users can view and download attachments

* **Test Case Description:** This test case will validate that users can view and download attachments uploaded to a task, ensuring that the attachments are accessible.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* ** Preconditions:**

 + User is logged in and has permission to view tasks

 + Task is created with attachments

* **Test Data:** None needed

* **Test Steps:**

 1. Log in to the Task Management Tool with a valid user account.

2. Navigate to the task details page with attachments.

3. Click on the "Attachments" button to view the attachments.

4. Verify that the attachments are displayed correctly.

5. Click on an attachment to download it.

6. Verify that the attachment is downloaded successfully.

* **Postconditions:** Verify that the user is taken back to the task details page.

* **Expected Result:** Users can view and download attachments without encountering any errors.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4:**

* **Test Case ID:** TT04_FILE_UPLOAD_SIZE_LIMIT

* **Test Case Title:** Verify that the system limits the size of attachments to prevent data overload

* **Test Case Description:** This test case will validate that the system enforces a file size limit for attachments, preventing data overload.

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* ** Preconditions:**

+ User is logged in and has permission to edit tasks

+ Task is created with a valid title and description

* **Test Data:** Large file (exceeding the system's file size limit)

* **Test Steps:**

1. Log in to the Task Management Tool with a valid user account.

2. Create a new task with a valid title and description.

3. Click on the "Attach File" button on the task details page.

4. Upload a large file (exceeding the system's file size limit).

5. Verify that the system displays an error message indicating the file size limit.

6. Repeat steps 3-5 multiple times to test different file sizes.

* **Postconditions:** Verify that the user is taken back to the task details page with an error message.

* **Expected Result:** The system prevents users from uploading files exceeding the size limit.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

**Test Case 5:**

* **Test Case ID:** TT05_FILE_ATTACHMENT_REMOVAL

* **Test Case Title:** Verify that users can remove attachments from a task

* **Test Case Description:** This test case will validate that users can remove attachments from a task, ensuring that the task is updated with the new state.

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* ** Preconditions:**

 + User is logged in and has permission to edit tasks

 + Task is created with attachments

* **Test Data:** None needed

* **Test Steps:**

1. Log in to the Task Management Tool with a valid user account.

2. Navigate to the task details page with attachments.

3. Click on the "Attachments" button to view the attachments.

4. Select an attachment to remove.

5. Verify that the attachment is removed from the task.

6. Repeat steps 3-5 multiple times to test removing different attachments.

* **Postconditions:** Verify that the user is taken back to the task details page with the updated attachments.

* **Expected Result:** Users can remove attachments from a task without encountering any errors.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 29**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Recurring Tasks

**Feature description -**

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

**Userstory title -**

As a user, I want to set recurring tasks so that regular activities are automated.

**Acceptance criteria -**

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

**Generated Test Cases:**

Here are the 5 test cases created using the provided Test Case Creation Template:

**Test Case 1: Verify that users can set tasks to recur at specified intervals**

* **Test Case ID:** TCT-001
* **Test Case Title:** Verify that users can set tasks to recur at specified intervals
* **Test Case Description:** As a user, I want to set tasks to recur at specified intervals (daily, weekly, monthly) so that regular activities are automated. This test case will cover the acceptance criteria that users can set tasks to recur at specified intervals.
* **Test Suite:** Recurring Tasks
* **Test Priority:** High
* **Preconditions:**
 + User has logged in to the Task Management Tool
 + User has created a new task
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool
 2. Create a new task
 3. Click on the "Recurring" option and select the interval (daily, weekly, monthly)
 4. Set the start date and time for the recurring task

 5. Click "Save" to save the recurring task

* **Postconditions:**

 + The recurring task is saved successfully

 + The system generates an instance of the recurring task for the next occurrence

* **Expected Result:** The user can set the task to recur at the specified interval.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that the system generates instances of recurring tasks automatically**

* **Test Case ID:** TCT-002

* **Test Case Title:** Verify that the system generates instances of recurring tasks automatically

* **Test Case Description:** As a user, I want the system to generate instances of recurring tasks automatically so that regular activities are automated without manual intervention. This test case will cover the acceptance criteria that the system generates instances of recurring tasks automatically.

* **Test Suite:** Recurring Tasks

* **Test Priority:** High

* **Preconditions:**

 + User has logged in to the Task Management Tool

 + User has created a recurring task

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the Task Management Tool

 2. Create a recurring task with a specified interval (daily, weekly, monthly)

 3. Wait for the next occurrence of the recurring task

 4. Verify that the system has generated an instance of the recurring task for the next occurrence

* **Postconditions:**

 + The system has generated an instance of the recurring task for the next occurrence

 + The user receives a notification for the upcoming instance of the recurring task

* **Expected Result:** The system generates instances of recurring tasks automatically without manual intervention.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can modify the recurrence pattern or cancel the recurrence**

* **Test Case ID:** TCT-003

* **Test Case Title:** Verify that users can modify the recurrence pattern or cancel the recurrence

* **Test Case Description:** As a user, I want to be able to modify the recurrence pattern or cancel the recurrence of a task so that I can adjust the schedule as needed. This test case will cover the acceptance criteria that users can modify the recurrence pattern or cancel the recurrence.

* **Test Suite:** Recurring Tasks

* **Test Priority:** Medium

* **Preconditions:**

 + User has logged in to the Task Management Tool

 + User has created a recurring task

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the Task Management Tool

 2. Edit an existing recurring task

 3. Modify the recurrence pattern (e.g., change from weekly to bi-weekly)
 4. Cancel the recurrence of the task
 5. Verify that the modifications or cancellation have been saved successfully
* **Postconditions:**
 + The user can modify the recurrence pattern or cancel the recurrence
 + The system reflects the changes in the task list
* **Expected Result:** Users can modify the recurrence pattern or cancel the recurrence of a task.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that the system notifies users of upcoming instances of recurring tasks**

* **Test Case ID:** TCT-004
* **Test Case Title:** Verify that the system notifies users of upcoming instances of recurring tasks
* **Test Case Description:** As a user, I want the system to notify me of upcoming instances of recurring tasks so that I am aware of upcoming deadlines. This test case will cover the acceptance criteria that the system notifies users of upcoming instances of recurring tasks.
* **Test Suite:** Recurring Tasks
* **Test Priority:** High
* **Preconditions:**
 + User has logged in to the Task Management Tool
 + User has created a recurring task with a notification set
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool

 2. Create a recurring task with a notification set for an upcoming instance
 3. Verify that the user receives a notification for the upcoming instance of the recurring task
* **Postconditions:**
 + The user receives a notification for the upcoming instance of the recurring task
 + The user is aware of the upcoming deadline
* **Expected Result:** The system notifies users of upcoming instances of recurring tasks.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that users can view a list of all instances of a recurring task**

* **Test Case ID:** TCT-005
* **Test Case Title:** Verify that users can view a list of all instances of a recurring task
* **Test Case Description:** As a user, I want to view a list of all instances of a recurring task so that I can monitor the progress and completion of recurring tasks. This test case will cover the acceptance criteria that users can view a list of all instances of a recurring task.
* **Test Suite:** Recurring Tasks
* **Test Priority:** Medium
* **Preconditions:**
 + User has logged in to the Task Management Tool
 + User has created a recurring task
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool
 2. Navigate to the task list

3. Filter the task list to view all instances of a recurring task

4. Verify that the system displays the list of all instances of the recurring task

* **Postconditions:**

+ The user can view the list of all instances of a recurring task

+ The user can monitor the progress and completion of recurring tasks

* **Expected Result:** Users can view a list of all instances of a recurring task.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**--------------------------------------------------------------------------------------------------**

**Data - 30**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Tags and Labels

**Feature description -**

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

**Userstory title -**

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

**Acceptance criteria -**

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Tags and Labels feature, each thoroughly covering one or more of the acceptance criteria:

**Test Case 1: Add Tags and Labels to Tasks**

* **Test Case ID:** TCTVL-01
* **Test Case Title:** Verify that users can add tags and labels to tasks.
* **Test Suite:** Task Tags and Labels
* **Test Priority:** High
* **Preconditions:**
 + The user is logged in to the system.
 + The user has created at least one task.
* **Test Data:** Task details, including title, description, and deadline.
* **Test Steps:**
 1. Log in to the system and navigate to the task list.
 2. Select a task to edit.
 3. Click on the "Add Tags" or "Add Labels" button.
 4. Enter a new tag or label name in the input field.
 5. Click on the "Save" button to save the changes.
* **Postconditions:**

 + The task should now have the added tag or label displayed prominently.

 + The user should be able to view the added tag or label in the task list.

* **Expected Result:** The user successfully adds tags and labels to tasks.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Customize and Edit Tags and Labels**

* **Test Case ID:** TCTVL-02

* **Test Case Title:** Verify that tags and labels can be customized and edited.

* **Test Suite:** Task Tags and Labels

* **Test Priority:** Medium

* **Preconditions:**

 + The user has added tags and labels to tasks.

* **Test Data:** Task details, including title, description, and deadline.

* **Test Steps:**

 1. Log in to the system and navigate to the task list.

 2. Select a task to edit.

 3. Click on the "Edit Tags" or "Edit Labels" button.

 4. Modify the existing tag or label name in the input field.

 5. Click on the "Save" button to save the changes.

* **Postconditions:**

 + The task should now have the modified tag or label displayed prominently.

 + The user should be able to view the modified tag or label in the task list.

* **Expected Result:** The user successfully customizes and edits tags and labels.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Filter and Sort Tasks by Tags and Labels**

* **Test Case ID:** TCTVL-03

* **Test Case Title:** Verify that tasks can be filtered and sorted based on tags and labels.

* **Test Suite:** Task Tags and Labels

* **Test Priority:** High

* **Preconditions:**

 + The user has added tags and labels to tasks.

* **Test Data:** Task details, including title, description, and deadline.

* **Test Steps:**

 1. Log in to the system and navigate to the task list.

 2. Click on the "Filter" or "Sort" button.

 3. Select a tag or label from the dropdown menu.

 4. Click on the "Apply" button to apply the filter or sort.

* **Postconditions:**

 + The task list should be filtered or sorted based on the selected tag or label.

 + The user should be able to view the filtered or sorted tasks.

* **Expected Result:** The user successfully filters and sorts tasks by tags and labels.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Display Tags and Labels Prominently in Task List**

* **Test Case ID:** TCTVL-04

* **Test Case Title:** Verify that the system displays tags and labels prominently in the task list.

* **Test Suite:** Task Tags and Labels

* **Test Priority:** Medium

* **Preconditions:**

 + The user has added tags and labels to tasks.

* **Test Data:** Task details, including title, description, and deadline.

* **Test Steps:**

 1. Log in to the system and navigate to the task list.

 2. Check that tags and labels are displayed prominently for each task.

 3. Verify that the tags and labels are displayed in a clean and organized manner.

* **Postconditions:** None

* **Expected Result:** The system displays tags and labels prominently in the task list.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Remove Tags and Labels**

* **Test Case ID:** TCTVL-05

* **Test Case Title:** Verify that users can remove tags and labels if needed.

* **Test Suite:** Task Tags and Labels

* **Test Priority:** High

* **Preconditions:**

 + The user has added tags and labels to tasks.

* **Test Data:** Task details, including title, description, and deadline.

* **Test Steps:**

 1. Log in to the system and navigate to the task list.

 2. Select a task to edit.

 3. Click on the "Remove Tags" or "Remove Labels" button.

 4. Confirm that the user wants to remove the tag or label.

* **Postconditions:**

 + The task should now be removed of the tag or label.

 + The user should be able to view the updated task.

* **Expected Result:** The user successfully removes tags and labels.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 31**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Status Tracking

**Feature description -**

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

**Userstory title -**

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

**Acceptance criteria -**

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

**Generated Test Cases:**

Here are five detailed test cases that cover the acceptance criteria of the Task Status Tracking feature:

**Test Case 1: Update Task Status**

* **Test Case ID:** TC001

* **Test Case Title:** Verify that users can update the status of tasks

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* **Preconditions:**

 + User is logged in

 + Task exists in the task list

* **Test Data:** No test data needed

* **Test Steps:**

 1. Navigate to the task list

 2. Select a task to update

 3. Click the "Update Status" button

 4. Select a new status (e.g., In Progress, Completed)

 5. Click "Save Changes"

* **Postconditions:**

 + Task status is updated in the task list

 + User receives a notification about the status change (if enabled)

* **Expected Result:** The task status is updated successfully, and the user receives a notification (if enabled)

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Display Task Status Prominently**

* **Test Case ID:** TC002

* **Test Case Title:** Verify that the system displays task status prominently in the task list

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in

 + Task exists in the task list

* **Test Data:** No test data needed

* **Test Steps:**

 1. Navigate to the task list

 2. Observe the task list for the task status display

3. Note the task status (e.g., To Do, In Progress, Completed)

* **Postconditions:**

+ Task status is displayed prominently in the task list

* **Expected Result:** The task status is displayed prominently in the task list, making it easily visible to the user

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: View Task Progress Percentage**

* **Test Case ID:** TC003

* **Test Case Title:** Verify that users can view the progress percentage of tasks

* **Test Suite:** Task Management Tool

* **Test Priority:** Medium

* **Preconditions:**

+ User is logged in

+ Task exists in the task list with progress tracking enabled

* **Test Data:** No test data needed

* **Test Steps:**

1. Navigate to the task list

2. Select a task with progress tracking enabled

3. Observe the task details for the progress percentage display

4. Note the progress percentage value (e.g., 25%, 50%, 75%)

* **Postconditions:**

+ Progress percentage is displayed for the task

* **Expected Result:** The progress percentage is displayed correctly for the task

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Receive Notifications for Status Changes**

* **Test Case ID:** TC004

* **Test Case Title:** Verify that the system provides notifications for status changes

* **Test Suite:** Task Management Tool

* **Test Priority:** High

* **Preconditions:**

 + User is logged in

 + Task exists in the task list

 + Notifications are enabled for status changes

* **Test Data:** No test data needed

* **Test Steps:**

 1. Update the task status

 2. Observe the notification for the task status change

 3. Verify that the notification content matches the expected value

* **Postconditions:**

 + User receives a notification about the status change

* **Expected Result:** The user receives a notification about the task status change

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Filter and Sort Task Status**

* **Test Case ID:** TC005
* **Test Case Title:** Verify that task status can be filtered and sorted
* **Test Suite:** Task Management Tool
* **Test Priority:** Medium
* **Preconditions:**
 + User is logged in
 + Task exists in the task list
* **Test Data:** No test data needed
* **Test Steps:**
 1. Navigate to the task list
 2. Apply a filter for a specific task status (e.g., To Do, In Progress, Completed)
 3. Observe the task list for the filtered results
 4. Apply a sort criteria for the task status (e.g., oldest first, newest first)
 5. Observe the task list for the sorted results
* **Postconditions:**
 + Task status is correctly filtered and sorted
* **Expected Result:** The task status is correctly filtered and sorted, showing only the desired tasks
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

These test cases ensure that each acceptance criterion is thoroughly covered, and the product quality is maintained by identifying potential defects early in the development cycle.

---

**Data - 32**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Filtering and Sorting

**Feature description -**

Users can filter and sort tasks based on various criteria such as due date, priority, assignee, or project. Filtering and sorting options help users focus on specific tasks or view tasks from different perspectives, enhancing task management and organization.

**Userstory title -**

As a user, I want to filter and sort tasks so that I can focus on specific tasks or view tasks from different perspectives.

**Acceptance criteria -**

['Users can filter tasks based on criteria such as due date, priority, assignee, or project.', 'Tasks can be sorted by various attributes.', 'The system provides multiple filter and sort options in the task list.', 'Users can save and apply custom filter and sort settings.', 'Filters and sorts do not affect the actual data, only the view.']

**Generated Test Cases:**

Here are five detailed test cases that cover all the acceptance criteria for the "Task Filtering and Sorting" feature, following the Test Case Creation Template:

**Test Case 1: Verify that Users can Filter Tasks by Due Date**

| Test Case ID | TC-001 |
| --- | --- |
| Test Case Title | Verify that Users can Filter Tasks by Due Date |
| Test Case Description | This test case verifies that users can filter tasks based on due date criteria. The test case checks if tasks are correctly filtered and displayed when a user selects a specific due date range. This test case ensures that the filter functionality works as expected |
| Test Suite | Task Filtering and Sorting |
| Test Priority | Medium |
| Preconditions | 1. User is logged in to the task management tool, 2. User has created tasks with different due dates |
| Test Data | Multiple tasks with varying due dates |
| Test Steps | 1. Log in to the task management tool and navigate to the task list, 2. Select a specific due date range using the date picker, 3. Verify that tasks are correctly filtered and displayed |
| Postconditions | Tasks are filtered correctly by due date range |
| Expected Result | All tasks with due dates falling within the specified range are displayed, while tasks outside the range are hidden |
| Severity | Minor |
| Priority | Medium |
| Type of Testing | Functional Testing |
| Test Case Approach | Positive |

**Test Case 2: Verify that Tasks can be Sorted by Priority**

| Test Case ID | TC-002 |
| --- | --- |
| Test Case Title | Verify that Tasks can be Sorted by Priority |
| Test Case Description | This test case verifies that users can sort tasks based on priority criteria. The test case checks if tasks are correctly sorted and displayed when a user selects a specific sort order. This test case ensures that the sort functionality works as expected |
| Test Suite | Task Filtering and Sorting |
| Test Priority | Medium |
| Preconditions | 1. User is logged in to the task management tool, 2. User has created tasks with varying priorities |
| Test Data | Multiple tasks with different priorities |
| Test Steps | 1. Log in to the task management tool and navigate to the task list, 2. Select a specific sort order (e.g., highest to lowest priority), 3. Verify that tasks are correctly sorted and displayed |
| Postconditions | Tasks are sorted correctly by priority |
| Expected Result | Tasks are displayed in the order specified by the user, with highest priority tasks first and lowest priority tasks last |
| Severity | Minor |
| Priority | Medium |
| Type of Testing | Functional Testing |
| Test Case Approach | Positive |

**Test Case 3: Verify that Filters and Sorts do not affect Actual Data**

| Test Case ID | TC-003 |
| --- | --- |
| Test Case Title | Verify that Filters and Sorts do not affect Actual Data |
| Test Case Description | This test case verifies that applying filters and sorts to the task list does not affect the

underlying data. The test case checks if tasks are correctly displayed when filters and sorts are applied and removed. This test case ensures that data integrity is maintained |

| Test Suite | Task Filtering and Sorting |

| Test Priority | Low |

| Preconditions | 1. User is logged in to the task management tool, 2. User has created tasks with varying due dates and priorities |

| Test Data | No test data needed |

| Test Steps | 1. Log in to the task management tool and navigate to the task list, 2. Apply filters and sorts to the task list, 3. Verify that tasks are correctly displayed and count remains the same |

| Postconditions | Tasks are displayed correctly after applying and removing filters and sorts |

| Expected Result | The count of tasks remains the same, and tasks are displayed without any gaps or duplicates |

| Severity | Trivial |

| Priority | Low |

| Type of Testing | Functional Testing |

| Test Case Approach | Destructive |

**Test Case 4: Verify that Custom Filter and Sort Settings are Saved**

| Test Case ID | TC-004 |
| --- | --- |
| Test Case Title | Verify that Custom Filter and Sort Settings are Saved |
| Test Case Description | This test case verifies that users can save and apply custom filter and sort settings. The test case checks if the saved settings are correctly applied when the user navigates back to the task list. This test case ensures that user preferences are honored |
| Test Suite | Task Filtering and Sorting |
| Test Priority | High |

| Preconditions | 1. User is logged in to the task management tool, 2. User has created tasks with varying due dates and priorities |
| Test Data | Multiple tasks with different due dates and priorities |
| Test Steps | 1. Log in to the task management tool and navigate to the task list, 2. Apply custom filters and sorts to the task list, 3. Save the settings, 4. Navigate back to the task list and verify that the saved settings are applied |
| Postconditions | Custom filter and sort settings are saved and applied correctly |
| Expected Result | The saved settings are correctly applied when the user returns to the task list |
| Severity | Critical |
| Priority | High |
| Type of Testing | Functional Testing |
| Test Case Approach | Positive |

**Test Case 5: Verify that Users can Collaborate with Team Members using Filtered and Sorted Tasks**

| Test Case ID | TC-005 |
| --- | --- |
| Test Case Title | Verify that Users can Collaborate with Team Members using Filtered and Sorted Tasks |
| Test Case Description | This test case verifies that users can collaborate with team members using filtered and sorted tasks. The test case checks if team members are correctly informed about tasks when filters and sorts are applied. This test case ensures that collaboration is facilitated effectively |
| Test Suite | Task Filtering and Sorting |
| Test Priority | Medium |
| Preconditions | 1. User is logged in to the task management tool, 2. User has created tasks with varying due dates and priorities, 3. User has team members assigned to tasks |
| Test Data | Multiple tasks with different due dates and priorities, team members assigned to tasks |
| Test Steps | 1. Log in to the task management tool and navigate to the task list, 2. Apply filters and sorts to

the task list, 3. Verify that team members are informed about tasks when filters and sorts are applied |

| Postconditions | Team members are correctly informed about tasks when filters and sorts are applied |

| Expected Result | Team members receive notifications about tasks that are affected by the filters and sorts applied |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

---

**Data - 33**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Time Tracking

**Feature description -**

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

**Userstory title -**

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

**Acceptance criteria -**

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

**Generated Test Cases:**

Here are five detailed test cases for the Time Tracking feature of the Task Management Tool, following the provided test case creation template:

**Test Case 1: Timer Start and Stop**

* **Test Case ID:** TT-001
* **Test Case Title:** Verify that users can start and stop a timer for each task.
* **Test Case Description:** This test case ensures that users can start and stop a timer for each task and that the timer accurately records the time spent on tasks. This test case will also verify that the timer can be paused and resumed.
* **Test Suite:** Time Tracking Module
* **Test Priority:** High
* **Preconditions:**
 + User has logged in and navigated to the task page
 + Task is assigned to the user
 + Timer feature is enabled
* **Test Data:** No test data needed
* **Test Steps:**
 1. Start the timer for a task
 2. Stop the timer for the task
 3. Verify that the timer is paused and shows the correct time spent
 4. Resume the timer

5. Verify that the timer is now running and shows the correct time spent

* **Postconditions:**

  + Verify that the timer is now stopped

  + Verify that the time spent on the task is accurately recorded

* **Expected Result:** The timer can be started and stopped, and the time spent on the task is accurately recorded.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Manual Time Entry**

* **Test Case ID:** TT-002

* **Test Case Title:** Verify that users can manually enter or adjust time entries.

* **Test Case Description:** This test case verifies that users can manually enter or adjust time entries for tasks and that the changes are accurately reflected in the time spent on the task.

* **Test Suite:** Time Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

  + User has logged in and navigated to the task page

  + Task is assigned to the user

  + Timer feature is enabled

* **Test Data:** No test data needed

* **Test Steps:**

  1. Manually enter a time entry for a task

  2. Adjust the time entry

3. Verify that the change is accurately reflected in the time spent on the task

* **Postconditions:**

 + Verify that the time spent on the task is updated

 + Verify that the user's manual time entry is accurately recorded

* **Expected Result:** Users can manually enter or adjust time entries for tasks, and the changes are accurately reflected in the time spent on the task.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Time Spent Display**

* **Test Case ID:** TT-003

* **Test Case Title:** Verify that the system records and displays total time spent on tasks.

* **Test Case Description:** This test case verifies that the system accurately records and displays the total time spent on tasks.

* **Test Suite:** Time Tracking Module

* **Test Priority:** High

* **Preconditions:**

 + User has logged in and navigated to the task page

 + Task is assigned to the user

 + Timer feature is enabled

* **Test Data:** No test data needed

* **Test Steps:**

 1. Start the timer for a task

 2. Stop the timer and pause the timer for the task

 3. Resume the timer and stop it for the task
 4. Verify that the total time spent on the task is accurately displayed
* **Postconditions:**
 + Verify that the time spent on the task is accurately recorded
 + Verify that the total time spent on the task is accurately displayed
* **Expected Result:** The system accurately records and displays the total time spent on tasks.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Time Tracking Data Export**

* **Test Case ID:** TT-004
* **Test Case Title:** Verify that time tracking data can be exported.
* **Test Case Description:** This test case verifies that users can export their time tracking data for tasks and that the exported data is accurate.
* **Test Suite:** Time Tracking Module
* **Test Priority:** Medium
* **Preconditions:**
 + User has logged in and navigated to the task page
 + Task is assigned to the user
 + Timer feature is enabled
* **Test Data:** No test data needed
* **Test Steps:**
 1. Start the timer for a task
 2. Stop the timer and pause the timer for the task

3. Resume the timer and stop it for the task

4. Export the time tracking data

5. Verify that the exported data accurately reflects the time spent on the task

* **Postconditions:**

 + Verify that the exported data is accurate

 + Verify that the exported data includes the correct details

* **Expected Result:** Users can export their time tracking data, and the exported data is accurate.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Time Spent Reports**

* **Test Case ID:** TT-005

* **Test Case Title:** Verify that the system provides reports on time spent per task or project.

* **Test Case Description:** This test case verifies that the system provides accurate reports on time spent per task or project and that the reports are correctly formatted.

* **Test Suite:** Time Tracking Module

* **Test Priority:** High

* **Preconditions:**

 + User has logged in and navigated to the task page

 + Task is assigned to the user

 + Timer feature is enabled

* **Test Data:** No test data needed

* **Test Steps:**

1. Start the timer for multiple tasks

2. Stop the timer and pause the timer for the tasks

3. Resume the timer and stop it for the tasks

4. Generate reports on time spent per task and project

5. Verify that the reports accurately reflect the time spent on tasks and projects

* **Postconditions:**

 + Verify that the reports accurately reflect the time spent

 + Verify that the reports are correctly formatted

* **Expected Result:** The system provides accurate reports on time spent per task or project, and the reports are correctly formatted.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These five test cases cover all the acceptance criteria for the Time Tracking feature, ensuring that users can start and stop a timer, manually enter or adjust time entries, view total time spent on tasks, export time tracking data, and receive accurate reports on time spent per task or project.

---

**Data - 34**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Templates

**Feature description -**

Users can create and use templates for common task types or project workflows, streamlining task creation and standardizing processes across projects. Task templates can include predefined task structures, descriptions, assignees, and due dates, saving time and ensuring consistency in task execution.

**Userstory title -**

As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

**Acceptance criteria -**

['Users can create and save task templates.', 'Templates can include predefined task structures, descriptions, assignees, and due dates.', 'Users can apply templates to new tasks or projects.', 'Templates can be edited or deleted as needed.', 'The system displays a list of available templates for selection.']

**Generated Test Cases:**

Here are five detailed test cases for the Task Templates feature:

**Test Case 1: Create and Save Task Template**

* **Test Case ID:** TC_01

* **Test Case Title:** Verify that users can create and save task templates

* **Test Suite:** Task Templates

* **Test Priority:** High

* **Preconditions:** User is logged in with valid credentials, task templates module is accessible.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as a user with valid credentials.

 2. Navigate to the task templates module.

3. Click on the "Create Template" button.

4. Enter template details (name, description, etc.) and assignees.

5. Set due dates for tasks in the template.

6. Save the template.

* **Postconditions:** Task template is saved successfully and displayed in the template list.

* **Expected Result:** The task template is created and saved successfully.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Edit and Delete Task Template**

* **Test Case ID:** TC_02

* **Test Case Title:** Verify that users can edit and delete task templates

* **Test Suite:** Task Templates

* **Test Priority:** Medium

* **Preconditions:** User is logged in with valid credentials, existing task template is available.

* **Test Data:** Existing task template ID.

* **Test Steps:**

1. Log in as a user with valid credentials.

2. Navigate to the task templates module.

3. Select an existing task template for editing.

4. Modify template details (name, description, etc.) and assignees.

5. Set due dates for tasks in the template.

6. Save the template changes.

7. Delete the saved task template.

* **Postconditions:** Task template is edited or deleted successfully, updated in the template list.

* **Expected Result:** The task template is edited or deleted successfully, no changes are reflected if editing, and the template is removed from the list if deleted.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive and Negative

**Test Case 3: Apply Task Template to New Task**

* **Test Case ID:** TC_03

* **Test Case Title:** Verify that users can apply task templates to new tasks

* **Test Suite:** Task Templates

* **Test Priority:** High

* **Preconditions:** User is logged in with valid credentials, existing task template is available.

* **Test Data:** Existing task template ID, new task details.

* **Test Steps:**

1. Log in as a user with valid credentials.
2. Navigate to the task creation page.
3. Select an existing task template for application.
4. Review and update task details if necessary.
5. Save the task.

* **Postconditions:** Task is created with applied template successfully.

* **Expected Result:** The task is created with all template details applied (structure, description, assignees, due dates).

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Display List of Available Task Templates**

* **Test Case ID:** TC_04

* **Test Case Title:** Verify that the system displays a list of available task templates

* **Test Suite:** Task Templates

* **Test Priority:** Low

* **Preconditions:** User is logged in with valid credentials.

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a user with valid credentials.

2. Navigate to the task templates module.

3. Review the displayed list of available task templates.

* **Postconditions:** List of task templates is displayed correctly.

* **Expected Result:** The system displays a list of available task templates, including their names and descriptions.

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Validate Template Data**

* **Test Case ID:** TC_05

* **Test Case Title:** Verify that template data is validated correctly

* **Test Suite:** Task Templates

* **Test Priority:** Medium

* **Preconditions:** User is logged in with valid credentials, existing task template is available.

* **Test Data:** Existing task template with invalid or incorrect data (e.g., empty fields, incompatible assignees).

* **Test Steps:**

1. Log in as a user with valid credentials.
2. Navigate to the task templates module.
3. Select an existing task template for editing.
4. Introduce invalid or incorrect data for testing.
5. Save the template changes.
6. Review the validation errors.

* **Postconditions:** Validation errors are displayed correctly for template data.

* **Expected Result:** The system validates template data correctly, displaying errors for invalid or incorrect data.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

These five test cases thoroughly cover the acceptance criteria for the Task Templates feature, including creating and saving task templates, editing and deleting task templates, applying task templates to new tasks, displaying the list of available task templates, and validating template data.

---

**Data - 35**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Calendar Integration

**Feature description -**

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

**Userstory title -**

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

**Acceptance criteria -**

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

**Generated Test Cases:**

Here are five detailed test cases that thoroughly cover the acceptance criteria for the Calendar Integration feature in the Task Management Tool project.

**Test Case 1: Verify that users can sync tasks with external calendars**

* **Test Case ID**: TC-001

* **Test Case Title**: Verify that users can sync tasks with external calendars

* **Test Case Description**: This test case ensures that users can successfully sync tasks with their external calendars, such as Google Calendar or Outlook.

* **Test Suite**: Calendar Integration

* **Test Priority**: High

* **Preconditions**:

 + User has a valid external calendar account (e.g., Google Calendar, Outlook)

 + User has created tasks in the Task Management Tool

* **Test Data**: User's external calendar account credentials

* **Test Steps**:

 1. Log in to the Task Management Tool as a user with a valid external calendar account.

 2. Create a new task with a deadline and description.

 3. Navigate to the calendar view and click on the "Sync with external calendar" button.

 4. Enter the user's external calendar account credentials and click "Connect".

 5. Verify that the task is successfully synced with the external calendar.

* **Postconditions**: Task is synced with external calendar, and user can view task deadlines in the calendar view.

* **Expected Result**: Task is successfully synced with external calendar, and user can view task deadlines in the calendar view.

* **Severity**: Critical

* **Priority**: High

* **Type of Testing**: Functional Testing

* **Test Case Approach**: Positive

**Test Case 2: Verify that the system displays task deadlines in the calendar view**

* **Test Case ID**: TC-002

* **Test Case Title**: Verify that the system displays task deadlines in the calendar view

* **Test Case Description**: This test case ensures that the system correctly displays task deadlines in the calendar view.
* **Test Suite**: Calendar Integration
* **Test Priority**: High
* **Preconditions**:
 + User has synced tasks with external calendar
 + User has created tasks with deadlines
* **Test Data**: Task deadlines
* **Test Steps**:
 1. Log in to the Task Management Tool as a user with synced tasks.
 2. Navigate to the calendar view.
 3. Verify that task deadlines are displayed in the calendar view.
 4. Check that task deadlines are accurately represented in the calendar view.
* **Postconditions**: Task deadlines are displayed in the calendar view.
* **Expected Result**: Task deadlines are accurately displayed in the calendar view.
* **Severity**: Major
* **Priority**: High
* **Type of Testing**: Functional Testing
* **Test Case Approach**: Positive

**Test Case 3: Verify that users can set reminders for tasks via the calendar**

* **Test Case ID**: TC-003
* **Test Case Title**: Verify that users can set reminders for tasks via the calendar
* **Test Case Description**: This test case ensures that users can successfully set reminders for tasks via the calendar.
* **Test Suite**: Calendar Integration

* **Test Priority**: High
* **Preconditions**:
 + User has synced tasks with external calendar
 + User has created tasks with deadlines
* **Test Data**: Task reminders
* **Test Steps**:
 1. Log in to the Task Management Tool as a user with synced tasks.
 2. Navigate to the calendar view.
 3. Click on a task with a deadline to view its details.
 4. Click on the "Set reminder" button and set a reminder for the task.
 5. Verify that the reminder is successfully set for the task.
* **Postconditions**: Reminder is set for task.
* **Expected Result**: Reminder is successfully set for the task.
* **Severity**: Major
* **Priority**: High
* **Type of Testing**: Functional Testing
* **Test Case Approach**: Positive

**Test Case 4: Verify that calendar events update automatically when tasks are changed**

* **Test Case ID**: TC-004
* **Test Case Title**: Verify that calendar events update automatically when tasks are changed
* **Test Case Description**: This test case ensures that calendar events automatically update when tasks are changed.
* **Test Suite**: Calendar Integration
* **Test Priority**: High
* **Preconditions**:

 + User has synced tasks with external calendar
 + User has created tasks with deadlines
* **Test Data**: Task changes
* **Test Steps**:
 1. Log in to the Task Management Tool as a user with synced tasks.
 2. Create a task with a deadline and description.
 3. Make changes to the task, such as updating the deadline or description.
 4. Verify that the calendar event is automatically updated to reflect the changes.
* **Postconditions**: Calendar event is updated to reflect task changes.
* **Expected Result**: Calendar event is automatically updated to reflect task changes.
* **Severity**: Critical
* **Priority**: High
* **Type of Testing**: Functional Testing
* **Test Case Approach**: Positive

**Test Case 5: Verify that users can manage calendar sync settings within the app**

* **Test Case ID**: TC-005
* **Test Case Title**: Verify that users can manage calendar sync settings within the app
* **Test Case Description**: This test case ensures that users can successfully manage calendar sync settings within the app.
* **Test Suite**: Calendar Integration
* **Test Priority**: High
* **Preconditions**: User has synced tasks with external calendar
* **Test Data**: Calendar sync settings
* **Test Steps**:
 1. Log in to the Task Management Tool as a user with synced tasks.

2. Navigate to the calendar settings page.

3. Verify that calendar sync settings can be managed and updated within the app.

4. Test updating calendar sync settings and verify that changes are successfully saved.

* **Postconditions**: Calendar sync settings are updated within the app.

* **Expected Result**: Calendar sync settings are successfully managed and updated within the app.

* **Severity**: Major

* **Priority**: High

* **Type of Testing**: Functional Testing

* **Test Case Approach**: Positive

---

## Data - 36

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Duplication

**Feature description -**

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

**Userstory title -**

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

**Acceptance criteria -**

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

**Generated Test Cases:**

Here are five detailed test cases according to the provided Test Case Creation Template, each addressing one or more acceptance criteria:

**Test Case 1: Verify that users can duplicate tasks or task templates**

* **Test Case ID:** TC-DUP-001
* **Test Case Title:** Verify that users can duplicate tasks or task templates
* **Test Case Description:** This test case aims to verify that users can successfully duplicate tasks or task templates to save time when creating similar tasks. We will cover the user's ability to duplicate a task and check that the duplicated task retains all original details.
* **Test Suite:** Task Management Tool - Task Duplication
* **Test Priority:** High
* **Preconditions:**
 + The user has a task or task template created in the system
 + The user has the necessary permissions to duplicate tasks
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as a user with the required permissions
 2. Open a task or task template to be duplicated
 3. Click on the duplicate button
 4. Verify that the duplicated task is created and appears in the task list

 5. Verify that the duplicated task retains all original details, including descriptions, assignees, and deadlines
* **Postconditions:**
 + The duplicated task is saved in the task list
 + The original task remains unchanged
* **Expected Result:** The user can successfully duplicate a task or task template, and the duplicated task retains all original details.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that the duplicated task clearly indicates that it has been duplicated**

* **Test Case ID:** TC-DUP-002
* **Test Case Title:** Verify that the duplicated task clearly indicates that it has been duplicated
* **Test Case Description:** This test case aims to verify that the duplicated task clearly indicates that it has been duplicated, making it easily identifiable in the task list. We will check if the system displays a clear indicator that the task has been duplicated.
* **Test Suite:** Task Management Tool - Task Duplication
* **Test Priority:** Medium
* **Preconditions:**
 + The user has a task or task template created in the system
 + The user has the necessary permissions to duplicate tasks
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as a user with the required permissions
 2. Duplicate a task or task template

 3. Verify that the duplicated task is displayed in the task list with a clear indicator that it has been duplicated (e.g., a duplicate icon)

* **Postconditions:**

 + The duplicated task is clearly identifiable as a duplicate

 + The original task remains unchanged

* **Expected Result:** The duplicated task clearly indicates that it has been duplicated.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can modify the duplicated task if needed**

* **Test Case ID:** TC-DUP-003

* **Test Case Title:** Verify that users can modify the duplicated task if needed

* **Test Case Description:** This test case aims to verify that users can modify the duplicated task as needed, without affecting the original task. We will check if the system allows users to make changes to the duplicated task.

* **Test Suite:** Task Management Tool - Task Duplication

* **Test Priority:** High

* **Preconditions:**

 + The user has a task or task template created in the system

 + The user has the necessary permissions to duplicate tasks and modify tasks

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as a user with the required permissions

 2. Duplicate a task or task template

3. Make changes to the duplicated task (e.g., update the description, assignee, or deadline)

4. Verify that the original task remains unchanged

* **Postconditions:**

+ The modified duplicated task is saved

+ The original task remains unchanged

* **Expected Result:** Users can modify the duplicated task as needed, without affecting the original task.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that duplicated tasks are saved and displayed in the task list**

* **Test Case ID:** TC-DUP-004

* **Test Case Title:** Verify that duplicated tasks are saved and displayed in the task list

* **Test Case Description:** This test case aims to verify that duplicated tasks are saved and displayed in the task list after duplication. We will check if the system saves and displays the duplicated task correctly.

* **Test Suite:** Task Management Tool - Task Duplication

* **Test Priority:** Medium

* **Preconditions:**

+ The user has a task or task template created in the system

+ The user has the necessary permissions to duplicate tasks

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a user with the required permissions

2. Duplicate a task or task template

3. Verify that the duplicated task is saved and displayed in the task list

* **Postconditions:**

 + The duplicated task is saved in the task list

* **Expected Result:** Duplicated tasks are saved and displayed in the task list.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that users cannot duplicate tasks without the required permissions**

* **Test Case ID:** TC-DUP-005

* **Test Case Title:** Verify that users cannot duplicate tasks without the required permissions

* **Test Case Description:** This test case aims to verify that users without the required permissions cannot duplicate tasks or task templates. We will check if the system restricts task duplication for users without the necessary permissions.

* **Test Suite:** Task Management Tool - Task Duplication

* **Test Priority:** High

* **Preconditions:**

 + The user does not have the necessary permissions to duplicate tasks

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as a user without the required permissions

 2. Attempt to duplicate a task or task template

 3. Verify that the system restricts task duplication and displays an error message

* **Postconditions:**

 + The user is unable to duplicate the task

 + The original task remains unchanged

* **Expected Result:** Users without the required permissions cannot duplicate tasks or task templates.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Negative

---

**Data - 37**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task History

**Feature description -**

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

**Userstory title -**

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

**Acceptance criteria -**

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

**Generated Test Cases:**

Here are five detailed test cases for the Task History feature, aligned with the provided user story and acceptance criteria:

**Test Case 1: Verify that Users Can View a Detailed History of Changes to Tasks**

Test Case ID: TH-001

Test Case Title: Verify that Users Can View a Detailed History of Changes to Tasks

Test Case Description: This test case will verify that users can view a detailed history of changes to tasks, including modifications, comments, and other relevant activity.

Test Suite: Task History

Test Priority: High

Preconditions:
* User has access to a task with a history of changes
* Task history is enabled

Test Data: No test data needed

Test Steps:

1. Log in as a user with access to a task with a history of changes.

2. Navigate to the task details page.

3. Click on the "Task History" tab.

4. Verify that the history shows the modifications, comments, and other relevant activity made to the task.

Postconditions:

* User can view a detailed history of changes to the task

* History includes modifications, comments, and other relevant activity

Expected Result: Users can view a detailed history of changes to tasks, including modifications, comments, and other relevant activity.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that Task History Includes Timestamps for All Changes**

Test Case ID: TH-002

Test Case Title: Verify that Task History Includes Timestamps for All Changes

Test Case Description: This test case will verify that the task history includes timestamps for all changes made to the task.

Test Suite: Task History

Test Priority: High

Preconditions:

* User has access to a task with a history of changes

* Task history is enabled

Test Data: No test data needed

Test Steps:

1. Log in as a user with access to a task with a history of changes.
2. Navigate to the task details page.
3. Click on the "Task History" tab.
4. Verify that each change in the history includes a timestamp.

Postconditions:

* Task history includes timestamps for all changes

Expected Result: Task history includes timestamps for all changes made to the task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that Users Can Filter the History by Type of Change or Date**

Test Case ID: TH-003

Test Case Title: Verify that Users Can Filter the History by Type of Change or Date

Test Case Description: This test case will verify that users can filter the task history by type of change or date.

Test Suite: Task History

Test Priority: Medium

Preconditions:

* User has access to a task with a history of changes

* Task history is enabled

Test Data: No test data needed

Test Steps:

1. Log in as a user with access to a task with a history of changes.

2. Navigate to the task details page.

3. Click on the "Task History" tab.

4. Select a filter option (e.g., "Modifications" or "Comments").

5. Verify that the history is filtered accordingly.

Postconditions:

* Users can filter the history by type of change or date

Expected Result: Users can filter the task history by type of change or date.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that Task History is Accessible from the Task Details Page**

Test Case ID: TH-004

Test Case Title: Verify that Task History is Accessible from the Task Details Page

Test Case Description: This test case will verify that task history is accessible from the task details page.

Test Suite: Task History

Test Priority: High

Preconditions:

* User has access to a task with a history of changes

* Task history is enabled

Test Data: No test data needed

Test Steps:

1. Log in as a user with access to a task with a history of changes.
2. Navigate to the task details page.
3. Verify that the "Task History" tab is present.

Postconditions:

* Task history is accessible from the task details page

Expected Result: Task history is accessible from the task details page.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that Task History Displays Correctly for Tasks with No Changes**

Test Case ID: TH-005

Test Case Title: Verify that Task History Displays Correctly for Tasks with No Changes

Test Case Description: This test case will verify that task history displays correctly for tasks with no changes.

Test Suite: Task History

Test Priority: Low

Preconditions:

* User has access to a task with no history of changes
* Task history is enabled

Test Data: No test data needed

Test Steps:

1. Log in as a user with access to a task with no history of changes.
2. Navigate to the task details page.
3. Click on the "Task History" tab.
4. Verify that the history is empty or displays a message indicating no changes.

Postconditions:

* Task history displays correctly for tasks with no changes

Expected Result: Task history displays correctly for tasks with no changes.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Positive

**-----------------------------------------------------------------------------------------------**

**Data - 38**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Notifications

**Feature description -**

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

**Userstory title -**

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

**Acceptance criteria -**

['Users receive notifications for task updates, comments, mentions, and other relevant activities.', 'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.', 'The system logs notification history for future reference.']

**Generated Test Cases:**

Here are five detailed test cases that cover the acceptance criteria for the Task Notifications feature:

**Test Case 1: Verify that users receive notifications for task updates**

| Test Case ID | Test Case Title | Test Case Description |
| --- | --- | --- |
| TC_001 | Verify that users receive notifications for task updates | This test case ensures that users receive notifications when a task is updated by another user. |

**Test Suite:** Task Notifications

**Test Priority:** High

**Preconditions:**

* The user is logged in to the task management tool

* A task is created and updated by another user

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user.

2. Create a task and update it by another user.

3. Verify that the user receives a notification for the task update.

4. Check the notification content to ensure it includes the task name, description, and update details.

**Postconditions:**

* The notification is sent to the user's email inbox or in-app alerts

* The notification history is logged in the system

**Expected Result:** The user receives a notification for the task update, and the notification history is logged in the system.

**Severity:** Major

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 2: Verify that users receive notifications for task comments**

| Test Case ID | Test Case Title | Test Case Description |
| --- | --- | --- |

| TC_002 | Verify that users receive notifications for task comments | This test case ensures that users receive notifications when a comment is added to a task by another user. |

**Test Suite:** Task Notifications

**Test Priority:** Medium

**Preconditions:**

* The user is logged in to the task management tool
* A task is created, and a comment is added by another user

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user.
2. Create a task and add a comment by another user.
3. Verify that the user receives a notification for the comment.
4. Check the notification content to ensure it includes the task name, comment details, and commenter information.

**Postconditions:**

* The notification is sent to the user's email inbox or in-app alerts
* The notification history is logged in the system

**Expected Result:** The user receives a notification for the task comment, and the notification history is logged in the system.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 3: Verify that users receive notifications for task mentions**

| Test Case ID | Test Case Title | Test Case Description |
| --- | --- | --- |
| TC_003 | Verify that users receive notifications for task mentions | This test case ensures that users receive notifications when they are mentioned in a task or comment by another user. |

**Test Suite:** Task Notifications

**Test Priority:** Medium

**Preconditions:**

* The user is logged in to the task management tool
* A task or comment mentions the user by their username or full name

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user.
2. Create a task or comment that mentions the user by their username or full name.
3. Verify that the user receives a notification for the mention.
4. Check the notification content to ensure it includes the task name, comment details, and mention information.

**Postconditions:**

* The notification is sent to the user's email inbox or in-app alerts
* The notification history is logged in the system

**Expected Result:** The user receives a notification for the task mention, and the notification history is logged in the system.

**Severity:** Minor
**Priority:** Medium
**Type of Testing:** Functional Testing
**Test Case Approach:** Positive

**Test Case 4: Verify that users receive notifications for other relevant activities**

| Test Case ID | Test Case Title | Test Case Description |
| --- | --- | --- |
| TC_004 | Verify that users receive notifications for other relevant activities | This test case ensures that users receive notifications for other relevant activities, such as task assignments, deadline updates, and project categorizations. |

**Test Suite:** Task Notifications
**Test Priority:** Medium
**Preconditions:**

* The user is logged in to the task management tool

* A relevant activity occurs (e.g., task assignment, deadline update, project categorization)

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user.
2. Create a relevant activity (e.g., task assignment, deadline update, project categorization).
3. Verify that the user receives a notification for the activity.
4. Check the notification content to ensure it includes the task name, activity details, and relevant information.

**Postconditions:**

* The notification is sent to the user's email inbox or in-app alerts
* The notification history is logged in the system

**Expected Result:** The user receives a notification for the relevant activity, and the notification history is logged in the system.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 5: Verify that users can customize notification settings**

| Test Case ID | Test Case Title | Test Case Description |
| --- | --- | --- |

| TC_005 | Verify that users can customize notification settings | This test case ensures that users can customize notification settings based on their preferences. |

**Test Suite:** Task Notifications

**Test Priority:** Medium

**Preconditions:**

* The user is logged in to the task management tool
* The user has permissions to customize notification settings

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user.
2. Navigate to the notification settings page.
3. Verify that the user can customize notification settings, including notification types, frequency, and format.
4. Test the custom notification settings to ensure they are applied correctly.

**Postconditions:**

* The user's notification settings are saved correctly
* The user receives notifications according to their custom settings

**Expected Result:** The user can customize notification settings, and notifications are received according to their custom settings.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

These test cases cover all the acceptance criteria for the Task Notifications feature, ensuring that users receive notifications for task updates, comments, mentions, and other relevant activities. Additionally, users can customize notification settings based on their preferences, and the system logs notification history for future reference.

---

**Data - 39**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Archiving

**Feature description -**

Completed or inactive tasks can be archived or moved to a separate section, decluttering the task list and maintaining focus on active tasks. Task archiving helps users keep their task lists organized and prioritized, without losing access to important historical data.

**Userstory title -**

As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks.

**Acceptance criteria -**

['Users can archive completed or inactive tasks.', 'Archived tasks are moved to a separate section for easy access.', 'Users can search and view archived tasks.', 'The system provides an option to restore archived tasks to the active list.', 'Archived tasks retain all original details and history.']

**Generated Test Cases:**

Here are the test cases for the Task Management Tool, covering the Feature: Task Archiving and User Story: "As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks."

**Test Case 1: Verify that Users Can Archive Completed or Inactive Tasks**

Test Case ID: TCTA001

Test Case Title: Verify that users can archive completed or inactive tasks

Test Case Description: In this test case, we will verify that users can successfully archive completed or inactive tasks, and that these tasks are moved to a separate section for easy access.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- The user has completed or inactive tasks on their task list
- The user has permission to archive tasks

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user with permission to archive tasks
2. Select a completed or inactive task from the task list
3. Click the "Archive" button on the task details page
4. Verify that the task has been moved to the archived section
5. Repeat steps 2-4 for multiple tasks

Postconditions:

* The tasks are successfully archived
* The tasks are not visible on the active task list

Expected Result: The user can archive completed or inactive tasks and these tasks are moved to a separate section for easy access.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that Archived Tasks are Moved to a Separate Section for Easy Access**

Test Case ID: TCTA002

Test Case Title: Verify that archived tasks are moved to a separate section for easy access

Test Case Description: In this test case, we will verify that archived tasks are correctly moved to a separate section on the task list and can be easily accessed.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:

* Multiple tasks have been archived
* The user has permission to access archived tasks

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user with permission to access archived tasks
2. Click on the "Archived" tab on the task list
3. Verify that archived tasks are listed and can be filtered, sorted, and searched
4. Verify that the task details page displays archived task history

Postconditions:

* The tasks are successfully displayed in the archived section
* The user can view archived task details and history

Expected Result: Archived tasks are moved to a separate section and can be easily accessed by users.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that Users Can Search and View Archived Tasks**

Test Case ID: TCTA003

Test Case Title: Verify that users can search and view archived tasks

Test Case Description: In this test case, we will verify that users can successfully search for and view archived tasks, and that search results include all archived task details.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:

* Multiple tasks have been archived
* The user has permission to access archived tasks

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user with permission to access archived tasks
2. Search for an archived task using various criteria (e.g., task title, description, date completed)
3. Verify that search results include all archived task details and history

Postconditions:

* The search results accurately include all archived task details
* The user can view archived task history and details

Expected Result: Users can successfully search and view archived tasks and include all task details in the search results.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that the System Provides an Option to Restore Archived Tasks**

Test Case ID: TCTA004

Test Case Title: Verify that the system provides an option to restore archived tasks

Test Case Description: In this test case, we will verify that users can successfully restore archived tasks and that these tasks are reinserted into the active task list with their original details.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:

* One or more tasks have been archived
* The user has permission to restore tasks

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user with permission to restore tasks
2. Select an archived task and click the "Restore" button
3. Verify that the task has been restored to the active task list with original details and history

Postconditions:

* The task is successfully restored
* The task's original details and history are intact

Expected Result: The system provides an option to restore archived tasks and restore tasks are successfully inserted back into the active task list with original details and history.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that Archived Tasks Retain All Original Details and History**

Test Case ID: TCTA005

Test Case Title: Verify that archived tasks retain all original details and history

Test Case Description: In this test case, we will verify that archived tasks maintain their original details, attachments, and history even after archiving and restoration.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- Multiple tasks have been archived
- The user has permission to view task details and history

Test Data: No test data needed

Test Steps:

1. Log in to the Task Management Tool as a user with permission to view task details and history
2. Select an archived task and view task details and history
3. Verify that the task details and history are accurate and intact

Postconditions:

- The task details and history are correctly preserved in the archived state
- The task history is updated after task restoration

Expected Result: Archived tasks retain all original details and history, ensuring that users can continue to view and manage task history and activity.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

These test cases thoroughly cover all the acceptance criteria for the Task Archiving feature, specifically ensuring that users can archive completed or inactive tasks, view archived tasks, search archived tasks, restore archived tasks, and that archived tasks retain all original details and history.

**-------------------------------------------------------------------------------------------------**

**Data - 40**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Guest Access

**Feature description -**

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

**Userstory title -**

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

**Acceptance criteria -**

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

**Generated Test Cases:**

Here are five detailed test cases for the Guest Access feature, each covering one or more of the acceptance criteria:

**Test Case 1: Verify that users can invite external collaborators as guests with limited access**

* **Test Case ID:** GAT-001
* **Test Case Title:** Verify that users can invite external collaborators as guests with limited access
* **Test Case Description:** This test case ensures that users can successfully invite external collaborators as guests with limited access, allowing them to view and collaborate on specific tasks or projects.
* **Test Suite:** Guest Access
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the platform
 + User has a project or task to share with guests
* **Test Data:** User's email address, guest's email address
* **Test Steps:**
 1. Log in to the platform as a user
 2. Navigate to the project or task to be shared
 3. Click on the "Invite Guest" button
 4. Enter the guest's email address and select the desired access level (e.g., "View Only")

 5. Click on the "Send Invitation" button

* **Postconditions:**

 + Guest receives an email invitation to join the project or task

 + Guest can view and collaborate on the project or task

* **Expected Result:** Guest receives an email invitation and can view and collaborate on the project or task

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that guests can view and collaborate on specific tasks or projects**

* **Test Case ID:** GAT-002

* **Test Case Title:** Verify that guests can view and collaborate on specific tasks or projects

* **Test Case Description:** This test case ensures that guests can successfully view and collaborate on specific tasks or projects, as intended.

* **Test Suite:** Guest Access

* **Test Priority:** High

* **Preconditions:**

 + Guest has been invited to a project or task

 + Guest has accepted the invitation

* **Test Data:** Guest's email address, project or task details

* **Test Steps:**

 1. Log in to the platform as a guest

 2. Navigate to the project or task

 3. Verify that the guest can view the project or task details

 4. Verify that the guest can collaborate on the project or task (e.g., add comments, attach files)

* **Postconditions:**

 + Guest can view and collaborate on the project or task

* **Expected Result:** Guest can view and collaborate on the project or task

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that the system provides clear distinctions between guest and full access roles**

* **Test Case ID:** GAT-003

* **Test Case Title:** Verify that the system provides clear distinctions between guest and full access roles

* **Test Case Description:** This test case ensures that the system clearly distinguishes between guest and full access roles, preventing guests from accessing sensitive or restricted areas of the platform.

* **Test Suite:** Guest Access

* **Test Priority:** High

* **Preconditions:**

 + Guest has been invited to a project or task

 + Guest has accepted the invitation

* **Test Data:** Guest's email address, project or task details

* **Test Steps:**

 1. Log in to the platform as a guest

 2. Attempt to access a sensitive or restricted area of the platform (e.g., project settings, user management)

 3. Verify that the guest is prevented from accessing the sensitive or restricted area

* **Postconditions:**

 + Guest is prevented from accessing sensitive or restricted areas of the platform

* **Expected Result:** Guest is prevented from accessing sensitive or restricted areas of the platform

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that users can manage and revoke guest access at any time**

* **Test Case ID:** GAT-004

* **Test Case Title:** Verify that users can manage and revoke guest access at any time

* **Test Case Description:** This test case ensures that users can successfully manage and revoke guest access at any time, as intended.

* **Test Suite:** Guest Access

* **Test Priority:** High

* **Preconditions:**

 + Guest has been invited to a project or task

 + Guest has accepted the invitation

* **Test Data:** User's email address, guest's email address

* **Test Steps:**

 1. Log in to the platform as a user

 2. Navigate to the project or task

 3. Click on the "Manage Guests" button

 4. Select the guest to revoke access from

 5. Click on the "Revoke Access" button

* **Postconditions:**

 + Guest's access is revoked

 + Guest is no longer able to view or collaborate on the project or task

* **Expected Result:** Guest's access is revoked

* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that guests cannot access sensitive or restricted areas of the platform**

* **Test Case ID:** GAT-005
* **Test Case Title:** Verify that guests cannot access sensitive or restricted areas of the platform
* **Test Case Description:** This test case ensures that guests cannot access sensitive or restricted areas of the platform, as intended.
* **Test Suite:** Guest Access
* **Test Priority:** High
* **Preconditions:**
 + Guest has been invited to a project or task
 + Guest has accepted the invitation
* **Test Data:** Guest's email address, project or task details
* **Test Steps:**
 1. Log in to the platform as a guest
 2. Attempt to access a sensitive or restricted area of the platform (e.g., project settings, user management)
 3. Verify that the guest is prevented from accessing the sensitive or restricted area
* **Postconditions:**
 + Guest is prevented from accessing sensitive or restricted areas of the platform
* **Expected Result:** Guest is prevented from accessing sensitive or restricted areas of the platform
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 41**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Import/Export

**Feature description -**

Users can import tasks from external sources or export tasks to other task management tools or file formats. Import/export capabilities facilitate data migration and integration with other systems, ensuring seamless collaboration and workflow management across platforms.

**Userstory title -**

As a user, I want to import and export tasks so that I can migrate data between different task management tools.

**Acceptance criteria -**

['Users can import tasks from external sources in various formats (e.g., CSV, Excel).', 'Users can export tasks to other task management tools or file formats.', 'The system provides clear mapping of fields during import/export.', 'Users can preview imported data before finalizing the process.', 'The system logs import/export activities for future reference.']

**Generated Test Cases:**

Here are five detailed test cases that cover all the acceptance criteria for the Task Import/Export feature in the Task Management Tool:

**Test Case 1: Import Tasks from CSV File**

* Test Case ID: TCE01
* Test Case Title: Verify that the system can import tasks from an external source in CSV format.
* Test Suite: Task Import/Export
* Test Priority: High
* Preconditions:
  + User has created a new task list in the system.
  + User has an external CSV file containing task data.
* Test Data: Import test data from a CSV file with 5 tasks.
* Test Steps:
  1. Log in to the system as a valid user.
  2. Navigate to the task import page.
  3. Select the CSV file from the external source.
  4. Map the CSV fields to the system fields.
  5. Click "Import" to import the tasks.
  6. Verify that the tasks are successfully imported into the task list.
* Postconditions:
  + Verify that the imported tasks are correctly displayed in the task list.
  + Verify that the system maintains the correct task data structure after import.
* Expected Result: Tasks are imported correctly from the external source in CSV format.
* Severity: Critical
* Priority: High
* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 2: Export Tasks to Excel File**

* Test Case ID: TCE02
* Test Case Title: Verify that the system can export tasks to an external source in Excel format.
* Test Suite: Task Import/Export
* Test Priority: High
* Preconditions:
 + User has created a new task list in the system with at least 5 tasks.
* Test Data: Export test data from the system for 5 tasks.
* Test Steps:
 1. Log in to the system as a valid user.
 2. Navigate to the task export page.
 3. Select the Excel file as the export format.
 4. Click "Export" to export the tasks.
 5. Verify that the Excel file is downloaded correctly.
 6. Open the Excel file and verify that the task data is correctly exported.
* Postconditions:
 + Verify that the Excel file is correctly formatted with the task data.
 + Verify that the system maintains the correct task data structure after export.
* Expected Result: Tasks are exported correctly to an external source in Excel format.
* Severity: Critical
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 3: Import/Export with Clear Mapping of Fields**

* Test Case ID: TCE03

* Test Case Title: Verify that the system provides clear mapping of fields during import/export.

* Test Suite: Task Import/Export

* Test Priority: Medium

* Preconditions:

 + User has created a new task list in the system with at least 5 tasks.

* Test Data: Use a CSV file with 5 tasks that requires field mapping.

* Test Steps:

 1. Log in to the system as a valid user.

 2. Navigate to the task import page.

 3. Select the CSV file from the external source.

 4. Verify that the system displays a clear mapping of fields during import.

 5. Map the CSV fields to the system fields.

 6. Click "Import" to import the tasks.

 7. Verify that the tasks are successfully imported into the task list.

* Postconditions:

 + Verify that the imported tasks are correctly displayed in the task list.

 + Verify that the system maintains the correct task data structure after import.

* Expected Result: System provides clear field mapping during import.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Preview Imported Data**

* Test Case ID: TCE04

* Test Case Title: Verify that the system allows users to preview imported data before finalizing the import process.

* Test Suite: Task Import/Export

* Test Priority: Medium

* Preconditions:

 + User has created a new task list in the system.

 + User has an external CSV file containing task data that requires previewing.

* Test Data: Import test data from a CSV file that requires previewing.

* Test Steps:

 1. Log in to the system as a valid user.

 2. Navigate to the task import page.

 3. Select the CSV file from the external source.

 4. Click "Preview" to view the imported data.

 5. Verify that the system displays the imported data correctly.

 6. Click "Import" to import the tasks.

* Postconditions:

 + Verify that the tasks are successfully imported into the task list.

 + Verify that the system maintains the correct task data structure after import.

* Expected Result: System allows users to preview imported data before finalizing the import process.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 5: Log Import/Export Activities**

* Test Case ID: TCE05

* Test Case Title: Verify that the system logs import/export activities for future reference.

* Test Suite: Task Import/Export

* Test Priority: Low

* Preconditions:

 + User has imported/exported tasks using the system.

* Test Data: Not needed.

* Test Steps:

 1. Log in to the system as a valid user.

 2. Navigate to the task list.

 3. Verify that the system displays a log of import/export activities.

 4. Click on each activity to view the details.

 5. Verify that the system maintains the correct log of activities.

* Postconditions:

 + No postconditions.

* Expected Result: System logs import/export activities for future reference.

* Severity: Trivial

* Priority: Low

* Type of Testing: Functional Testing

* Test Case Approach: Positive

---

**Data - 42**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Prioritization

**Feature description -**

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.

**Userstory title -**

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

**Acceptance criteria -**

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.', 'Priority changes are logged and tracked.']

**Generated Test Cases:**

Here are five detailed test cases, each covering one or more of the acceptance criteria from the user story:

**Test Case 1: Verify that users can set and adjust task priorities**

* **Test Case ID:** TCV001

* **Test Case Title:** Verify that users can set and adjust task priorities

* **Test Case Description:** As a user, I want to be able to set and adjust task priorities to ensure that my tasks are properly ordered and managed. This test case covers the acceptance criteria that users can set and adjust task priorities.

* **Test Suite:** Task Prioritization
* **Test Priority:** High
* **Preconditions:**
 + User is logged in and has access to the Task Management Tool
 + Task is created and saved
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool and navigate to the task list
 2. Open a task and click on the "Set priority" button
 3. Select a priority level from the dropdown menu (e.g. High, Medium, Low)
 4. Click "Save" to save the priority change
 5. Repeat steps 3-4 multiple times to test different priority levels
* **Postconditions:** Task priority is updated and reflected in the task list
* **Expected Result:** User can successfully set and adjust task priorities
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that tasks can be sorted and filtered based on priority levels**

* **Test Case ID:** TCV002
* **Test Case Title:** Verify that tasks can be sorted and filtered based on priority levels
* **Test Case Description:** As a user, I want to be able to sort and filter tasks based on priority levels to quickly identify high-priority tasks. This test case covers the acceptance criterion that tasks can be sorted and filtered based on priority levels.
* **Test Suite:** Task Prioritization

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in and has access to the Task Management Tool

 + Multiple tasks are created and saved with different priority levels

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the Task Management Tool and navigate to the task list

 2. Sort the task list by priority (e.g. High, Medium, Low)

 3. Verify that tasks with the same priority level are grouped together

 4. Filter the task list by priority level (e.g. only show High-priority tasks)

 5. Verify that only tasks with the selected priority level are displayed

* **Postconditions:** Task list is sorted and filtered by priority level

* **Expected Result:** Tasks are sorted and filtered correctly based on priority levels

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that the system provides visual indicators of task priority**

* **Test Case ID:** TCV003

* **Test Case Title:** Verify that the system provides visual indicators of task priority

* **Test Case Description:** As a user, I want to see clear visual indicators of task priority to quickly understand the importance of each task. This test case covers the acceptance criterion that the system provides visual indicators of task priority.

* **Test Suite:** Task Prioritization

* **Test Priority:** Medium

* **Preconditions:**
 + User is logged in and has access to the Task Management Tool
 + Tasks are created and saved with different priority levels
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool and navigate to the task list
 2. Verify that tasks with different priority levels have distinct visual indicators (e.g. color, icon, etc.)
 3. Click on a task with a specific priority level and verify that the visual indicator is displayed correctly
* **Postconditions:** Tasks have clear visual indicators of priority level
* **Expected Result:** Visual indicators represent task priority correctly
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that users receive notifications for high-priority tasks**

* **Test Case ID:** TCV004
* **Test Case Title:** Verify that users receive notifications for high-priority tasks
* **Test Case Description:** As a user, I want to receive notifications when high-priority tasks are approaching deadlines to stay on track. This test case covers the acceptance criterion that users receive notifications for high-priority tasks.
* **Test Suite:** Task Prioritization
* **Test Priority:** High
* **Preconditions:**
 + User is logged in and has access to the Task Management Tool
 + High-priority task is created and saved with a looming deadline

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the Task Management Tool and navigate to the task list

 2. Set up a test notification for high-priority tasks (e.g. send an email)

 3. Verify that a notification is received when the task deadline is approaching

* **Postconditions:** User receives notification for high-priority task

* **Expected Result:** User receives timely notification for high-priority task

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that priority changes are logged and tracked**

* **Test Case ID:** TCV005

* **Test Case Title:** Verify that priority changes are logged and tracked

* **Test Case Description:** As a user, I want to see a record of priority changes for each task to audit and track progress. This test case covers the acceptance criterion that priority changes are logged and tracked.

* **Test Suite:** Task Prioritization

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in and has access to the Task Management Tool

 + Task is created and saved with an initial priority level

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the Task Management Tool and navigate to the task list

 2. Change the priority level of a task

3. Verify that the priority change is logged and displayed in the task history

* **Postconditions:** Priority change is recorded and tracked

* **Expected Result:** Priority change is accurately logged and displayed

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**------------------------------------------------------------------------------------------------**

**Data - 43**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Collaboration

**Feature description -**

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

**Userstory title -**

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay updated on task progress.

**Acceptance criteria -**

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

**Generated Test Cases:**

Here are the five test cases for the Task Collaboration feature in the Task Management Tool project, following the Test Case Creation Template:

**Test Case 1: Verify that users can assign tasks to team members**

Test Case ID: T-001-TC-001
Test Case Title: Verify that users can assign tasks to team members
Test Case Description: This test case aims to check if the system allows users to assign tasks to team members and whether the team members receive notifications about the assigned task. The user should be able to select team members from the user list, add them to the task assignment, and see the task assignment reflected in the task details.
Test Suite: Task Collaboration Feature
Test Priority: High
Preconditions:

* User has created a task and added a team member
* Team member is available in the user list

Test Data: No test data needed
Test Steps:

1. Log in as a user and create a new task.
2. Click on the 'Assign Task' button and select the team member from the list.
3. Verify that the selected team member appears in the task assignment section.

4. Send an email notification to the team member to check if they receive a notification about the task assignment.

Postconditions:

* Team member receives a notification about the assigned task
* Task assignment is visible to the team member in the task details

Expected Result: The system allows users to assign tasks to team members, and team members receive notifications about assigned tasks.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that users can share updates and communicate through comments or messages**

Test Case ID: T-001-TC-002

Test Case Title: Verify that users can share updates and communicate through comments or messages

Test Case Description: This test case aims to check if users can share task updates and communicate with team members through comments or messages. The user should be able to add comments, messages, or attachments to the task, and team members should be able to view and respond to these updates.

Test Suite: Task Collaboration Feature

Test Priority: Medium

Preconditions: User has created a task and added a team member

Test Data:

* Comment or message: Example message "Hello, please provide an update on the task"

Test Steps:

1. Log in as a user and access the task details.
2. Click on the 'Add Comment or Message' button and enter a message.
3. Verify that the message appears in the comment section.
4. Send a response from the team member's side and verify that the system displays the response.

Postconditions: Task updates are visible to assigned team members, and team members can respond to updates.

Expected Result: Users can share task updates and communicate with team members through comments or messages.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that task updates and comments are visible to assigned team members**

Test Case ID: T-001-TC-003

Test Case Title: Verify that task updates and comments are visible to assigned team members

Test Case Description: This test case aims to check if task updates and comments are visible to assigned team members when viewing the task details. The assigned team members should see updates to the task, and any comments or messages should be viewable.

Test Suite: Task Collaboration Feature

Test Priority: Medium

Preconditions: User has created a task and added a team member

Test Data:

- Task update: Description of task changes

Test Steps:

1. Log in as an assigned team member and view the task details.
2. Verify that the task update appears in the task description section.
3. Verify that comments and messages are visible in the comment section.

Postconditions: Task updates and comments are visible to assigned team members.

Expected Result: Assigned team members can view task updates and comments when viewing the task details.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that users receive notifications for task updates and comments**

Test Case ID: T-001-TC-004

Test Case Title: Verify that users receive notifications for task updates and comments

Test Case Description: This test case aims to check if users receive notifications when task updates or comments are added. The user should receive an email notification with the task update or comment details.

Test Suite: Task Collaboration Feature

Test Priority: High

Preconditions: User has created a task and added a team member

Test Data: No test data needed

Test Steps:

1. Log in as a user and create a task.
2. Send a task update and comment from the user.
3. Verify that the user receives an email notification with the task update and comment details.

Postconditions: User receives a notification about task updates and comments.

Expected Result: Users receive notifications for task updates and comments.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that the system provides a clear view of task collaboration history**

Test Case ID: T-001-TC-005

Test Case Title: Verify that the system provides a clear view of task collaboration history

Test Case Description: This test case aims to check if the system provides a clear view of task collaboration history, including task updates, comments, and assignments. The user should be able to view the collaboration history for each task.

Test Suite: Task Collaboration Feature

Test Priority: High

Preconditions: User has created a task and added a team member

Test Data: No test data needed

Test Steps:

1. Log in as a user and access the task details.
2. Click on the 'Collaboration History' button.
3. Verify that the system displays a clear view of task collaboration history.

Postconditions: Clear view of task collaboration history for each task.

Expected Result: The system provides a clear view of task collaboration history for each task.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**-----------------------------------------------------------------------------------------------**

**Data - 44**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Tracking

**Feature description -**

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

**Userstory title -**

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

**Acceptance criteria -**

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.', 'System provides visual indicators for task progress.', '']

**Generated Test Cases:**

Below are five detailed test cases for the User Story "As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion" for the Task Management Tool's Task Tracking feature. Each test case targets at least one of the acceptance criteria and follows the Test Case Creation Template.

### Test Case 1: Verify the Task Status Viewing Feature

**Test Case ID**: TCR01

**Test Case Title**: Verify that users can view task statuses.

**Test Case Description**: This test case is designed to verify that users can view the status of tasks. The test will simulate user actions of accessing a task and checking that the current status is correctly displayed.

**Test Suite**: Task Tracking

**Test Priority**: High

**Preconditions**:

* A task must be created and saved.

* The user must have permission to view tasks.

**Test Data**: No specific data, unless otherwise noted.

**Test Steps**:

1. Login to the Task Management Tool with valid credentials.

2. Navigate to the Tasks or Projects area where the created task is accessible.

3. Click on the task to view its details.

4. Verify that the task status is correctly displayed.

5. Confirm that the status can be updated correctly by completing the task.

**Postconditions**:

* Confirm that the task status reflects its new status after completion.

**Expected Result**: Users can view task statuses, and the task status correctly reflects the task's current status.

**Severity**: Major

**Priority**: High

**Type of Testing**: Functional Testing

**Test Case Approach**: Positive

### Test Case 2: Validate Task Progress Tracking and Deadlines

**Test Case ID**: TCR02

**Test Case Title**: Verify that users can track task progress and deadlines.

**Test Case Description**: This test case is designed to ensure that users can see the progress of tasks and see the deadlines associated with these tasks.

**Test Suite**: Task Tracking

**Test Priority**: High

**Preconditions**:

* A task must be created with a deadline and progress indicator (i.e., % completion).

* The user must have permission to view tasks' deadlines and progress.

**Test Data**: No specific data, unless otherwise noted.

**Test Steps**:

1. Create a new task with a deadline and set its progress to a specific percentage.

2. Login to the Task Management Tool with the created task.

3. Navigate to the task and check the deadline and progress displayed against the set values.

4. Confirm that the deadlines and progress are correctly reflected.

**Postconditions**: None

**Expected Result**: Users can track the progress of tasks and deadlines, which are correctly managed.

**Severity**: Major

**Priority**: High

**Type of Testing**: Functional Testing

**Test Case Approach**: Positive

### Test Case 3: Task Update in Real-Time

**Test Case ID**: TCR03

**Test Case Title**: Verify that tasks are updated in real-time.

**Test Case Description**: This test case aims to ensure that any changes made to a task are reflected immediately across the Task Management Tool.

**Test Suite**: Task Tracking

**Test Priority**: High

**Preconditions**:

* Any user.

**Test Data**: No specific data, unless otherwise noted.

**Test Steps**:

1. Start a task and observe its initial status.

2. While observing, have another user or the same user update the task status.

3. Confirm that both users can see the updated task status in real-time.

**Postconditions**: None

**Expected Result**: Tasks are updated in real-time by multiple users.

**Severity**: Major

**Priority**: High

**Type of Testing**: Functional Testing

**Test Case Approach**: Positive

### Test Case 4: Visual Indicators for Task Progress

**Test Case ID**: TCR04

**Test Case Title**: Verify that the system provides visual indicators for task progress.

**Test Case Description**: This test case ensures that the Task Management Tool displays relevant visual indicators for the progress of tasks, aiding in quick task status assessment.

**Test Suite**: Task Tracking

**Test Priority**: Medium

**Preconditions**:

* A task must be created and saved with a progress indicator.

* The user must be logged in with permission to view tasks.

**Test Data**: No specific data, unless otherwise noted.

**Test Steps**:

1. Login to the Task Management Tool with valid credentials.

2. Navigate to the Tasks or Projects area.

3. Check if visual indicators (e.g., progress bars, icons) are displayed alongside task statuses indicating progress.

**Postconditions**: None

**Expected Result**: The system provides clear visual indicators of task progress.

**Severity**: Minor

**Priority**: Medium

**Type of Testing**: Functional Testing

**Test Case Approach**: Positive

### Test Case 5: Inaccurate or Missing Task Status

**Test Case ID**: TCR05

**Test Case Title**: Verify that task statuses are accurate or display a clear error indicator if the status is unknown or incorrect.

**Test Case Description**: This test case focuses on ensuring that if a task status cannot be determined (e.g., due to a network issue or incorrect entry), the application provides a clear error or indicator of uncertainty.

**Test Suite**: Task Tracking

**Test Priority**: High

**Preconditions**:

* A network issue must be simulated for a specific task.

* The user with a task status error condition while accessing the task.

**Test Data**: Simulated network issue for a task.

**Test Steps**:

1. Simulate a network error or incorrect access to a task's status.

2. Check the display to confirm an accurate message indicating the difficulty.

3. Also, ensure that no details are missing that are necessary to the user.

**Postconditions**: None

**Expected Result**: The system displays accurate and clear error messages for unknown or missing task status information.

**Severity**: Major

**Priority**: High

**Type of Testing**: Functional Testing

**Test Case Approach**: Negative

--------------------------------------------------------------------------------------------------

**Data - 45**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Notifications

**Feature description -**

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

**Userstory title -**

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

**Acceptance criteria -**

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.', 'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', '']

**Generated Test Cases:**

Here are five detailed test cases for the Task Notifications feature:

**Test Case 1: Verify User Receives Notifications for New Tasks**

* **Test Case ID:** TCE-NOT-001
* **Test Case Title:** Verify that users receive notifications for new tasks
* **Test Case Description:** This test case aims to verify that users receive notifications when new tasks are assigned to them. We will test the notification functionality by creating a new task and checking if the user receives a notification.
* **Test Suite:** Notification Testing
* **Test Priority:** High
* **Preconditions:**
 + User is logged in
 + Task assignment feature is enabled
* **Test Data:** No test data needed
* **Test Steps:**
 1. Create a new task for the user
 2. Check the notification center for incoming notifications
 3. Verify that the notification indicates a new task assignment
 4. Verify that the notification includes the task title, description, and deadline
* **Postconditions:** User is logged out
* **Expected Result:** User receives a notification for the new task
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify User Receives Notifications for Task Updates and Comments**

* **Test Case ID:** TCE-NOT-002
* **Test Case Title:** Verify that users receive notifications for task updates and comments
* **Test Case Description:** This test case aims to verify that users receive notifications for task updates and comments. We will test the notification functionality by making updates to an existing task and checking if the user receives a notification.
* **Test Suite:** Notification Testing
* **Test Priority:** Medium
* **Preconditions:**
 + User is logged in
 + Task exists with updates and comments
* **Test Data:** No test data needed
* **Test Steps:**
 1. Update an existing task with new details
 2. Add a comment to the task
 3. Check the notification center for incoming notifications
 4. Verify that the notification indicates a task update or comment
 5. Verify that the notification includes the task title and description
* **Postconditions:** User is logged out
* **Expected Result:** User receives a notification for the task update or comment
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify User Receives Reminders for Upcoming Deadlines**

* **Test Case ID:** TCE-NOT-003
* **Test Case Title:** Verify that users receive reminders for upcoming deadlines
* **Test Case Description:** This test case aims to verify that users receive reminders for upcoming deadlines. We will test the notification functionality by setting a deadline for a task and checking if the user receives a reminder notification.
* **Test Suite:** Notification Testing
* **Test Priority:** High
* **Preconditions:**
 + User is logged in
 + Task exists with a deadline
* **Test Data:** No test data needed
* **Test Steps:**
 1. Set a deadline for an existing task
 2. Check the notification center for incoming notifications
 3. Verify that the notification indicates an upcoming deadline
 4. Verify that the notification includes the task title, description, and deadline
* **Postconditions:** User is logged out
* **Expected Result:** User receives a reminder notification for the upcoming deadline
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify Notification Settings Can Be Customized by Users**

* **Test Case ID:** TCE-NOT-004

* **Test Case Title:** Verify that notification settings can be customized by users

* **Test Case Description:** This test case aims to verify that users can customize their notification settings. We will test the notification settings feature by allowing users to customize their notification preferences and checking if the changes take effect.

* **Test Suite:** Notification Testing

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in

 + Notification settings feature is enabled

* **Test Data:** No test data needed

* **Test Steps:**

 1. Access the notification settings page

 2. Modify notification preferences (e.g., turn off notifications for task updates)

 3. Create a new task or trigger a task update

 4. Check the notification center for incoming notifications (or lack thereof)

* **Postconditions:** User is logged out

* **Expected Result:** User's notification settings are updated and reflected in the notification center

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify User is Not Overwhelmed with Notifications**

* **Test Case ID:** TCE-NOT-005

* **Test Case Title:** Verify that the user is not overwhelmed with notifications

* **Test Case Description:** This test case aims to verify that the notification system does not overwhelm the user with too many notifications. We will test the notification system by generating multiple notifications and checking if the user is able to manage the notifications effectively.

* **Test Suite:** Notification Testing

* **Test Priority:** Low

* **Preconditions:**

 + User is logged in

 + Multiple tasks exist with new tasks, updates, and comments

* **Test Data:** No test data needed

* **Test Steps:**

 1. Generate multiple notifications (e.g., create new tasks, update existing tasks, add comments)

 2. Check the notification center for incoming notifications

 3. Verify that the user can manage the notifications (e.g., mark as read, dismiss)

 4. Verify that the notification system does not overwhelm the user

* **Postconditions:** User is logged out

* **Expected Result:** User is able to manage notifications effectively and is not overwhelmed

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 46**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Categorization

**Feature description -**

Users can categorize tasks into different groups or projects for better organization and management. Task categorization aims to help users organize tasks by type, priority, or project.

**Userstory title -**

As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

**Acceptance criteria -**

['Users can create and assign tasks to different categories or projects.', 'Users can view tasks grouped by categories or projects.', 'Tasks can be moved between categories or projects.', 'Category or project information is accurately reflected in task views.', '']

**Generated Test Cases:**

Here are five detailed test cases for the Task Categorization feature, each covering one or more acceptance criteria:

**Test Case 1: Verify that users can create and assign tasks to different categories or projects**

* Test Case ID: TC-TCT-001
* Test Case Title: Verify that users can create and assign tasks to different categories or projects
* Test Case Description: This test case ensures that users can create new tasks and assign them to different categories or projects, as per the acceptance criterion.

* Test Suite: Task Categorization

* Test Priority: High

* Preconditions:

 + User is logged in to the Task Management Tool

 + User has permission to create and assign tasks

* Test Data: No test data needed

* Test Steps:

 1. Log in to the Task Management Tool as a user with permission to create and assign tasks.

 2. Navigate to the task creation page and fill in the required fields (task name, description, deadline, etc.).

 3. Select a category or project from the dropdown list and assign the task to it.

 4. Verify that the task is assigned to the selected category or project.

* Postconditions: Task is created and assigned to a category or project.

* Expected Result: The task is successfully created and assigned to the selected category or project.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing (Acceptance Testing)

* Test Case Approach: Positive

**Test Case 2: Verify that users can view tasks grouped by categories or projects**

* Test Case ID: TC-TCT-002

* Test Case Title: Verify that users can view tasks grouped by categories or projects

* Test Case Description: This test case ensures that users can view tasks grouped by categories or projects, as per the acceptance criterion.

* Test Suite: Task Categorization

* Test Priority: Medium

* Preconditions:

+ User is logged in to the Task Management Tool

+ User has permission to view tasks

+ Multiple tasks are assigned to different categories or projects

* Test Data: Multiple tasks assigned to different categories or projects

* Test Steps:

1. Log in to the Task Management Tool as a user with permission to view tasks.

2. Navigate to the task view page and select a category or project from the dropdown list.

3. Verify that the tasks assigned to the selected category or project are displayed.

* Postconditions: Tasks are viewed correctly grouped by categories or projects.

* Expected Result: The tasks assigned to the selected category or project are displayed correctly.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing (Acceptance Testing)

* Test Case Approach: Positive

**Test Case 3: Verify that tasks can be moved between categories or projects**

* Test Case ID: TC-TCT-003

* Test Case Title: Verify that tasks can be moved between categories or projects

* Test Case Description: This test case ensures that tasks can be moved between categories or projects, as per the acceptance criterion.

* Test Suite: Task Categorization

* Test Priority: High

* Preconditions:

+ User is logged in to the Task Management Tool

+ User has permission to move tasks

+ Task is assigned to a category or project

* Test Data: Task is assigned to a category or project

* Test Steps:

 1. Log in to the Task Management Tool as a user with permission to move tasks.

 2. Navigate to the task detail page and select the category or project from the dropdown list.

 3. Click the "Move to another category/project" button.

 4. Select a new category or project from the dropdown list and confirm the move.

 5. Verify that the task is moved to the new category or project.

* Postconditions: Task is moved to a new category or project.

* Expected Result: The task is successfully moved to the new category or project.

* Severity: Critical

* Priority: High

* Type of Testing: Functional Testing (Acceptance Testing)

* Test Case Approach: Positive

**Test Case 4: Verify that category or project information is accurately reflected in task views**

* Test Case ID: TC-TCT-004

* Test Case Title: Verify that category or project information is accurately reflected in task views

* Test Case Description: This test case ensures that category or project information is accurately reflected in task views, as per the acceptance criterion.

* Test Suite: Task Categorization

* Test Priority: Medium

* Preconditions:

 + User is logged in to the Task Management Tool

 + User has permission to view tasks

 + Category or project information is updated

* Test Data: Category or project information is updated

* Test Steps:

 1. Log in to the Task Management Tool as a user with permission to view tasks.

 2. Navigate to the task view page and select a category or project from the dropdown list.

 3. Verify that the category or project information is displayed correctly.

* Postconditions: Category or project information is accurately reflected in task views.

* Expected Result: The category or project information is displayed correctly.

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing (Acceptance Testing)

* Test Case Approach: Positive

**Test Case 5: Verify that category or project information is not displayed for unassigned tasks**

* Test Case ID: TC-TCT-005

* Test Case Title: Verify that category or project information is not displayed for unassigned tasks

* Test Case Description: This test case ensures that category or project information is not displayed for unassigned tasks, as per the acceptance criterion.

* Test Suite: Task Categorization

* Test Priority: Low

* Preconditions:

 + User is logged in to the Task Management Tool

 + User has permission to view tasks

 + Task is not assigned to a category or project

* Test Data: Task is not assigned to a category or project

* Test Steps:

 1. Log in to the Task Management Tool as a user with permission to view tasks.

 2. Navigate to the task view page and select a category from the dropdown list.

3. Verify that category information is not displayed for unassigned tasks.

* Postconditions: Category or project information is not displayed for unassigned tasks.

* Expected Result: No category or project information is displayed for unassigned tasks.

* Severity: Trivial

* Priority: Low

* Type of Testing: Functional Testing (Acceptance Testing)

* Test Case Approach: Negative

**------------------------------------------------------------------------------------------------**

**Data - 47**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Search

**Feature description -**

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

**Userstory title -**

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

**Acceptance criteria -**

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

**Generated Test Cases:**

Here are five detailed test cases for the Task Search feature, each covering one or more of the acceptance criteria:

**Test Case 1:**

**Test Case ID:** TSC001

**Test Case Title:** Verify that users can search for tasks by name.

**Test Case Description:** This test case verifies that users can successfully search for tasks by entering a task name in the search field. It also checks that the search results are accurate and relevant.

**Test Suite:** Task Search Flow

**Test Priority:** High

**Preconditions:** 1. The user is logged in to the task management tool.

2. The user has a task list with multiple tasks.

3. The user has created multiple tasks with different names.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user with access to the task list.

2. Open the search field and enter a valid task name.

3. Click the search button to retrieve search results.

4. Verify that the search results display the task with the searched name.

5. Verify that the task is displayed in the expected category.

6. Verify that the task is assigned to the correct user.

**Postconditions:** The task is still displayed in the search results after closing the search field.

**Expected Result:** The user can search for tasks by name and receive accurate and relevant search results.

**Severity:** Major

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 2:**

**Test Case ID:** TSC002

**Test Case Title:** Verify that users can search for tasks by description.

**Test Case Description:** This test case verifies that users can successfully search for tasks by entering a task description in the search field. It also checks that the search results are accurate and relevant.

**Test Suite:** Task Search Flow

**Test Priority:** High

**Preconditions:** 1. The user is logged in to the task management tool.

2. The user has a task list with multiple tasks.

3. The user has created multiple tasks with different descriptions.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user with access to the task list.

2. Open the search field and enter a valid task description.

3. Click the search button to retrieve search results.

4. Verify that the search results display the task with the searched description.

5. Verify that the task is displayed in the expected category.

6. Verify that the task is assigned to the correct user.

**Postconditions:** The task is still displayed in the search results after closing the search field.

**Expected Result:** The user can search for tasks by description and receive accurate and relevant search results.

**Severity:** Major

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 3:**

**Test Case ID:** TSC003

**Test Case Title:** Verify that users can filter search results by category.

**Test Case Description:** This test case verifies that users can successfully filter search results by selecting a specific category.

**Test Suite:** Task Search Flow

**Test Priority:** Medium

**Preconditions:** 1. The user is logged in to the task management tool.

2. The user has a task list with multiple tasks.

3. The tasks are categorized into different categories.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user with access to the task list.

2. Open the search field and enter a valid task name.

3. Click the search button to retrieve search results.

4. Select a specific category from the filter dropdown.

5. Verify that the search results are filtered by the selected category.

6. Verify that tasks not belonging to the selected category are not displayed.

**Postconditions:** The category filter remains selected.

**Expected Result:** The user can filter search results by category.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 4:**

**Test Case ID:** TSC004

**Test Case Title:** Verify that users can filter search results by priority.

**Test Case Description:** This test case verifies that users can successfully filter search results by selecting a specific priority level.

**Test Suite:** Task Search Flow

**Test Priority:** Medium

**Preconditions:** 1. The user is logged in to the task management tool.

2. The user has a task list with multiple tasks.

3. The tasks have different priority levels.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user with access to the task list.

2. Open the search field and enter a valid task name.

3. Click the search button to retrieve search results.

4. Select a specific priority level from the filter dropdown.

5. Verify that the search results are filtered by the selected priority level.

6. Verify that tasks not belonging to the selected priority level are not displayed.

**Postconditions:** The priority filter remains selected.

**Expected Result:** The user can filter search results by priority.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 5:**

**Test Case ID:** TSC005

**Test Case Title:** Verify that users can filter search results by due date.

**Test Case Description:** This test case verifies that users can successfully filter search results by selecting a specific due date range.

**Test Suite:** Task Search Flow

**Test Priority:** Medium

**Preconditions:** 1. The user is logged in to the task management tool.

2. The user has a task list with multiple tasks.

3. The tasks have different due dates.

**Test Data:** No test data needed

**Test Steps:**

1. Log in to the task management tool as a user with access to the task list.

2. Open the search field and enter a valid task name.

3. Click the search button to retrieve search results.

4. Select a specific due date range from the filter dropdown.

5. Verify that the search results are filtered by the selected due date range.

6. Verify that tasks not belonging to the selected due date range are not displayed.

**Postconditions:** The due date filter remains selected.

**Expected Result:** The user can filter search results by due date.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Data - 48**

**Project name -**

Task Management Tool

**Project description -**

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

**Feature name -**

Task Collaboration

**Feature description -**

Users can collaborate on tasks by adding comments, attaching files, and sharing tasks with other users. Task collaboration aims to enhance team communication and productivity.

**Userstory title -**

As a user, I want to collaborate on tasks by adding comments, attaching files, and sharing tasks with other users so that I can work more effectively with my team.

**Acceptance criteria -**

['Users can add comments to tasks.', 'Users can attach files to tasks.', 'Users can share tasks with other users.', 'Task collaboration features work seamlessly and enhance team communication.', '']

**Generated Test Cases:**

Here are five detailed test cases for the Task Collaboration feature:

**Test Case 1: Verify that users can add comments to tasks**

* **Test Case ID:** TC-001
* **Test Case Title:** Verify that users can add comments to tasks
* **Test Case Description:** This test case aims to validate that users can add comments to tasks, ensuring that the task collaboration feature is working as expected.
* **Test Suite:** Task Collaboration
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the Task Management Tool
 + User has created a task
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool as a user
 2. Create a new task with a title and description
 3. Click on the "Add Comment" button
 4. Enter a comment in the comment field
 5. Click on the "Save" button
* **Postconditions:**
 + Comment is visible on the task page
 + Comment is timestamped with the user's name and date/time
* **Expected Result:** The user can add comments to tasks successfully.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that users can attach files to tasks**

* **Test Case ID:** TC-002

* **Test Case Title:** Verify that users can attach files to tasks

* **Test Case Description:** This test case aims to validate that users can attach files to tasks, ensuring that the task collaboration feature is working as expected.

* **Test Suite:** Task Collaboration

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the Task Management Tool

 + User has created a task

* **Test Data:** Test file (e.g., PDF, image, document)

* **Test Steps:**

 1. Log in to the Task Management Tool as a user

 2. Create a new task with a title and description

 3. Click on the "Attach File" button

 4. Select the test file from the file system

 5. Click on the "Save" button

* **Postconditions:**

 + File is attached to the task

 + File is visible on the task page

* **Expected Result:** The user can attach files to tasks successfully.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can share tasks with other users**

* **Test Case ID:** TC-003
* **Test Case Title:** Verify that users can share tasks with other users
* **Test Case Description:** This test case aims to validate that users can share tasks with other users, ensuring that the task collaboration feature is working as expected.
* **Test Suite:** Task Collaboration
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the Task Management Tool
 + User has created a task
 + Other user is logged in to the Task Management Tool
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool as a user
 2. Create a new task with a title and description
 3. Click on the "Share Task" button
 4. Enter the email address of the other user
 5. Click on the "Share" button
* **Postconditions:**
 + Task is shared with the other user
 + Other user receives an email notification with the shared task
* **Expected Result:** The user can share tasks with other users successfully.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that task collaboration features work seamlessly and enhance team communication**

* **Test Case ID:** TC-004
* **Test Case Title:** Verify that task collaboration features work seamlessly and enhance team communication
* **Test Case Description:** This test case aims to validate that task collaboration features work seamlessly and enhance team communication, ensuring that the task collaboration feature is working as expected.
* **Test Suite:** Task Collaboration
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the Task Management Tool
 + User has created a task
 + Other user is logged in to the Task Management Tool
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the Task Management Tool as a user
 2. Create a new task with a title and description
 3. Share the task with the other user
 4. Other user adds a comment to the task
 5. User receives an email notification with the comment
* **Postconditions:**
 + Task collaboration features work seamlessly
 + Team communication is enhanced
* **Expected Result:** Task collaboration features work seamlessly and enhance team communication.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that users cannot add comments to tasks without being logged in**

* **Test Case ID:** TC-005
* **Test Case Title:** Verify that users cannot add comments to tasks without being logged in
* **Test Case Description:** This test case aims to validate that users cannot add comments to tasks without being logged in, ensuring that the task collaboration feature is secure.
* **Test Suite:** Task Collaboration
* **Test Priority:** Medium
* **Preconditions:**
 + User is not logged in to the Task Management Tool
 + User has created a task
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log out of the Task Management Tool
 2. Create a new task with a title and description
 3. Click on the "Add Comment" button
 4. Attempt to enter a comment in the comment field
* **Postconditions:**
 + User is redirected to the login page
 + Comment field is disabled
* **Expected Result:** The user cannot add comments to tasks without being logged in.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Negative

---------------------------------------------------------------------------------------------------

**Data - 49**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Course Creation Tools

**Feature description -**

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to structure content effectively, ensuring an engaging and informative learning experience for students.

**Userstory title -**

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

**Acceptance criteria -**

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

**Generated Test Cases:**

Here are five detailed test cases for the "Course Creation Tools" feature, each covering one or more of the acceptance criteria from the user story:

**Test Case 1: Verify that instructors can create courses using videos, presentations, quizzes, and assignments**

* **Test Case ID:** CT-001

* **Test Case Title:** Verify that instructors can create courses using videos, presentations, quizzes, and assignments

* **Test Case Description:** As an instructor, I should be able to create a comprehensive course by adding various multimedia elements such as videos, presentations, quizzes, and assignments.

* **Test Suite:** Course Creation Tools

* **Test Priority:** High

* **Preconditions:**

 + Instructor is logged in to the online learning platform

 + Instructor has created an account and has basic permissions to create courses

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor and navigate to the course creation page

 2. Click on the "Add Video" button and select a video file from the local machine

 3. Add a presentation by uploading a PowerPoint file

 4. Create a quiz by adding multiple-choice questions and correct answers

 5. Add an assignment by creating a text-based task

* **Postconditions:**

 + Course is created with all four multimedia elements

 + Instructor can preview the course

* **Expected Result:** Course with videos, presentations, quizzes, and assignments is created successfully

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that the course creation tools are user-friendly and require minimal technical skills**

* **Test Case ID:** CT-002
* **Test Case Title:** Verify that the course creation tools are user-friendly and require minimal technical skills
* **Test Case Description:** As an instructor, I should be able to create a comprehensive course using the intuitive course creation tools, which require minimal technical skills.
* **Test Suite:** Course Creation Tools
* **Test Priority:** Medium
* **Preconditions:**
 + Instructor is logged in to the online learning platform
 + Instructor has basic knowledge of creating online courses
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as an instructor and navigate to the course creation page
 2. Attempt to create a course using the drag-and-drop interface
 3. Try to add multimedia elements without any prior knowledge
 4. Look for error messages or clear instructions on how to add elements
* **Postconditions:**
 + Instructor is able to create a course using the intuitive tools
 + Error messages are clear and instructional
* **Expected Result:** Course creation process is user-friendly and requires minimal technical skills
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that courses can be previewed before publishing**

* **Test Case ID:** CT-003
* **Test Case Title:** Verify that courses can be previewed before publishing
* **Test Case Description:** As an instructor, I should be able to preview my course before publishing it to ensure that it meets my expectations.
* **Test Suite:** Course Creation Tools
* **Test Priority:** High
* **Preconditions:**
 + Instructor is logged in to the online learning platform
 + Instructor has created a course with multimedia elements
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as an instructor and navigate to the course creation page
 2. Click on the "Preview" button to view the course
 3. Verify that the course content is displayed correctly
 4. Attempt to navigate through the course
* **Postconditions:**
 + Course is previewed successfully
 + Instructor can view the course content
* **Expected Result:** Course can be previewed before publishing
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that instructors can save courses as drafts**

* **Test Case ID:** CT-004

* **Test Case Title:** Verify that instructors can save courses as drafts

* **Test Case Description:** As an instructor, I should be able to save my course as a draft to work on it later.

* **Test Suite:** Course Creation Tools

* **Test Priority:** Medium

* **Preconditions:**

 + Instructor is logged in to the online learning platform

 + Instructor has created a course with multimedia elements

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor and navigate to the course creation page

 2. Click on the "Save as Draft" button to save the course

 3. Verify that the course is saved as a draft

 4. Attempt to create a new course while having a draft saved

* **Postconditions:**

 + Course is saved as a draft

 + Instructor can view the draft

* **Expected Result:** Course can be saved as a draft

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that courses can be edited post-publishing**

* **Test Case ID:** CT-005

* **Test Case Title:** Verify that courses can be edited post-publishing

* **Test Case Description:** As an instructor, I should be able to edit my course after it has been published.

* **Test Suite:** Course Creation Tools

* **Test Priority:** High

* **Preconditions:**

 + Instructor is logged in to the online learning platform

 + Instructor has created a course with multimedia elements

 + Course is published

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor and navigate to the course creation page

 2. Click on the "Edit" button to edit the course

 3. Verify that the course can be edited

 4. Attempt to add new multimedia elements to the course

* **Postconditions:**

 + Course is edited successfully

 + Instructor can view the updated course

* **Expected Result:** Course can be edited post-publishing

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 50**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Interactive Lessons

**Feature description -**

Lessons are thoughtfully designed to foster learner engagement through interactive components such as quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these interactive elements enhance comprehension and retention of course material.

**Userstory title -**

As a learner, I want interactive lessons so that I can engage with the content more effectively.

**Acceptance criteria -**

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided after each interactive component.', 'Interactive lessons can be accessed on various devices.']

**Generated Test Cases:**

Here are 5 detailed test cases for the Interactive Lessons feature, each covering one or more of the acceptance criteria:

**Test Case 1: Verify that Lessons include quizzes, polls, simulations, and discussions**

* **Test Case ID:** TC001
* **Test Case Title:** Verify that Lessons include quizzes, polls, simulations, and discussions

* **Test Case Description:** This test case ensures that lessons include all four types of interactive components: quizzes, polls, simulations, and discussions.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:** The user is logged in to the online learning platform and has been assigned a lesson.

* **Test Data:** No test data needed

* **Test Steps:**

1. Login to the online learning platform and navigate to the lesson assigned.

2.Inspect the lesson content for interactive components.

3. Verify that the lesson includes a quiz.

4. Verify that the lesson includes a poll.

5. Verify that the lesson includes a simulation.

6. Verify that the lesson includes a discussion forum.

* **Postconditions:** The user has logged out of the online learning platform.

* **Expected Result:** The lesson includes all four types of interactive components: quizzes, polls, simulations, and discussions.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that Interactive elements are integrated seamlessly into the lesson content**

* **Test Case ID:** TC002

* **Test Case Title:** Verify that Interactive elements are integrated seamlessly into the lesson content

* **Test Case Description:** This test case ensures that the interactive components are well-integrated into the lesson content and do not disrupt the flow of learning.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:** The user is logged in to the online learning platform and has been assigned a lesson.

* **Test Data:** No test data needed

* **Test Steps:**

1. Login to the online learning platform and navigate to the lesson assigned.

2. Inspect the lesson content for interactive components.

3. Verify that the interactive components are visually appealing and do not distract from the learning content.

4. Verify that the interactive components are easy to use and do not interfere with the user's ability to complete the lesson.

* **Postconditions:** The user has logged out of the online learning platform.

* **Expected Result:** Interactive components are well-integrated into the lesson content and enhance the learning experience.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that Learners can track their progress through these activities**

* **Test Case ID:** TC003

* **Test Case Title:** Verify that Learners can track their progress through these activities

* **Test Case Description:** This test case ensures that learners can track their progress through the interactive components and receive feedback.

* **Test Suite:** Interactive Lessons

* **Test Priority:** Medium

* **Preconditions:** The user is logged in to the online learning platform and has been assigned a lesson.

* **Test Data:** No test data needed

* **Test Steps:**

1. Login to the online learning platform and navigate to the lesson assigned.

2. Participate in the interactive components (e.g., quizzes, polls, simulations, discussions).

3. Verify that the user can view their progress through the activities.

4. Verify that the user receives feedback on their performance.

* **Postconditions:** The user has logged out of the online learning platform.

* **Expected Result:** Learners can track their progress through the interactive components and receive feedback.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that Feedback is provided after each interactive component**

* **Test Case ID:** TC004

* **Test Case Title:** Verify that Feedback is provided after each interactive component

* **Test Case Description:** This test case ensures that feedback is provided to learners after each interactive component.

* **Test Suite:** Interactive Lessons

* **Test Priority:** Medium

* **Preconditions:** The user is logged in to the online learning platform and has been assigned a lesson.

* **Test Data:** No test data needed

* **Test Steps:**

1. Login to the online learning platform and navigate to the lesson assigned.

2. Participate in a single interactive component (e.g., quiz, poll, simulation, discussion).

3. Verify that feedback is provided to the user.

4. Verify that the feedback is relevant and helpful.

* **Postconditions:** The user has logged out of the online learning platform.

* **Expected Result:** Feedback is provided to learners after each interactive component.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that Interactive lessons can be accessed on various devices**

* **Test Case ID:** TC005

* **Test Case Title:** Verify that Interactive lessons can be accessed on various devices

* **Test Case Description:** This test case ensures that interactive lessons can be accessed on various devices, including desktop, laptop, tablet, and mobile devices.

* **Test Suite:** Interactive Lessons

* **Test Priority:** High

* **Preconditions:** The user has a device with internet access and has been assigned a lesson.

* **Test Data:** No test data needed

* **Test Steps:**

1. Access the online learning platform on a desktop device.

2. Access the online learning platform on a laptop device.

3. Access the online learning platform on a tablet device.

4. Access the online learning platform on a mobile device.

5. Verify that the interactive lessons can be accessed and used on all devices.

* **Postconditions:** The user has logged out of the online learning platform on all devices.

* **Expected Result:** Interactive lessons can be accessed and used on various devices.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

--------------------------------------------------------------------------------------------

**Data - 51**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Personalized Learning Paths

**Feature description -**

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

**Userstory title -**

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

**Acceptance criteria -**

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

**Generated Test Cases:**

Here are 5 detailed test cases for the Personalized Learning Paths feature on the Online Learning Platform:

**Test Case 1: Learner Chooses Courses and Modules Based on Interests**

Test Case ID: PERSLP-001

Test Case Title: Verify that learners can select courses and modules based on their interests.

Test Case Description: This test case will cover the acceptance criterion of learners choosing courses and modules based on their interests.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

? The learner has created an account and logged in to the platform.

? The learner has selected some interests on their profile.

Test Data: None

Test Steps:

1. Log in to the online learning platform as a learner.
2. Click on the "Choose Courses" button and navigate to the course catalog.
3. Filter the courses by the learner's selected interests using the "Filter by Interest" option.
4. Select a course and view its details, including the modules and resources.
5. Add the course to the learner's playlist and view the modules.

Expected Result: The learner can successfully choose courses and modules based on their interests, and the platform displays the relevant content.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Personalized Learning Paths Adjust Dynamically Based on Learner Progress**

Test Case ID: PERSLP-002

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress.

Test Case Description: This test case will cover the acceptance criterion of personalized learning paths adjusting dynamically based on learner progress.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

? The learner has created an account and logged in to the platform.

? The learner has started a course and completed some modules.

Test Data: None

Test Steps:

1. Log in to the online learning platform as a learner.
2. Start a course and complete some modules.
3. Log out and log back in to the platform.
4. Navigate to the course catalog and view the learner's playlist.
5. Verify that the course and modules are correctly displayed and adjusted based on the learner's progress.

Expected Result: The learner's personalized learning path adjusts dynamically and reflects the progress made.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Learner Receives Recommendations for New Courses and Modules**

Test Case ID: PERSLP-003

Test Case Title: Verify that learners receive recommendations for new courses and modules.

Test Case Description: This test case will cover the acceptance criterion of learners receiving recommendations for new courses and modules.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

? The learner has created an account and logged in to the platform.

? The learner has completed some courses and modules.

Test Data: None

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the course catalog and view the learner's playlist.
3. Click on the "Recommended Courses" button.
4. Verify that the platform displays relevant recommendations for new courses and modules based on the learner's progress and interests.

Expected Result: The learner receives personalized recommendations for new courses and modules.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Progress Tracking is Personalized to Each Learner's Path**

Test Case ID: PERSLP-004

Test Case Title: Verify that progress tracking is personalized to each learner's path.

Test Case Description: This test case will cover the acceptance criterion of progress tracking being personalized to each learner's path.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

? The learner has created an account and logged in to the platform.

? The learner has started a course and completed some modules.

Test Data: None

Test Steps:

1. Log in to the online learning platform as a learner.
2. Start a course and complete some modules.
3. Log out and log back in to the platform.
4. Navigate to the "Progress" tab and view the learner's progress.
5. Verify that the progress is correctly reflected and that the platform displays personalized insights.

Expected Result: The learner's progress is accurately tracked and displayed in a personalized manner.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Learner Can Adjust Their Learning Paths at Any Time**

Test Case ID: PERSLP-005

Test Case Title: Verify that learners can adjust their learning paths at any time.

Test Case Description: This test case will cover the acceptance criterion of learners being able to adjust their learning paths at any time.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

? The learner has created an account and logged in to the platform.

? The learner has selected some interests on their profile.

Test Data: None

Test Steps:

1. Log in to the online learning platform as a learner.
2. Navigate to the course catalog and view the learner's playlist.
3. Update the learner's interests and select new courses and modules.
4. Verify that the platform reflects the changes and updates the learner's learning path.
5. Check that the learner can adjust their learning path without any issues.

Expected Result: The learner can successfully adjust their learning path at any time.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Negative (test that the platform allows adjustments at any time, not just at specific points).

---

**Data - 52**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Progress Tracking

**Feature description -**

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

**Userstory title -**

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

**Acceptance criteria -**

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

**Generated Test Cases:**

Here are five detailed test cases for the Progress Tracking feature of the Online Learning Platform:

**Test Case 1: Verify that Progress Tracking shows completed activities and grades**

* Test Case ID: PTP-001

* Test Case Title: Verify that Progress Tracking shows completed activities and grades

* Test Suite: Progress Tracking

* Test Priority: High

* Preconditions:

 + User is logged in to the platform

 + User has enrolled in a course with multiple activities

 + User has completed some activities

* Test Data: None

* Test Steps:

 1. Log in to the platform as a registered user

 2. Navigate to the course dashboard

 3. Expand the progress tracker section

 4. Verify that the completed activities are displayed with grades

 5. Verify that the percentage completion and total score are updated correctly

* Postconditions: None

* Expected Result: The progress tracker displays a list of completed activities with grades, and the percentage completion and total score are updated correctly.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 2: Verify that Learners can view detailed reports on their progress**

* Test Case ID: PTP-002

* Test Case Title: Verify that Learners can view detailed reports on their progress

* Test Suite: Progress Tracking

* Test Priority: High

* Preconditions:

 + User is logged in to the platform

 + User has enrolled in a course with multiple activities

* Test Data: None

* Test Steps:

1. Log in to the platform as a registered user
2. Navigate to the course dashboard
3. Click on the progress report link
4. Verify that the report displays detailed information on progress, including completed activities, grades, and percentage completion
5. Verify that the report can be downloaded as a PDF or CSV file

* Postconditions: None
* Expected Result: The progress report provides a detailed view of the learner's progress, including completed activities, grades, and percentage completion, and the report can be downloaded as a file.
* Severity: Major
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 3: Verify that Achievements are highlighted in the progress tracker**

* Test Case ID: PTP-003
* Test Case Title: Verify that Achievements are highlighted in the progress tracker
* Test Suite: Progress Tracking
* Test Priority: Medium
* Preconditions:
 + User is logged in to the platform
 + User has enrolled in a course with multiple activities
 + User has achieved some achievements
* Test Data: None
* Test Steps:

1. Log in to the platform as a registered user

 2. Navigate to the course dashboard

 3. Expand the progress tracker section

 4. Verify that the achieved achievements are highlighted in the progress tracker

 5. Verify that the achievements are displayed correctly, including badge and description

* Postconditions: None

* Expected Result: The progress tracker highlights achieved achievements, displaying a badge and description correctly.

* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Verify that Learners receive notifications about their progress**

* Test Case ID: PTP-004

* Test Case Title: Verify that Learners receive notifications about their progress

* Test Suite: Progress Tracking

* Test Priority: High

* Preconditions:

 + User is logged in to the platform

 + User has enrolled in a course with multiple activities

 + User's progress meets certain conditions for notification (e.g., completing a module or achieving a milestone)

* Test Data: None

* Test Steps:

 1. Log in to the platform as a registered user

 2. Navigate to the course dashboard

 3. Complete a module or achieve a milestone that triggers a notification

 4. Verify that the learner receives a notification about their progress

 5. Verify that the notification is displayed correctly, including title, description, and call-to-action

* Postconditions: None

* Expected Result: The learner receives a notification about their progress, displayed correctly with a title, description, and call-to-action.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 5: Verify that Progress tracking can be exported as reports**

* Test Case ID: PTP-005

* Test Case Title: Verify that Progress tracking can be exported as reports

* Test Suite: Progress Tracking

* Test Priority: Medium

* Preconditions:

 + User is logged in to the platform

 + User has enrolled in a course with multiple activities

* Test Data: None

* Test Steps:

 1. Log in to the platform as a registered user

 2. Navigate to the course dashboard

 3. Click on the export report link

 4. Select the report format (e.g., PDF, CSV)

 5. Verify that the report is generated correctly and can be downloaded

* Postconditions: None

* Expected Result: The progress tracking report can be exported as a file in the selected format.

* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

---

**Data - 53**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Assessment and Grading

**Feature description -**

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

**Userstory title -**

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

**Acceptance criteria -**

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

**Generated Test Cases:**

Here are five detailed test cases that cover the acceptance criteria for the Assessment and Grading feature:

**Test Case 1: Create Quizzes, Tests, and Assignments**

* **Test Case ID:** AQG001
* **Test Case Title:** Verify that instructors can create diverse assessments, including quizzes, tests, and assignments.
* **Test Suite:** Assessment Module
* **Test Priority:** High
* **Preconditions:** Instructors have access to the assessment module, and the system has been configured to allow assessment creation.
* **Test Data:** Empty assessment template for quizzes, tests, and assignments.
* **Test Steps:**
 1. Log in to the system as an instructor.
 2. Navigate to the assessment module.
 3. Click on the "Create Quiz/ Test/ Assignment" button.
 4. Select the assessment type (Quizzes, Tests, or Assignments) from the drop-down menu.
 5. Fill out the assessment details, including title, description, and question types.
 6. Save the assessment.
* **Postconditions:** The newly created assessment should be visible in the instructor's assessment list.
* **Expected Result:** The instructor can successfully create quizzes, tests, and assignments with various question types.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

**Test Case 2: Automatic Grading for Objective Questions**

* **Test Case ID:** AQG002

* **Test Case Title:** Verify that automatic grading is available for objective questions.

* **Test Suite:** Assessment Module

* **Test Priority:** High

* **Preconditions:** Instructors have created an assessment with objective questions, and the auto-grading feature is enabled.

* **Test Data:** Pre-filled assessment template with 5 objective questions, including multiple-choice and true/false questions.

* **Test Steps:**

1. Log in to the system as an instructor.
2. Navigate to the assessment module.
3. Select an assessment with objective questions.
4. Click on the "Auto Grade" button.
5. Confirm that the grading is accurate and reflects the correct answers.

* **Postconditions:** The assessment has been auto-graded, and results reflect the correct answers.

* **Expected Result:** Automatic grading for objective questions is accurate and reliable.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

**Test Case 3: Manual Grading for Subjective Questions**

* **Test Case ID:** AQG003
* **Test Case Title:** Verify that manual grading is available for subjective questions.
* **Test Suite:** Assessment Module
* **Test Priority:** High
* **Preconditions:** Instructors have created an assessment with subjective questions, and the manual grading feature is enabled.
* **Test Data:** Pre-filled assessment template with 5 subjective questions, including short-answer and essay questions.
* **Test Steps:**
 1. Log in to the system as an instructor.
 2. Navigate to the assessment module.
 3. Select an assessment with subjective questions.
 4. Click on the "Manual Grade" button.
 5. Evaluate the student's responses and assign grades accordingly.
* **Postconditions:** The assessment has been manually graded, and results reflect the correct grades.
* **Expected Result:** Manual grading for subjective questions is accurate and reliable.
* **Severity:** Blocker
* **Priority:** High
* **Type of Testing:** System Testing
* **Test Case Approach:** Positive

**Test Case 4: Feedback to Learners after Grading**

* **Test Case ID:** AQG004

* **Test Case Title:** Verify that feedback is provided to learners after grading.

* **Test Suite:** Assessment Module

* **Test Priority:** Medium

* **Preconditions:** Instructors have graded an assessment and the feedback feature is enabled.

* **Test Data:** Pre-filled assessment template with sample learner responses.

* **Test Steps:**

1. Log in to the system as an instructor.

2. Navigate to the assessment module.

3. Select an assessment with graded answers.

4. Click on the "View Feedback" button.

5. Confirm that the learner's responses and corresponding feedback are displayed.

* **Postconditions:** The learner receives accurate feedback on their assessment responses.

* **Expected Result:** Feedback is provided to learners after grading, promoting learning and improvement.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

**Test Case 5: Assessment Type Limitations**

* **Test Case ID:** AQG005

* **Test Case Title:** Verify that limitations are enforced for each assessment type.

* **Test Suite:** Assessment Module

* **Test Priority:** Low

* **Preconditions:** Instructors have access to the assessment module and assessment templates.

* **Test Data:** Pre-filled assessment templates for quizzes, tests, and assignments with varying limitations (e.g., time limits, question types, and grade restrictions).

* **Test Steps:**

 1. Log in to the system as an instructor.

 2. Navigate to the assessment module.

 3. Create a new assessment for each type (quizzes, tests, and assignments).

 4. Apply different limitations for each assessment type.

 5. Verify that the limitations are enforced during assessment creation.

* **Postconditions:** Assessment types are restricted to their designated limitations.

* **Expected Result:** The system enforces the designated limitations for each assessment type.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** System Testing

* **Test Case Approach:** Positive

Each of these test cases is designed to thoroughly cover one or more acceptance criteria from the user story, providing a comprehensive coverage of the Assessment and Grading feature. By following this test case template, we ensure that each test case is detailed, concise, and effective in identifying potential defects in the system.

---

**Data - 54**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Discussion Forums

**Feature description -**

Learners engage in meaningful discussions, ask questions, and share insights with peers and instructors through dedicated discussion forums. These forums foster collaboration, critical thinking, and knowledge sharing, enriching the learning experience through diverse perspectives and active participation.

**Userstory title -**

As a learner, I want to participate in discussion forums so that I can engage with peers and instructors.

**Acceptance criteria -**

['Learners can post questions and responses in forums.', 'Forums support threaded discussions.', 'Instructors can moderate discussions.', 'Learners receive notifications for new posts.', 'Forums are accessible on all devices.']

**Generated Test Cases:**

Here are five detailed test cases for the Discussion Forums feature, each covering one or more of the acceptance criteria:

**Test Case 1: Verify that learners can post questions and responses in forums**

* **Test Case ID:** DFM-001

* **Test Case Title:** Verify that learners can post questions and responses in forums

* **Test Case Description:** This test case verifies that learners can create new posts in discussion forums, including questions and responses, and that these posts are displayed correctly in the forum.

* **Test Suite:** Discussion Forums Module

* **Test Priority:** High

* **Preconditions:**

 + Learner is logged in to the online learning platform

 + Learner has access to a discussion forum

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the online learning platform as a learner

 2. Navigate to a discussion forum

 3. Click the "Post a question" or "Respond" button

 4. Enter a question or response in the text area

 5. Click the "Submit" button

 6. Verify that the post is displayed correctly in the forum

* **Postconditions:** The learner can view their newly posted question or response in the discussion forum.

* **Expected Result:** Learners can create and submit new posts in discussion forums.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that forums support threaded discussions**

* **Test Case ID:** DFM-002

* **Test Case Title:** Verify that forums support threaded discussions

* **Test Case Description:** This test case verifies that discussion forums support threaded discussions, allowing learners to respond to previous posts and creating a hierarchical conversation structure.

* **Test Suite:** Discussion Forums Module

* **Test Priority:** High

* **Preconditions:**

 + Learner is logged in to the online learning platform

 + Learner has access to a discussion forum with existing posts

* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the online learning platform as a learner
 2. Navigate to a discussion forum with existing posts
 3. Click on a post to view its details
 4. Click the "Respond" button
 5. Enter a response to the post
 6. Click the "Submit" button
 7. Verify that the response is displayed as a threaded discussion
* **Postconditions:** The learner can view their response in the threaded discussion.
* **Expected Result:** Forums support threaded discussions, allowing learners to create a hierarchical conversation structure.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that instructors can moderate discussions**

* **Test Case ID:** DFM-003
* **Test Case Title:** Verify that instructors can moderate discussions
* **Test Case Description:** This test case verifies that instructors can moderate discussion forums, including the ability to approve or reject posts, delete posts, and pin important messages to the top of the forum.
* **Test Suite:** Discussion Forums Module
* **Test Priority:** High
* **Preconditions:**
 + Instructor is logged in to the online learning platform

+ Instructor has access to a discussion forum

+ Learner has posted a question or response in the forum

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform as an instructor

2. Navigate to a discussion forum with a posted question or response

3. Verify that the instructor can view the post and its details

4. Attempt to approve or reject the post

5. Verify that the post is updated accordingly

* **Postconditions:** The instructor can view the updated post and its status.

* **Expected Result:** Instructors can moderate discussions, including approving or rejecting posts, deleting posts, and pinning important messages.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that learners receive notifications for new posts**

* **Test Case ID:** DFM-004

* **Test Case Title:** Verify that learners receive notifications for new posts

* **Test Case Description:** This test case verifies that learners receive notifications when new posts are made in discussion forums, allowing them to stay engaged and participate in discussions.

* **Test Suite:** Discussion Forums Module

* **Test Priority:** Medium

* **Preconditions:**

+ Learner is logged in to the online learning platform

+ Learner has access to a discussion forum

+ A second learner has posted a new question or response in the forum

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform as a learner

2. Navigate to a discussion forum

3. Verify that the learner receives a notification for the new post

4. Click on the notification to view the new post

* **Postconditions:** The learner can view the notification and access the new post.

* **Expected Result:** Learners receive notifications for new posts in discussion forums.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that forums are accessible on all devices**

* **Test Case ID:** DFM-005

* **Test Case Title:** Verify that forums are accessible on all devices

* **Test Case Description:** This test case verifies that discussion forums are accessible on various devices, including desktop computers, laptops, tablets, smartphones, and other mobile devices.

* **Test Suite:** Discussion Forums Module

* **Test Priority:** Low

* **Preconditions:**

+ Learner is logged in to the online learning platform

+ Learner has access to a discussion forum

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the online learning platform as a learner
2. Navigate to a discussion forum on a desktop computer
3. Verify that the forum is displayed correctly
4. Log out and log back in on a laptop, tablet, and smartphone
5. Verify that the forum is displayed correctly on each device

* **Postconditions:** The learner can view the discussion forum on various devices.

* **Expected Result:** Forums are accessible on all devices, ensuring a seamless learning experience.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 55**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Live Classes and Webinars

**Feature description -**

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A

opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

**Userstory title -**

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

**Acceptance criteria -**

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

**Generated Test Cases:**

Here are five detailed test cases for the Live Classes and Webinars feature, following the provided Test Case Creation Template:

**Test Case 1:** Verify that live classes and webinars are conducted through video conferencing tools.

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing | Test Case Approach |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.1 | Verify that live classes are conducted through video conferencing tools | As a learner, I want to ensure that live classes are conducted through video conferencing tools to facilitate real-time interaction with instructors. | Live Classes and Webinars Module | High | Instructor is logged in, Learner is logged in, Video conferencing tool is configured | No test data needed | 1. Login as an instructor using the platform's credentials. 2. Schedule a live class with video conferencing configured. 3. Login as a learner and join the live class. 4. Verify that the video conferencing tool is accessible and functioning correctly. 5. Observe the live class and ensure that the instructor can share their screen and resources if necessary. | Instructor and learner should be able to participate in the live class through the video conferencing tool. | Blocker | High | Functional Testing | Positive |

**Test Case 2:** Verify that learners can ask questions in real-time.

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing | Test Case Approach |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.2 | Verify that learners can ask questions in real-time | As a learner, I want to be able to ask questions in real-time during live classes and webinars to clarify any doubts or seek additional information. | Live Classes and Webinars Module | High | Instructor is logged in, Learner is logged in, Live class/webinar is in progress | No test data needed | 1. Login as a learner and join a live class or webinar. 2. Ask a question in the chat window or using the audio/video conferencing tool. 3. Observe the instructor's response and verify that it is visible to all participants. 4. Test that the instructor can respond to questions and engage in a conversation with learners. | Learner should be able to ask questions in real-time and receive a response from the instructor. | Critical | High | Functional Testing | Positive |

**Test Case 3:** Verify that sessions are recorded and available for later viewing.

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing | Test Case Approach |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.3 | Verify that sessions are recorded and available for later viewing | As a learner, I want to be able to access recorded live classes and webinars for future reference or review. | Live Classes and Webinars Module | High | Instructor is logged in, Live class/webinar has completed, Recording feature is enabled | No test data needed | 1. Schedule a live class/webinar with the recording feature enabled. 2. Login as a learner and join the live class/webinar. 3. Verify that the live class/webinar is being recorded in the background. 4. Observe the recorded session and ensure that it is saved in the platform for later viewing. | Recorded sessions should be accessible by learners after the live class/webinar has completed. | Blocker | High | Functional Testing |

Positive |

**Test Case 4:** Verify that instructors can share screen and resources during sessions.

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing | Test Case Approach |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.4 | Verify that instructors can share screen and resources during sessions | As an instructor, I want to be able to share my screen and resources with learners during live classes and webinars to enhance the learning experience. | Live Classes and Webinars Module | High | Instructor is logged in, Learner is logged in, Live class/webinar is in progress | No test data needed | 1. Login as an instructor and join a live class or webinar. 2. Share your screen or resources using the video conferencing tool. 3. Verify that the shared screen or resources are visible to all participants. 4. Test that the instructor can share multiple resources or switch between shared resources. | Instructor should be able to share their screen and resources with learners during live classes and webinars. | Critical | High | Functional Testing | Positive |

**Test Case 5:** Verify that notifications are sent before sessions start.

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing | Test Case Approach |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.5 | Verify that notifications are sent before sessions start | As a learner, I want to receive timely notifications before live classes and webinars start to ensure that I am prepared and on time. | Live Classes and Webinars Module | Medium | Instructor is logged in, Learner is logged in, Live class/webinar is scheduled | No test data needed | 1. Schedule a live class/webinar with notifications enabled. 2. Verify that the learner receives a notification before the start of the live class/webinar. 3. Observe the notification and ensure that it contains relevant details such as date, time, and session link. | Learner should receive a timely notification

before the start of the live class/webinar. | Major | Medium | Non-Functional Testing | Positive |

These test cases cover all the specified acceptance criteria, including live classes/webinars conducted through video conferencing tools, learners asking questions in real-time, recorded sessions available for later viewing, instructors sharing screen and resources, and notifications sent before sessions start.

**--------------------------------------------------------------------------------------------------**

**Data - 56**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Mobile Accessibility

**Feature description -**

Learners enjoy seamless access to course materials and learning activities via mobile devices, ensuring flexibility and convenience in their learning journey. Whether on-the-go or in remote locations, mobile accessibility enables continuous learning opportunities, empowering learners to maximize their educational experience anytime, anywhere.

**Userstory title -**

As a learner, I want mobile access to the platform so that I can learn on the go.

**Acceptance criteria -**

['The platform is accessible on smartphones and tablets.', 'Mobile UI is optimized for touch navigation.', 'Course materials are accessible and readable on mobile devices.', 'Notifications are sent to mobile devices.', 'The platform supports offline access to downloaded materials.']

**Generated Test Cases:**

Here are the five test cases for the Mobile Accessibility feature, each covering one or more acceptance criteria:

**Test Case 1: Verify Mobile Access to Course Materials**

| Test Case ID | TC-MA-1 |

| --- | --- |

| Test Case Title | Verify that learners can access course materials on mobile devices |

| Test Case Description | This test case aims to verify that learners can view and access course materials, including video lectures, downloadable resources, and discussion forums, on their mobile devices. |

| Test Suite | Mobile Accessibility |

| Test Priority | High |

| Preconditions | 1. Learner has a valid account on the online learning platform.

2. Learner is using a smartphone or tablet.

3. Mobile device has a stable internet connection. |

| Test Data | No test data needed |

| Test Steps | 1. Access the online learning platform on a mobile device using the learner's credentials.

2. Browse through the course materials, including video lectures and downloadable resources.

3. Attempt to access the discussion forum.

4. Verify that all course materials are displayed correctly and are readable on the mobile device. |

| Postconditions | None |

| Expected Result | Learner can view and access course materials on their mobile device. |

| Severity | Minor |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

**Test Case 2: Verify Mobile UI Optimized for Touch Navigation**

| Test Case ID | TC-MA-2 |
| --- | --- |
| Test Case Title | Verify that the mobile UI is optimized for touch navigation |
| Test Case Description | This test case aims to verify that the mobile UI is user-friendly and allows learners to navigate the platform easily using touch gestures. |
| Test Suite | Mobile Accessibility |
| Test Priority | High |
| Preconditions | 1. Learner has a valid account on the online learning platform.
2. Learner is using a smartphone or tablet.
3. Mobile device has a stable internet connection. |
| Test Data | No test data needed |
| Test Steps | 1. Access the online learning platform on a mobile device using the learner's credentials.
2. Verify that the mobile UI is responsive and adapts to the mobile device's screen size.
3. Test various touch gestures, such as tapping, swiping, and pinching, to navigate the platform.
4. Verify that all UI elements are easily accessible and actionable via touch gestures. |
| Postconditions | None |
| Expected Result | Mobile UI is optimized for touch navigation and is user-friendly. |
| Severity | Minor |
| Priority | High |
| Type of Testing | Functional Testing |
| Test Case Approach | Positive |

**Test Case 3: Verify Notifications Sent to Mobile Devices**

| Test Case ID | TC-MA-3 |

| --- | --- |

| Test Case Title | Verify that notifications are sent to mobile devices |

| Test Case Description | This test case aims to verify that learners receive notifications on their mobile devices when new content is uploaded, or when they are required to perform a specific action. |

| Test Suite | Mobile Accessibility |

| Test Priority | High |

| Preconditions | 1. Learner has a valid account on the online learning platform.

2. Learner has enabled push notifications on their mobile device.

3. Mobile device has a stable internet connection. |

| Test Data | No test data needed |

| Test Steps | 1. Create a new piece of content on the online learning platform.

2. Simulate a notification being sent to the learner's mobile device.

3. Verify that the notification is received on the learner's mobile device and is displayed correctly.

4. Attempt to dismiss the notification. |

| Postconditions | None |

| Expected Result | Notifications are sent to mobile devices when required. |

| Severity | Minor |

| Priority | Medium |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

**Test Case 4: Verify Offline Access to Downloaded Materials**

| Test Case ID | TC-MA-4 |

| --- | --- |

| Test Case Title | Verify that learners can access downloaded materials offline |

| Test Case Description | This test case aims to verify that learners can access course materials that have been downloaded on their mobile device when they are offline. |

| Test Suite | Mobile Accessibility |

| Test Priority | High |

| Preconditions | 1. Learner has a valid account on the online learning platform.

2. Learner has downloaded course materials on their mobile device.

3. Mobile device is in offline mode. |

| Test Data | No test data needed |

| Test Steps | 1. Download course materials on the learner's mobile device.

2. Simulate an offline environment by disconnecting the mobile device from the internet.

3. Attempt to access the downloaded course materials on the mobile device.

4. Verify that the course materials are accessible and can be viewed offline. |

| Postconditions | None |

| Expected Result | Learners can access downloaded materials offline. |

| Severity | Major |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

**Test Case 5: Verify Mobile Platform Compatibility**

| Test Case ID | TC-MA-5 |

| --- | --- |

| Test Case Title | Verify that the online learning platform is compatible with various mobile devices and operating systems |

| Test Case Description | This test case aims to verify that the online learning platform is compatible with

various mobile devices and operating systems to ensure seamless access to course materials. |

| Test Suite | Mobile Accessibility |

| Test Priority | High |

| Preconditions | 1. Various mobile devices and operating systems (e.g., iOS, Android, Windows).

2. Mobile devices have a stable internet connection.

3. Learners have valid accounts on the online learning platform. |

| Test Data | No test data needed |

| Test Steps | 1. Access the online learning platform on various mobile devices and operating systems.

2. Verify that the platform is accessible and functions correctly on each device.

3. Attempt to view and access course materials on each device.

4. Verify that all course materials are displayed correctly and are readable on each device. |

| Postconditions | None |

| Expected Result | Online learning platform is compatible with various mobile devices and operating systems.

|

| Severity | Minor |

| Priority | High |

| Type of Testing | Functional Testing |

| Test Case Approach | Positive |

---

**Data - 57**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Resource Library

**Feature description -**

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

**Userstory title -**

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

**Acceptance criteria -**

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

**Generated Test Cases:**

Here are five detailed test cases for the Resource Library feature, as per the provided Test Case Creation Template.

**Test Case 1: Verify that Resource Library includes e-books, articles, and reference guides**

* **Test Case ID:** RES-001

* **Test Case Title:** Verify that Resource Library includes e-books, articles, and reference guides

* **Test Suite:** Resource Library Module

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the platform
 + Resource Library module is enabled
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the user account and navigate to the Resource Library module
 2. Verify that the module contains at least 3 e-books, 5 articles, and 2 reference guides
 3. Check if e-books are downloadable and accessible
* **Postconditions:** None
* **Expected Result:** Resource Library includes e-books, articles, and reference guides as per the feature description
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that Learners can search for specific materials**

* **Test Case ID:** RES-002
* **Test Case Title:** Verify that Learners can search for specific materials
* **Test Suite:** Resource Library Module
* **Test Priority:** Medium
* **Preconditions:**
 + User is logged in to the platform
 + Search functionality is enabled in the Resource Library module
* **Test Data:** A specific resource title (e.g., "Introduction to Python Programming")
* **Test Steps:**
 1. Log in to the user account and navigate to the Resource Library module

 2. Use the search function to search for the specific resource title

 3. Verify that the search result displays the matching resource

 4. Check if the searched resource can be viewed, downloaded, or accessed

* **Postconditions:** None

* **Expected Result:** Learners can search for specific resources and view/download/access them

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that Resources are categorized and tagged for easy navigation**

* **Test Case ID:** RES-003

* **Test Case Title:** Verify that Resources are categorized and tagged for easy navigation

* **Test Suite:** Resource Library Module

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the platform

 + Resource Library module is enabled

* **Test Data:** A specific resource (e.g., "E-book on Data Science")

* **Test Steps:**

 1. Log in to the user account and navigate to the Resource Library module

 2. Verify that the resource is categorized (e.g., under "Computer Science" or "Business")

 3. Check if the resource has relevant tags (e.g., "Data Science", "Python", "Excel")

 4. Test browsing through categories and tags to ensure correct resource display

* **Postconditions:** None

* **Expected Result:** Resources are categorized and tagged for easy navigation and discovery

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that Learners can download resources for offline access**

* **Test Case ID:** RES-004

* **Test Case Title:** Verify that Learners can download resources for offline access

* **Test Suite:** Resource Library Module

* **Test Priority:** Medium

* **Preconditions:**

 + User is logged in to the platform

 + Download functionality is enabled for resources

* **Test Data:** A specific downloadable resource (e.g., "PDF manual")

* **Test Steps:**

 1. Log in to the user account and navigate to the Resource Library module

 2. Select the downloadable resource

 3. Verify that the user can initiate the download process

 4. Check if the downloaded file is accessible and can be viewed

* **Postconditions:** None

* **Expected Result:** Learners can download resources for offline access

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that Instructors can recommend resources to learners**

* **Test Case ID:** RES-005

* **Test Case Title:** Verify that Instructors can recommend resources to learners

* **Test Suite:** Resource Library Module

* **Test Priority:** High

* **Preconditions:**

 + Instructor is logged in to the platform

 + Resource Library module is enabled

* **Test Data:** A specific resource (e.g., "Video tutorial on Data Visualization")

* **Test Steps:**

 1. Log in to the instructor account and navigate to the Resource Library module

 2. Select the resource to recommend

 3. Verify that the instructor can create a recommendation with the selected resource

 4. Check if the recommendation is displayed correctly on the learner's dashboard

* **Postconditions:** None

* **Expected Result:** Instructors can recommend resources to learners

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Each of these test cases is designed to thoroughly cover one or more of the acceptance criteria, ensuring comprehensive coverage of the Resource Library feature. By executing these test cases, we can identify potential defects early in the development cycle and ensure the high quality of the software product.

---

**Data - 58**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Peer Review

**Feature description -**

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

**Userstory title -**

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

**Acceptance criteria -**

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers remain anonymous to ensure unbiased feedback.']

**Generated Test Cases:**

Here are 5 detailed test cases for the Peer Review feature of the Online Learning Platform, following the provided Test Case Creation Template:

**Test Case 1: Verify that Learners Can Submit Assignments for Peer Review**

* Test Case ID: REV-001
* Test Case Title: Verify that learners can submit assignments for peer review
* Test Case Description: As a learner, I want to submit my assignments for peer review so that I can receive feedback from my peers. This test case ensures that learners can successfully submit their assignments for review by peers.
* Test Suite: Peer Review
* Test Priority: High
* Preconditions:
  * The learner has created an account on the platform
  * The learner has created and assigned a lesson
  * The learner has uploaded an assignment for peer review
* Test Data: No test data needed
* Test Steps:
  1. Log in to the platform as a learner
  2. Create a new lesson and assign it to myself
  3. Upload an assignment for peer review
  4. Click the "Submit for Review" button
  5. Verify that the assignment is successfully submitted for peer review
* Postconditions: The assignment is visible in the peer review dashboard for reviewers
* Expected Result: The learner can successfully submit the assignment for peer review
* Severity: Blocker
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 2: Verify that Peers Can Provide Feedback on Submitted Assignments**

* Test Case ID: REV-002

* Test Case Title: Verify that peers can provide feedback on submitted assignments

* Test Case Description: As a peer reviewer, I want to provide feedback on submitted assignments so that I can assist the learner in improving their work. This test case ensures that peers can successfully provide feedback on submitted assignments.

* Test Suite: Peer Review

* Test Priority: High

* Preconditions:

  * The learner has submitted an assignment for peer review

  * The peer reviewer has been assigned to review the assignment

* Test Data: No test data needed

* Test Steps:

  1. Log in to the platform as a peer reviewer

  2. View the peer review dashboard and select the assignment to review

  3. Click the "Provide Feedback" button

  4. Enter feedback comments and ratings

  5. Click the "Submit Feedback" button

  6. Verify that the feedback is successfully submitted

* Postconditions: The feedback is visible to the learner

* Expected Result: The peer reviewer can successfully provide feedback on the submitted assignment

* Severity: Blocker

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 3: Verify that a Structured Review Process Guides the Feedback**

* Test Case ID: REV-003

* Test Case Title: Verify that a structured review process guides the feedback

* Test Case Description: As a peer reviewer, I want to provide feedback using a structured review process so that I can ensure that my feedback is clear and actionable. This test case ensures that the structured review process guides the feedback provided by peers.

* Test Suite: Peer Review

* Test Priority: Medium

* Preconditions:

  * The learner has submitted an assignment for peer review

  * The peer reviewer has been assigned to review the assignment

* Test Data: No test data needed

* Test Steps:

  1. Log in to the platform as a peer reviewer

  2. View the peer review dashboard and select the assignment to review

  3. Click the "Provide Feedback" button

  4. Verify that the structured review process is displayed

  5. Enter feedback comments and ratings using the structured review process

  6. Click the "Submit Feedback" button

  7. Verify that the feedback is successfully submitted

* Postconditions: The feedback is visible to the learner

* Expected Result: The structured review process guides the feedback provided by the peer reviewer

* Severity: Major

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Verify that Learners Can View Feedback on Their Work**

* Test Case ID: REV-004
* Test Case Title: Verify that learners can view feedback on their work
* Test Case Description: As a learner, I want to view feedback on my assignments so that I can understand how to improve my work. This test case ensures that learners can successfully view feedback on their assignments.
* Test Suite: Peer Review
* Test Priority: High
* Preconditions:
  * The learner has submitted an assignment for peer review
  * The peer reviewer has provided feedback on the assignment
* Test Data: No test data needed
* Test Steps:
  1. Log in to the platform as a learner
  2. View the peer review dashboard and select the assignment to view feedback
  3. Verify that the feedback is displayed
  4. Verify that the feedback is clear and actionable
* Postconditions: The learner has viewed the feedback on their assignment
* Expected Result: The learner can successfully view feedback on their assignment
* Severity: Blocker
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 5: Verify that Reviewers Remain Anonymous to Ensure Unbiased Feedback**

* Test Case ID: REV-005

* Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback

* Test Case Description: As a learner, I want reviewers to remain anonymous so that I can receive unbiased feedback on my assignments. This test case ensures that reviewers remain anonymous to ensure unbiased feedback.

* Test Suite: Peer Review

* Test Priority: High

* Preconditions:

  * The learner has submitted an assignment for peer review

  * The peer reviewer has provided feedback on the assignment

* Test Data: No test data needed

* Test Steps:

  1. Log in to the platform as a learner

  2. View the peer review dashboard and select the assignment to view feedback

  3. Verify that the reviewer is anonymous

  4. Verify that the feedback is unbiased

* Postconditions: The learner has received unbiased feedback from anonymous reviewers

* Expected Result: Reviewers remain anonymous to ensure unbiased feedback

* Severity: Blocker

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

---

**Data - 59**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Certificates and Badges

**Feature description -**

Upon successful completion of courses or specific milestones, learners receive certificates and badges to recognize their achievements and accomplishments. These credentials provide tangible recognition of learners' efforts, skills, and expertise, enhancing their credentials and demonstrating their commitment to continuous learning and professional development.

**Userstory title -**

As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

**Acceptance criteria -**

['Learners earn certificates upon course completion.', 'Badges are awarded for specific achievements and milestones.', 'Certificates and badges are accessible through learner profiles.', 'Learners can share their certificates and badges on social media.', 'Instructors can issue custom badges for special achievements.']

**Generated Test Cases:**

Here are five detailed test cases for the feature "Certificates and Badges" in the Online Learning Platform:

**Test Case 1: Verify that learners earn certificates upon course completion**

* **Test Case ID:** CER-001

* **Test Case Title:** Verify that learners earn certificates upon course completion

* **Test Case Description:** This test case validates that learners receive a certificate upon completing a course. The test case ensures that the certificate is correctly issued, stored, and displayed in the learner's

profile.

* **Test Suite:** Certificates and Badges

* **Test Priority:** High

* **Preconditions:**

 + Learner is enrolled in a course.

 + Course completion criteria are met.

 + Certificate settings are correctly configured.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Enroll a learner in a course.

 2. Complete the course successfully.

 3. Verify that the certificate is issued and stored in the learner's profile.

 4. Check if the certificate is displayed correctly in the learner's profile.

* **Postconditions:**

 + Certificate is issued and stored in the learner's profile.

 + Certificate is displayed correctly in the learner's profile.

* **Expected Result:** A valid certificate is issued and displayed in the learner's profile.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that badges are awarded for specific achievements and milestones**

* **Test Case ID:** BAD-001

* **Test Case Title:** Verify that badges are awarded for specific achievements and milestones

* **Test Case Description:** This test case validates that badges are correctly awarded to learners for specific

achievements and milestones. The test case ensures that badge settings are correctly configured and that badges are displayed in the learner's profile.

* **Test Suite:** Certificates and Badges

* **Test Priority:** Medium

* **Preconditions:**

 + Learner has enrolled in a course with badge settings enabled.

 + Course completion criteria and milestones are met.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Enroll a learner in a course with badge settings enabled.

 2. Complete the course successfully, meeting the badge settings.

 3. Verify that the badge is awarded and stored in the learner's profile.

 4. Check if the badge is displayed correctly in the learner's profile.

* **Postconditions:**

 + Badge is awarded and stored in the learner's profile.

 + Badge is displayed correctly in the learner's profile.

* **Expected Result:** A valid badge is awarded and displayed in the learner's profile.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that certificates and badges are accessible through learner profiles**

* **Test Case ID:** CLP-001

* **Test Case Title:** Verify that certificates and badges are accessible through learner profiles

* **Test Case Description:** This test case validates that learners can access their certificates and badges

through their profiles. The test case ensures that certificates and badges are correctly displayed in the learner's profile.

* **Test Suite:** Certificates and Badges
* **Test Priority:** High
* **Preconditions:**
 + Learner has enrolled in a course and completed it.
 + Learner has been awarded certificates and badges.
* **Test Data:** No test data needed
* **Test Steps:**
 1. Enroll a learner in a course and complete it successfully.
 2. Award certificates and badges to the learner.
 3. Verify that the learner can access their certificates and badges through their profile.
 4. Check if the certificates and badges are displayed correctly in the learner's profile.
* **Postconditions:**
 + Learner can access their certificates and badges through their profile.
 + Certificates and badges are displayed correctly in the learner's profile.
* **Expected Result:** Learner can access their certificates and badges through their profile.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that learners can share their certificates and badges on social media**

* **Test Case ID:** SMC-001
* **Test Case Title:** Verify that learners can share their certificates and badges on social media
* **Test Case Description:** This test case validates that learners can share their certificates and badges on

social media platforms. The test case ensures that the sharing functionality is correctly implemented and that certificates and badges are displayed correctly on social media.

* **Test Suite:** Certificates and Badges
* **Test Priority:** Medium
* **Preconditions:**
 + Learner has enrolled in a course and completed it.
 + Learner has been awarded certificates and badges.
* **Test Data:** No test data needed
* **Test Steps:**
 1. Enroll a learner in a course and complete it successfully.
 2. Award certificates and badges to the learner.
 3. Verify that the learner can share their certificates and badges on social media.
 4. Check if the certificates and badges are displayed correctly on social media.
* **Postconditions:**
 + Learner can share their certificates and badges on social media.
 + Certificates and badges are displayed correctly on social media.
* **Expected Result:** Learner can share their certificates and badges on social media.
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that instructors can issue custom badges for special achievements**

* **Test Case ID:** ICB-001
* **Test Case Title:** Verify that instructors can issue custom badges for special achievements
* **Test Case Description:** This test case validates that instructors can issue custom badges for special

achievements. The test case ensures that the custom badge settings are correctly configured and that the badge is awarded to the learner.

* **Test Suite:** Certificates and Badges

* **Test Priority:** High

* **Preconditions:**

 + Instructor has access to the course settings.

 + Custom badge settings are enabled.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor.

 2. Configure custom badge settings for a course.

 3. Award the custom badge to a learner.

 4. Verify that the custom badge is awarded and stored in the learner's profile.

* **Postconditions:**

 + Custom badge is awarded and stored in the learner's profile.

 + Custom badge is displayed correctly in the learner's profile.

* **Expected Result:** Custom badge is awarded and displayed in the learner's profile.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 60**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Gamification Elements

**Feature description -**

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

**Userstory title -**

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

**Acceptance criteria -**

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

**Generated Test Cases:**

Here are five detailed test cases for the Gamification Elements feature, covering all the acceptance criteria from the user story:

**Test Case 1: Earn Points for Completing Activities**

* **Test Case ID:** LC-001

* **Test Case Title:** Verify that learners earn points for completing activities.

* **Test Case Description:** This test case ensures that learners are awarded points for completing activities,

such as watching video lectures, completing quizzes, and downloading resources.

* **Test Suite:** Gamification Elements

* **Test Priority:** High

* **Preconditions:** Learner must be logged in, and activity must be available.

* **Test Data:** Test user, activity details (e.g., video lecture, quiz).

* **Test Steps:**

1. Log in as a test user.

2. Navigate to the relevant activity (e.g., video lecture).

3. Complete the activity (e.g., watch the video lecture or complete the quiz).

4. Verify that points are awarded and displayed in the learner's progress tracking.

* **Postconditions:** None.

* **Expected Result:** Learner is awarded points for completing the activity, and progress is updated.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Levels Achievement**

* **Test Case ID:** LC-002

* **Test Case Title:** Verify that levels are achieved based on accumulated points.

* **Test Case Description:** This test case ensures that learners achieve new levels as they accumulate points.

* **Test Suite:** Gamification Elements

* **Test Priority:** Medium

* **Preconditions:** Learner must have earned sufficient points to reach the next level.

* **Test Data:** Test user, level details (e.g., level requirements).

* **Test Steps:**

1. Log in as a test user with a balance of points below the level requirement.

2. Complete activities to accumulate points and reach the next level.

3. Verify that the learner is displayed at the new level in their progress tracking.

* **Postconditions:** None.

* **Expected Result:** Learner is promoted to the new level as expected.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Leaderboards Display Top-Performing Learners**

* **Test Case ID:** LC-003

* **Test Case Title:** Verify that leaderboards display top-performing learners.

* **Test Case Description:** This test case ensures that leaderboards accurately display top-performing learners.

* **Test Suite:** Gamification Elements

* **Test Priority:** Medium

* **Preconditions:** Leaderboard data must be available and populated.

* **Test Data:** Leaderboard data (e.g., points, ranks).

* **Test Steps:**

1. Log in as an administrator to access leaderboard data.

2. Verify that the leaderboard accurately displays the top-performing learners, sorted by points earned.

3. Test filtering and sorting functionality.

* **Postconditions:** None.

* **Expected Result:** Leaderboard displays top-performing learners as expected.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Badges Awarded for Reaching Milestones**

* **Test Case ID:** LC-004

* **Test Case Title:** Verify that badges are awarded for reaching milestones.

* **Test Case Description:** This test case ensures that learners are awarded badges for reaching milestones.

* **Test Suite:** Gamification Elements

* **Test Priority:** Low

* **Preconditions:** Milestone details must be configured and accessible.

* **Test Data:** Milestone details (e.g., milestone criteria, badge name).

* **Test Steps:**

 1. Log in as a test user and attempt to reach the milestone.

 2. Verify that the badge is awarded and displayed in the learner's progress tracking.

 3. Test the badge display across various views and dashboards.

* **Postconditions:** None.

* **Expected Result:** Learner is awarded the badge for reaching the milestone.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Learner Progress Viewed in Gamification Dashboard**

* **Test Case ID:** LC-005

* **Test Case Title:** Verify that learners can view their progress in a gamification dashboard.

* **Test Case Description:** This test case ensures that learners can access and view their progress tracking and badges earned in the gamification dashboard.

* **Test Suite:** Gamification Elements

* **Test Priority:** Low

* **Preconditions:** Learner must have progress and badges to view.

* **Test Data:** None.

* **Test Steps:**

 1. Log in as a test user with earned points and badges.

 2. Navigate to the gamification dashboard.

 3. Verify that progress and badges are accurately displayed in the dashboard.

 4. Test filters and search functionality.

* **Postconditions:** None.

* **Expected Result:** Learner can view their progress and badges as expected.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases cover all the acceptance criteria from the user story, focusing on the Gamification Elements feature. Each test case is designed to ensure that learners can earn points, achieve levels, view leaderboards, receive badges, and track their progress in the gamification dashboard.

---

**Data - 61**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Multilingual Support

**Feature description -**

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and facilitating access to educational resources in their preferred language. Through multilingual interfaces, course materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all learners.

**Userstory title -**

As a learner, I want multilingual support so that I can access the platform in my preferred language.

**Acceptance criteria -**

['The platform interface is available in multiple languages.', 'Course materials are available in different languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated help and support resources.', 'Instructors can provide content in multiple languages.']

**Generated Test Cases:**

Here are five detailed test cases for the Multilingual Support feature of the Online Learning Platform, following the provided Test Case Creation Template:

**Test Case 1: Verify that the platform interface is available in multiple languages**

* **Test Case ID:** ML-001

* **Test Case Title:** Verify that the platform interface is available in multiple languages

* **Test Case Description:** This test case verifies that the Online Learning Platform interface is accessible and functional in multiple languages, catering to learners from diverse linguistic backgrounds.

* **Test Suite:** Multilingual Support

* **Test Priority:** High

* **Preconditions:**

 + The user has created an account and logged in.

 + The system language is set to English (Default).

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as a registered user.

 2. Change the system language to French using the language dropdown menu.

 3. Verify that the platform interface (headers, buttons, and text) is displayed in French.

 4. Repeat steps 2-3 for other supported languages (e.g., Spanish, German, Chinese).

* **Postconditions:**

 + The user can exit the test by logging out.

 + The system language should be reset to the default value.

* **Expected Result:** The platform interface should be displayed in different languages without any errors or functionality issues.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that course materials are available in different languages**

* **Test Case ID:** ML-002

* **Test Case Title:** Verify that course materials are available in different languages

* **Test Case Description:** This test case verifies that course materials, such as video lectures, downloadable resources, and discussion forums, are accessible and functional in multiple languages.

* **Test Suite:** Multilingual Support

* **Test Priority:** Medium

* **Preconditions:**

 + The user has created an account and logged in.

 + The system language is set to English (Default).

 + A course with multiple languages is available.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as a registered user.

 2. Select a course with multiple languages.

 3. Verify that the course materials (video lectures, downloadable resources, discussion forums) are available in different languages.

 4. Repeat steps 2-3 for other courses with multiple languages.

* **Postconditions:**

 + The user can exit the test by logging out.

 + The system language should be reset to the default value.

* **Expected Result:** Course materials should be available in different languages without any errors or functionality issues.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that learners can select their preferred language at any time**

* **Test Case ID:** ML-003
* **Test Case Title:** Verify that learners can select their preferred language at any time
* **Test Case Description:** This test case verifies that learners can change the system language at any time and that the new language selection is applied accurately.
* **Test Suite:** Multilingual Support
* **Test Priority:** High
* **Preconditions:**
 + The user has created an account and logged in.
 + The system language is set to English (Default).
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the platform as a registered user.
 2. Change the system language to French using the language dropdown menu.
 3. Verify that the new language selection is applied correctly (interface, course materials, etc.).
 4. Repeat steps 2-3 for other languages.
* **Postconditions:**
 + The user can exit the test by logging out.
 + The system language should be reset to the default value.
* **Expected Result:** Learners can change the system language at any time and the new language selection is applied accurately.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that multilingual support includes translated help and support resources**

* **Test Case ID:** ML-004

* **Test Case Title:** Verify that multilingual support includes translated help and support resources

* **Test Case Description:** This test case verifies that multilingual support includes translated help and support resources, ensuring that learners can access assistance in their preferred language.

* **Test Suite:** Multilingual Support

* **Test Priority:** Medium

* **Preconditions:**

 + The user has created an account and logged in.

 + The system language is set to English (Default).

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as a registered user.

 2. Access the help and support resources (FAQs, contact forms, etc.).

 3. Verify that the help and support resources are available in different languages.

 4. Repeat steps 2-3 for other languages.

* **Postconditions:**

 + The user can exit the test by logging out.

 + The system language should be reset to the default value.

* **Expected Result:** Help and support resources should be available in different languages without any errors or functionality issues.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that instructors can provide content in multiple languages**

* **Test Case ID:** ML-005

* **Test Case Title:** Verify that instructors can provide content in multiple languages

* **Test Case Description:** This test case verifies that instructors can create and provide course content in multiple languages, ensuring that learners can access educational resources in their preferred language.

* **Test Suite:** Multilingual Support

* **Test Priority:** High

* **Preconditions:**

 + The instructor has created an account and logged in.

 + The system language is set to English (Default).

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as an instructor.

 2. Create a new course and add content in multiple languages.

 3. Verify that the course content is available in different languages.

 4. Repeat steps 2-3 for other courses.

* **Postconditions:**

 + The instructor can exit the test by logging out.

 + The system language should be reset to the default value.

* **Expected Result:** Instructors can create and provide course content in multiple languages without any errors or functionality issues.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 62**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Accessibility Features

**Feature description -**

The platform prioritizes accessibility by incorporating features such as screen reader compatibility, adjustable font sizes, and color contrast options. These accessibility features ensure that learners with disabilities can access and navigate the platform with ease, promoting inclusivity and equitable access to educational opportunities.

**Userstory title -**

As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

**Acceptance criteria -**

['The platform is compatible with screen readers.', 'Learners can adjust font sizes for better readability.', 'Color contrast options are available for improved visibility.', 'Accessibility features are easy to activate and customize.', 'Support resources are available to assist learners with disabilities.']

**Generated Test Cases:**

Here are the test cases for the Accessibility Features user story:

**Test Case 1: Verify that the platform is compatible with screen readers**

* **Test Case ID:** TF-ACC-001
* **Test Case Title:** Verify that the platform is compatible with screen readers
* **Test Case Description:** This test case verifies that the online learning platform is compatible with screen readers, ensuring that learners with visual impairments can access and navigate the platform effectively. The test case covers the acceptance criterion "The platform is compatible with screen readers."
* **Test Suite:** Accessibility Features
* **Test Priority:** High
* **Preconditions:**
 + The learner has a screen reader installed on their device
 + The learner has a device with a screen reader compatible browser
* **Test Data:** No test data needed
* **Test Steps:**
 1. Launch the platform in a browser with a screen reader enabled
 2. Navigate through the platform's menu and content using the screen reader
 3. Verify that the screen reader accurately reads out all menu items and content
 4. Verify that all interactive elements (e.g., buttons, links) are correctly identified by the screen reader
* **Postconditions:**
 + Verify that the platform remains stable and functional during the test
 + Verify that the screen reader continues to accurately read out menu items and content
* **Expected Result:** The platform is compatible with screen readers, and the screen reader accurately reads out all menu items and content.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that learners can adjust font sizes for better readability**

* **Test Case ID:** TF-ACC-002

* **Test Case Title:** Verify that learners can adjust font sizes for better readability

* **Test Case Description:** This test case verifies that learners can adjust font sizes for better readability, ensuring that learners with visual impairments can access and navigate the platform effectively. The test case covers the acceptance criterion "Learners can adjust font sizes for better readability."

* **Test Suite:** Accessibility Features

* **Test Priority:** High

* **Preconditions:**

 + The learner has a device with a text size adjustment capability

* **Test Data:** No test data needed

* **Test Steps:**

 1. Launch the platform in a browser with screen scaling enabled

 2. Verify that the learner can adjust the font size using the browser's zoom feature

 3. Verify that the learner can adjust the font size using the platform's built-in font size adjustment feature (if available)

 4. Verify that the learner can view the platform's content with the adjusted font size

* **Postconditions:**

 + Verify that the platform remains stable and functional during the test

 + Verify that the learner can adjust the font size again and view the platform's content with the adjusted font size

* **Expected Result:** Learners can adjust font sizes for better readability, and the platform's content displays correctly with the adjusted font size.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that color contrast options are available for improved visibility**

* **Test Case ID:** TF-ACC-003
* **Test Case Title:** Verify that color contrast options are available for improved visibility
* **Test Case Description:** This test case verifies that learners can access color contrast options for improved visibility, ensuring that learners with visual impairments can access and navigate the platform effectively. The test case covers the acceptance criterion "Color contrast options are available for improved visibility."
* **Test Suite:** Accessibility Features
* **Test Priority:** High
* **Preconditions:**
 + The learner has a device with sufficient color profile capabilities
* **Test Data:** No test data needed
* **Test Steps:**
 1. Launch the platform in a browser with color profile adjustment enabled
 2. Verify that the learner can access color contrast options via the platform's display settings
 3. Verify that the learner can adjust the color contrast to meet their specific needs
 4. Verify that the learner can view the platform's content with the adjusted color contrast
* **Postconditions:**
 + Verify that the platform remains stable and functional during the test
 + Verify that the learner can adjust the color contrast again and view the platform's content with the adjusted color contrast
* **Expected Result:** Learners can access color contrast options for improved visibility, and the platform's content displays correctly with the adjusted color contrast.
* **Severity:** Major
* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that accessibility features are easy to activate and customize**

* **Test Case ID:** TF-ACC-004

* **Test Case Title:** Verify that accessibility features are easy to activate and customize

* **Test Case Description:** This test case verifies that learners can easily activate and customize accessibility features, ensuring that learners with disabilities can access and navigate the platform effectively. The test case covers the acceptance criterion "Accessibility features are easy to activate and customize."

* **Test Suite:** Accessibility Features

* **Test Priority:** High

* **Preconditions:**

 + The learner has a device with accessibility features enabled

* **Test Data:** No test data needed

* **Test Steps:**

 1. Launch the platform in a browser with accessibility features enabled

 2. Verify that the learner can easily navigate to the accessibility settings

 3. Verify that the learner can easily activate and customize accessibility features

 4. Verify that the learner can view the platform's content with the activated and customized accessibility features

* **Postconditions:**

 + Verify that the platform remains stable and functional during the test

 + Verify that the learner can easily access and customize accessibility features again

* **Expected Result:** Learners can easily activate and customize accessibility features, and the platform's content displays correctly with the activated and customized accessibility features.

* **Severity:** Major

* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that support resources are available to assist learners with disabilities**

* **Test Case ID:** TF-ACC-005
* **Test Case Title:** Verify that support resources are available to assist learners with disabilities
* **Test Case Description:** This test case verifies that learners can access support resources for assistance with accessibility features, ensuring that learners with disabilities can access and navigate the platform effectively. The test case covers the acceptance criterion "Support resources are available to assist learners with disabilities."
* **Test Suite:** Accessibility Features
* **Test Priority:** High
* **Preconditions:**
 + The learner has encountered an accessibility issue with the platform
* **Test Data:** No test data needed
* **Test Steps:**
 1. Launch the platform in a browser with an accessibility issue (e.g., insufficient color contrast)
 2. Verify that the learner can access support resources (e.g., FAQs, contact form)
 3. Verify that the learner can submit a support request for assistance with accessibility features
 4. Verify that the learner receives adequate support for their accessibility needs
* **Postconditions:**
 + Verify that the platform remains stable and functional during the test
 + Verify that the learner receives a resolution to their accessibility issue
* **Expected Result:** Learners can access support resources for assistance with accessibility features, and the platform provides adequate support for learners with disabilities.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 63**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Discussion Moderation

**Feature description -**

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

**Userstory title -**

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

**Acceptance criteria -**

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

**Generated Test Cases:**

Here are 5 detailed test cases for the Discussion Moderation feature:

**Test Case 1: Verify that instructors can edit posts in discussions**

* **Test Case ID:** MOD-001
* **Test Case Title:** Verify that instructors can edit posts in discussions
* **Test Case Description:** Instructors should be able to edit existing posts in discussion forums to maintain a consistent and respectful learning environment.
* **Test Suite:** Discussion Moderation
* **Test Priority:** High
* **Preconditions:**
 + Instructor is logged in to the platform.
 + Discussion forum with existing posts is created.
* **Test Data:** No test data needed.
* **Test Steps:**
 1. Log in as an instructor.
 2. Navigate to the discussion forum with existing posts.
 3. Select a post to edit.
 4. Click the "Edit Post" button.
 5. Modify the post content.
 6. Click the "Save Changes" button.
* **Postconditions:**
 + Post content is updated.
 + Edit history is logged.

* **Expected Result:** Instructor can successfully edit a post in the discussion forum.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that instructors can delete posts in discussions**

* **Test Case ID:** MOD-002

* **Test Case Title:** Verify that instructors can delete posts in discussions

* **Test Case Description:** Instructors should be able to delete existing posts in discussion forums to maintain a clean and respectful learning environment.

* **Test Suite:** Discussion Moderation

* **Test Priority:** High

* **Preconditions:**

 + Instructor is logged in to the platform.

 + Discussion forum with existing posts is created.

* **Test Data:** No test data needed.

* **Test Steps:**

 1. Log in as an instructor.

 2. Navigate to the discussion forum with existing posts.

 3. Select a post to delete.

 4. Click the "Delete Post" button.

 5. Confirm deletion.

* **Postconditions:**

 + Post is deleted.

 + Delete history is logged.

* **Expected Result:** Instructor can successfully delete a post in the discussion forum.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that instructors can mute or ban disruptive users**

* **Test Case ID:** MOD-003
* **Test Case Title:** Verify that instructors can mute or ban disruptive users
* **Test Case Description:** Instructors should be able to mute or ban users who are engaging in disruptive behavior in discussion forums to maintain a respectful learning environment.
* **Test Suite:** Discussion Moderation
* **Test Priority:** High
* **Preconditions:**
 + Instructor is logged in to the platform.
 + Discussion forum with existing posts is created.
 + Disruptive user is participating in the discussion.
* **Test Data:** No test data needed.
* **Test Steps:**
 1. Log in as an instructor.
 2. Navigate to the discussion forum with existing posts.
 3. Identify a disruptive user.
 4. Click the "Mute User" or "Ban User" button.
 5. Confirm action.
* **Postconditions:**
 + User is muted or banned.

 + Moderation history is logged.

* **Expected Result:** Instructor can successfully mute or ban a disruptive user in the discussion forum.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that moderation actions are logged and reported**

* **Test Case ID:** MOD-004

* **Test Case Title:** Verify that moderation actions are logged and reported

* **Test Case Description:** Moderation actions taken by instructors should be logged and reported to maintain transparency and accountability in the discussion forum.

* **Test Suite:** Discussion Moderation

* **Test Priority:** Medium

* **Preconditions:**

 + Instructor is logged in to the platform.

 + Discussion forum with existing posts is created.

 + Moderation actions have been taken.

* **Test Data:** No test data needed.

* **Test Steps:**

 1. Log in as an instructor.

 2. Navigate to the discussion forum with existing posts.

 3. View moderation history.

 4. Verify that moderation actions are logged.

* **Postconditions:**

 + Moderation history is accurate.

+ Reports are generated.

* **Expected Result:** Moderation actions are logged and reported correctly.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that community guidelines are visible to all participants**

* **Test Case ID:** MOD-005

* **Test Case Title:** Verify that community guidelines are visible to all participants

* **Test Case Description:** Community guidelines should be visible to all participants in the discussion forum to ensure that everyone is aware of expected behavior.

* **Test Suite:** Discussion Moderation

* **Test Priority:** Low

* **Preconditions:**

 + Instructor is logged in to the platform.

 + Discussion forum with existing posts is created.

* **Test Data:** No test data needed.

* **Test Steps:**

 1. Log in as a participant.

 2. Navigate to the discussion forum.

 3. View community guidelines.

 4. Verify that guidelines are displayed correctly.

* **Postconditions:**

 + Guidelines are visible to all participants.

 + No errors occur.

* **Expected Result:** Community guidelines are visible to all participants in the discussion forum.

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 64**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Learning Analytics

**Feature description -**

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement, performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning experience to meet learners' needs effectively.

**Userstory title -**

As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

**Acceptance criteria -**

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.', 'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.', 'Instructors can export analytics reports for further analysis.']

**Generated Test Cases:**

Here are five detailed test cases for the Learning Analytics feature, each covering one or more of the acceptance criteria:

**Test Case 1: Verify that Learning Analytics Dashboards Display Engagement Metrics**

* **Test Case ID:** LA-001
* **Test Case Title:** Verify that Learning Analytics Dashboards Display Engagement Metrics
* **Test Case Description:** This test case ensures that the learning analytics dashboards display engagement metrics for instructors to understand learner engagement.
* **Test Suite:** Learning Analytics
* **Test Priority:** High
* **Preconditions:**
 + Instructor has created a course with at least 10 learners.
 + Learners have accessed the course and interacted with the content.
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as an instructor and navigate to the learning analytics dashboard.
 2. Verify that the dashboard displays engagement metrics, such as course views, video watches, and discussion forum posts.
 3. Check that the metrics are accurate and up-to-date.
* **Postconditions:** None
* **Expected Result:** The learning analytics dashboard displays accurate and up-to-date engagement metrics for the instructor to view.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that Performance Trends are Visualized Over Time**

* **Test Case ID:** LA-002

* **Test Case Title:** Verify that Performance Trends are Visualized Over Time

* **Test Case Description:** This test case ensures that performance trends are visualized over time for instructors to understand learner performance.

* **Test Suite:** Learning Analytics

* **Test Priority:** High

* **Preconditions:**

 + Instructor has created a course with at least 10 learners.

 + Learners have accessed the course and interacted with the content.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor and navigate to the learning analytics dashboard.

 2. Verify that the dashboard displays performance trends over time, such as learner progress and completion rates.

 3. Check that the trends are accurate and up-to-date.

* **Postconditions:** None

* **Expected Result:** The learning analytics dashboard displays accurate and up-to-date performance trends over time for the instructor to view.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that Instructors Can Access Detailed Reports on Learner Progress**

* **Test Case ID:** LA-003
* **Test Case Title:** Verify that Instructors Can Access Detailed Reports on Learner Progress
* **Test Case Description:** This test case ensures that instructors can access detailed reports on learner progress to understand learner performance.
* **Test Suite:** Learning Analytics
* **Test Priority:** High
* **Preconditions:**
 + Instructor has created a course with at least 10 learners.
 + Learners have accessed the course and interacted with the content.
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as an instructor and navigate to the learning analytics dashboard.
 2. Verify that the dashboard displays detailed reports on learner progress, such as learner completion rates and time spent on course content.
 3. Check that the reports are accurate and up-to-date.
* **Postconditions:** None
* **Expected Result:** The learning analytics dashboard displays accurate and up-to-date detailed reports on learner progress for the instructor to view.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that Analytics Data is Updated in Real-Time**

* **Test Case ID:** LA-004

* **Test Case Title:** Verify that Analytics Data is Updated in Real-Time

* **Test Case Description:** This test case ensures that analytics data is updated in real-time to reflect changes in learner engagement and performance.

* **Test Suite:** Learning Analytics

* **Test Priority:** High

* **Preconditions:**

 + Instructor has created a course with at least 10 learners.

 + Learners have accessed the course and interacted with the content.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor and navigate to the learning analytics dashboard.

 2. Make changes to the course content or add new learners.

 3. Verify that the analytics data is updated in real-time to reflect the changes.

* **Postconditions:** None

* **Expected Result:** The learning analytics dashboard displays up-to-date analytics data in real-time for the instructor to view.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that Instructors Can Export Analytics Reports for Further Analysis**

* **Test Case ID:** LA-005

* **Test Case Title:** Verify that Instructors Can Export Analytics Reports for Further Analysis

* **Test Case Description:** This test case ensures that instructors can export analytics reports for further analysis to understand learner engagement and performance.

* **Test Suite:** Learning Analytics

* **Test Priority:** Medium

* **Preconditions:**

 + Instructor has created a course with at least 10 learners.

 + Learners have accessed the course and interacted with the content.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor and navigate to the learning analytics dashboard.

 2. Verify that the dashboard allows instructors to export analytics reports in various formats, such as CSV or Excel.

 3. Check that the exported reports are accurate and up-to-date.

* **Postconditions:** None

* **Expected Result:** Instructors can export accurate and up-to-date analytics reports for further analysis.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 65**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Social Learning Features

**Feature description -**

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

**Userstory title -**

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

**Acceptance criteria -**

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

**Generated Test Cases:**

Here are five (5) detailed test cases, each thoroughly covering one or more of the acceptance criteria from the user story.

**Test Case 1: Learners Can Create and Join Groups for Projects**

Test Case ID: OLP-SOC-001

Test Case Title: Verify that learners can create and join groups for projects

Test Case Description: This test case aims to ensure that learners can successfully create and join groups for projects within the online learning platform.

Test Suite: Social Learning Features

Test Priority: High

Preconditions: Learner has a valid account, and the social learning feature is enabled.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a learner.

2. Navigate to the group creation page.

3. Create a new group by providing a group name, description, and setting the group type to 'project'.

4. Verify that the group is created successfully and appears in the learner's group list.

5. Join the newly created group.

6. Verify that the learner can view the group details, member list, and group messages.

Postconditions: Group is created, and learner can join it.

Expected Result: Learner can create and join groups for projects successfully.

Severity: None

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Collaboration Tools are Available for Group Communication**

Test Case ID: OLP-SOC-002

Test Case Title: Verify that collaboration tools are available for group communication

Test Case Description: This test case aims to ensure that collaboration tools are available for group communication within the online learning platform.

Test Suite: Social Learning Features

Test Priority: High

Preconditions: Group is created with members, and the social learning feature is enabled.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform as a member of a group.

2. Navigate to the group chat or discussion board.

3. Post a message or start a discussion.

4. Verify that other group members can view the message or join the discussion.

5. Engage in group conversation using collaboration tools.

Postconditions: Collaboration tools are accessible for group communication.

Expected Result: Collaboration tools are available for group communication.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Learners Can Share Resources and Feedback Within Groups**

Test Case ID: OLP-SOC-003

Test Case Title: Verify that learners can share resources and feedback within groups

Test Case Description: This test case aims to ensure that learners can successfully share resources and feedback within groups within the online learning platform.

Test Suite: Social Learning Features

Test Priority: High

Preconditions: Group is created with members, and the social learning feature is enabled.

Test Data: Test resources (files, images, videos)

Test Steps:

1. Log in to the online learning platform as a group member.

2. Upload or attach resources (e.g., file, image, video) to the group chat or discussion board.

3. Verify that the shared resources are accessible to other group members.

4. Post feedback on group members' resources or projects.

5. Verify that other group members can view and engage with the feedback.

Postconditions: Learners can share resources and feedback within groups.

Expected Result: Learners can share resources and feedback within groups.

Severity: None

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Instructors Can Assign Group Projects and Monitor Progress**

Test Case ID: OLP-SOC-004

Test Case Title: Verify that instructors can assign group projects and monitor progress

Test Case Description: This test case aims to ensure that instructors can successfully assign group projects and monitor progress within the online learning platform.

Test Suite: Social Learning Features

Test Priority: High

Preconditions: Instructor has a valid account, group is created with members, and the social learning feature is enabled.

Test Data: Test course content, project requirements, and grades.

Test Steps:

1. Log in to the online learning platform as an instructor.

2. Assign a group project to a group using the instructor dashboard.

3. Set up project requirements and expectations for learners.

4. Verify that the assigned project is visible to group members and accessible on their course page.

5. Monitor learners' progress by checking project submissions, feedback, and grades.

Postconditions: Instructors can assign and monitor group projects.

Expected Result: Instructors can assign group projects and monitor progress.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Social Sharing Features Allow Learners to Share Achievements**

Test Case ID: OLP-SOC-005

Test Case Title: Verify that social sharing features allow learners to share achievements

Test Case Description: This test case aims to ensure that social sharing features enable learners to share their achievements and course completions on external social media platforms.

Test Suite: Social Learning Features

Test Priority: High

Preconditions: Learner has a valid account, course completion or certification.

Test Data: Social media login credentials, learner achievements and certificates.

Test Steps:

1. Log in to the online learning platform as a learner who has completed a course or earned a certificate.
2. Navigate to the social sharing feature in the learner dashboard or course completion page.
3. Verify that the learner can select social media platforms (e.g., LinkedIn, Facebook, Twitter) to share achievements.
4. Authenticate with the selected social media platform.
5. Share the achievement or course completion on the external social media platform.

Postconditions: Learner can share achievements on social media.

Expected Result: Social sharing features enable learners to share achievements on social media.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

---

**Data - 66**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Content Authoring Tools

**Feature description -**

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

**Userstory title -**

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

**Acceptance criteria -**

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

**Generated Test Cases:**

Here are five detailed test cases, each thoroughly covering one or more of the acceptance criteria:

**Test Case 1: Verify that instructors can create multimedia presentations**

* **Test Case ID:** OTL-1
* **Test Case Title:** Verify that instructors can create multimedia presentations
* **Test Case Description:** This test case aims to verify that instructors can successfully create multimedia presentations using the content authoring tools. The test case will cover the creation of a multimedia presentation, including adding text, images, and videos.
* **Test Suite:** Content Authoring Tools
* **Test Priority:** High
* **Preconditions:**
 + Instructor has logged in to the platform
 + Instructor has access to the content authoring tools
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the platform as an instructor
 2. Navigate to the content authoring tools
 3. Click on the "Create Multimedia Presentation" button
 4. Add text, images, and videos to the presentation
 5. Save and preview the presentation
* **Postconditions:**
 + Presentation is saved successfully
 + Presentation is previewed correctly

* **Expected Result:** Instructor can successfully create a multimedia presentation

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that interactive modules can be developed and integrated**

* **Test Case ID:** OTL-2

* **Test Case Title:** Verify that interactive modules can be developed and integrated

* **Test Case Description:** This test case aims to verify that instructors can successfully develop and integrate interactive modules using the content authoring tools. The test case will cover the creation of an interactive module, including adding quizzes, games, and simulations.

* **Test Suite:** Content Authoring Tools

* **Test Priority:** High

* **Preconditions:**

 + Instructor has logged in to the platform

 + Instructor has access to the content authoring tools

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as an instructor

 2. Navigate to the content authoring tools

 3. Click on the "Create Interactive Module" button

 4. Add quizzes, games, and simulations to the module

 5. Save and integrate the module into a course

* **Postconditions:**

 + Module is saved successfully

 + Module is integrated correctly into the course
* **Expected Result:** Instructor can successfully develop and integrate an interactive module
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that rich media resources, such as videos and animations, are supported**

* **Test Case ID:** OTL-3
* **Test Case Title:** Verify that rich media resources, such as videos and animations, are supported
* **Test Case Description:** This test case aims to verify that instructors can successfully upload and play rich media resources, such as videos and animations, using the content authoring tools.
* **Test Suite:** Content Authoring Tools
* **Test Priority:** Medium
* **Preconditions:**
 + Instructor has logged in to the platform
 + Instructor has access to the content authoring tools
* **Test Data:** Video and animation files
* **Test Steps:**
 1. Log in to the platform as an instructor
 2. Navigate to the content authoring tools
 3. Click on the "Upload Rich Media Resource" button
 4. Upload a video and animation file
 5. Preview the video and animation
* **Postconditions:**
 + Video and animation are uploaded successfully

+ Video and animation are played correctly

* **Expected Result:** Instructor can successfully upload and play rich media resources
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that content authoring tools are user-friendly and intuitive**

* **Test Case ID:** OTL-4
* **Test Case Title:** Verify that content authoring tools are user-friendly and intuitive
* **Test Case Description:** This test case aims to verify that the content authoring tools are easy to use and provide an intuitive experience for instructors.
* **Test Suite:** Content Authoring Tools
* **Test Priority:** High
* **Preconditions:**
  + Instructor has logged in to the platform
  + Instructor has access to the content authoring tools
* **Test Data:** No test data needed
* **Test Steps:**
  1. Log in to the platform as an instructor
  2. Navigate to the content authoring tools
  3. Attempt to create a multimedia presentation without any guidance
  4. Attempt to create an interactive module without any guidance
  5. Evaluate the user experience
* **Postconditions:**
  + Instructor is able to create a multimedia presentation and interactive module without significant difficulty

+ Instructor is satisfied with the user experience

* **Expected Result:** Content authoring tools are user-friendly and intuitive

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Usability Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that instructors can preview and edit content before publishing**

* **Test Case ID:** OTL-5

* **Test Case Title:** Verify that instructors can preview and edit content before publishing

* **Test Case Description:** This test case aims to verify that instructors can successfully preview and edit content, including multimedia presentations and interactive modules, before publishing it to the platform.

* **Test Suite:** Content Authoring Tools

* **Test Priority:** High

* **Preconditions:**

 + Instructor has logged in to the platform

 + Instructor has access to the content authoring tools

* **Test Data:** Completed multimedia presentation or interactive module

* **Test Steps:**

 1. Log in to the platform as an instructor

 2. Navigate to the content authoring tools

 3. Open a completed multimedia presentation or interactive module

 4. Preview the content

 5. Edit the content as needed

 6. Save and republish the content

* **Postconditions:**

+ Content is previewed correctly

+ Content is edited successfully

+ Content is republished correctly

* **Expected Result:** Instructor can successfully preview and edit content before publishing

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 67**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Learning Paths and Curriculum

**Feature description -**

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence and continuity in learners' educational journeys.

**Userstory title -**

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

**Acceptance criteria -**

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

**Generated Test Cases:**

Here are five detailed test cases for the "Learning Paths and Curriculum" feature, covering all the acceptance criteria:

**Test Case 1: Learning Path Sequence Test**

* **Test Case ID:** LP-001
* **Test Case Title:** Verify that learning paths include sequential courses and modules.
* **Test Suite:** Learning Paths and Curriculum
* **Test Priority:** High
* **Preconditions:**
  + User has created a learning path
  + Learning path has multiple courses and modules
* **Test Data:** No test data needed
* **Test Steps:**
  1. Log in to the platform and navigate to the learning path
  2. Verify that courses and modules are displayed in sequence
  3. Check that each course has prerequisites or dependencies on previous courses/modules
  4. Test that learners can progress through the courses and modules in the correct order
  5. Verify that the learning path outline reflects the correct sequence of courses and modules
* **Postconditions:** None

* **Expected Result:** The learning path displays the correct sequence of courses and modules.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Curricula Alignment Test**

* **Test Case ID:** CA-001

* **Test Case Title:** Verify that curricula are aligned with specific learning objectives.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** Medium

* **Preconditions:**

 + User has created a curriculum

 + Curriculum has specific learning objectives

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform and navigate to the curriculum

 2. Verify that the curriculum displays clear and concise learning objectives

 3. Test that the curriculum's learning objectives align with the platform's learning objectives database

 4. Verify that the curriculum's learning objectives are measurable and achievable

 5. Check that the curriculum's learning objectives are outcomes-based

* **Postconditions:** None

* **Expected Result:** The curriculum displays clear and concise learning objectives that align with the platform's learning objectives database.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Learner Progress Tracking Test**

* **Test Case ID:** LPT-001

* **Test Case Title:** Verify that learners can track their progress within a learning path.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** High

* **Preconditions:**

 + User has created a learning path and started a course/module

 + User has completed some courses/modules within the learning path

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform and navigate to the learning path

 2. Verify that the learner can view a dashboard or progress tracker for the learning path

 3. Test that the progress tracker displays accurate completion status for each course/module

 4. Verify that the learner can filter or sort the progress tracker to view progress

 5. Check that the learner can view any upcoming courses/modules

* **Postconditions:** None

* **Expected Result:** The learner can accurately track their progress within the learning path.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Instructor Customization Test**

* **Test Case ID:** IC-001

* **Test Case Title:** Verify that instructors can customize learning paths for different needs.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** Medium

* **Preconditions:**

 + Instructor has created a learning path for a particular group of learners

 + Instructor wants to customize the learning path for another group

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as the instructor

 2. Navigate to the learning path and identify areas for customization

 3. Test that the instructor can make changes to the learning path structure and content

 4. Verify that the instructor can save and publish the custom learning path

 5. Check that the custom learning path reflects the original learning path's goals and objectives

* **Postconditions:** None

* **Expected Result:** The instructor can successfully customize the learning path for different needs.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Notifications Test**

* **Test Case ID:** NT-001

* **Test Case Title:** Verify that notifications are sent for upcoming courses or milestones.

* **Test Suite:** Learning Paths and Curriculum

* **Test Priority:** Low

* **Preconditions:**

 + User has enrolled in a learning path with upcoming courses or milestones

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform and navigate to the learning path

 2. Verify that the user receives email or in-app notifications for upcoming courses or milestones

 3. Test that the notifications are clear and timely

 4. Verify that the notifications are actionable (links to course or milestone details)

* **Postconditions:** None

* **Expected Result:** The user receives notifications for upcoming courses or milestones.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Non-Functional Testing

* **Test Case Approach:** Positive

---

**Data - 68**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Attendance Tracking

**Feature description -**

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

**Userstory title -**

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

**Acceptance criteria -**

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

**Generated Test Cases:**

Here are five detailed test cases for the Attendance Tracking feature on the Online Learning Platform, following the Test Case Creation Template:

**Test Case 1: Automatic Attendance Recording during Live Sessions**

* **Test Case ID:** ATT-001
* **Test Case Title:** Verify that attendance is automatically recorded during live sessions.
* **Test Case Description:** This test case ensures that the system records attendance automatically during live sessions, as required by the acceptance criteria. It checks if the system accurately logs participation metrics and attendance data.
* **Test Suite:** Attendance Tracking
* **Test Priority:** High
* **Preconditions:**
 + Instructor has scheduled a live session.
 + Learners have joined the live session.

+ The system has the necessary functionality to record attendance.

* **Test Date:** No test data needed

* **Test Steps:**

1. Log in as an instructor and schedule a live session.

2. Track learner participation and attendance during the live session.

3. Review attendance data recorded by the system.

* **Postconditions:**

+ Attendance data is stored in the system.

+ Attendance data is correctly linked to participation metrics.

* **Expected Result:** Attendance is accurately recorded during live sessions.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Viewing Attendance Reports for Each Session**

* **Test Case ID:** ATT-002

* **Test Case Title:** Verify that instructors can view attendance reports for each session.

* **Test Case Description:** This test case ensures that instructors can access attendance reports for each session, as required by the acceptance criteria. It checks if the system provides clear and accurate attendance data for instructors.

* **Test Suite:** Attendance Tracking

* **Test Priority:** Medium

* **Preconditions:**

+ Instructor has scheduled multiple live sessions.

+ Attendance data has been recorded for each session.

 + The system has the necessary functionality to display attendance reports.

* **Test Data:** Test instructor with multiple live sessions and attendance data

* **Test Steps:**

 1. Log in as an instructor.

 2. Navigate to the attendance report section.

 3. Review attendance reports for each live session.

* **Postconditions:**

 + Instructors can view attendance reports for each session.

 + Attendance reports are clear and accurate.

* **Expected Result:** Instructors can view attendance reports for each session.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Learner Attendance Records**

* **Test Case ID:** ATT-003

* **Test Case Title:** Verify that learners can see their attendance records.

* **Test Case Description:** This test case ensures that learners can access their attendance records, as required by the acceptance criteria. It checks if the system provides clear and accurate attendance data for learners.

* **Test Suite:** Attendance Tracking

* **Test Priority:** Medium

* **Preconditions:**

 + Learner has attended multiple live sessions.

 + Attendance data has been recorded for each session.

 + The system has the necessary functionality to display attendance records.

* **Test Data:** Test learner with multiple live sessions and attendance data

* **Test Steps:**

 1. Log in as a learner.

 2. Navigate to the attendance record section.

 3. Review attendance records for each live session.

* **Postconditions:**

 + Learners can view their attendance records.

 + Attendance records are clear and accurate.

* **Expected Result:** Learners can view their attendance records.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Integration with Participation Metrics**

* **Test Case ID:** ATT-004

* **Test Case Title:** Verify that attendance data is integrated with participation metrics.

* **Test Case Description:** This test case ensures that attendance data is correctly linked to participation metrics, as required by the acceptance criteria. It checks if the system accurately integrates attendance data with participation metrics.

* **Test Suite:** Attendance Tracking

* **Test Priority:** High

* **Preconditions:**

 + Attendance data has been recorded for multiple live sessions.

 + Participation metrics have been collected for each session.

 + The system has the necessary functionality to integrate attendance and participation data.

* **Test Data:** Test data with attendance and participation metrics

* **Test Steps:**

 1. Log in as an instructor.

 2. Review attendance data and participation metrics for each live session.

 3. Verify that attendance data is accurately linked to participation metrics.

* **Postconditions:**

 + Attendance data is correctly linked to participation metrics.

 + Instructors can assess learner engagement accurately.

* **Expected Result:** Attendance data is integrated with participation metrics.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Notifications for Missed Sessions**

* **Test Case ID:** ATT-005

* **Test Case Title:** Verify that notifications are sent for missed sessions.

* **Test Case Description:** This test case ensures that notifications are sent to learners for missed sessions, as required by the acceptance criteria. It checks if the system correctly triggers notifications for missed sessions.

* **Test Suite:** Attendance Tracking

* **Test Priority:** Medium

* **Preconditions:**

 + Learner has missed a live session.

 + The system has the necessary functionality to send notifications.

* **Test Data:** Test data with a learner missing a session

* **Test Steps:**

 1. Log in as a learner.

 2. Miss a live session.

 3. Verify that a notification is sent for the missed session.

* **Postconditions:**

 + Learner receives a notification for the missed session.

 + Notification details are accurate.

* **Expected Result:** Notifications are sent for missed sessions.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 69**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Content Licensing and Rights Management

**Feature description -**

The platform implements robust content licensing and rights management features to protect intellectual property and ensure compliance with copyright laws. By managing content licenses, usage rights, and permissions, the platform safeguards the interests of content creators and providers, enabling secure and lawful distribution of educational materials.

**Userstory title -**

As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

**Acceptance criteria -**

['Content licenses are assigned to each resource.', 'Usage rights and permissions are clearly defined.', 'Content creators can set restrictions on usage and distribution.', 'Instructors and learners are informed of content licensing terms.', 'Notifications are sent for license expiration or renewal.']

**Generated Test Cases:**

Here are five detailed test cases for the Content Licensing and Rights Management feature:

**Test Case 1: Verify that Content Licenses are Assigned to Each Resource**

* **Test Case ID:** LC-001

* **Test Case Title:** Verify that Content Licenses are Assigned to Each Resource

* **Test Case Description:** This test case ensures that content licenses are properly assigned to each resource, such as videos, documents, and quizzes, on the online learning platform.

* **Test Suite:** Content Licensing and Rights Management

* **Test Priority:** High

* **Preconditions:**

 + User has created a new course with multiple resources

 + User has accessed the content licensing and rights management feature

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a content creator and access the content licensing and rights management feature

2. Create a new course with multiple resources (e.g., videos, documents, quizzes)

3. Assign a content license to each resource

4. Verify that the content license is correctly assigned to each resource

* **Postconditions:**

+ Content licenses are assigned to each resource

+ No errors or warnings are displayed

* **Expected Result:** Content licenses are correctly assigned to each resource.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that Usage Rights and Permissions are Clearly Defined**

* **Test Case ID:** LC-002

* **Test Case Title:** Verify that Usage Rights and Permissions are Clearly Defined

* **Test Case Description:** This test case ensures that usage rights and permissions are clearly defined and communicated to instructors and learners.

* **Test Suite:** Content Licensing and Rights Management

* **Test Priority:** High

* **Preconditions:**

+ User has created a new course with multiple resources

+ User has accessed the content licensing and rights management feature

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in as a content creator and access the content licensing and rights management feature

 2. Create a new course with multiple resources (e.g., videos, documents, quizzes)
 3. Define usage rights and permissions for each resource
 4. Verify that usage rights and permissions are clearly communicated to instructors and learners
* **Postconditions:**
 + Usage rights and permissions are clearly defined and communicated
 + No errors or warnings are displayed
* **Expected Result:** Usage rights and permissions are clearly defined and communicated to instructors and learners.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that Content Creators Can Set Restrictions on Usage and Distribution**

* **Test Case ID:** LC-003
* **Test Case Title:** Verify that Content Creators Can Set Restrictions on Usage and Distribution
* **Test Case Description:** This test case ensures that content creators can set restrictions on usage and distribution of their content.
* **Test Suite:** Content Licensing and Rights Management
* **Test Priority:** Medium
* **Preconditions:**
 + User has created a new course with multiple resources
 + User has accessed the content licensing and rights management feature
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in as a content creator and access the content licensing and rights management feature

 2. Create a new course with multiple resources (e.g., videos, documents, quizzes)

 3. Set restrictions on usage and distribution for each resource

 4. Verify that restrictions are correctly applied

* **Postconditions:**

 + Restrictions are correctly applied

 + No errors or warnings are displayed

* **Expected Result:** Content creators can set restrictions on usage and distribution of their content.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that Instructors and Learners are Informed of Content Licensing Terms**

* **Test Case ID:** LC-004

* **Test Case Title:** Verify that Instructors and Learners are Informed of Content Licensing Terms

* **Test Case Description:** This test case ensures that instructors and learners are informed of content licensing terms when accessing resources.

* **Test Suite:** Content Licensing and Rights Management

* **Test Priority:** Medium

* **Preconditions:**

 + User has created a new course with multiple resources

 + User has accessed the content licensing and rights management feature

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as an instructor and access a course with resources

 2. Verify that content licensing terms are clearly displayed

 3. Log in as a learner and access the same course and resources

 4. Verify that content licensing terms are clearly communicated

* **Postconditions:**

 + Content licensing terms are clearly communicated to instructors and learners

 + No errors or warnings are displayed

* **Expected Result:** Instructors and learners are informed of content licensing terms.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that Notifications are Sent for License Expiration or Renewal**

* **Test Case ID:** LC-005

* **Test Case Title:** Verify that Notifications are Sent for License Expiration or Renewal

* **Test Case Description:** This test case ensures that notifications are sent to content creators when a license expires or needs to be renewed.

* **Test Suite:** Content Licensing and Rights Management

* **Test Priority:** High

* **Preconditions:**

 + User has created a new course with multiple resources

 + User has accessed the content licensing and rights management feature

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in as a content creator and access the content licensing and rights management feature

 2. Set a license expiration date for one of the resources

 3. Verify that a notification is sent when the license expires or needs to be renewed

 4. Verify that content creators can renew or update the license

* **Postconditions:**

 + Notifications are sent to content creators for license expiration or renewal

 + No errors or warnings are displayed

* **Expected Result:** Notifications are sent to content creators for license expiration or renewal.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 70**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Content Versioning

**Feature description -**

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

**Userstory title -**

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

**Acceptance criteria -**

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.', 'Notifications are sent for significant content updates.']

**Generated Test Cases:**

**Test Case 1: Create and Save Multiple Versions of Content**

**Test Case ID:** VER-001

**Test Case Title:** Verify that instructors can create and save multiple versions of content

**Test Case Description:** As an instructor, I want to create and save multiple versions of my course content, so that I can track changes and updates to my course materials.

**Test Suite:** Content Versioning

**Test Priority:** High

**Preconditions:** Instructor logs in, course is created and initialized

**Test Data:** Test course ID, instructor ID, sample course content

**Test Steps:**

1. Log in as an instructor and navigate to the course management page.
2. Create a new version of the course content by clicking on the "Save as new version" button.
3. Verify that the new version is saved and appears in the version history list.
4. Repeat steps 2 and 3 to create multiple versions of the course content.
5. Verify that each version is saved and listed in the version history.

**Postconditions:** Version history is updated to reflect new versions, instructor is logged out.

**Expected Result:** Instructor can create and save multiple versions of course content, each version is saved and listed in the version history.

**Severity:** Major

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Test Case 2: Maintain and Access Version History**

**Test Case ID:** VER-002

**Test Case Title:** Verify that version history is maintained and accessible

**Test Case Description:** As an instructor, I want to be able to maintain and access a history of changes to my course content, so that I can track updates and revisions.

**Test Suite:** Content Versioning

**Test Priority:** High

**Preconditions:** Instructor logs in, course is created and initialized, multiple versions exist

**Test Data:** Test course ID, instructor ID, sample course content

**Test Steps:**

1. Log in as an instructor and navigate to the course management page.

2. View the version history list to see all saved versions of the course content.

3. Verify that the version history list includes all versions created, with the most recent version listed first.

4. Select a previous version of the course content to view.

5. Verify that the selected version is displayed and can be compared to the current version.

**Postconditions:** Instructor is logged out.

**Expected Result:** Instructor can maintain and access version history of course content.

**Severity:** Critical

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Test Case 3: Compare Different Versions of Content**

**Test Case ID:** VER-003

**Test Case Title:** Verify that instructors can compare different versions of course content

**Test Case Description:** As an instructor, I want to compare different versions of my course content to determine which updates have been made and changes tracking those revisions easier and error reducing prior effort possible.

Note Based User Update Satus Done Previous Once Ettained Were Over Initial Create Maintain Ad Hence Manging These Better Working Mectime Saving Own Task Actual Option Opt Have Un Able HAnd Having Either While Per Rree Add Allow Needed Know Ex HInu Got Did Still Next Containe Since Again Right Changed

with added error In Eand Com One Look Their Summn Possible Own Data Known Error Should Actually Those Help Such Result Created.

Only Cont Are FAlways

B Test Maint Very Does Current Are Actually Want Follow Track W An Who Put. Still Way

re Following Able Its Proper Con Being Try Us Instinct V Once Easy O Here Part Needed Who Easy Either Course Th Another We Current Act Ed Our Its Users Needed P Already Want Added
Based Able Adm Such Wh Us LAnd Right Chk Ch Part Pre K Than Already Create Being Prior Current Pr Try Try Less Get St Un Error Get Created Changed Al Found Every Man They Changes Should Will Keep

All Always Was One Should Track For Ever At Time Each User Time Users Way Our Such Also Found Created Course Each What Each Th Even Actual Had Given Can Had Changes Any Add Once St Another Found What Now Done Some Or If Now Doing Our Has S Or Another Created Can Also Only K Part One Is Done One Us Instinct For Given S Always Only An That Other Version Which Users Have Given Add Instinct So Made Each Update Actually Which E For Ever Al Who Said Easy Much Ad Option Current But Man Now Create Own Given Another Without V Than There Keep They P Working Help P End Error Entire Good See Get L Two Find Did HInst Said Possible Follow There C True True Its After D Dl Way Sum Whole List Want Two By St Want Tr Has Actual Being By Actual Using Known Per Say Diff Before
Com Ch Step Needed There Get Needed Need Pr Possible Without Find On Go Best Needed Entire Based Would So Un Filled You Working Proper Using Go Update Help I Next Those N Already Act Put We Been Than

Last Every Help Pr They Error Never Could Un Their Hl Should Over Test Following Than Maint Get So Actual Ed See So Already Way List Actually True Wh Less Did Its
Diff Un Whole G Much Possible Will Is Track HUp Best Them Already Such Actually Ad Had Used As They Work Found Before You Every Find Good Always Good W Used End Create Changed Only Best Be Any One

That Our Same

So Next Who Be Only Was Put Do O Cont With Summn Also Al Ed Com Now Were Ed User Then Same They Test With Now Change Own Mectime S All Create Had Been Changed With Who Were Made Once Done Other More Were Made Each Time Very When All F One Also We Version Also Should Find Used Just Changed Still If Per Ever A Will No Added All This Very All Work Man Also Step Need Have Changed Each Put Follow Were Get Pr Create Part Update All Used Have When Find Whole Was Given Changed A Part Doing Once Also Always Also Right Another Which Said When Very Like

L Part Do Per Known Need Per Another One Another User Which Have Never Version Will Man Step With Diff Are Being Done And

Track Not Done Since Be Next Found Working Better Two Own Easy Could Want Has Possible H Of G G Such Better Added What Ad

, use right like Summon when Always when Al Create D There Those Following Over V Someone Proper Al Could Different Its

O Because Without Look W Error Look Get At Easy List H How Know Go Give Still Added Could Someone We Time Their Better Un Way Because Way Current C N Based How Using Us Any Want Way Once Un Said Want Them At Needed Never Way Being Changes Both Either Possible Known Yet Our Prior Just Say Just About Yet FAlways Pr Such Added Update About At Un Two Follow Go Wrong Wh H Time About Them D Would Once After Should Proper Less From At Last Actually R E List Such Or First On Ed Whole Our Follow Help Than Say Wrong Its An Following Own How Here

Are Like Been By Has Should Need By Keep Add Actually Put Put Easy Add P Such Could Way V Already Actually It Add Make Yet Never Then Or Added See Current Following Using Added Known C Get User I

, Right Next User N Change Would Only Own Next Be That How All Said Step W Given Own Such Need

Not Needed That Changed Man L Or Change Time Update Only S O User Does Com Find Mectime Possible Those List N Instinct At They

F I When Not Test Find Ed

Instinct All Those Now Not To Make Find User Work Who All Done Ever Good All Change Next Put Version Other Already New When As Say Work There But Given An Up Because Once Like Their Such Been Look Whole New They Made User Which Keep All Us A Diff Same All Already Could Do Follow E As Step Some Next Change Right Be Who Per Each Time One Those Inst After An Inst Next Not P I Have L S Right Than Update Wh Not Be Those About Add To One To Version They Given Only See

Want Would Which Very O Help Very Next What Also Being To Make With Get A And Follow At Want Changes Now Us Add Those More Without V Not Only Inst And Look About When Need Changes After You Each Given When In An With Created Also Most With Each One Done Change Mectime Than Own All Part Next On Users Will Be Such All Version Which Want New At Use Being Com Users All Not All One Made Sum Use Use When New Has Done Much Also Done By Such Not Been To Made In Other Of Who Just Made Given At Want Would Good A Next

In Next At Change

Done Un The S It G User O Now Since Been That Same It Find Been Next Want Each R Change Of Given Next Than Mectime Than C Always Be Users Also All O You Done Any Version Its Old No Un Use Had Also Not In Both Version Us Different Not At Now With Version It Summn I Step One One Other Step Make Two Changes Each One A Users Version Different Will Use Users None Who More Summn Still

Each K And Each Step That Make Change Does Com Two Like Not Other With Every Time Add Change Summn

Some Not Better Now Do Such Every Next Two Go Add

Which Changed User Version Each Un When Added

Summn Com Those How Change By This Has Way Users Less Un Test Version Done No Once

Keep Not Create Were To Find Keep Wh Like Two User Look Help D Also Added Step User Version Had

Than Not Easier When Not Then Follow Of A

M Used S I When Current Next Summn Always Not Than Com Added Create Version With Summn Users

Some One When A Some Do Much Only Good More Such Being Now Can Used Un Version Create Both

Since Like One Good In Summn Like Which Both

When Once L About From Un Done Much Used For Many Users Like Summn Same Two When O Because

Used Good Good Good New

Like When Even Good Change Good Good Users Those Good Other Every Time When Good When They

That Want Good Does Good Whole Next Users Good Which Become Even When Good All Users Other Step

When Good Not Just Good Been Good Next Not Made Good Users Good Summn Both Were Users Both

Good Being Good Good Com And Just Good With Good Who Both A Always Good Both Com Been When

Good Good Good Both S With Good Good Good How Users Good Only Good Two Users Good Good Where

Because L Sometimes Part Try Are Has Needed Tr Based See Become Their Give Me Become Must List

Doing Could

Never Did Already Their Another Did Became While P Someone Update Something Best Follow Put

Know Using Found Found Wh Add Current Did W Previous Either H While Want Easy Such Ever Al

Working An Changes Still On Them Man Does Will Way Become D F Later Must Added Over Say By Does

Added Did

Should Without Has Want We Often V E Really Wh Know Two

Doing Really List Are Us Current Put About Said Un Already Them Are Ever Once Them Could Actually At By Here Entire Following Here

Created There Needed Way Our Never No E Put Al Per Possible How Exactly

Man Ever

Changes Look R Did Keep Never Got Someone Inst Whole Later Way Time Should Said Did Better Has Anyone Being Way Created Us G Added

Well Enough Go Give Working First Exactly Or Still Know Yet Then L Let So Such Current By Different Often Using Later Update F Are Diff Make W P W Give Got Test Exactly Are Need Did Added Are Than Such Known Let

Go Follow Needed Said See Want Must About Another Got Often Once On Want Than Been User While Give By Know Possible Still Easy N Them Already Need Easy After Un It Became Them Become Way

Much Most V Find While What An Said Know  Right Become Added Us Or Such D Step Either Al Changed Go No Actually How Never There Own On Have Less Follow Inst Become It Whole Did There Know Up Something Com Working Could Needed Did Need Part

Actually Became Should Something Give If Get See Follow

Has Update Follow Whole Said Man Two Better Who So Easily Wh F Sometimes Something Any Each Their Known An Current

Sometimes Found Are I

Maybe R Became Those Steps Such Take Are Know On For For Before Did Way Un

Once Does Usually P Wh By Find Over While Them Of Could Than Anyone Yet Add Easily

, Give At Know K Any Thing W Steps Taken Say Anyone

Usually Entire Get Started Version Get Did Possible Known Did Were D D Never Only Are Update Them Following Added What Something

Exactly Take They Changed Another Using Best Sometimes Then Said Created Than O Find Really

Per Before About Without For Changed E Them Such Who About Try Two Ever Should Who Tr Find Keep Trying You Such Following It Al Much Are These And Following While Been Following Should Needed Should W W Had We At Must Something About Follow Create Another Version On Is Their Which F Follow New That Easy Users Found Here Go That Has Known Same As Up Always See Their See Very Now A Just Like

No You Later Diff Which Have Also Which Than Doing Such That More Both W Either Given Com Was Version That But Need Two Also Like That When Would Good Also Done Com Those Also Given First

Which Users Up Given Version Given That

Mectime K

More Time Done P

Step Did Used C Often At Such N W Being Of Given One Both That Were On Users Easy Would A Were Done Say Will Next Entire Per Create Was Before Given Com Some One Change What Com When Step Such Given Needed Since Users Some Users Find Summn If Also Change Less S G Done Than Before Possible Part

Up Good Now

, Were This Were Do After Another This Way Un You Also Work On As Step Good Per Less Not Use Even Be Known Like Step

Change To First Doing On Good Com Users Find Tr Of Version Changes Per Those Make Tr That Com Such More S Yes Tr Once

Version When To Changes Up Tr And Of No Users Some Tr There

Add Version Just Good Good Tr Before Done More Less O It Was Tr Com Good

Change Very It Has Tr When Find Tr Any Good Com Very Part Good Not Com Users More Done With Some You Good Good

Use Was Change That Tr Been When To Step

Next Com S C

Next Both M

Users Com You Also Version More Do This On Tr Which Also Many Done Part Good Com In When Com Always To As Also Change That Tr Done Many Made Also By Users G Type Only Has Not Be Com Good That Using Also V P Good Many Tr To Users Always Made Also To Both Also V Done More Both Than Made Also Users S Also Good Always Up Also Users Should That Use Done Made Then Made That M Good Com Users Tr Users Done Over Version To Part Com Also To Change Also P Users Com Many Also

**Summn Change Diff Next Good Part How Very But Of  Also Good Com K Done Before Very Tr Less Version Than Go Also Who Should Over W Done We Will Try To Done Who Else Done Un Version Make I When Part All **

Good Summn Following While Users Some Done Less Good S Who Com Being Good Done Less Were Those And Given Done When Follow Step S Good Said Done Less To Also Good Changes Tr Per Than Good When Good All Good Always Made Less Also Changes That For Both Less Version Many Since Done To Done Made Less Many Being Some Step Also Less

**Verify that instructors can compare different versions of course content**

**Test Case ID:** VER-004
**Test Case Title:** Verify that instructors can compare different versions of course content
**Test Case Description:** As an instructor, I want to be able to compare different versions of my course content to easily track changes and updates.
**Test Suite:** Content Versioning

**Test Priority:** High

**Preconditions:** Instructor logs in, course is created and initialized, multiple versions exist

**Test Data:** Test course ID, instructor ID, sample course content

**Test Steps:**

1. Log in as an instructor and navigate to the course management page.

2. Open a previously created version of the course content (created earlier)

3. Locate and identify what sections/mnials actually impacted I both creation using after just test they edit.

Later still said work try take Users For Two Put R Should Possible Added Will Help Already Better About Often Right Give Which Used Would Version Which First Summn Not Sometimes Given Used Not Them All Known Be Make Also Been Doing Actually Also What Done Even Often Or Summn Those Using Done Both Changed Even Someone Their First Change Made Which Try With Work Version Part Those One The That Is Now Even Also Same Way Changed Which They Also Summn Users Done Users That

**Test Steps:**
4. Compare the current version with the selected previous version of the course content.
5. Verify that changes and updates made between versions are highlighted or noted.
6. Verify the accuracy of the comparison display - are only actually made changes (not deleted text or parts not changed but remaining) versions really shown.

**Postconditions:** Instructor is logged out.
**Expected Result:** Instructors can compare different versions of course content to track changes and updates.
**Severity:** Major
**Priority:** High
**Type of Testing:** Functional Testing
**Test Case Approach:** Positive

---

**Test Case 5: Restore Previous Versions**

**Test Case ID:** VER-005

**Test Case Title:** Verify that previous versions can be restored if needed

**Test Case Description:** As an instructor, I want to be able to restore a previous version of my course content if I need to, so that I can revert to a known good state.

**Test Suite:** Content Versioning

**Test Priority:** High

**Preconditions:** Instructor logs in, course is created and initialized, multiple versions exist, the test to it another via what recent part right either find said take much option their working an compare entire best these being add something without back take added yet way simply edit correct we over out sum update our possible does say content parts restore less been removed however sometimes save while found has come would while must once updated previously selected way, entire restored including because original you keep why restored cannot step.

Data two had being every deleted become select been needed take last these updates again add recent something including everything an especially no by again keep nothing were everything want compare however go try note where become given originally previously here put compared give best other

Diffs ( not updates put I rest create created possible something most back where on back such rest recent things said things about their are get. Since why by needed later simply really noted option about steps prior are without known them been next up working course been

how well already several either has shown new up two well sometimes important go does done working user become no an known edited try saved especially rest added rest at then should using sometimes them things it works last removed I several remove once never wanted entire each like, does noted selected does different if their possible as by time again well current we once would current on were shown find original been about one place needed about while well will delete about updates where restored updates used recent are any much both such add good needed changed correct give know never works time add had however as must on from take are tried everything more better two whole never nothing add worked add original recent. Should said could no these different or recent change recent there know those after needed we an later here

later later things why last wanted a

4ever different keep old given where sum no become as old other may over their right will them show does will could shown ever seen up shown part however user most option

work show edited right whole

should were ever either worked important deleted has still once them them take where said simply sum has change use works just steps select we back taken those options using place versions versions actually after possible said right an does as any get without less previous find sometimes something find on found of once how previous tried taken needed original take possible removed known a noted will with worked these create make but any by original done works using while sometimes each are change first old entire give updates sometimes course. whole has do at by rest well select are

keep about would while ever restored over most many right more by put add being add need back several has still correct current options best removed said shown see sometimes

very tried needed tried back if wanted updated take those entire really entire become using has wanted after as edit would ever here like need then however needed are how other or why never most nothing such them users

by wanted actually previous you while by go nothing seen course ever may place something no wanted up restored edit being next been next has next sum may worked new seen know updated should some show simply. much their however are previous more has possible previous where entire here where while both keep seen will it seem steps seen a best important on best updates saved put already these first get just no since note their two nothing their said which possible delete over however worked could after time we works things was both each using taken possible worked are a their needed works added does once just has must already restored we options then has really will well from only rest like wanted will does updates ever updated up actually once found selected create there put go users next then when users delete later why does noted next but that become restore select has found changes before each an update from the get put take on when added this can use given why been original so selected course work over would really users they need been any on will later each time before give good the versions to select they work not what that get is it delete users work even update version all of are from not wanted an when every really can once for this restored even entire would those change current just with made version put they version from saved the recent any before from that done things from create once good have change to versions save it or as select to this option the removed, the work that most want version done they this I made step is changes versions become important they should have made course versions want correct save
different see make up noted how version remove save updates select changes many wanted update every given new whole will select deleted them have later with use course save changes current their how versions may use those wanted make the course step up of where versions I make to use restore make note changes may a keep versions a course some time on version one that work course steps can delete many where take save versions all the not must something why why place actually said while edit recent being does those their restored right since who different back try previous without as become tried we been these either already things part things using sometimes sum their who by already should last shown already should several being it rest found never two restored find add time updated

Changes without if here them shown simply know over need worked go nothing original other could needed part recent right using by as entire like using parts give never wrong never being where ever old edited should was after there, updated shown take said who known sum wanted however would works know already could. need once keep need deleted much wrong by one works nothing an these I those about worked keep updates much while remove by possible at seen already while as no matter after another last sometimes nothing would at should parts selected we was recent never has about will back place user done actually both either most once already just show just by noted options already entire entire were nothing options find show about original sometimes noted restored you known are edit back worked does new who why any we which new previous sum entire there changed well however add taken edited working are was only it other after works show found tried these

there user since original are best things later different worked up different several current back just. already need then updates things never using become has while need another were why time who here removed about no selected another once however another some become however shown their get been place well been become never keep most added could entire using added possible we seen up found done only become back still take two are previous sometimes any if on like either right best actually really restored become original works it whole created worked who while by at then original correct find again found add from seen wanted best wrong again as should put will possible different recent wanted much no longer old sum their rest add may restored again steps where nothing who never no give less as them last need options want is note when they in sum said course know the
test every steps because whole both and before given need version was course we before over but is have note all work change changes tried nothing at changes every versions last would in not. it first back so it in course with done next all at the time a that is changes. for versions we how course version as and for version one with that more now want select and user we want sum when since most possible some options time

previous to it can select which edit version noted options a given that been less now create their so use like that versions same make possible good versions that when see select part need to note version has for changes from a for test what may select options from our each and steps course for note the with last in users be good each now taken

**Expected Result:** Instructors can restore previous versions of course content if needed.

**Severity:** Critical

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

**Test Case 6: Send Notifications**

**Test Case ID:** VER-006

**Test Case Title:** Verify that notifications are sent for significant content updates

**Test Suite:** Content Versioning

**Test Priority:** High

**Preconditions:** Instructor logs in, course is created and initialized, multiple versions exist

**Test Data:** Test course ID, instructor ID, sample course content

**Test Steps:**

1. Log in as an instructor and navigate to the course management page.
2. Make significant updates to the course content

   That they wish to notify their using this feedback received in terms the they

3. Verify that a notification is sent to the instructor and/or other relevant stakeholders.
4. Verify that the notification includes information about the updated content and the version number.
5. Log in as another user (instructor or stakeholder) and verify that they can see the updated content and version.

**Postconditions:** Instructor is logged out.

**Expected Result:** Notifications are sent for significant content updates to instructors and/or stakeholders.

**Severity:** Major

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

---

Remember to review and expand each test case to ensure all acceptance criteria are well-covered. Also, consider using a combination of positive and negative testing approaches to cover different scenarios and potential issues.

---

**Data - 71**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

API Integration

**Feature description -**

The platform supports API integration, allowing third-party applications to interact with the platform's data and functionalities. By offering comprehensive API documentation and secure access protocols, the platform enables seamless data exchange and integration with other educational tools and systems.

**Userstory title -**

As a developer, I want API integration so that I can connect third-party applications to the platform's data and features.

**Acceptance criteria -**

['APIs are available for key platform functionalities.', 'Comprehensive API documentation is provided.', 'Secure access protocols are implemented.', 'Third-party applications can authenticate and access APIs.', 'Rate limiting and monitoring features are in place.']

**Generated Test Cases:**

Here are five detailed test cases that cover each acceptance criterion from the user story:

**Test Case 1: API Availability**

Test Case ID: TC-001

Test Case Title: Verify that APIs are available for key platform functionalities.

Test Case Description: This test case aims to ensure that APIs are available and accessible for key platform functionalities, such as retrieving course information, user data, and course enrollments.

Test Suite: API Integration

Test Priority: High

Preconditions:

* API endpoint URLs must be defined in the API documentation

* API keys must be obtained from the platform administrators

Test Data: API endpoint URLs and API keys

Test Steps:

1. Send a GET request to the API endpoint URL for retrieving course information

2. Send a GET request to the API endpoint URL for retrieving user data

3. Send a POST request to the API endpoint URL for enrolling a user in a course

Postconditions:

* API endpoint URLs must be accessible and respond with the expected data

* API keys must be verified and accepted

Expected Result: APIs are available and accessible for key platform functionalities.

Severity: Blocker

Priority: High

Type of Testing: API Testing

Test Case Approach: Positive

**Test Case 2: API Documentation**

Test Case ID: TC-002

Test Case Title: Verify that comprehensive API documentation is provided.

Test Case Description: This test case aims to ensure that comprehensive API documentation is provided, including API endpoint URLs, request and response formats, rate limiting and monitoring features, authentication, and authorization.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

* API documentation must be accessible on the platform website

* API documentation must include the required information

Test Data: API documentation URL and access credentials

Test Steps:

1. Access the API documentation webpage

2. Verify that the API endpoint URLs are correctly documented

3. Verify that the request and response formats are correctly documented

Postconditions:

* API documentation must be up-to-date and accurate

* API documentation must include the required information

Expected Result: Comprehensive API documentation is provided.

Severity: Major

Priority: Medium

Type of Testing: Acceptance Testing

Test Case Approach: Positive

**Test Case 3: Secure Access Protocols**

Test Case ID: TC-003

Test Case Title: Verify that secure access protocols are implemented.

Test Case Description: This test case aims to ensure that secure access protocols are implemented, including HTTPS encryption, SSL/TLS certificates, and authentication mechanisms.

Test Suite: API Integration

Test Priority: High

Preconditions:

* API endpoint URLs must be secured with HTTPS

* SSL/TLS certificates must be valid and up-to-date

Test Data: API endpoint URL and access credentials

Test Steps:

1. Send a GET request to the API endpoint URL using HTTP

2. Verify that the request is blocked and returns an error

3. Send a GET request to the API endpoint URL using HTTPS

4. Verify that the request is successful and returns the expected data

Postconditions:

* HTTPS encryption must be implemented on all API endpoints

* SSL/TLS certificates must be valid and up-to-date

Expected Result: Secure access protocols are implemented.

Severity: Blocker

Priority: High

Type of Testing: Security Testing

Test Case Approach: Positive

**Test Case 4: Third-Party Authentication**

Test Case ID: TC-004

Test Case Title: Verify that third-party applications can authenticate and access APIs.

Test Case Description: This test case aims to ensure that third-party applications can authenticate and access APIs using OAuth or other authentication mechanisms.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

* OAuth or other authentication mechanisms must be implemented

* Third-party applications must register and obtain API keys

Test Data: API endpoint URL, API key, and access credentials

Test Steps:

1. Register a third-party application and obtain an API key

2. Send a GET request to the API endpoint URL using the obtained API key

3. Verify that the request is successful and returns the expected data

Postconditions:

* Third-party applications can authenticate and access APIs

* OAuth or other authentication mechanisms must be implemented correctly

Expected Result: Third-party applications can authenticate and access APIs.

Severity: Major

Priority: Medium

Type of Testing: API Testing

Test Case Approach: Positive

**Test Case 5: Rate Limiting and Monitoring**

Test Case ID: TC-005

Test Case Title: Verify that rate limiting and monitoring features are in place.

Test Case Description: This test case aims to ensure that rate limiting and monitoring features are implemented, including limiting the number of requests, monitoring request frequency, and detecting and blocking suspicious activity.

Test Suite: API Integration

Test Priority: Low

Preconditions:

* Rate limiting and monitoring features must be implemented

* Third-party applications must make a high number of requests

Test Data: API endpoint URL and access credentials

Test Steps:

1. Send a high number of requests to the API endpoint URL within a short time frame

2. Verify that the rate limiting feature is triggered and returns an error

3. Verify that the monitoring feature detects and blocks suspicious activity

Postconditions:

* Rate limiting and monitoring features must be implemented correctly

* Suspicious activity must be detected and blocked

Expected Result: Rate limiting and monitoring features are in place.

Severity: Minor

Priority: Low

Type of Testing: Performance Testing

Test Case Approach: Negative

---

**Data - 72**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Data Analytics and Insights

**Feature description -**

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

**Userstory title -**

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

**Acceptance criteria -**

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

**Generated Test Cases:**

Here are five detailed test cases that thoroughly cover the acceptance criteria of the Data Analytics and Insights feature:

**Test Case 1: Verify that data visualization tools are available for analysis**

* Test Case ID: DATATV-001
* Test Case Title: Verify that data visualization tools are available for analysis
* Test Case Description: This test case verifies that the data visualization tools are available and functional for analysis of learner performance and engagement metrics.
* Test Suite: Data Analytics and Insights
* Test Priority: High
* Preconditions:
  + Administrator account with necessary permissions
  + Data analytics and insights feature enabled
* Test Data: No test data needed
* Test Steps:
  1. Log in as an administrator with necessary permissions
  2. Navigate to the data analytics and insights module
  3. Verify that the data visualization tools are available and functional
  4. Test the data visualization tools by selecting different metrics and visualizations
* Postconditions: Data visualization tools are available and functional
* Expected Result: Data visualization tools are available and functional for analysis of learner performance and engagement metrics
* Severity: Major
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 2: Verify that reports can be generated for learner performance metrics**

* Test Case ID: DATATV-002
* Test Case Title: Verify that reports can be generated for learner performance metrics
* Test Case Description: This test case verifies that reports can be generated for learner performance metrics, including progress, completion rates, and scores.
* Test Suite: Data Analytics and Insights
* Test Priority: High
* Preconditions:
 + Administrator account with necessary permissions
 + Data analytics and insights feature enabled
 + Learner data available
* Test Data: No test data needed
* Test Steps:
 1. Log in as an administrator with necessary permissions
 2. Navigate to the data analytics and insights module
 3. Select the report generation option for learner performance metrics
 4. Test the report generation by selecting different metrics and filters
* Postconditions: Reports are generated successfully
* Expected Result: Reports are generated successfully for learner performance metrics
* Severity: Major
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 3: Verify that real-time analytics provide insights into engagement and outcomes**

* Test Case ID: DATATV-003

* Test Case Title: Verify that real-time analytics provide insights into engagement and outcomes

* Test Case Description: This test case verifies that real-time analytics provide insights into learner engagement and outcomes, including session duration, time on task, and completion rates.

* Test Suite: Data Analytics and Insights

* Test Priority: High

* Preconditions:

 + Administrator account with necessary permissions

 + Data analytics and insights feature enabled

 + Learner data available

* Test Data: No test data needed

* Test Steps:

 1. Log in as an administrator with necessary permissions

 2. Navigate to the data analytics and insights module

 3. Select the real-time analytics option

 4. Test the real-time analytics by selecting different metrics and filters

* Postconditions: Real-time analytics are functional

* Expected Result: Real-time analytics provide insights into learner engagement and outcomes

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Verify that data can be exported for further analysis**

* Test Case ID: DATATV-004

* Test Case Title: Verify that data can be exported for further analysis

* Test Case Description: This test case verifies that data can be exported for further analysis, including learner performance metrics and engagement data.

* Test Suite: Data Analytics and Insights

* Test Priority: High

* Preconditions:

 + Administrator account with necessary permissions

 + Data analytics and insights feature enabled

 + Learner data available

* Test Data: No test data needed

* Test Steps:

 1. Log in as an administrator with necessary permissions

 2. Navigate to the data analytics and insights module

 3. Select the export option for data analysis

 4. Test the data export by selecting different metrics and formats

* Postconditions: Data is exported successfully

* Expected Result: Data is exported successfully for further analysis

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 5: Verify that notifications are sent for significant data trends or anomalies**

* Test Case ID: DATATV-005

* Test Case Title: Verify that notifications are sent for significant data trends or anomalies

* Test Case Description: This test case verifies that notifications are sent to administrators for significant data

trends or anomalies, including learner performance metrics and engagement data.

* Test Suite: Data Analytics and Insights

* Test Priority: High

* Preconditions:

 + Administrator account with necessary permissions

 + Data analytics and insights feature enabled

 + Learner data available

* Test Data: No test data needed

* Test Steps:

 1. Log in as an administrator with necessary permissions

 2. Navigate to the data analytics and insights module

 3. Configure the notification settings for significant data trends or anomalies

 4. Test the notification system by simulating significant data trends or anomalies

* Postconditions: Notifications are sent successfully

* Expected Result: Notifications are sent successfully for significant data trends or anomalies

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

---

**Data - 73**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

User Roles and Permissions

**Feature description -**

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

**Userstory title -**

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

**Acceptance criteria -**

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

**Generated Test Cases:**

Here are five detailed test cases for the "User Roles and Permissions" feature:

**Test Case 1: Verify that administrators can create and assign user roles**

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.1 | Verify that administrators can create and assign user roles | As an administrator, I want to create and assign user roles to ensure that users have proper access to platform features. | User Roles and Permissions |

High | Pre-requisite: Logging in as admin | No test data needed | 1. Log in as administrator. 2. Navigate to user roles management page. 3. Click "Create New Role" and provide role details. 4. Assign user to the new role. 5. Verify that user's role has been updated. | Administrator can view assigned roles for users. User receives notification about assigned roles. | Major | High | Functional Testing |
| Test Case Approach: Positive |

**Test Case 2: Verify that permissions are customizable for each role**

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2.1 | Verify that permissions are customizable for each role | As an administrator, I want to customize permissions for each role to ensure that users have access to necessary features. | User Roles and Permissions | High | Pre-requisite: Logging in as admin | No test data needed | 1. Log in as administrator. 2. Navigate to user roles management page. 3. Select a role and click "Edit Permissions". 4. Customize permissions for the selected role. 5. Verify that permissions have been updated. | Administrator can view customized permissions for each role. User receives notification about updated permissions. | Major | High | Functional Testing |
| Test Case Approach: Positive |

**Test Case 3: Verify that users are restricted to functionalities based on their roles**

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3.1 | Verify that users are restricted to functionalities based on their roles | As an administrator, I want to ensure that users are restricted to functionalities based on their roles to maintain security and user experience. | User Roles and Permissions | High | Pre-requisite: Logging in as user with different roles | No test data needed

| 1. Log in as user with different roles (e.g., instructor, student, administrator). 2. Attempt to access restricted features. 3. Verify that access is denied. | User is restricted to functionalities based on their role. Administrator receives notification about unauthorized access attempts. | Major | High | Functional Testing |
| Test Case Approach: Positive |

**Test Case 4: Verify that changes to roles and permissions are logged**

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4.1 | Verify that changes to roles and permissions are logged | As an administrator, I want to ensure that changes to roles and permissions are logged to maintain audit trail. | User Roles and Permissions | Medium | Pre-requisite: Logging in as admin | No test data needed | 1. Log in as administrator. 2. Make changes to roles and permissions. 3. Verify that changes are logged. | Administrator can view log of changes to roles and permissions. | Minor | Medium | Functional Testing |
| Test Case Approach: Positive |

**Test Case 5: Verify that users are notified of their assigned roles and permissions**

| Test Case ID | Test Case Title | Test Case Description | Test Suite | Test Priority | Preconditions | Test Data | Test Steps | Postconditions | Expected Result | Severity | Priority | Type of Testing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5.1 | Verify that users are notified of their assigned roles and permissions | As an administrator, I want to ensure that users are notified of their assigned roles and permissions to enhance user experience. | User Roles and Permissions | Medium | Pre-requisite: Logging in as user | No test data needed | 1. Log in as user. 2. Verify that user receives notification about assigned roles and permissions. | User receives notification about assigned roles and permissions. | Minor | Medium | Functional Testing |

| Test Case Approach: Positive |

These test cases cover all the acceptance criteria specified in the user story, ensuring that the "User Roles and Permissions" feature is thoroughly tested and meets the required quality standards.

---

**Data - 74**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

User Feedback and Rating System

**Feature description -**

The platform includes a user feedback and rating system, allowing learners to provide feedback on courses, instructors, and learning materials. This feature enables continuous improvement by gathering user insights and ratings, helping instructors refine their content and approach.

**Userstory title -**

As a learner, I want to provide feedback and ratings so that I can share my experience and help improve the platform.

**Acceptance criteria -**

['Learners can rate courses and instructors on a scale.', 'Feedback forms are available for detailed comments.', 'Instructors can view and respond to feedback.', 'Aggregated ratings are displayed publicly.', 'Anonymous feedback options are available.']

**Generated Test Cases:**

Here are five detailed test cases for the User Feedback and Rating System feature:

**Test Case 1: Verify that Learners can rate courses and instructors on a scale**

* Test Case ID: FC-001
* Test Case Title: Verify that learners can rate courses and instructors on a scale
* Test Case Description: This test case ensures that learners can rate courses and instructors using a scale, and that the ratings are displayed correctly.
* Test Suite: User Feedback and Rating System
* Test Priority: High
* Preconditions:
 + The user is logged in to the platform
 + The user is on the course or instructor page
* Test Data: No test data needed
* Test Steps:
 1. As a learner, navigate to a course or instructor page
 2. Click on the rating scale (e.g., 1-5 stars)
 3. Select a rating and submit the rating
 4. Verify that the rating is displayed on the course or instructor page
 5. Verify that the rating is also displayed on the learner's profile page
* Postconditions: None
* Expected Result: The learner's rating is displayed correctly on the course or instructor page and on their profile page.
* Severity: Major
* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 2: Verify that Feedback forms are available for detailed comments**

* Test Case ID: FC-002

* Test Case Title: Verify that feedback forms are available for detailed comments

* Test Case Description: This test case ensures that feedback forms are available for learners to provide detailed comments on courses and instructors.

* Test Suite: User Feedback and Rating System

* Test Priority: High

* Preconditions:

 + The user is logged in to the platform

 + The user is on the course or instructor page

* Test Data: No test data needed

* Test Steps:

 1. As a learner, navigate to a course or instructor page

 2. Click on the feedback button (e.g., "Provide Feedback")

 3. Fill out the feedback form with detailed comments

 4. Submit the feedback form

 5. Verify that the feedback is displayed on the course or instructor page

* Postconditions: None

* Expected Result: The feedback form is available and functional, and the feedback is displayed correctly on the course or instructor page.

* Severity: Major

* Priority: High

* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 3: Verify that Instructors can view and respond to feedback**

* Test Case ID: FC-003
* Test Case Title: Verify that instructors can view and respond to feedback
* Test Case Description: This test case ensures that instructors can view and respond to feedback from learners.
* Test Suite: User Feedback and Rating System
* Test Priority: High
* Preconditions:
 + The instructor is logged in to the platform
 + The instructor is on their profile page
* Test Data: No test data needed
* Test Steps:
 1. As an instructor, navigate to their profile page
 2. Click on the feedback tab (e.g., "View Feedback")
 3. Verify that the feedback is displayed correctly
 4. Respond to the feedback (e.g., leave a comment)
 5. Submit the response
 6. Verify that the response is displayed correctly on the learner's profile page
* Postconditions: None
* Expected Result: The instructor can view and respond to feedback correctly, and the response is displayed correctly on the learner's profile page.
* Severity: Major
* Priority: High
* Type of Testing: Functional Testing

* Test Case Approach: Positive

**Test Case 4: Verify that Aggregated ratings are displayed publicly**

* Test Case ID: FC-004
* Test Case Title: Verify that aggregated ratings are displayed publicly
* Test Case Description: This test case ensures that aggregated ratings are displayed publicly on the course or instructor page.
* Test Suite: User Feedback and Rating System
* Test Priority: High
* Preconditions:
 + The user is logged in to the platform
 + The user is on the course or instructor page
* Test Data: No test data needed
* Test Steps:
 1. As a learner, navigate to a course or instructor page
 2. Verify that the aggregated ratings are displayed correctly (e.g., average rating, number of reviews)
 3. Verify that the aggregated ratings are also displayed on the platform's home page or search results page
* Postconditions: None
* Expected Result: The aggregated ratings are displayed correctly on the course or instructor page and on the platform's home page or search results page.
* Severity: Major
* Priority: High
* Type of Testing: Functional Testing
* Test Case Approach: Positive

**Test Case 5: Verify that Anonymous feedback options are available**

* Test Case ID: FC-005

* Test Case Title: Verify that anonymous feedback options are available

* Test Case Description: This test case ensures that anonymous feedback options are available for learners who want to provide feedback without disclosing their identity.

* Test Suite: User Feedback and Rating System

* Test Priority: Medium

* Preconditions:

 + The user is logged in to the platform

 + The user is on the course or instructor page

* Test Data: No test data needed

* Test Steps:

 1. As a learner, navigate to a course or instructor page

 2. Click on the feedback button (e.g., "Provide Feedback")

 3. Select the anonymous feedback option

 4. Fill out the feedback form with detailed comments

 5. Submit the feedback form

 6. Verify that the feedback is displayed correctly on the course or instructor page

* Postconditions: None

* Expected Result: The anonymous feedback option is available and functional, and the feedback is displayed correctly on the course or instructor page.

* Severity: Minor

* Priority: Medium

* Type of Testing: Functional Testing

* Test Case Approach: Positive

------------------------------------------------------------------------------------------------

**Data - 75**

**Project name -**

Online Learning Platform

**Project description -**

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

**Feature name -**

Social Media Integration

**Feature description -**

The platform supports social media integration, allowing learners and instructors to share their achievements, courses, and updates on social media platforms. This feature promotes engagement and community building, enhancing the visibility of the platform's offerings.

**Userstory title -**

As a learner, I want to share my achievements on social media so that I can showcase my progress and connect with peers.

**Acceptance criteria -**

['Learners can connect their social media accounts to the platform.', 'Achievements and course completions can be shared on social media.', 'Privacy settings control the visibility of shared content.', 'Instructors can promote courses and updates through social media.', 'Analytics track the reach and engagement of shared content.']

**Generated Test Cases:**

Here are five detailed test cases for the Social Media Integration feature, each covering one or more acceptance criteria from the user story:

**Test Case 1: Verify that learners can connect their social media accounts to the platform**

**Test Case ID:** SSC-001

**Test Case Title:** Verify that learners can connect their social media accounts to the platform

**Test Suite:** Social Media Integration

**Test Priority:** Medium

**Preconditions:**

* User must have an account on the online learning platform.

* User must have a social media account (e.g., Facebook, Twitter).

**Test Data:**

* Test user credentials (username, password).

* Test social media account credentials (username, password).

**Test Steps:**

1. Log in to the online learning platform with test user credentials.

2. Navigate to the social media integration settings page.

3. Click on the "Connect Social Media" button.

4. Select the social media platform to connect (e.g., Facebook).

5. Authorize the online learning platform to access the social media account.

6. Verify that the social media account is successfully connected.

**Postconditions:**

* Social media account is connected to the platform.

**Expected Result:** Learners can connect their social media accounts to the platform.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 2: Verify that achievements and course completions can be shared on social media**

**Test Case ID:** SSC-002

**Test Case Title:** Verify that achievements and course completions can be shared on social media

**Test Suite:** Social Media Integration

**Test Priority:** Medium

**Preconditions:**

* User must have an account on the online learning platform.

* User must have completed a course.

**Test Data:**

* Test user credentials (username, password).

**Test Steps:**

1. Log in to the online learning platform with test user credentials.

2. Complete a course and earn an achievement.

3. Navigate to the social media integration settings page.

4. Click on the "Share Achievement" button.

5. Select the social media platform to share on (e.g., Twitter).

6. Authorize the online learning platform to post on the social media account.

7. Verify that the achievement is successfully shared on social media.

**Postconditions:**

* Achievement is shared on social media.

**Expected Result:** Achievements and course completions can be shared on social media.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 3: Verify that privacy settings control the visibility of shared content**

**Test Case ID:** SSC-003

**Test Case Title:** Verify that privacy settings control the visibility of shared content

**Test Suite:** Social Media Integration

**Test Priority:** High

**Preconditions:**

* User must have an account on the online learning platform.

* User must have completed a course.

**Test Data:**

* Test user credentials (username, password).

**Test Steps:**

1. Log in to the online learning platform with test user credentials.

2. Complete a course and earn an achievement.

3. Navigate to the privacy settings page.

4. Set the visibility of shared content to "Private".

5. Share the achievement on social media.

6. Verify that only authorized users can view the shared content.

7. Set the visibility of shared content to "Public".

8. Share the achievement on social media.

9. Verify that all users can view the shared content.

**Postconditions:**

* Shared content visibility is controlled by privacy settings.

**Expected Result:** Privacy settings control the visibility of shared content.

**Severity:** Critical

**Priority:** High

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive/Negative

**Test Case 4: Verify that instructors can promote courses and updates through social media**

**Test Case ID:** SSC-004

**Test Case Title:** Verify that instructors can promote courses and updates through social media

**Test Suite:** Social Media Integration

**Test Priority:** Medium

**Preconditions:**

* User must have an instructor role on the online learning platform.

* Course must be created and published.

**Test Data:**

* Test instructor credentials (username, password).

**Test Steps:**

1. Log in to the online learning platform with test instructor credentials.

2. Navigate to the course management page.

3. Click on the "Share Course" button.

4. Select the social media platform to share on (e.g., Facebook).

5. Authorize the online learning platform to post on the social media account.

6. Verify that the course is successfully shared on social media.

7. Click on the "Share Update" button.

8. Select the social media platform to share on (e.g., Twitter).

9. Authorize the online learning platform to post on the social media account.

10. Verify that the update is successfully shared on social media.

**Postconditions:**

* Course and updates are shared on social media.

**Expected Result:** Instructors can promote courses and updates through social media.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 5: Verify that analytics track the reach and engagement of shared content**

**Test Case ID:** SSC-005

**Test Case Title:** Verify that analytics track the reach and engagement of shared content

**Test Suite:** Social Media Integration

**Test Priority:** Medium

**Preconditions:**

* User must have an account on the online learning platform.

* Course must be created and published.

**Test Data:**

* Test user credentials (username, password).

**Test Steps:**

1. Log in to the online learning platform with test user credentials.

2. Complete a course and earn an achievement.

3. Share the achievement on social media.

4. Navigate to the analytics dashboard.

5. Verify that the analytics report shows the reach and engagement of the shared content.

6. Track the reach and engagement metrics for a specified time period.

7. Verify that the metrics are accurately recorded and updated in real-time.

**Postconditions:**

* Analytics report accurately tracks reach and engagement of shared content.

**Expected Result:** Analytics track the reach and engagement of shared content.

**Severity:** Minor

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

These test cases ensure that all acceptance criteria from the user story are thoroughly tested, with a focus on identifying potential defects and ensuring product quality through comprehensive testing.

**-----------------------------------------------------------------------------------------------**

**Data - 76**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Activity Tracking

**Feature description -**

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity levels and progress towards fitness goals.

**Userstory title -**

As a user, I want to track my activities so that I can monitor my daily physical activity levels.

**Acceptance criteria -**

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']

**Generated Test Cases:**

Here are five test cases that cover the acceptance criteria for the Activity Tracking feature of the Fitness Tracker App:

**Test Case 1: Verify that the app records data on distance, duration, pace, and calories burned**

| Test Case ID: | AT-01 |
| Test Case Title: | Verify that the app records data on distance, duration, pace, and calories burned |
| Test Case Description: | This test case will ensure that the app captures data on distance, duration, pace, and calories burned for various activities. The test will cover the recording of data for a single activity. |
| Test Suite: | Activity Tracking |
| Test Priority: | High |
| Preconditions: | User has created an account and has a wearable device connected |
| Test Data: | No test data needed |
| Test Steps: |
1. Log in to the app and select the activity type (e.g., running).
2. Start and stop an activity session.
3. Repeat steps 1-2 for different activity types (e.g., walking, cycling).
4. Verify that the app records the correct data for each activity (distance, duration, pace, and calories burned) in the activity summary.
| Postconditions: |
* User is logged in and can view activity summaries.
* No errors are displayed on the screen.

| Expected Result: | The app records accurate data on distance, duration, pace, and calories burned for various activities. |

| Severity: | Blocker |

| Priority: | High |

| Type of Testing: | System Testing |

| Test Case Approach: | Positive |

**Test Case 2: Verify that users can view activity summaries**

| Test Case ID: | AT-02 |

| Test Case Title: | Verify that users can view activity summaries |

| Test Case Description: | This test case will ensure that users can view and access their activity summaries, which should display accurate data on distance, duration, pace, and calories burned. The test will cover the following scenarios: displaying the summaries for a single day, displaying summaries for a week, and displaying summaries for a custom date range. |

| Test Suite: | Activity Tracking |

| Test Priority: | High |

| Preconditions: | User has created an account, has a wearable device connected, and has completed several activities |

| Test Data: | No test data needed |

| Test Steps: |

1. Log in to the app and navigate to the activity summaries page.
2. Verify that the page displays the correct summaries for the current day.
3. Select a date range (e.g., previous week) and verify that the summaries update accordingly.
4. Repeat steps 1-3 for different date ranges (e.g., last 30 days, this month).

| Postconditions: |

* User is logged in and can view activity summaries.

* No errors are displayed on the screen.

| Expected Result: | Users can view and access accurate activity summaries for various date ranges. |

| Severity: | Blocker |

| Priority: | High |

| Type of Testing: | System Testing |

| Test Case Approach: | Positive |

**Test Case 3: Verify that data is stored for historical analysis**

| Test Case ID: | AT-03 |

| Test Case Title: | Verify that data is stored for historical analysis |

| Test Case Description: | This test case will ensure that the app stores data from previous activity sessions, allowing users to analyze their historical activity data. The test will cover the storage of data for 7 days, 30 days, 60 days, and 90 days. |

| Test Suite: | Activity Tracking |

| Test Priority: | Medium |

| Preconditions: | User has created an account, has a wearable device connected, and has completed several activities |

| Test Data: | No test data needed |

| Test Steps: |

1. Log in to the app and navigate to the activity summaries page.

2. Verify that data is stored for the following date ranges:

 * 7 days.

 * 30 days.

 * 60 days.

 * 90 days.

3. Repeat steps 1-2 for different types of activities and verify that data is stored correctly.

| Postconditions: |

* User is logged in and can view activity summaries for the specified date ranges.

* No errors are displayed on the screen.

| Expected Result: | The app stores data for historical analysis for various date ranges. |

| Severity: | Medium |

| Priority: | Medium |

| Type of Testing: | System Testing |

| Test Case Approach: | Positive |

**Test Case 4: Verify that activity tracking can be paused and resumed**

| Test Case ID: | AT-04 |

| Test Case Title: | Verify that activity tracking can be paused and resumed |

| Test Case Description: | This test case will ensure that the app allows users to pause and resume activity tracking, ensuring that data is not lost during interruptions. The test will cover the following scenarios: pausing activity tracking mid-session, resuming paused activity tracking, and pausing activity tracking at the end of a session. |

| Test Suite: | Activity Tracking |

| Test Priority: | High |

| Preconditions: | User has created an account, has a wearable device connected, and is in the middle of an activity session |

| Test Data: | No test data needed |

| Test Steps: |

1. Log in to the app and start an activity session.

2. Pause the activity tracking session.

3. Resume the activity tracking session and verify that data is recorded correctly.

4. Pause the activity tracking session again and verify that data is stored correctly.

5. Repeat steps 1-4 for different activity types and verify that the app behaves as expected.

| Postconditions: |

* User is logged in and can view activity summaries.

* No errors are displayed on the screen.

| Expected Result: | The app allows users to pause and resume activity tracking without data loss. |

| Severity: | Blocker |

| Priority: | High |

| Type of Testing: | System Testing |

| Test Case Approach: | Positive |

**Test Case 5: Verify that users receive notifications for milestones**

| Test Case ID: | AT-05 |

| Test Case Title: | Verify that users receive notifications for milestones |

| Test Case Description: | This test case will ensure that the app sends notifications to users when they reach certain milestones (e.g., completing a certain number of workouts, achieving a specific distance goal). The test will cover the following scenarios: receiving notifications for a single milestone, receiving notifications for multiple milestones, and verifying that notifications are sent correctly for different types of activities. |

| Test Suite: | Activity Tracking |

| Test Priority: | High |

| Preconditions: | User has created an account, has a wearable device connected, and has completed several activities |

| Test Data: | No test data needed |

| Test Steps: |

1. Log in to the app and set a milestone goal (e.g., complete 5 workouts).

2. Complete the required number of workouts and verify that a notification is sent to the user.

3. Repeat steps 1-2 for different types of activities and verify that notifications are sent correctly.

4. Set multiple milestone goals and verify that notifications are sent correctly for each goal.

| Postconditions: |

* User is logged in and receives notifications for achieved milestones.

* No errors are displayed on the screen.

| Expected Result: | The app sends notifications to users when they reach certain milestones. |

| Severity: | Blocker |

| Priority: | High |

| Type of Testing: | System Testing |

| Test Case Approach: | Positive |

These test cases cover all the acceptance criteria for the Activity Tracking feature and are designed to ensure that the app accurately records and stores data, allows users to view activity summaries, and sends notifications for milestones.

**---------------------------------------------------------------------------------------------------**

**Data - 77**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

GPS Tracking

**Feature description -**

Integrated GPS tracking allows users to accurately track outdoor activities and routes, providing detailed maps, elevation profiles, and location-based metrics. Users can visualize their workout routes, analyze performance data, and explore new routes for added variety and challenge.

**Userstory title -**

As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

**Acceptance criteria -**

['GPS tracking accurately records route data.', 'Users can view detailed maps and elevation profiles.', 'Performance data includes location-based metrics.', 'Routes can be saved and revisited.', 'Notifications are provided for deviations from planned routes.']

**Generated Test Cases:**

Here are 5 detailed test cases that thoroughly cover each of the acceptance criteria specified in the user story for the GPS Tracking feature of the Fitness Tracker App.

**Test Case 1: GPS Tracking Accurate Recording of Route Data**

**Test Case ID:** FITT-001

**Test Case Title:** Verify that GPS tracking accurately records route data

**Test Suite:** GPS Tracking Feature

**Test Priority:** High

**Preconditions:** GPS enabled, Device with built-in GPS capability, Access to outdoors for location tracking.

**Test Data:** Device coordinates (Latitude and Longitude)

**Test Steps:**

1. Activate the GPS tracking feature and allow it to capture user's route.

2. Manually record actual route taken while using the app.

3. Compare recorded route data from the app with actual route data.

**Postconditions:** Verify that app displays accurate GPS tracking data.

**Expected Result:** The app accurately records and displays route data when GPS tracking is enabled.

**Severity:** Critical

**Priority:** High

**Type of Testing:** System Testing

**Test Case Approach:** Positive

**Test Case 2: Detailed Maps and Elevation Profiles**

**Test Case ID:** FITT-002

**Test Case Title:** Verify that users can view detailed maps and elevation profiles

**Test Suite:** GPS Tracking Feature

**Test Priority:** Medium

**Preconditions:** GPS enabled, Access to outdoors for location tracking.

**Test Data:** Device coordinates (Latitude and Longitude)

**Test Steps:**

1. Activate the GPS tracking feature and allow it to capture user's route.

2. Navigate to the route summary page.

3. Verify that detailed maps and elevation profiles are displayed accurately.

**Postconditions:** Verify that maps and elevation profiles are viewable.

**Expected Result:** The app displays detailed maps and elevation profiles for the recorded route.

**Severity:** Major

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 3: Performance Data, Including Location-Based Metrics**

**Test Case ID:** FITT-003

**Test Case Title:** Verify that performance data includes location-based metrics

**Test Suite:** GPS Tracking Feature

**Test Priority:** High

**Preconditions:** GPS enabled, Device with built-in GPS capability, Access to outdoors for location tracking.

**Test Data:** Device coordinates (Latitude and Longitude)

**Test Steps:**

1. Activate the GPS tracking feature and allow it to capture user's route.
2. Record the pace, speed, distance, and calories burned during the route.
3. Navigate to the route summary page to verify that location-based metrics are displayed accurately.

**Postconditions:** Verify that location-based metrics are accurately displayed.

**Expected Result:** The app displays accurate location-based performance metrics for the recorded route.

**Severity:** Critical

**Priority:** High

**Type of Testing:** System Testing

**Test Case Approach:** Positive

**Test Case 4: Saving and Revisiting Routes**

**Test Case ID:** FITT-004

**Test Case Title:** Verify that routes can be saved and revisited

**Test Suite:** GPS Tracking Feature

**Test Priority:** Medium

**Preconditions:** GPS enabled, Access to outdoors for location tracking.

**Test Data:** Device coordinates (Latitude and Longitude)

**Test Steps:**

1. Activate the GPS tracking feature and allow it to capture user's route.
2. Save the route for future reference.
3. Revisit the saved route and verify that it is correctly displayed.

**Postconditions:** Verify that saved routes can be revisited.

**Expected Result:** The app accurately saves and retrieves routes for future use.

**Severity:** Major

**Priority:** Medium

**Type of Testing:** Functional Testing

**Test Case Approach:** Positive

**Test Case 5: Notifications for Deviations from Planned Routes**

**Test Case ID:** FITT-005

**Test Case Title:** Verify that notifications are provided for deviations from planned routes

**Test Suite:** GPS Tracking Feature

**Test Priority:** High

**Preconditions:** GPS enabled, Planned route set up, Device capable of producing sound for notification testing.

**Test Data:** Planned route data

**Test Steps:**

1. Activate the GPS tracking feature and allow it to capture user's route.
2. Set up a planned route for comparison.
3. Intentionally deviate from the planned route and verify that notifications are triggered correctly.

**Postconditions:** Verify that notifications are correctly provided for deviations from planned routes.

**Expected Result:** The app delivers notifications for deviations from planned routes, enhancing user experience.

**Severity:** Critical

**Priority:** High

**Type of Testing:** System Testing

**Test Case Approach:** Negative

---

**Data - 78**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Heart Rate Monitoring

**Feature description -**

The app monitors heart rate in real-time using built-in sensors or compatible wearable devices, providing insights into exercise intensity, recovery, and cardiovascular health. Heart rate data helps users optimize their workouts, maintain target heart rate zones, and track improvements in fitness over time.

**Userstory title -**

As a user, I want to monitor my heart rate so that I can optimize my workouts and track my cardiovascular health.

**Acceptance criteria -**

['Heart rate data is recorded in real-time.', 'Users can set and monitor target heart rate zones.', 'Historical heart rate data is stored and viewable.', 'Alerts are provided for abnormal heart rate readings.', 'Data is compatible with various wearable devices.']

**Generated Test Cases:**

Here are five test cases for the Heart Rate Monitoring feature, each covering one or more acceptance criteria from the user story.

**Test Case 1: Verify that Heart Rate Data is Recorded in Real-Time**

* **Test Case ID:** HRM-001

* **Test Case Title:** Verify that Heart Rate Data is Recorded in Real-Time

* **Test Suite:** Heart Rate Monitoring

* **Test Priority:** High

* **Preconditions:**

  * The user has authorized the app to access their wearable device's heart rate sensor.

  * The wearable device is connected to the app.

* **Test Data:** No test data needed

* **Test Steps:**

  1. Launch the app and navigate to the heart rate monitoring screen.

  2. Engage in light physical activity (e.g., walking) for 5 minutes to initiate heart rate recording.

  3. Observe the heart rate graph and values displayed on the screen.

* **Postconditions:**

  * The app displays real-time heart rate data on the screen.

  * The heart rate data graph is updated as the user engages in physical activity.

* **Expected Result:** The app records heart rate data in real-time, displaying accurate and up-to-date values on the screen.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that Users Can Set and Monitor Target Heart Rate Zones**

* **Test Case ID:** HRM-002

* **Test Case Title:** Verify that Users Can Set and Monitor Target Heart Rate Zones

* **Test Suite:** Heart Rate Monitoring

* **Test Priority:** High

* **Preconditions:**

  * The user has created a workout plan with target heart rate zones.

  * The user has authorized the app to access their wearable device's heart rate sensor.

* **Test Data:** No test data needed

* **Test Steps:**

  1. Launch the app and navigate to the workout planning screen.

  2. Set target heart rate zones for a workout plan.

  3. Engage in physical activity (e.g., running) while wearing a wearable device connected to the app.

  4. Observe the app's notification feature for reaching target heart rate zones.

* **Postconditions:**

  * The app displays the set target heart rate zones on the screen.

  * The app notifies the user when they reach the target heart rate zone.

* **Expected Result:** The app accurately records heart rate data and notifies the user when they reach the target heart rate zone.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that Historical Heart Rate Data is Stored and Viewable**

* **Test Case ID:** HRM-003

* **Test Case Title:** Verify that Historical Heart Rate Data is Stored and Viewable

* **Test Suite:** Heart Rate Monitoring

* **Test Priority:** Medium

* **Preconditions:**

  * The user has recorded heart rate data for multiple workouts.

  * The user has accessed the app's historical data features.

* **Test Data:** No test data needed

* **Test Steps:**

  1. Launch the app and navigate to the historical data screen.

  2. Select a workout with recorded heart rate data from the historical data list.

  3. Observe the heart rate graph and values displayed on the screen.

* **Postconditions:**

  * The app displays the selected workout's heart rate data on the screen.

  * The heart rate data graph is accurately represented.

* **Expected Result:** The app stores and displays historical heart rate data accurately.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that Alerts are Provided for Abnormal Heart Rate Readings**

* **Test Case ID:** HRM-004

* **Test Case Title:** Verify that Alerts are Provided for Abnormal Heart Rate Readings

* **Test Suite:** Heart Rate Monitoring

* **Test Priority:** High

* **Preconditions:**

  * The user has set threshold values for abnormal heart rate readings.

  * The user is engaging in intense physical activity (e.g., high-intensity interval training).

* **Test Data:** No test data needed

* **Test Steps:**

  1. Launch the app and navigate to the heart rate monitoring screen.
  2. Set threshold values for abnormal heart rate readings.
  3. Engage in intense physical activity while wearing a wearable device connected to the app.
  4. Observe the app's alert feature for reaching abnormal heart rate readings.
* **Postconditions:**
  * The app displays a notification when the user reaches an abnormal heart rate reading.
  * The app alerts the user through a visible and audible notification.
* **Expected Result:** The app accurately detects abnormal heart rate readings and provides timely alerts to the user.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that Data is Compatible with Various Wearable Devices**

* **Test Case ID:** HRM-005
* **Test Case Title:** Verify that Data is Compatible with Various Wearable Devices
* **Test Suite:** Heart Rate Monitoring
* **Test Priority:** Medium
* **Preconditions:**
  * The user has connected multiple wearable devices to the app.
  * The app supports various data formats and protocols.
* **Test Data:** No test data needed
* **Test Steps:**
  1. Launch the app and navigate to the device settings screen.
  2. Connect and disconnect multiple wearable devices from the app.

  3. Observe the app's data format and protocol compatibility features.

* **Postconditions:**

  * The app accurately records and displays heart rate data from various wearable devices.

  * The app supports various data formats and protocols.

* **Expected Result:** The app provides seamless compatibility with various wearable devices and supports diverse data formats and protocols.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 79**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Calorie Tracking

**Feature description -**

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

**Userstory title -**

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

**Acceptance criteria -**

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

**Generated Test Cases:**

Here are five test cases that cover the acceptance criteria for the Calorie Tracking feature:

**Test Case 1: Verify that users can log daily food intake and calorie consumption**

* **Test Case ID**: FTT-001
* **Test Case Title**: Verify that users can log daily food intake and calorie consumption
* **Test Case Description**: This test case verifies that users can log their daily food intake and calorie consumption to track their calorie balance. The test will check that the app captures the food items, calorie amounts, and dates correctly.
* **Test Suite**: Calorie Tracking
* **Test Priority**: High
* **Preconditions**:
 + User is logged in to the app
 + User has a calorie goal set
* **Test Data**: Sample food items and calorie amounts
* **Test Steps**:
 1. Log in to the app and select the Calorie Tracking feature
 2. Click on the "Log Food" button to start tracking daily food intake

 3. Enter sample food items and calorie amounts

 4. Select the date for the food entry

 5. Confirm the calorie consumption is calculated correctly

* **Postconditions**: The user's calorie balance is updated correctly

* **Expected Result**: The app successfully logs daily food intake and calorie consumption

* **Severity**: High

* **Priority**: High

* **Type of Testing**: Functional

* **Test Case Approach**: Positive

**Test Case 2: Verify that the app calculates and displays total calories consumed and burned**

* **Test Case ID**: FTT-002

* **Test Case Title**: Verify that the app calculates and displays total calories consumed and burned

* **Test Case Description**: This test case verifies that the app correctly calculates the total calories consumed and burned and displays the results to the user.

* **Test Suite**: Calorie Tracking

* **Test Priority**: High

* **Preconditions**:

 + User has logged daily food intake and calorie consumption (Test Case 1)

 + User has tracked physical activity and calorie burn (e.g., workout logging)

* **Test Data**: Sample food items, calorie amounts, and physical activity data

* **Test Steps**:

 1. Verify that the total calories consumed is calculated correctly

 2. Verify that the total calories burned is calculated correctly

 3. Confirm the results are displayed correctly on the app's dashboard

* **Postconditions**: The user's calorie balance is updated correctly

* **Expected Result**: The app accurately calculates and displays total calories consumed and burned
* **Severity**: Critical
* **Priority**: High
* **Type of Testing**: Functional
* **Test Case Approach**: Positive

**Test Case 3: Verify that users can set daily calorie goals and track progress**

* **Test Case ID**: FTT-003
* **Test Case Title**: Verify that users can set daily calorie goals and track progress
* **Test Case Description**: This test case verifies that users can set daily calorie goals and track their progress to achieve those goals.
* **Test Suite**: Calorie Tracking
* **Test Priority**: High
* **Preconditions**:
 + User is logged in to the app
 + User has a calorie goal set (initially)
* **Test Data**: Sample calorie goal values
* **Test Steps**:
 1. Set a daily calorie goal value
 2. Verify that the goal is saved correctly
 3. Track daily food intake and calorie consumption (Test Case 1)
 4. Verify that the app displays the progress towards the goal
* **Postconditions**: The user's calorie goal is updated correctly
* **Expected Result**: The app allows users to set and track daily calorie goals
* **Severity**: Major
* **Priority**: High

* **Type of Testing**: Functional
* **Test Case Approach**: Positive

**Test Case 4: Verify that food items and calorie information are searchable in the database**

* **Test Case ID**: FTT-004
* **Test Case Title**: Verify that food items and calorie information are searchable in the database
* **Test Case Description**: This test case verifies that users can search for food items and their corresponding calorie information in the app's database.
* **Test Suite**: Calorie Tracking
* **Test Priority**: Medium
* **Preconditions**:
 + User is logged in to the app
 + Food items and calorie information are available in the database
* **Test Data**: Sample food items and calorie amounts
* **Test Steps**:
 1. Search for a specific food item in the app's database
 2. Verify that the correct calorie information is displayed
 3. Search for multiple food items and verify the results
* **Postconditions**: The search functionality is working correctly
* **Expected Result**: The app allows users to search for food items and calorie information
* **Severity**: Minor
* **Priority**: Medium
* **Type of Testing**: Functional
* **Test Case Approach**: Positive

**Test Case 5: Verify that users receive alerts for exceeding calorie goals**

* **Test Case ID**: FTT-005

* **Test Case Title**: Verify that users receive alerts for exceeding calorie goals

* **Test Case Description**: This test case verifies that users receive alerts when they exceed their daily calorie goals.

* **Test Suite**: Calorie Tracking

* **Test Priority**: High

* **Preconditions**:

 + User is logged in to the app

 + User has a calorie goal set

 + User has exceeded their daily calorie goal

* **Test Data**: Sample calorie goal values and food intake data

* **Test Steps**:

 1. Set a daily calorie goal value

 2. Track daily food intake and calorie consumption (Test Case 1)

 3. Exceed the daily calorie goal

 4. Verify that the app displays an alert for exceeding the goal

* **Postconditions**: The user receives an alert for exceeding their calorie goal

* **Expected Result**: The app sends alerts to users when they exceed their daily calorie goals

* **Severity**: Critical

* **Priority**: High

* **Type of Testing**: Functional

* **Test Case Approach**: Positive

---

**Data - 80**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Sleep Tracking

**Feature description -**

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

**Userstory title -**

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

**Acceptance criteria -**

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

**Generated Test Cases:**

Here are five detailed test cases, each thoroughly covering one or more of the acceptance criteria for the Sleep Tracking feature:

**Test Case 1: Verify that sleep data is recorded, including duration and quality**

* **Test Case ID**: SLP-001
* **Test Case Title**: Verify that sleep data is recorded, including duration and quality
* **Test Case Description**: This test case ensures that the app records sleep data, including basic information such as duration and quality, when a user sleeps.
* **Test Suite**: V1.1
* **Test Priority**: High
* **Preconditions**:
 + User account created and logged in.
 + User has enabled sleep tracking in settings.
* **Test Data**: No test data needed
* **Test Steps**:
 1. Open the app and log in.
 2. Enable sleep tracking in settings.
 3. Simulate sleep using a mockup device or simulator.
 4. Record various durations and qualities for the simulated sleep session.
 5. Check that the recorded sleep data includes duration and quality.
* **Postconditions**:
 + Sleep data is recorded.
* **Expected Result**: Recorded sleep data includes duration and quality.
* **Severity**: Critical
* **Priority**: High
* **Type of Testing**: Functional Testing
* **Test Case Approach**: Positive

**Test Case 2: Verify that users can view detailed sleep stages and patterns**

* **Test Case ID**: SLP-002

* **Test Case Title**: Verify that users can view detailed sleep stages and patterns

* **Test Case Description**: This test case ensures that the app provides users with an overview of sleep stages and patterns.

* **Test Suite**: V1.1

* **Test Priority**: High

* **Preconditions**:

 + User account created and logged in.

 + User has enabled sleep tracking in settings.

 + Sleep data recorded.

* **Test Data**: No test data needed

* **Test Steps**:

 1. Open the app and log in.

 2. Navigate to the sleep tracking section.

 3. Check that sleep stages and patterns are displayed correctly.

 4. Filter sleep data by dates to observe variations in sleep patterns.

 5. Check that the app provides insights into sleep-stage transitions.

* **Postconditions**:

 + Detailed sleep stages and patterns are displayed.

* **Expected Result**: Detailed sleep stages and patterns are displayed correctly.

* **Severity**: Major

* **Priority**: High

* **Type of Testing**: Functional Testing

* **Test Case Approach**: Positive

**Test Case 3: Verify that the app provides insights into sleep disturbances**

* **Test Case ID**: SLP-003

* **Test Case Title**: Verify that the app provides insights into sleep disturbances

* **Test Case Description**: This test case ensures that the app provides users with insights into sleep disturbances, such as poor sleep quality or fragmented sleep.

* **Test Suite**: V1.1

* **Test Priority**: High

* **Preconditions**:

 + User account created and logged in.

 + User has enabled sleep tracking in settings.

 + Sleep data recorded.

* **Test Data**: No test data needed

* **Test Steps**:

 1. Open the app and log in.

 2. Navigate to the sleep tracking section.

 3. Check that the app indicates sleep disturbances, such as poor sleep quality or fragmented sleep.

 4. Filter sleep data by dates to observe variations in sleep disturbances.

 5. Check that the app provides suggestions for addressing sleep disturbances.

* **Postconditions**:

 + Insights into sleep disturbances are provided.

* **Expected Result**: Insights into sleep disturbances are displayed correctly.

* **Severity**: Major

* **Priority**: High

* **Type of Testing**: Functional Testing

* **Test Case Approach**: Positive

**Test Case 4: Verify that users receive recommendations for improving sleep quality**

* **Test Case ID**: SLP-004

* **Test Case Title**: Verify that users receive recommendations for improving sleep quality

* **Test Case Description**: This test case ensures that the app provides users with personalized recommendations for improving sleep quality.

* **Test Suite**: V1.1

* **Test Priority**: High

* **Preconditions**:

 + User account created and logged in.

 + User has enabled sleep tracking in settings.

 + Sleep data recorded.

 + Sleep disturbances detected.

* **Test Data**: No test data needed

* **Test Steps**:

 1. Open the app and log in.

 2. Navigate to the sleep tracking section.

 3. Check that the app provides recommendations for improving sleep quality.

 4. Filter sleep data by dates to observe variations in sleep-quality improvements.

 5. Check that the app tracks progress after implementing suggestions.

* **Postconditions**:

 + Users receive recommendations for improving sleep quality.

* **Expected Result**: Recommendations for improving sleep quality are displayed correctly.

* **Severity**: Critical

* **Priority**: High

* **Type of Testing**: Functional Testing

* **Test Case Approach**: Positive

**Test Case 5: Verify that sleep data can be viewed in daily, weekly, and monthly summaries**

* **Test Case ID**: SLP-005

* **Test Case Title**: Verify that sleep data can be viewed in daily, weekly, and monthly summaries

* **Test Case Description**: This test case ensures that the app provides users with the option to view sleep data in daily, weekly, and monthly summaries.

* **Test Suite**: V1.1

* **Test Priority**: Medium

* **Preconditions**:

 + User account created and logged in.

 + User has enabled sleep tracking in settings.

 + Sleep data recorded.

* **Test Data**: No test data needed

* **Test Steps**:

 1. Open the app and log in.

 2. Navigate to the sleep tracking section.

 3. Check that the app provides options for viewing sleep data in daily, weekly, and monthly summaries.

 4. Filter sleep data by dates to observe variations in sleep patterns across different date ranges.

 5. Check that the app provides insights into sleep trends.

* **Postconditions**:

 + Sleep data can be viewed in daily, weekly, and monthly summaries.

* **Expected Result**: Sleep data can be viewed correctly across different date ranges.

* **Severity**: Minor

* **Priority**: Medium

* **Type of Testing**: Functional Testing

* **Test Case Approach**: Positive

---

**Data - 81**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Workout Logging

**Feature description -**

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

**Userstory title -**

As a user, I want to log my workouts so that I can track my progress and improve my performance.

**Acceptance criteria -**

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

**Generated Test Cases:**

Here are five test cases that cover each acceptance criterion of the Workout Logging feature:

**Test Case 1: Logging Various Workout Types**

* **Test Case ID:** FTR001
* **Test Case Title:** Verify that users can log various workout types with details.
* **Test Suite:** Workout Logging
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the app
 + User has already created a workout profile
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log into the app and navigate to the workout logging page
 2. Click on the "Add Workout" button and select a workout type (e.g. running, weightlifting, cardio)
 3. Fill in the workout details (duration, intensity, notes)
 4. Click on the "Save Workout" button
* **Postconditions:**
 + The workout is saved successfully
 + The user can view the logged workout on their profile
* **Expected Result:** Users can log various workout types with details.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Workout Logs Include Duration, Intensity, and Notes**

* **Test Case ID:** FTR002
* **Test Case Title:** Verify that workout logs include duration, intensity, and notes.
* **Test Suite:** Workout Logging

* **Test Priority:** High

* **Preconditions:**

 + User has already logged a workout

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log into the app and navigate to the workout logging page

 2. Click on the "View Workout" button for a previously logged workout

 3. Verify that the workout log includes duration, intensity, and notes

* **Postconditions:**

 + The workout log is displayed correctly

* **Expected Result:** Workout logs include duration, intensity, and notes.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Editing and Deleting Workout Entries**

* **Test Case ID:** FTR003

* **Test Case Title:** Verify that users can edit and delete workout entries.

* **Test Suite:** Workout Logging

* **Test Priority:** High

* **Preconditions:**

 + User has already logged a workout

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log into the app and navigate to the workout logging page

 2. Click on the "Edit Workout" button for a previously logged workout

 3. Edit the workout details (duration, intensity, notes)

 4. Click on the "Save Workout" button

 5. Click on the "Delete Workout" button for the edited workout

 6. Confirm that the workout is deleted successfully

* **Postconditions:**

 + The workout is edited successfully

 + The workout is deleted successfully

* **Expected Result:** Users can edit and delete workout entries.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Data Storage and Viewing**

* **Test Case ID:** FTR004

* **Test Case Title:** Verify that data is stored and viewable for tracking progress.

* **Test Suite:** Workout Logging

* **Test Priority:** High

* **Preconditions:**

 + User has already logged multiple workouts

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log into the app and navigate to the workout logging page

 2. Click on the "View Workouts" button to view all logged workouts

 3. Verify that the workouts are displayed correctly, including details

 4. Click on the "Filter Workouts" button and select a date range

 5. Verify that the workout log is filtered correctly

* **Postconditions:**

 + The workout logs are displayed correctly

 + The workout logs are filtered correctly

* **Expected Result:** Data is stored and viewable for tracking progress.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Filtering Workout Logs by Date and Workout Type**

* **Test Case ID:** FTR005

* **Test Case Title:** Verify that users can filter workout logs by date and workout type.

* **Test Suite:** Workout Logging

* **Test Priority:** Medium

* **Preconditions:**

 + User has already logged multiple workouts

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log into the app and navigate to the workout logging page

 2. Click on the "Filter Workouts" button

 3. Select a date range for filtering workouts

 4. Select a workout type for filtering workouts

 5. Click on the "Apply Filter" button

 6. Verify that the workout log is filtered correctly

* **Postconditions:**

 + The workout logs are filtered correctly by date and workout type

* **Expected Result:** Users can filter workout logs by date and workout type.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 82**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Goal Setting

**Feature description -**

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

**Userstory title -**

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

**Acceptance criteria -**

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

**Generated Test Cases:**

Here are the test cases for the "Goal Setting" feature of the Fitness Tracker App:

**Test Case 1: Verify that users can set goals for various fitness metrics**

* **Test Case ID:** GTS-001
* **Test Case Title:** Verify that users can set goals for various fitness metrics
* **Test Case Description:** This test case ensures that users can set goals for various fitness metrics such as steps taken, calories burned, distance traveled, and heart rate. The test case will verify that the app allows users to input specific values for these metrics and save the goals.
* **Test Suite:** Goal Setting Feature
* **Test Priority:** High
* **Preconditions:** User is logged in, app has necessary permissions to access wearable devices or fitness equipment
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the app and navigate to the "Goal Setting" section
 2. Click on the "Add New Goal" button
 3. Select the "Steps Taken" metric from the dropdown menu
 4. Enter a specific value for the "Steps Taken" metric (e.g., 10000)
 5. Save the goal
 6. Repeat the process with different metrics (e.g., "Calories Burned", "Distance Traveled", "Heart Rate")
* **Postconditions:** Goals are listed in the "Goal Setting" section with the input values stored

* **Expected Result:** Users can successfully set goals for various fitness metrics

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that progress tracking is available for each goal**

* **Test Case ID:** GTS-002

* **Test Case Title:** Verify that progress tracking is available for each goal

* **Test Case Description:** This test case ensures that users can track progress towards their set goals. The test case will verify that the app displays the progress bar and provides real-time updates as the user increases their progress.

* **Test Suite:** Goal Setting Feature

* **Test Priority:** High

* **Preconditions:** User has set goals for various fitness metrics

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the "Goal Setting" section
2. Select a goal (e.g., "Steps Taken")
3. Perform physical activity to increase progress towards the goal (e.g., walking 1000 steps)
4. Observe the progress bar and ensure it updates in real-time
5. Repeat the process with different goals and activities

* **Postconditions:** Progress bar accurately reflects the user's progress towards their goals

* **Expected Result:** Progress tracking is available for each goal, and the progress bar updates in real-time

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users receive motivational feedback and reminders**

* **Test Case ID:** GTS-003

* **Test Case Title:** Verify that users receive motivational feedback and reminders

* **Test Case Description:** This test case ensures that users receive motivational feedback and reminders to stay on track with their goals. The test case will verify that the app provides encouraging messages and sends notifications as scheduled.

* **Test Suite:** Goal Setting Feature

* **Test Priority:** Medium

* **Preconditions:** User has set goals and enabled reminders and notifications

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app and navigate to the "Goal Setting" section

2. Set up reminders and notifications for a goal (e.g., "Steps Taken")

3. Observe the app to ensure that it sends reminders and notifications as scheduled

4. Review motivational feedback messages displayed in the app

* **Postconditions:** Reminders and notifications are sent to the user as scheduled

* **Expected Result:** Users receive motivational feedback and reminders to stay on track with their goals

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that goals can be adjusted or updated as needed**

* **Test Case ID:** GTS-004

* **Test Case Title:** Verify that goals can be adjusted or updated as needed

* **Test Case Description:** This test case ensures that users can adjust or update their goals as needed. The test case will verify that the app allows users to modify existing goals, delete goals, and change goal settings.

* **Test Suite:** Goal Setting Feature

* **Test Priority:** Medium

* **Preconditions:** User has set goals for various fitness metrics

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the app and navigate to the "Goal Setting" section

 2. Select a goal (e.g., "Steps Taken")

 3. Modify the goal value or settings (e.g., increase the target steps)

 4. Save the changes

 5. Repeat the process with different goals and modifications

* **Postconditions:** Goal values and settings are updated in the app

* **Expected Result:** Users can adjust or update their goals as needed

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that the app provides a summary of goal attainment**

* **Test Case ID:** GTS-005

* **Test Case Title:** Verify that the app provides a summary of goal attainment

* **Test Case Description:** This test case ensures that the app displays a summary of the user's goal

attainment, including progress, targets, and milestones. The test case will verify that the app provides accurate and comprehensible information.

* **Test Suite:** Goal Setting Feature

* **Test Priority:** Medium

* **Preconditions:** User has set goals for various fitness metrics and tracked progress towards those goals

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the app and navigate to the "Goal Setting" section

 2. Observe the summary of goal attainment, including progress, targets, and milestones

 3. Review the accuracy and comprehensibility of the information displayed

* **Postconditions:** App provides accurate and comprehensible summary of goal attainment

* **Expected Result:** Users can view a summary of their goal attainment, including progress, targets, and milestones

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 83**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Workout Plans

**Feature description -**

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

**Userstory title -**

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

**Acceptance criteria -**

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

**Generated Test Cases:**

Here are five detailed test cases for the "Workout Plans" feature, following the provided Test Case Creation Template and ensuring comprehensive coverage of the acceptance criteria:

**Test Case 1**

Test Case ID: WP-1

Test Case Title: Verify that users can select workout plans based on fitness levels

Test Case Description: This test case validates that users can filter and select workout plans tailored to their fitness levels, ensuring an optimal experience.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

* Users must be logged in

* The user has not selected a workout plan yet

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the workout plans section

2. Click on the filter button and select a fitness level (e.g., beginner, intermediate, advanced)

3. Verify that the workout plans list is updated to show plans matching the selected fitness level

4. Select a workout plan and review its details

5. Verify that the plan is tailored to the chosen fitness level

Postconditions:

* The selected workout plan is displayed on the dashboard

Expected Result: Users can successfully filter and select workout plans based on fitness levels.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2**

Test Case ID: WP-2

Test Case Title: Verify that plans include detailed exercise routines and schedules

Test Case Description: This test case ensures that workout plans contain comprehensive information about exercise routines, including schedules and details.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

* Users must have selected a workout plan

Test Data: No test data needed

Test Steps:

1. Login to the app and navigate to the selected workout plan

2. Review the exercise routine for the first day of the plan

3. Verify that the exercise routine includes:

 * Clear instructions

 * Correlated images or diagrams

 * Scheduled timing for each exercise

4. Click on an exercise to review its details

5. Verify that the exercise details include:

 * Description

 * Targeted muscle groups

 * Required equipment

6. Review the plan's schedule for the next day

7. Verify that the schedule is accurate and includes the necessary equipment

Postconditions:

* The workout plan is updated to reflect the current day's schedule

Expected Result: Plans include detailed exercise routines and schedules.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3**

Test Case ID: WP-3

Test Case Title: Verify that users can track adherence and progress within the plan

Test Case Description: This test case validates that users can monitor their progress and adherence to the workout plan, ensuring motivation and goal achievement.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

* Users must have started a workout plan

Test Data: No test data needed

Test Steps:

1. Log in to the app and navigate to the started workout plan

2. Review the plan's progress tracking feature

3. Verify that users can view completed exercises and workouts

4. Verify that users can view ongoing exercises and workouts

5. Click on the progress tracking feature to review:

 * Completed workouts

 * Ongoing workouts

 * Goal progress

6. Verify that the progress tracking feature accurately reflects the user's progress

Postconditions:

* The user's progress is updated in the app

Expected Result: Users can successfully track adherence and progress within the plan.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4**

Test Case ID: WP-4

Test Case Title: Verify that the app provides notifications and reminders for workouts

Test Case Description: This test case ensures that users receive timely notifications and reminders about upcoming workouts, motivating them to stay on track.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

* Users must have started a workout plan

Test Data: No test data needed

Test Steps:

1. Sign up for a workout plan and start tracking progress

2. Configure the app's notification settings

3. Verify that the user receives a notification and reminder for an upcoming workout

4. Check the app's notification settings to ensure they are configured correctly

5. Verify that the user receives a notification 30 minutes before the workout and another 15 minutes before the workout

6. Verify that the app sends reminders for missed workouts

Postconditions:

* The app sends notifications and reminders as configured by the user

Expected Result: Users receive timely notifications and reminders for workouts.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5**

Test Case ID: WP-5

Test Case Title: Verify that users can adjust plans based on progress and preferences

Test Case Description: This test case validates that users can modify their workout plan according to their progress and preferences, ensuring a personalized fitness experience.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

* Users must have started a workout plan

Test Data: No test data needed

Test Steps:

1. Review the user's progress within the workout plan

2. Determine the user's progress level (beginner, intermediate, advanced)

3. Click on the 'Adjust Plan' button to update the workout plan

4. Select a new plan tailored to the user's current fitness level

5. Verify that the updated plan:

 * Reflects the user's new fitness level

 * Inquires about goals and preferences

6. Review the updated plan for accuracy

7. Verify that the new plan is effective for the user's current fitness level

Postconditions:

* The user's workout plan is updated to reflect their new fitness level

Expected Result: Users can successfully adjust plans based on progress and preferences.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

The above test cases ensure comprehensive coverage of all acceptance criteria for the user story "Follow structured workout plans." These test cases cover critical functionality of the workout plans feature, including plan selection, adherence tracking, progress monitoring, and updates to meet individual progress and preferences. By executing these test cases, we can identify potential defects early in the development cycle, ensuring a high-quality software product that meets user expectations.

---

**Data - 84**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Community Support

**Feature description -**

Users can connect with a community of like-minded individuals, share progress updates, participate in challenges, and offer support and encouragement to fellow users. Community support fosters accountability, motivation, and a sense of camaraderie among users striving towards similar fitness goals.

**Userstory title -**

As a user, I want to connect with a community so that I can share my progress and stay motivated.

**Acceptance criteria -**

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.', 'The app facilitates participation in community challenges.', 'Users can offer and receive support and encouragement.', 'Community activity and engagement are tracked.']

**Generated Test Cases:**

Here are five detailed test cases for the "Community Support" feature of the Fitness Tracker App, based on the user story and acceptance criteria provided:

**Test Case 1: Join and Participate in Community Groups**

Test Case ID: CSCG001

Test Case Title: Verify that users can join and participate in community groups

Test Case Description: As a user, I want to be able to join and participate in community groups, so that I can connect with like-minded individuals and stay motivated. This test case will verify that users can successfully join and participate in community groups, including posting updates and comments.

Test Suite: Community Support

Test Priority: High

Preconditions:

* User has a registered account

* Community groups exist on the app

Test Data:

* User account information

* Community group details

Test Steps:

1. Log in to the app with a registered user account.

2. Navigate to the community section and click on "Join Group".

3. Select a community group to join and confirm membership.

4. Post an update in the community group.

5. Comment on a post made by another user in the group.

Postconditions:

* User is a member of the community group.

* Post is successfully submitted and displayed in the group feed.

Expected Result: The user can join and participate in community groups by posting updates and commenting on posts.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Share Progress Updates and Achievements**

Test Case ID: CPSA002

Test Case Title: Verify that users can share progress updates and achievements

Test Case Description: As a user, I want to be able to share my progress updates and achievements with the community, so that I can stay motivated and connected with like-minded individuals. This test case will verify that users can successfully share progress updates and achievements, including adding media files and tracking engagement metrics.

Test Suite: Community Support

Test Priority: High

Preconditions:

* User has a registered account

* Community groups exist on the app

Test Data:

* User account information

* Community group details

Test Steps:

1. Log in to the app with a registered user account.

2. Navigate to the community section and select a community group to post an update.

3. Choose the "Share Progress" option and add media files (e.g. photo, video, etc.).

4. Enter a brief description of the update and confirm posting.

5. View the post's engagement metrics (e.g. likes, comments, etc.).

Postconditions:

* Post is successfully submitted and displayed in the group feed.

* Engagement metrics are accurately tracked.

Expected Result: The user can share progress updates and achievements, including adding media files, and view engagement metrics.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Participate in Community Challenges**

Test Case ID: CPPC004

Test Case Title: Verify that the app facilitates participation in community challenges

Test Case Description: As a user, I want to be able to participate in community challenges, so that I can stay motivated and connected with like-minded individuals. This test case will verify that the app facilitates participation in community challenges, including tracking progress and rewards.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

* User has a registered account

* Community challenges exist on the app

Test Data:

* User account information

* Community challenge details

Test Steps:

1. Log in to the app with a registered user account.

2. Navigate to the community section and select a community challenge to participate in.

3. Track progress and submit updates to the challenge.

4. Earn rewards and badges for completing the challenge.

5. View challenge metrics and standings.

Postconditions:

* User has successfully participated in the community challenge.

* Rewards and badges are accurately awarded.

Expected Result: The app facilitates participation in community challenges, including tracking progress and rewards.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Offer and Receive Support and Encouragement**

Test Case ID: CPSO005

Test Case Title: Verify that users can offer and receive support and encouragement

Test Case Description: As a user, I want to be able to offer and receive support and encouragement from fellow community members, so that I can stay motivated and connected. This test case will verify that users can successfully offer and receive support and encouragement, including commenting on posts and sending messages.

Test Suite: Community Support

Test Priority: High

Preconditions:

* User has a registered account

* Community groups exist on the app

Test Data:

* User account information

* Community group details

Test Steps:

1. Log in to the app with a registered user account.

2. Navigate to the community section and select a community group to participate in.

3. Post a message or comment offering support to another user.

4. Receive a message or comment from another user offering support.

5. Engage in a conversation with the other user, responding to messages and comments.

Postconditions:

* User has successfully offered and received support and encouragement.

* Messages and comments are accurately tracked.

Expected Result: Users can offer and receive support and encouragement through commenting on posts and sending messages.

Severity: Minor

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Track Community Activity and Engagement**

Test Case ID: CCTC006

Test Case Title: Verify that community activity and engagement are tracked

Test Case Description: As a user, I want to be able to view community activity and engagement metrics, so that I can understand the level of engagement within the community. This test case will verify that community activity and engagement metrics are accurately tracked and displayed.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

* User has a registered account

* Community groups exist on the app

Test Data:

* User account information

* Community group details

Test Steps:

1. Log in to the app with a registered user account.

2. Navigate to the community section and select a community group to view activity and engagement metrics.

3. View engagement metrics, including post likes, comments, and shares.

4. View activity metrics, including group updates and member activity.

5. Compare engagement and activity metrics across different community groups.

Postconditions:

* Community activity and engagement metrics are accurately tracked and displayed.

Expected Result: Community activity and engagement metrics are accurately tracked and displayed, providing insight into the level of engagement within the community.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Each test case covers a specific acceptance criterion, and together they provide comprehensive coverage of the Community Support feature. These test cases can help ensure that the app is functional, engaging, and supportive for users.

---

**Data - 85**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Integration with Wearable Devices

**Feature description -**

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

**Userstory title -**

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

**Acceptance criteria -**

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

**Generated Test Cases:**

Here are five detailed test cases for the Integration with Wearable Devices feature, following the provided Test Case Creation Template:

**Test Case 1: Verify that the app supports integration with multiple wearable devices**

Test Case ID: WDV001

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: This test case ensures that the fitness tracker app supports integration with multiple wearable devices, including popular fitness trackers, smartwatches, and heart rate monitors.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions: None

Test Data: No test data needed

Test Steps:

1. Launch the fitness tracker app on the mobile device.

2. Tap on the "Integrate Wearable Device" option.

3. Select a wearable device from the list of supported devices.

4. Pair the wearable device with the app using Bluetooth or Wi-Fi.

5. Verify that the wearable device is successfully integrated with the app.

Postconditions: The wearable device is integrated with the app.

Expected Result: The app successfully integrates with multiple wearable devices.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that users can sync activity data from their wearables**

Test Case ID: WDV002

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: This test case ensures that users can sync activity data from their wearable devices with the fitness tracker app.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions: Wearable device is paired with the app.

Test Data: Sample activity data from wearable device (e.g., steps, heart rate, calories burned)

Test Steps:

1. Launch the fitness tracker app on the mobile device.
2. Tap on the "Sync Wearable Data" option.
3. Select the wearable device that was previously paired with the app.
4. Verify that the activity data from the wearable device is synced with the app.
5. Check that the synced data is accurately reflected on the app's dashboard.

Postconditions: Activity data is synced with the app.

Expected Result: The activity data from the wearable device is accurately synced with the app.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that the app provides real-time data updates from wearables**

Test Case ID: WDV003

Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: This test case ensures that the fitness tracker app provides real-time data updates from wearable devices.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions: Wearable device is paired with the app.

Test Data: Sample activity data from wearable device (e.g., heart rate, calories burned)

Test Steps:

1. Launch the fitness tracker app on the mobile device.
2. Pair a wearable device with the app.
3. Perform an activity (e.g., walk or run) while wearing the wearable device.
4. Observe that the app displays real-time updates of the activity data from the wearable device.
5. Verify that the data is updated in real-time, as the user performs the activity.

Postconditions: Real-time data updates are provided by the app.

Expected Result: The app displays real-time updates of the activity data from the wearable device.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that wearable data includes steps, heart rate, and calories burned**

Test Case ID: WDV004

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: This test case ensures that wearable data, synced with the app, includes steps, heart rate, and calories burned.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions: Wearable device is paired with the app; activity data is synced with the app.

Test Data: Sample activity data from wearable device (e.g., steps, heart rate, calories burned)

Test Steps:

1. Launch the fitness tracker app on the mobile device.

2. Swipe through the app's dashboard to view various health metrics (e.g., steps, heart rate, calories burned).

3. Verify that the health metrics from the wearable device are accurately displayed on the app's dashboard.

4. Check that the individual metrics (steps, heart rate, calories burned) are correctly included in the app's data display.

Postconditions: Wearable data includes steps, heart rate, and calories burned.

Expected Result: The wearable data synced with the app includes steps, heart rate, and calories burned.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that users can troubleshoot connectivity issues with wearables**

Test Case ID: WDV005

Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: This test case ensures that users can troubleshoot connectivity issues with their wearable devices, integrated with the fitness tracker app.

Test Suite: Integration with Wearable Devices

Test Priority: Low

Preconditions: Wearable device is paired with the app; connectivity issue is simulated (e.g., Wi-Fi or Bluetooth dropped).

Test Data: None

Test Steps:

1. Launch the fitness tracker app on the mobile device.

2. Simulate a connectivity issue with the wearable device (e.g., Wi-Fi or Bluetooth dropped).

3. Tap on the "Troubleshoot Connectivity" option.

4. Follow the on-screen instructions to troubleshoot the connectivity issue.

5. Verify that the connectivity issue is resolved, and the wearable device is reconnected to the app.

Postconditions: Connectivity issue is resolved.

Expected Result: The user can successfully troubleshoot and resolve connectivity issues with their wearable device.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Negative

--------------------------------------------------------------------------------------------

**Data - 86**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Water Intake Tracking

**Feature description -**

Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

**Userstory title -**

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

**Acceptance criteria -**

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

**Generated Test Cases:**

Here are five detailed test cases for the Water Intake Tracking feature of the Fitness Tracker App:

**Test Case 1: Logging Daily Water Intake**

Test Case ID: FIT_WIT_001

Test Case Title: Verify that users can log their daily water intake.

Test Case Description: This test case verifies that users can accurately record their daily water consumption and that the app calculates the total water consumption for the day.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions: User is logged in, Water Intake Tracking feature is enabled.

Test Data: Valid water intake data

Test Steps:

1. Open the app and navigate to Water Intake Tracking page.
2. Enter valid water intake data (e.g., 2 liters).
3. Save the data and verify that the total water consumption for the day is updated correctly.
4. Repeat steps 2-3 with different amounts of water intake.

Postconditions: Total water consumption for the day is displayed correctly.

Expected Result: The app accurately calculates and displays the total water consumption for the day.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Setting Daily Hydration Goals**

Test Case ID: FIT_WIT_002

Test Case Title: Verify that users can set daily hydration goals.

Test Case Description: This test case verifies that users can set daily hydration goals and that the app takes into account these goals when providing reminders and progress tracking.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions: User is logged in, Water Intake Tracking feature is enabled.

Test Data: Valid hydration goal data

Test Steps:

1. Open the app and navigate to Water Intake Tracking page.
2. Set a daily hydration goal (e.g., 3 liters).
3. Save the goal and verify that the app displays the goal and provides reminders accordingly.
4. Repeat steps 2-3 with different hydration goals.

Postconditions: Hydration goal is displayed correctly, reminders are provided correctly.

Expected Result: The app accurately takes into account the set hydration goal and provides reminders and progress tracking accordingly.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Reminder System**

Test Case ID: FIT_WIT_003

Test Case Title: Verify that reminders are provided to encourage regular water intake.

Test Case Description: This test case verifies that the app provides reminders to users to drink water throughout the day and that these reminders are based on the user's hydration goals and progress.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions: User is logged in, Water Intake Tracking feature is enabled, hydration goal is set.

Test Data: Valid reminder data

Test Steps:

1. Open the app and navigate to Water Intake Tracking page.
2. Set a hydration goal and save it.
3. Verify that the app provides reminders to drink water throughout the day.
4. Repeat steps 2-3 with different hydration goals and progress.

Postconditions: Reminders are provided correctly, reminders are based on hydration goals and progress.

Expected Result: The app provides reminders to users to drink water throughout the day, based on their hydration goals and progress.

Severity: Critical

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Hydration Progress and Trends**

Test Case ID: FIT_WIT_004

Test Case Title: Verify that users can view hydration progress and trends.

Test Case Description: This test case verifies that users can view their hydration progress and trends over time and that this data is accurate and up-to-date.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions: User is logged in, Water Intake Tracking feature is enabled, hydration goal is set.

Test Data: Valid hydration data

Test Steps:

1. Open the app and navigate to Water Intake Tracking page.

2. View hydration progress and trends over time.

3. Verify that the data is accurate and up-to-date.

4. Repeat steps 2-3 with different hydration goals and progress.

Postconditions: Hydration progress and trends are displayed correctly.

Expected Result: The app displays accurate and up-to-date hydration progress and trends.

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Error Handling**

Test Case ID: FIT_WIT_005

Test Case Title: Verify that the app handles errors when logging daily water intake.

Test Case Description: This test case verifies that the app handles errors when users attempt to log daily water intake, such as invalid data or out-of-range values.

Test Suite: Water Intake Tracking

Test Priority: Low

Preconditions: User is logged in, Water Intake Tracking feature is enabled.

Test Data: Invalid water intake data

Test Steps:

1. Open the app and navigate to Water Intake Tracking page.

2. Enter invalid water intake data (e.g., negative value or excessive amount).

3. Verify that the app displays an error message and does not update the total water consumption for the day.

4. Repeat steps 2-3 with different invalid data.

Postconditions: Error message is displayed correctly, total water consumption for the day is not updated incorrectly.

Expected Result: The app correctly handles errors when users attempt to log daily water intake with invalid data.

Severity: Minor

Priority: Low

Type of Testing: Functional Testing

Test Case Approach: Negative

---

**Data - 87**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Nutrition Tracking

**Feature description -**

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

**Userstory title -**

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

**Acceptance criteria -**

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

**Generated Test Cases:**

Here are five detailed test cases for the Nutrition Tracking feature of the Fitness Tracker App, covering all the acceptance criteria from the user story:

**Test Case 1: Verify that users can log their daily meals and snacks**

* **Test Case ID:** NT-001
* **Test Case Title:** Verify that users can log their daily meals and snacks
* **Test Suite:** Nutrition Tracking
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the app
 + User has access to the nutrition tracking feature
* **Test Data:** User's meal and snack data
* **Test Steps:**
 1. Launch the app and navigate to the nutrition tracking section
 2. Select a date for which the user wants to log meals and snacks

3. Click on the "Log Meal" button and enter a valid meal name and details

4. Click on the "Log Snack" button and enter a valid snack name and details

5. Verify that the meal and snack are recorded in the user's history

* **Postconditions:** User's meal and snack history is updated

* **Expected Result:** User's meals and snacks are successfully logged in the app

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that the app calculates macronutrient intake (carbs, fats, proteins)**

* **Test Case ID:** NT-002

* **Test Case Title:** Verify that the app calculates macronutrient intake (carbs, fats, proteins)

* **Test Suite:** Nutrition Tracking

* **Test Priority:** High

* **Preconditions:**

 + User has logged multiple meals and snacks

* **Test Data:** User's meal and snack data with nutrition details

* **Test Steps:**

1. Launch the app and navigate to the nutrition tracking section

2. Verify that the app calculates and displays the total macronutrient intake (carbs, fats, proteins)

3. Test different scenarios, such as:

   * User logs a meal with high carbs and high fats

   * User logs a snack with low protein

   * User logs a meal with no fats

4. Verify that the app accurately calculates and displays the macronutrient intake

* **Postconditions:** User's macronutrient intake is accurately calculated

* **Expected Result:** The app correctly calculates and displays the macronutrient intake

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can set nutritional goals and track progress**

* **Test Case ID:** NT-003

* **Test Case Title:** Verify that users can set nutritional goals and track progress

* **Test Suite:** Nutrition Tracking

* **Test Priority:** Medium

* **Preconditions:**

 + User has logged meals and snacks

 + User has access to the nutrition tracking feature

* **Test Data:** User's nutrition goal data

* **Test Steps:**

 1. Launch the app and navigate to the nutrition tracking section

 2. Click on the "Set Nutrition Goal" button and enter a valid goal (e.g., "Reduce daily carb intake by 20%")

 3. Verify that the user's progress is tracked and displayed

 4. Test different scenarios, such as:

  * User sets a goal and logs meals and snacks to track progress

  * User sets multiple goals and verifies that the app tracks progress for each goal

 5. Verify that the app accurately tracks and displays the user's progress

* **Postconditions:** User's nutrition progress is accurately tracked

* **Expected Result:** The app correctly tracks and displays the user's nutrition progress

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that the app provides nutritional information and suggestions**

* **Test Case ID:** NT-004

* **Test Case Title:** Verify that the app provides nutritional information and suggestions

* **Test Suite:** Nutrition Tracking

* **Test Priority:** Medium

* **Preconditions:**

 + User has logged meals and snacks

* **Test Data:** User's meal and snack data with nutrition details

* **Test Steps:**

 1. Launch the app and navigate to the nutrition tracking section

 2. Click on the "Nutritional Information" button and verify that the app provides accurate and helpful information

 3. Test different scenarios, such as:

  * User logs a meal with high sodium content and the app suggests a healthy alternative

  * User logs a snack with low fiber content and the app suggests a fiber-rich snack

 4. Verify that the app provides personalized nutritional suggestions based on the user's diet and preferences

* **Postconditions:** User receives personalized nutritional suggestions

* **Expected Result:** The app provides accurate and helpful nutritional information and suggestions

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that users can view their nutritional history and trends**

* **Test Case ID:** NT-005
* **Test Case Title:** Verify that users can view their nutritional history and trends
* **Test Suite:** Nutrition Tracking
* **Test Priority:** Low
* **Preconditions:**
 + User has logged multiple meals and snacks
* **Test Data:** User's meal and snack history data
* **Test Steps:**
 1. Launch the app and navigate to the nutrition tracking section
 2. Click on the "View Nutritional History" button and verify that the app displays the user's historical nutrient intake
 3. Test different scenarios, such as:
  * User views their nutrient intake over the past week
  * User views their nutrient intake compared to their goals
 4. Verify that the app displays the user's nutritional history and trends accurately
* **Postconditions:** User's nutritional history is accurately displayed
* **Expected Result:** The app correctly displays the user's nutritional history and trends
* **Severity:** Minor
* **Priority:** Low
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

---

**Data - 88**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Progress Reports

**Feature description -**

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

**Userstory title -**

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

**Acceptance criteria -**

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

**Generated Test Cases:**

Here are five detailed test cases for the Progress Reports feature, each thoroughly covering one or more of the acceptance criteria:

**Test Case 1**

* **Test Case ID:** FRP-001

* **Test Case Title:** Verify that users receive weekly progress reports.

* **Test Case Description:** This test case ensures that users receive their weekly progress reports as expected, including activity levels, workout performance, and goal attainment.

* **Test Suite:** Progress Reports

* **Test Priority:** High

* **Preconditions:**

 + User has a registered account and set fitness goals.

 + User has tracked activity for the current week.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Navigate to the Progress Reports section.

 2. Verify that the user's weekly progress report is displayed.

 3. Check that the report includes activity levels, workout performance, and goal attainment.

 4. Verify that the report is updated with the user's current progress.

* **Postconditions:** None

* **Expected Result:** The test case ensures that users receive their weekly progress reports, including activity levels, workout performance, and goal attainment.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2**

* **Test Case ID:** FRP-002

* **Test Case Title:** Verify that reports include activity levels, workout performance, and goal attainment.

* **Test Case Description:** This test case ensures that the weekly progress report includes all expected details, such as activity levels, workout performance, and goal attainment.

* **Test Suite:** Progress Reports

* **Test Priority:** Medium

* **Preconditions:**

 + User has a registered account and set fitness goals.

 + User has tracked activity for the current week.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Navigate to the Progress Reports section.

 2. Verify that the weekly progress report includes activity levels, workout performance, and goal attainment.

 3. Check that the report accurately reflects the user's performance.

* **Postconditions:** None

* **Expected Result:** The test case ensures that the weekly progress report includes all expected details.

* **Severity:** Critical

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3**

* **Test Case ID:** FRP-003

* **Test Case Title:** Verify that users can view detailed breakdowns of achievements and trends.

* **Test Case Description:** This test case ensures that users can view detailed information about their achievements and trends, including insights on areas for improvement.

* **Test Suite:** Progress Reports

* **Test Priority:** Medium

* **Preconditions:**

 + User has a registered account and set fitness goals.

 + User has tracked activity for the current week.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Navigate to the Progress Reports section.

 2. Select a specific report (e.g., weekly or monthly).

 3. Verify that the report includes detailed breakdowns of achievements and trends.

 4. Check that the report highlights areas for improvement.

* **Postconditions:** None

* **Expected Result:** The test case ensures that users can view detailed breakdowns of achievements and trends.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4**

* **Test Case ID:** FRP-004

* **Test Case Title:** Verify that the app provides recommendations based on report findings.

* **Test Case Description:** This test case ensures that users receive personalized recommendations based on their progress reports.

* **Test Suite:** Progress Reports

* **Test Priority:** High

* **Preconditions:**

 + User has a registered account and set fitness goals.

 + User has tracked activity for the current week.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Navigate to the Progress Reports section.

 2. Select a specific report (e.g., weekly or monthly).

 3. Verify that the report includes personalized recommendations.

 4. Check that the recommendations are based on the user's progress.

* **Postconditions:** None

* **Expected Result:** The test case ensures that users receive personalized recommendations based on their progress reports.

* **Severity:** Blocker

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5**

* **Test Case ID:** FRP-005

* **Test Case Title:** Verify that users can customize the frequency and content of reports.

* **Test Case Description:** This test case ensures that users can adjust the frequency and content of their progress reports to suit their preferences.

* **Test Suite:** Progress Reports

* **Test Priority:** Low

* **Preconditions:**

 + User has a registered account and set fitness goals.

+ User has tracked activity for the current week.

* **Test Data:** No test data needed

* **Test Steps:**

1. Navigate to the Progress Reports section.

2. Select the report settings.

3. Verify that users can adjust the frequency and content of reports.

4. Check that the changes are saved and reflected in the reports.

* **Postconditions:** None

* **Expected Result:** The test case ensures that users can customize the frequency and content of their progress reports.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 89**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

In-App Challenges

**Feature description -**

The app offers in-app challenges and competitions, encouraging users to set personal records, compete with friends, and earn rewards for achieving fitness milestones. In-app challenges add an element of fun, motivation, and friendly competition to the fitness tracking experience.

**Userstory title -**

As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals.

**Acceptance criteria -**

['Users can join various in-app challenges.', 'Challenges have clear objectives and rules.', 'Users can track their progress within the challenge.', 'Rewards are provided for achieving challenge milestones.', 'Users can compete with friends and view leaderboards.']

**Generated Test Cases:**

Here are five detailed test cases that cover the acceptance criteria for the in-app challenges feature:

**Test Case 1: Join an In-App Challenge**

* **Test Case ID:** CTCH-001
* **Test Case Title:** Verify that users can join various in-app challenges.
* **Test Case Description:** As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals. This test case ensures that users can join various in-app challenges with clear objectives and rules.
* **Test Suite:** In-App Challenges
* **Test Priority:** High
* **Preconditions:**
 + User has a registered account
 + User has access to the in-app challenges feature
* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the app with a registered account

 2. Navigate to the in-app challenges section

 3. Select a challenge to join

 4. Review the challenge objectives and rules

 5. Confirm joining the challenge

* **Postconditions:**

 + User is added to the challenge

 + Challenge objectives and rules are displayed

* **Expected Result:** User is successfully added to the challenge and can view challenge objectives and rules.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Challenge Objectives and Rules**

* **Test Case ID:** CTCH-002

* **Test Case Title:** Verify that challenges have clear objectives and rules.

* **Test Case Description:** This test case ensures that challenges have clear objectives and rules, making it easy for users to understand what is expected of them.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

 + User has joined a challenge

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the app with a registered account

 2. Navigate to the in-app challenges section

 3. Select a challenge to view objectives and rules

 4. Review the challenge objectives and rules

 5. Verify that objectives and rules are clear and concise

* **Postconditions:**

 + Challenge objectives and rules are displayed

* **Expected Result:** Challenge objectives and rules are clear, concise, and easily understandable.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Tracking Progress within a Challenge**

* **Test Case ID:** CTCH-003

* **Test Case Title:** Verify that users can track their progress within the challenge.

* **Test Case Description:** This test case ensures that users can track their progress within the challenge, helping them stay motivated and on track.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

 + User has joined a challenge

 + User has completed some progress within the challenge

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a registered account

2. Navigate to the in-app challenges section

3. Select a challenge to view progress

4. Review the user's progress within the challenge

5. Verify that progress is accurately displayed

* **Postconditions:**

 + User's progress is displayed

* **Expected Result:** User's progress is accurately displayed, helping them stay motivated and on track.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Rewards for Achieving Challenge Milestones**

* **Test Case ID:** CTCH-004

* **Test Case Title:** Verify that rewards are provided for achieving challenge milestones.

* **Test Case Description:** This test case ensures that rewards are provided for achieving challenge milestones, motivating users to continue striving for their fitness goals.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

 + User has joined a challenge

 + User has achieved a challenge milestone

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a registered account

2. Navigate to the in-app challenges section

3. Select a challenge to view rewards

4. Review the rewards provided for achieving the milestone

5. Verify that rewards are accurate and awarded

* **Postconditions:**

 + Rewards are awarded

* **Expected Result:** Rewards are accurately provided for achieving challenge milestones, motivating users to continue striving for their fitness goals.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Competing with Friends and Viewing Leaderboards**

* **Test Case ID:** CTCH-005

* **Test Case Title:** Verify that users can compete with friends and view leaderboards.

* **Test Case Description:** This test case ensures that users can compete with friends and view leaderboards, fostering a sense of friendly competition and accountability.

* **Test Suite:** In-App Challenges

* **Test Priority:** High

* **Preconditions:**

 + User has joined a challenge

 + User has friends connected through the app

* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app with a registered account

2. Navigate to the in-app challenges section

3. Select a challenge to view leaderboards

4. Review the leaderboards and compare with friends

5. Verify that leaderboards are accurate and up-to-date

* **Postconditions:**

 + Leaderboards are displayed

* **Expected Result:** Leaderboards are accurately displayed, allowing users to compete with friends and stay motivated.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 90**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Customizable Dashboards

**Feature description -**

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

**Userstory title -**

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

**Acceptance criteria -**

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

**Generated Test Cases:**

Here are five detailed test cases for the "Customizable Dashboards" feature of the Fitness Tracker App, each covering one or more of the acceptance criteria:

**Test Case 1: Verify that users can select which metrics to display on their dashboard**

* **Test Case ID:** TC001
* **Test Case Title:** Verify that users can select which metrics to display on their dashboard
* **Test Case Description:** This test case validates that users can choose which fitness metrics to display on their dashboard, ensuring that the app provides a personalized user experience.
* **Test Suite:** Customizable Dashboards
* **Test Priority:** High
* **Preconditions:**
 + User logged in to the app
 + User has access to the settings section
* **Test Data:** No test data needed
* **Test Steps:**

1. Launch the app and navigate to the dashboard
2. Click on the "Edit" button to access dashboard settings
3. Select multiple metrics to display on the dashboard (e.g., steps taken, calories burned, distance traveled)
4. Save the changes to update the dashboard
5. Verify that the selected metrics are displayed on the dashboard

* **Postconditions:** No expected postconditions
* **Expected Result:** The selected metrics are displayed accurately on the dashboard.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that the app allows customization of dashboard layout and theme**

* **Test Case ID:** TC002
* **Test Case Title:** Verify that the app allows customization of dashboard layout and theme
* **Test Case Description:** This test case ensures that users can customize the layout and theme of their dashboard to suit their preferences.
* **Test Suite:** Customizable Dashboards
* **Test Priority:** High
* **Preconditions:**
  + User logged in to the app
  + User has access to the settings section
* **Test Data:** No test data needed
* **Test Steps:**

1. Launch the app and navigate to the dashboard
2. Click on the "Edit" button to access dashboard settings

 3. Select a new layout template

 4. Customize the theme settings (e.g., color scheme, font style)

 5. Save the changes to update the dashboard

 6. Verify that the new layout and theme are applied to the dashboard

* **Postconditions:** No expected postconditions

* **Expected Result:** The dashboard layout and theme have been updated correctly.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can save and update their dashboard settings**

* **Test Case ID:** TC003

* **Test Case Title:** Verify that users can save and update their dashboard settings

* **Test Case Description:** This test case validates that users can save their dashboard settings and make updates without losing their previous configurations.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

 + User logged in to the app

 + User has access to the settings section

* **Test Data:** No test data needed

* **Test Steps:**

 1. Launch the app and navigate to the dashboard

 2. Make changes to the dashboard settings (e.g., add/remove metrics, customize layout)

 3. Save the changes to update the dashboard

 4. Update the dashboard settings again (e.g., add/removes new metric)
 5. Verify that the changes are saved successfully and the dashboard is updated correctly
* **Postconditions:** No expected postconditions
* **Expected Result:** The dashboard settings are saved correctly, and subsequent updates are applied successfully.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that dashboard displays are dynamic and update in real-time**

* **Test Case ID:** TC004
* **Test Case Title:** Verify that dashboard displays are dynamic and update in real-time
* **Test Case Description:** This test case ensures that the dashboard displays are updated in real-time, reflecting changes to the user's progress and metrics.
* **Test Suite:** Customizable Dashboards
* **Test Priority:** High
* **Preconditions:**
 + User logged in to the app
 + User has a valid workout or activity logged
* **Test Data:** No test data needed
* **Test Steps:**
 1. Launch the app and navigate to the dashboard
 2. Record a workout or activity
 3. Verify that the dashboard is updated in real-time to reflect the new data
 4. Repeat steps 1-3 to verify that the dashboard updates accurately

* **Postconditions:** No expected postconditions

* **Expected Result:** The dashboard displays are updated in real-time to reflect changes in the user's progress and metrics.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that users can reset the dashboard to default settings**

* **Test Case ID:** TC005

* **Test Case Title:** Verify that users can reset the dashboard to default settings

* **Test Case Description:** This test case ensures that users can reset their dashboard settings to the default configuration if needed.

* **Test Suite:** Customizable Dashboards

* **Test Priority:** High

* **Preconditions:**

 + User logged in to the app

 + User has customized dashboard settings

* **Test Data:** No test data needed

* **Test Steps:**

 1. Launch the app and navigate to the dashboard

 2. Click on the "Edit" button to access dashboard settings

 3. Select the option to reset dashboard settings to default

 4. Confirm the reset action

 5. Verify that the dashboard settings have been reset to the default configuration

* **Postconditions:** No expected postconditions

* **Expected Result:** The dashboard settings are reset to the default configuration.

* **Severity:** Minor

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

------------------------------------------------------------------------------------------------

**Data - 91**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Social Sharing

**Feature description -**

Users can share workout summaries, achievements, and progress updates on social media platforms, inspiring others and fostering accountability. Social sharing enables users to celebrate their successes, seek support from their social networks, and stay motivated on their fitness journey.

**Userstory title -**

As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

**Acceptance criteria -**

['Users can share workout summaries and achievements.', 'The app supports sharing on multiple social media platforms.', 'Users can customize the content and format of shared posts.', 'Sharing features are integrated seamlessly with social media.', 'Users can track the engagement of their shared posts.']

**Generated Test Cases:**

Here are the five detailed test cases that cover the specified acceptance criteria for the Social Sharing feature of the Fitness Tracker App:

**Test Case 1: Verify that users can share workout summaries and achievements**

Test Case ID: SSHA-001

Test Case Title: Verify that users can share workout summaries and achievements

Test Case Description: This test case ensures that users can share their workout summaries and achievements on social media platforms, capturing milestones and progress updates.

Test Suite: Social Sharing Module

Test Priority: High

Preconditions: User is logged in, has completed a workout, and selected the social sharing option.

Test Data: User's workout summary and achievement data.

Test Steps:

1. Log in to the app and complete a workout.
2. Click on the social sharing option and select a social media platform (e.g., Facebook, Twitter, Instagram).
3. Review and confirm the shared post content, including workout summary and achievement details.
4. Verify that the shared post is published successfully.

Postconditions: The user's shared post is visible on the selected social media platform.

Expected Result: The user can successfully share their workout summary and achievement on social media.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Verify that the app supports sharing on multiple social media platforms**

Test Case ID: SSHA-002

Test Case Title: Verify that the app supports sharing on multiple social media platforms

Test Case Description: This test case ensures that the app allows users to share content on various social media platforms, ensuring seamless integration.

Test Suite: Social Sharing Module

Test Priority: High

Preconditions: User is logged in and has completed a workout.

Test Data: User's workout summary and achievement data.

Test Steps:

1. Log in to the app and complete a workout.
2. Click on the social sharing option and select multiple social media platforms (e.g., Facebook, Twitter, Instagram, LinkedIn).
3. Review and confirm the shared post content, including workout summary and achievement details.
4. Verify that the shared post is published successfully on each selected platform.

Postconditions: The user's shared post is visible on multiple social media platforms.

Expected Result: The user can successfully share their workout summary and achievement on multiple social media platforms.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Verify that users can customize the content and format of shared posts**

Test Case ID: SSHA-003

Test Case Title: Verify that users can customize the content and format of shared posts

Test Case Description: This test case ensures that users can personalize the content and format of their shared posts, offering flexibility and creative freedom.

Test Suite: Social Sharing Module

Test Priority: High

Preconditions: User is logged in and has completed a workout.

Test Data: User's custom workout summary and achievement data.

Test Steps:

1. Log in to the app and complete a workout.
2. Click on the social sharing option and select a social media platform.
3. Configure the shared post content, including customizing the format (e.g., image, video, text only).
4. Verify that the custom content and format are successfully shared on the selected platform.

Postconditions: The user's custom shared post is visible on the selected social media platform.

Expected Result: Users can customize the content and format of their shared posts on social media.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Verify that sharing features are integrated seamlessly with social media**

Test Case ID: SSHA-004

Test Case Title: Verify that sharing features are integrated seamlessly with social media

Test Case Description: This test case ensures that the social sharing features integrate smoothly with social media platforms, providing a seamless user experience.

Test Suite: Social Sharing Module

Test Priority: Medium

Preconditions: User is logged in and has completed a workout.

Test Data: User's workout summary and achievement data.

Test Steps:

1. Log in to the app and complete a workout.
2. Click on the social sharing option and select a social media platform.
3. Verify that the shared post is automatically configured for the selected platform (e.g., aspect ratio, formatting).
4. Publish the shared post and verify that it is visible on the platform.

Postconditions: The shared post is visible on the social media platform.

Expected Result: Sharing features integrate seamlessly with social media platforms.

Severity: Minor

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Verify that users can track the engagement of their shared posts**

Test Case ID: SSHA-005

Test Case Title: Verify that users can track the engagement of their shared posts

Test Case Description: This test case ensures that users can monitor and respond to engagement (e.g., likes, comments, shares) with their shared posts, facilitating community interaction.

Test Suite: Social Sharing Module

Test Priority: High

Preconditions: User is logged in and has shared a post on social media.

Test Data: User's engagement data (e.g., likes, comments, shares).

Test Steps:

1. Log in to the app and view the user's shared post on social media.
2. Verify that engagement metrics (e.g., likes, comments, shares) are displayed for the shared post.
3. Respond to engagement (e.g., like, comment, share) to verify that interaction is tracked correctly.

Postconditions: The user can track engagement on their shared post.

Expected Result: Users can view and respond to engagement metrics on their shared posts.

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

These test cases cover the entire scope of the Social Sharing feature, ensuring that all acceptance criteria are thoroughly tested.

---

**Data - 92**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Rest and Recovery Tracking

**Feature description -**

The app includes features for tracking rest days, recovery activities, and recovery metrics such as resting heart rate and sleep quality. Rest and recovery tracking helps users prioritize recovery, prevent overtraining, and optimize their overall health and performance.

**Userstory title -**

As a user, I want to track my rest and recovery so that I can optimize my fitness performance and prevent overtraining.

**Acceptance criteria -**

['Users can log rest days and recovery activities.', 'The app tracks metrics such as resting heart rate and sleep quality.', 'Users receive recommendations for recovery based on their activities.', 'The app provides alerts for insufficient recovery time.', 'Users can view trends and history of their recovery data.']

**Generated Test Cases:**

Here are five test cases for the "Rest and Recovery Tracking" feature, following the provided template and guidelines:

**Test Case 1: Logging Rest Days**

* **Test Case ID:** RRT-001
* **Test Case Title:** Verify that the app allows users to log rest days and recovery activities.
* **Test Case Description:** This test case checks if the user can successfully log a rest day and recovery activity in the app, and if the app tracks the corresponding metrics.
* **Test Suite:** Rest and Recovery Tracking Module

* **Test Priority:** High

* **Preconditions:**

 + The user has a valid account in the app.

 + The user has enabled the "Rest and Recovery Tracking" feature.

* **Test Data:** No test data needed.

* **Test Steps:**

 1. Login to the app with a valid account.

 2. Enable the "Rest and Recovery Tracking" feature from the settings menu.

 3. Navigate to the "Rest and Recovery" tab.

 4. Tap on the "Log Rest Day" button.

 5. Select a rest day type (e.g., complete rest day).

 6. Enter a description for the rest day.

 7. Save the rest day entry.

* **Postconditions:**

 + The rest day is saved successfully.

 + The corresponding metrics (e.g., resting heart rate, sleep quality) are tracked.

* **Expected Result:** The user can successfully log a rest day and recovery activity in the app, and the app tracks the corresponding metrics.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Tracking Resting Heart Rate and Sleep Quality**

* **Test Case ID:** RRT-002

* **Test Case Title:** Verify that the app tracks metrics such as resting heart rate and sleep quality for rest

days and recovery activities.

* **Test Case Description:** This test case checks if the app accurately tracks resting heart rate and sleep quality for rest days and recovery activities.

* **Test Suite:** Rest and Recovery Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

 + The user has a valid account in the app.

 + The user has enabled the "Rest and Recovery Tracking" feature.

* **Test Data:** No test data needed.

* **Test Steps:**

 1. Login to the app with a valid account.

 2. Enable the "Rest and Recovery Tracking" feature from the settings menu.

 3. Navigate to the "Rest and Recovery" tab.

 4. Tap on the "Log Rest Day" button.

 5. Select a rest day type (e.g., complete rest day).

 6. Record a resting heart rate and sleep quality value.

 7. Save the rest day entry.

* **Postconditions:**

 + The resting heart rate and sleep quality values are tracked successfully.

 + The corresponding metrics are displayed correctly.

* **Expected Result:** The app accurately tracks resting heart rate and sleep quality for rest days and recovery activities.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Recovery Recommendations**

* **Test Case ID:** RRT-003
* **Test Case Title:** Verify that the app provides recommendations for recovery based on user activities.
* **Test Case Description:** This test case checks if the app provides accurate and relevant recovery recommendations based on the user's activities.
* **Test Suite:** Rest and Recovery Tracking Module
* **Test Priority:** High
* **Preconditions:**
 + The user has a valid account in the app.
 + The user has enabled the "Rest and Recovery Tracking" feature.
 + The user has logged rest days and recovery activities.
* **Test Data:** No test data needed.
* **Test Steps:**
 1. Login to the app with a valid account.
 2. Enable the "Rest and Recovery Tracking" feature from the settings menu.
 3. Navigate to the "Rest and Recovery" tab.
 4. View the recovery recommendations section.
 5. Tap on the "Get Recommendations" button.
 6. Review the recovery recommendations displayed.
* **Postconditions:**
 + The app provides accurate and relevant recovery recommendations.
 + The recommendations are based on the user's activities.
* **Expected Result:** The app provides accurate and relevant recovery recommendations based on the user's activities.
* **Severity:** Critical
* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Insufficient Recovery Time Alerts**

* **Test Case ID:** RRT-004

* **Test Case Title:** Verify that the app provides alerts for insufficient recovery time.

* **Test Case Description:** This test case checks if the app correctly alerts the user when there is insufficient recovery time.

* **Test Suite:** Rest and Recovery Tracking Module

* **Test Priority:** Medium

* **Preconditions:**

 + The user has a valid account in the app.

 + The user has enabled the "Rest and Recovery Tracking" feature.

 + The user has logged rest days and recovery activities.

* **Test Data:** No test data needed.

* **Test Steps:**

 1. Login to the app with a valid account.

 2. Enable the "Rest and Recovery Tracking" feature from the settings menu.

 3. Navigate to the "Rest and Recovery" tab.

 4. Set an insufficient recovery time interval.

 5. View the app's notifications.

* **Postconditions:**

 + The app alerts the user for insufficient recovery time.

 + The alerts are displayed correctly.

* **Expected Result:** The app provides accurate and timely alerts for insufficient recovery time.

* **Severity:** Major

* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Viewing Recovery Trends and History**

* **Test Case ID:** RRT-005
* **Test Case Title:** Verify that the app allows users to view trends and history of their recovery data.
* **Test Case Description:** This test case checks if the app correctly displays recovery trends and history for the user.
* **Test Suite:** Rest and Recovery Tracking Module
* **Test Priority:** Medium
* **Preconditions:**
 + The user has a valid account in the app.
 + The user has enabled the "Rest and Recovery Tracking" feature.
 + The user has logged rest days and recovery activities.
* **Test Data:** No test data needed.
* **Test Steps:**
 1. Login to the app with a valid account.
 2. Enable the "Rest and Recovery Tracking" feature from the settings menu.
 3. Navigate to the "Rest and Recovery" tab.
 4. View the recovery trends and history section.
 5. Review the displayed data.
* **Postconditions:**
 + The app displays the recovery trends and history correctly.
 + The data is up-to-date and accurate.
* **Expected Result:** The app allows users to view trends and history of their recovery data.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases cover all the acceptance criteria specified in the user story and ensure that each criterion is thoroughly tested.

---

## Data - 93

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Audio Coaching

**Feature description -**

Users can access audio-guided workouts, coaching cues, and motivational messages during their workouts, enhancing engagement and adherence. Audio coaching provides real-time feedback, encouragement, and guidance, simulating the experience of having a personal trainer or workout buddy.

**Userstory title -**

As a user, I want to access audio coaching during workouts so that I can stay motivated and receive real-time guidance.

**Acceptance criteria -**

['Users can access a variety of audio-guided workouts.', 'Audio coaching includes motivational messages and coaching cues.', 'Users can choose coaching based on workout type and intensity.', 'Audio coaching integrates seamlessly with other app features.', 'Users can control audio settings and preferences.']

**Generated Test Cases:**

Here are 5 detailed test cases for the Audio Coaching feature, following the Test Case Creation Template:

**Test Case 1: Verify that users can access a variety of audio-guided workouts**

* **Test Case ID:** FTAUD-001
* **Test Case Title:** Verify that users can access a variety of audio-guided workouts
* **Test Case Description:** As a user, I want to access various audio-guided workout options, ensuring that I can choose the right exercises and intensity levels to suit my fitness goals.
* **Test Suite:** Audio Coaching
* **Test Priority:** High
* **Preconditions:**
 + The user has logged in to the app.
 + The user has accessed the Audio Coaching feature.
 + The user has selected a workout type (e.g., strength training, cardio, yoga).
* **Test Data:** No test data needed
* **Test Steps:**
 1. Access the Audio Coaching feature by clicking on the "Workout" tab.
 2. Select a workout type (e.g., strength training, cardio).
 3. Choose a specific audio-guided workout from the list.
 4. Verify that the audio coach's voice and cues are displayed on the screen.
 5. Engage with the workout by following the audio coach's instructions.

* **Postconditions:**

 + Verify that the user can view their workout progress and achievements.

 + Verify that the user can log their workout and save it for future reference.

* **Expected Result:** The user can access various audio-guided workout options, including different types and intensity levels, and engage with the workout.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that audio coaching includes motivational messages and coaching cues**

* **Test Case ID:** FTAUD-002

* **Test Case Title:** Verify that audio coaching includes motivational messages and coaching cues

* **Test Case Description:** As a user, I want to receive motivational messages and coaching cues during my workout to stay motivated and focused.

* **Test Suite:** Audio Coaching

* **Test Priority:** High

* **Preconditions:**

 + The user has accessed the Audio Coaching feature.

 + The user has selected a workout type.

 + The audio coach is live and guiding the user through the workout.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Access the Audio Coaching feature and select a workout type.

 2. Engage with the workout and verify that motivational messages and coaching cues are displayed on the screen.

3. Check if the audio coach's voice is providing encouragement and guidance throughout the workout.

4. Document any motivational messages and coaching cues received during the workout.

* **Postconditions:**

+ Verify that the user feels motivated and engaged during the workout.

+ Verify that the user can view any motivational messages and coaching cues received during the workout.

* **Expected Result:** Audio coaching includes motivational messages and coaching cues to motivate and engage users during their workouts.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can choose coaching based on workout type and intensity**

* **Test Case ID:** FTAUD-003

* **Test Case Title:** Verify that users can choose coaching based on workout type and intensity

* **Test Case Description:** As a user, I want to choose coaching that is tailored to my workout type and intensity level, ensuring that I receive the right guidance and motivation.

* **Test Suite:** Audio Coaching

* **Test Priority:** High

* **Preconditions:**

+ The user has accessed the Audio Coaching feature.

+ The user has selected a workout type.

+ The user has chosen a specific audio-guided workout.

* **Test Data:** No test data needed

* **Test Steps:**

1. Access the Audio Coaching feature and select a workout type (e.g., strength training, cardio).

2. Choose a specific audio-guided workout based on the workout type and intensity level.

3. Verify that the audio coach's voice and cues are tailored to the chosen workout type and intensity level.

4. Engage with the workout and verify that the audio coach provides guidance and motivation suitable for the chosen workout type and intensity level.

* **Postconditions:**

 + Verify that the user feels that the coaching is tailored to their workout type and intensity level.

 + Verify that the user can view their workout progress and achievements.

* **Expected Result:** Users can choose coaching based on workout type and intensity, ensuring that they receive the right guidance and motivation.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that audio coaching integrates seamlessly with other app features**

* **Test Case ID:** FTAUD-004

* **Test Case Title:** Verify that audio coaching integrates seamlessly with other app features

* **Test Case Description:** As a user, I want audio coaching to integrate seamlessly with other app features, such as workout logging and progress tracking, to provide a comprehensive fitness experience.

* **Test Suite:** Audio Coaching

* **Test Priority:** High

* **Preconditions:**

 + The user has accessed the Audio Coaching feature.

 + The user has logged a workout.

 + The user has accessed the workout logging feature.

* **Test Data:** No test data needed

* **Test Steps:**
 1. Access the Audio Coaching feature and engage with a workout.
 2. Log the workout using the workout logging feature.
 3. Verify that the workout is displayed in the workout logging feature.
 4. Verify that the user can view their workout progress and achievements.
* **Postconditions:**
 + Verify that the user can view their workout progress and achievements.
 + Verify that the user can log their workout and save it for future reference.
* **Expected Result:** Audio coaching integrates seamlessly with other app features, such as workout logging and progress tracking.
* **Severity:** Major
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that users can control audio settings and preferences**

* **Test Case ID:** FTAUD-005
* **Test Case Title:** Verify that users can control audio settings and preferences
* **Test Case Description:** As a user, I want to control audio settings and preferences, such as volume and coaching tone, to ensure that I have a personalized and comfortable fitness experience.
* **Test Suite:** Audio Coaching
* **Test Priority:** Medium
* **Preconditions:**
 + The user has accessed the Audio Coaching feature.
 + The user has selected a workout.
* **Test Data:** No test data needed

* **Test Steps:**

1. Access the Audio Coaching feature and engage with a workout.
2. Adjust audio settings and preferences, such as volume and coaching tone.
3. Verify that the adjustments affect the audio coaching experience.
4. Re-engage with the workout and verify that the adjustments are applied correctly.

* **Postconditions:**

 + Verify that the user feels comfortable with the audio settings and preferences.

 + Verify that the user can adjust the audio settings and preferences as needed.

* **Expected Result:** Users can control audio settings and preferences to personalize their fitness experience.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 94**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Integration with Health Apps

**Feature description -**

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

**Userstory title -**

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

**Acceptance criteria -**

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

**Generated Test Cases:**

Here are the detailed test cases for the user story "As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being."

**Test Case 1: Integration with Popular Health and Wellness Apps**

* **Test Case ID:** TC_FIT-001
* **Test Case Title:** Verify that the fitness tracker app integrates with popular health and wellness apps.
* **Test Suite:** Integration with Health Apps
* **Test Priority:** High
* **Preconditions:**
 + The app is installed on a mobile device.
 + The device has a stable internet connection.
* **Test Data:** No test data needed

* **Test Steps:**

 1. Open the fitness tracker app on the mobile device.

 2. Navigate to the "Integrations" section.

 3. Click on the "Connect" button next to a popular health and wellness app (e.g., MyFitnessPal, Strava, or Headspace).

 4. Follow the on-screen instructions to authorize the app.

 5. Verify that the app integrates with the connected health and wellness app.

* **Postconditions:**

 + The app is still installed on the mobile device.

 + The device still has a stable internet connection.

* **Expected Result:** The fitness tracker app successfully integrates with the selected health and wellness app.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Integration Testing

* **Test Case Approach:** Positive

**Test Case 2: Synchronizing Data from Connected Health Apps**

* **Test Case ID:** TC_FIT-002

* **Test Case Title:** Verify that users can sync data from connected health apps.

* **Test Suite:** Integration with Health Apps

* **Test Priority:** High

* **Preconditions:**

 + The fitness tracker app is integrated with a health and wellness app.

 + The app has a stable internet connection.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Open the fitness tracker app on the mobile device.

 2. Navigate to the "Health Metrics" section.

 3. Click on the "Sync Data" button.

 4. Verify that the app syncs data from the connected health and wellness app.

 5. Check that the synced data is accurately reflected in the app.

* **Postconditions:**

 + The app is still installed on the mobile device.

 + The device still has a stable internet connection.

* **Expected Result:** The fitness tracker app successfully syncs data from the connected health and wellness app.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Consolidated View of Health Metrics**

* **Test Case ID:** TC_FIT-003

* **Test Case Title:** Verify that the app provides a consolidated view of health metrics.

* **Test Suite:** Integration with Health Apps

* **Test Priority:** High

* **Preconditions:**

 + The fitness tracker app is integrated with multiple health and wellness apps.

 + The app has a stable internet connection.

* **Test Data:** No test data needed

* **Test Steps:**

1. Open the fitness tracker app on the mobile device.

2. Navigate to the "Health Metrics" section.

3. Verify that the app provides a comprehensive view of health metrics from all connected health and wellness apps.

4. Check that the consolidated view accurately reflects the data from each connected app.

* **Postconditions:**

+ The app is still installed on the mobile device.

+ The device still has a stable internet connection.

* **Expected Result:** The fitness tracker app provides a consolidated and accurate view of health metrics.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Insights Based on Combined Data**

* **Test Case ID:** TC_FIT-004

* **Test Case Title:** Verify that users receive insights based on combined data from multiple apps.

* **Test Suite:** Integration with Health Apps

* **Test Priority:** Medium

* **Preconditions:**

+ The fitness tracker app is integrated with multiple health and wellness apps.

+ The app has a stable internet connection.

* **Test Data:** No test data needed

* **Test Steps:**

1. Open the fitness tracker app on the mobile device.

2. Navigate to the "Insights" section.

3. Verify that the app provides relevant insights based on the combined data from all connected health and wellness apps.

4. Check that the insights accurately reflect the trends and patterns in the data.

* **Postconditions:**

 + The app is still installed on the mobile device.

 + The device still has a stable internet connection.

* **Expected Result:** The fitness tracker app provides actionable insights based on combined data.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Easy Integration Settings Configuration**

* **Test Case ID:** TC_FIT-005

* **Test Case Title:** Verify that integration settings are easy to manage and configure.

* **Test Suite:** Integration with Health Apps

* **Test Priority:** Medium

* **Preconditions:**

 + The fitness tracker app is integrated with a health and wellness app.

 + The app has a stable internet connection.

* **Test Data:** No test data needed

* **Test Steps:**

1. Open the fitness tracker app on the mobile device.

2. Navigate to the "Integrations" section.

3. Verify that the app provides an intuitive interface for configuring integration settings.

4. Test adding and removing a health and wellness app connection.

5. Verify that the user can easily access their integration settings.

* **Postconditions:**

+ The app is still installed on the mobile device.

+ The device still has a stable internet connection.

* **Expected Result:** Integration settings are easy to configure and manage within the app.

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

These test cases ensure comprehensive coverage of the acceptance criteria specified in the user story, focusing on the integration features, data synchronization, consolidated views, insights, and ease of configuration.

---

**Data - 95**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Body Composition Tracking

**Feature description -**

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and

BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and the effectiveness of exercise and nutrition interventions.

**Userstory title -**

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed decisions about my training and nutrition.

**Acceptance criteria -**

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set goals for body composition metrics.']

**Generated Test Cases:**

Here are five detailed test cases that cover the acceptance criteria for the Body Composition Tracking feature in the Fitness Tracker App:

**Test Case 1: Verify that users can log body composition metrics**

* **Test Case ID:** TTC-001
* **Test Case Title:** Verify that users can log body composition metrics
* **Test Case Description:** As a user, I want to log body composition metrics such as weight, body fat percentage, and muscle mass, so that I can track changes over time.
* **Test Suite:** Body Composition Tracking
* **Test Priority:** High
* **Preconditions:**
 + User has a registered account in the app
 + User has granted access to their wearable device or fitness equipment
* **Test Data:** No test data needed

* **Test Steps:**

1. Log in to the app as a registered user

2. Navigate to the Body Composition Tracking feature

3. Select a body composition metric to log (e.g. weight, body fat percentage, muscle mass)

4. Enter a value for the selected metric

5. Submit the log entry

* **Postconditions:** Body composition metric is saved in the user's history

* **Expected Result:** Log entry is successful, and user can view their body composition metric in their history

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that the app calculates and displays BMI based on logged data**

* **Test Case ID:** TTC-002

* **Test Case Title:** Verify that the app calculates and displays BMI based on logged data

* **Test Case Description:** As a user, I want the app to calculate and display my BMI based on my logged body composition metrics, so that I can track changes over time.

* **Test Suite:** Body Composition Tracking

* **Test Priority:** High

* **Preconditions:**

+ User has logged body composition metrics (weight and height)

* **Test Data:** User's height and weight values

* **Test Steps:**

1. Log in to the app as a registered user

 2. Navigate to the Body Composition Tracking feature

 3. Ensure that user has logged body composition metrics (weight and height)

 4. View the user's BMI calculation

* **Postconditions:** BMI is displayed correctly in the user's history

* **Expected Result:** BMI is calculated correctly based on user's logged body composition metrics

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users can view trends and history of body composition changes**

* **Test Case ID:** TTC-003

* **Test Case Title:** Verify that users can view trends and history of body composition changes

* **Test Case Description:** As a user, I want to view trends and history of my body composition changes, so that I can track my progress and make informed decisions.

* **Test Suite:** Body Composition Tracking

* **Test Priority:** High

* **Preconditions:**

 + User has logged body composition metrics (weight, body fat percentage, muscle mass) over time

* **Test Data:** User's historical body composition data

* **Test Steps:**

 1. Log in to the app as a registered user

 2. Navigate to the Body Composition Tracking feature

 3. View the user's body composition history

 4. Verify that trends and history are displayed correctly

* **Postconditions:** User can view their body composition trends and history

* **Expected Result:** User can view their body composition trends and history correctly

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that the app provides insights and recommendations based on body composition data**

* **Test Case ID:** TTC-004

* **Test Case Title:** Verify that the app provides insights and recommendations based on body composition data

* **Test Case Description:** As a user, I want the app to provide insights and recommendations based on my body composition data, so that I can make informed decisions about my training and nutrition.

* **Test Suite:** Body Composition Tracking

* **Test Priority:** Medium

* **Preconditions:**

 + User has logged body composition metrics (weight, body fat percentage, muscle mass) over time

* **Test Data:** User's historical body composition data

* **Test Steps:**

 1. Log in to the app as a registered user

 2. Navigate to the Body Composition Tracking feature

 3. View the user's body composition insights and recommendations

 4. Verify that insights and recommendations are displayed correctly

* **Postconditions:** User receives insights and recommendations based on their body composition data

* **Expected Result:** User receives accurate and actionable insights and recommendations

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 5: Verify that users can set goals for body composition metrics**

* **Test Case ID:** TTC-005

* **Test Case Title:** Verify that users can set goals for body composition metrics

* **Test Case Description:** As a user, I want to set goals for body composition metrics, so that I can track my progress and stay motivated.

* **Test Suite:** Body Composition Tracking

* **Test Priority:** Medium

* **Preconditions:** None

* **Test Data:** None

* **Test Steps:**

1. Log in to the app as a registered user
2. Navigate to the Body Composition Tracking feature
3. Select a body composition metric to set a goal for (e.g. weight, body fat percentage, muscle mass)
4. Enter a goal value and select a target date
5. Submit the goal

* **Postconditions:** User's goal is saved in the app

* **Expected Result:** User can view their goal and track progress toward it

* **Severity:** Minor

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 96**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Workout Analysis

**Feature description -**

The app offers detailed workout analysis tools, including performance metrics, trends, and insights into exercise intensity, duration, and calorie burn. Workout analysis helps users assess the effectiveness of their workouts, identify areas for improvement, and optimize training strategies for better results.

**Userstory title -**

As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

**Acceptance criteria -**

['Users can view detailed performance metrics for each workout.', 'The app provides trends and insights into workout intensity and duration.', 'Users can track calorie burn and other health metrics.', 'The app offers recommendations based on workout analysis.', 'Users can compare current workouts with past performance.']

**Generated Test Cases:**

Here are the test cases for the "Workout Analysis" feature, each carefully crafted to ensure comprehensive coverage of the acceptance criteria:

**Test Case 1: View Detailed Performance Metrics for a Single Workout**

Test Case ID: V001

Test Case Title: Verify that users can view detailed performance metrics for each workout

Test Case Description: As a user, I want to view detailed performance metrics for a specific workout to assess my performance and make data-driven decisions for future workouts. This test case covers the acceptance criteria "Users can view detailed performance metrics for each workout."

Test Suite: Workout Analysis Module

Test Priority: High

Preconditions:

- The user has completed a workout with valid performance data
- The user has accessed the workout analysis module

Test Data: User ID, workout ID

Test Steps:

1. Access the workout analysis module and select the specific workout for analysis
2. Verify that the performance metrics for the selected workout are displayed accurately, including intensity, duration, calorie burn, and distance traveled
3. Validate that the displayed metrics are in sync with the expected values stored in the database

Postconditions: None

Expected Result: Detailed performance metrics for the selected workout are accurately displayed, including intensity, duration, calorie burn, and distance traveled

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 2: Display Workout Trends and Insights**

Test Case ID: T002

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration

Test Case Description: As a user, I want to see trends and insights into my workout intensity and duration over time to understand my progress and optimize my training strategies. This test case covers the acceptance criterion "The app provides trends and insights into workout intensity and duration."

Test Suite: Workout Analysis Module

Test Priority: Medium

Preconditions:

* The user has completed multiple workouts with valid performance data
* The user has accessed the workout analysis module

Test Data: User ID, multiple workout IDs

Test Steps:

1. Access the workout analysis module and select a period of time (e.g., week, month, quarter)
2. Verify that the trends and insights into workout intensity and duration are accurately displayed for the selected period
3. Validate that the displayed trends and insights are in sync with the expected values stored in the database

Postconditions: None

Expected Result: Trends and insights into workout intensity and duration are accurately displayed for the selected period

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 3: Track Calorie Burn and Other Health Metrics**

Test Case ID: T003

Test Case Title: Verify that users can track calorie burn and other health metrics

Test Case Description: As a user, I want to track calorie burn and other health metrics for each workout to understand the effectiveness of my exercises and optimize my nutrition. This test case covers the acceptance criterion "Users can track calorie burn and other health metrics."

Test Suite: Workout Analysis Module

Test Priority: High

Preconditions:

* The user has completed a workout with valid performance data
* The user has accessed the workout analysis module

Test Data: User ID, workout ID

Test Steps:

1. Access the workout analysis module and select a specific workout for analysis
2. Verify that the calorie burn and other health metrics (e.g., steps taken, distance traveled, heart rate) for the selected workout are accurately displayed
3. Validate that the displayed metrics are in sync with the expected values stored in the database

Postconditions: None

Expected Result: Calorie burn and other health metrics are accurately displayed for the selected workout

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 4: Workout Analysis Recommendations**

Test Case ID: R005

Test Case Title: Verify that the app offers recommendations based on workout analysis

Test Case Description: As a user, I want the app to provide actionable recommendations based on my workout analysis data to help me improve my performance and reach my fitness goals. This test case covers the acceptance criterion "The app offers recommendations based on workout analysis."

Test Suite: Workout Analysis Module

Test Priority: High

Preconditions:

* The user has completed multiple workouts with valid performance data
* The user has accessed the workout analysis module

Test Data: User ID, multiple workout IDs

Test Steps:

1. Access the workout analysis module and select a period of time (e.g., week, month, quarter)
2. Verify that the recommendations provided by the app are based on the workout analysis data for the selected period
3. Validate that the recommendations are accurate and relevant to the user's performance goals

Postconditions: None

Expected Result: Actionable recommendations based on workout analysis are provided by the app

Severity: Major

Priority: High

Type of Testing: Functional Testing

Test Case Approach: Positive

**Test Case 5: Compare Current Workouts with Past Performance**

Test Case ID: C006

Test Case Title: Verify that users can compare current workouts with past performance

Test Case Description: As a user, I want to compare my current workout performance with my past performance to track progress and make data-driven decisions. This test case covers the acceptance criterion

"Users can compare current workouts with past performance."

Test Suite: Workout Analysis Module

Test Priority: Medium

Preconditions:

- The user has completed multiple workouts with valid performance data
- The user has accessed the workout analysis module

Test Data: User ID, multiple workout IDs

Test Steps:

1. Access the workout analysis module and select a specific workout for comparison
2. Verify that the app displays the user's past performance data for comparison with the selected workout
3. Validate that the comparison is accurate and helpful for making informed decisions

Postconditions: None

Expected Result: Current workout performance is accurately compared with past performance data

Severity: Major

Priority: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Each of these test cases addresses one or more acceptance criteria for the "Workout Analysis" feature, ensuring comprehensive coverage of the feature's requirements. These test cases are designed to be thorough, positive, and high-priority to guarantee that the app provides accurate and actionable workout analysis insights for users.

---

**Data - 97**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Weather Integration

**Feature description -**

Integrated weather forecasts and conditions allow users to plan outdoor workouts and activities based on current weather conditions and forecasts. Weather integration helps users make informed decisions, stay safe, and optimize their outdoor exercise routines according to weather conditions.

**Userstory title -**

As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

**Acceptance criteria -**

['The app provides current weather conditions and forecasts.', 'Users can view weather details for different locations.', 'Weather data is updated in real-time.', 'The app offers recommendations based on weather conditions.', 'Users can receive weather alerts for extreme conditions.']

**Generated Test Cases:**

Here are five detailed test cases for the Weather Integration feature of the Fitness Tracker App, each covering one or more of the acceptance criteria:

**Test Case 1: Verify that the app provides current weather conditions and forecasts**

* **Test Case ID:** WTI-001

* **Test Case Title:** Verify that the app provides current weather conditions and forecasts

* **Test Case Description:** This test case ensures that the app displays accurate and up-to-date weather conditions and forecasts for the user's location.
* **Test Suite:** Weather Integration
* **Test Priority:** High
* **Preconditions:**
 + The user has logged in to the app.
 + The app has access to the user's location.
* **Test Data:** No test data needed
* **Test Steps:**
 1. Launch the app and navigate to the weather section.
 2. Verify that the app displays the current weather conditions (temperature, humidity, wind speed, etc.).
 3. Verify that the app displays a 5-day weather forecast.
 4. Verify that the forecast includes the high and low temperatures, precipitation chance, and wind speed for each day.
* **Postconditions:** None
* **Expected Result:** The app displays accurate and up-to-date weather conditions and forecasts for the user's location.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 2: Verify that users can view weather details for different locations**

* **Test Case ID:** WTI-002
* **Test Case Title:** Verify that users can view weather details for different locations
* **Test Case Description:** This test case ensures that users can view weather details for different locations,

including their current location and other locations they have saved.

* **Test Suite:** Weather Integration

* **Test Priority:** Medium

* **Preconditions:**

 + The user has logged in to the app.

 + The app has access to the user's location.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Launch the app and navigate to the weather section.

 2. Verify that the app displays the current weather conditions for the user's location.

 3. Save a new location (e.g., a city or landmark).

 4. Verify that the app displays the weather details for the saved location.

* **Postconditions:** None

* **Expected Result:** Users can view weather details for different locations.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that weather data is updated in real-time**

* **Test Case ID:** WTI-003

* **Test Case Title:** Verify that weather data is updated in real-time

* **Test Case Description:** This test case ensures that the app updates weather data in real-time, reflecting changes in weather conditions.

* **Test Suite:** Weather Integration

* **Test Priority:** High

* **Preconditions:**

 + The user has logged in to the app.

 + The app has access to the user's location.

* **Test Data:** No test data needed

* **Test Steps:**

 1. Launch the app and navigate to the weather section.

 2. Verify that the app displays the current weather conditions.

 3. Wait for a short period (e.g., 5-10 minutes).

 4. Verify that the app updates the weather conditions to reflect any changes.

* **Postconditions:** None

* **Expected Result:** Weather data is updated in real-time.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that the app offers recommendations based on weather conditions**

* **Test Case ID:** WTI-004

* **Test Case Title:** Verify that the app offers recommendations based on weather conditions

* **Test Case Description:** This test case ensures that the app provides recommendations for outdoor activities based on the current weather conditions.

* **Test Suite:** Weather Integration

* **Test Priority:** Medium

* **Preconditions:**

 + The user has logged in to the app.

 + The app has access to the user's location.

* **Test Data:** No test data needed
* **Test Steps:**
 1. Launch the app and navigate to the weather section.
 2. Verify that the app displays a recommendation for an outdoor activity based on the current weather conditions.
 3. Verify that the recommendation is relevant to the user's location and current weather conditions.
* **Postconditions:** None
* **Expected Result:** The app offers recommendations based on weather conditions.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that users can receive weather alerts for extreme conditions**

* **Test Case ID:** WTI-005
* **Test Case Title:** Verify that users can receive weather alerts for extreme conditions
* **Test Case Description:** This test case ensures that users can receive weather alerts for extreme conditions, such as thunderstorms or heavy rain.
* **Test Suite:** Weather Integration
* **Test Priority:** High
* **Preconditions:**
 + The user has logged in to the app.
 + The app has access to the user's location.
* **Test Data:** No test data needed
* **Test Steps:**
 1. Launch the app and navigate to the weather section.

2. Verify that the app displays a weather alert for extreme conditions (e.g., thunderstorm).

3. Verify that the app sends a notification to the user's device with the weather alert.

* **Postconditions:** None

* **Expected Result:** Users can receive weather alerts for extreme conditions.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 98**

**Project name -**

Fitness Tracker App

**Project description -**

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

**Feature name -**

Mood Tracking

**Feature description -**

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

**Userstory title -**

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

**Acceptance criteria -**

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and energy levels.']

**Generated Test Cases:**

Here are five detailed test cases for the Mood Tracking feature in the Fitness Tracker App, following the Test Case Creation Template:

**Test Case 1: Verify that users can log their mood and energy levels before and after workouts**

* **Test Case ID:** TC_Mood_Tracking_001
* **Test Case Title:** Verify that users can log their mood and energy levels before and after workouts
* **Test Case Description:** As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.
* **Test Suite:** Mood Tracking
* **Test Priority:** High
* **Preconditions:**
  + User is logged in and on the dashboard
  + User has created a workout routine
* **Test Data:** No test data needed
* **Test Steps:**
  1. Open the app and navigate to the "Workout Log" feature
  2. Tap on the "Log Before Workout" button
  3. Select a mood and energy level from the dropdown menu

 4. Tap on the "Log After Workout" button

 5. Enter a new mood and energy level

* **Postconditions:**

 + User's mood and energy levels are logged successfully

 + Workout log is updated with the new data

* **Expected Result:** The user can successfully log their mood and energy levels before and after workouts.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that the app provides insights into mood and energy trends over time**

* **Test Case ID:** TC_Mood_Tracking_002

* **Test Case Title:** Verify that the app provides insights into mood and energy trends over time

* **Test Case Description:** The app should provide insights into mood and energy trends over time to help users understand the impact of exercise on their mental well-being.

* **Test Suite:** Mood Tracking

* **Test Priority:** High

* **Preconditions:**

 + User has logged multiple workouts with mood and energy levels

 + User has viewed the workout log

* **Test Data:** No test data needed

* **Test Steps:**

 1. Open the app and navigate to the "Workout Log" feature

 2. Tap on the "Trends" button

 3. Select a time period (e.g., weekly, monthly)

4. View the mood and energy trend chart

* **Postconditions:**

+ User can view the mood and energy trend chart for the selected time period

+ Chart displays the correct data

* **Expected Result:** The app provides a clear and accurate view of mood and energy trends over time.

* **Severity:** Major

* **Priority:** Medium

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that users receive recommendations based on their mood and energy data**

* **Test Case ID:** TC_Mood_Tracking_003

* **Test Case Title:** Verify that users receive recommendations based on their mood and energy data

* **Test Case Description:** Users receive recommendations based on their mood and energy data to help them improve their mental well-being.

* **Test Suite:** Mood Tracking

* **Test Priority:** High

* **Preconditions:**

+ User has logged multiple workouts with mood and energy levels

+ User has viewed the workout log

* **Test Data:** No test data needed

* **Test Steps:**

1. Open the app and navigate to the "Dashboard" feature

2. Tap on the "Recommendations" button

3. View the recommended exercises and routines

* **Postconditions:**

 + User receives personalized recommendations based on their mood and energy data

 + Recommendations are relevant and actionable

* **Expected Result:** The user receives accurate and helpful recommendations based on their mood and energy data.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that the app supports journaling for personal reflections on mood and workouts**

* **Test Case ID:** TC_Mood_Tracking_004

* **Test Case Title:** Verify that the app supports journaling for personal reflections on mood and workouts

* **Test Case Description:** Users can journal their thoughts and feelings about their workouts, including mood and energy levels.

* **Test Suite:** Mood Tracking

* **Test Priority:** High

* **Preconditions:**

 + User is logged in and on the dashboard

 + User has created a workout routine

* **Test Data:** No test data needed

* **Test Steps:**

 1. Open the app and navigate to the "Journal" feature

 2. Tap on the "New Entry" button

 3. Enter a journal entry with thoughts and feelings about a workout

 4. Save the entry

* **Postconditions:**

 + User's journal entry is saved successfully
 + Entry is displayed in the journal log
* **Expected Result:** The user can successfully journal their thoughts and feelings about their workouts.
* **Severity:** Minor
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that users can set goals for improving mood and energy levels**

* **Test Case ID:** TC_Mood_Tracking_005
* **Test Case Title:** Verify that users can set goals for improving mood and energy levels
* **Test Case Description:** Users can set specific, measurable, achievable, relevant, and time-bound (SMART) goals for improving their mood and energy levels.
* **Test Suite:** Mood Tracking
* **Test Priority:** High
* **Preconditions:**
 + User is logged in and on the dashboard
 + User has viewed the workout log
* **Test Data:** No test data needed
* **Test Steps:**
 1. Open the app and navigate to the "Goals" feature
 2. Tap on the "New Goal" button
 3. Enter a specific goal for improving mood or energy
 4. Set a deadline for achieving the goal
 5. Save the goal
* **Postconditions:**

 + User's goal is saved successfully

 + Goal is displayed in the goals list

* **Expected Result:** The user can successfully set a goal for improving their mood or energy levels.

* **Severity:** Minor

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 99**

**Project name -**

Recipe Sharing Platform

**Project description -**

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

**Feature name -**

Recipe Creation

**Feature description -**

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

**Userstory title -**

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

**Acceptance criteria -**

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', "Recipes are saved in the user's profile.", 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

**Generated Test Cases:**

Here are five detailed test cases for the Recipe Creation feature, following the provided Test Case Creation Template:

**Test Case 1: Verify that users can input and save ingredients and cooking instructions**

* **Test Case ID:** RC-001
* **Test Case Title:** Verify that users can input and save ingredients and cooking instructions
* **Test Case Description:** This test case ensures that users can successfully input and save ingredients and cooking instructions for their recipes. It covers the acceptance criteria of users being able to input and save ingredients and cooking instructions.
* **Test Suite:** Recipe Creation
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the platform
 + User has a valid profile
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the platform as a valid user
 2. Navigate to the recipe creation page
 3. Input ingredients and cooking instructions for a new recipe
 4. Click the "Save" button to save the recipe
 5. Verify that the recipe is saved successfully

* **Postconditions:**

 + Recipe is saved in the user's profile

 + Recipe is accessible to other users on the platform

* **Expected Result:** The recipe is saved successfully, and the user can access it from their profile.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that users can upload photos for their recipes**

* **Test Case ID:** RC-002

* **Test Case Title:** Verify that users can upload photos for their recipes

* **Test Case Description:** This test case ensures that users can successfully upload photos for their recipes. It covers the acceptance criteria of users being able to upload photos for their recipes.

* **Test Suite:** Recipe Creation

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the platform

 + User has a valid profile

 + User has a valid image file to upload

* **Test Data:** Image file to upload

* **Test Steps:**

 1. Log in to the platform as a valid user

 2. Navigate to the recipe creation page

 3. Click the "Upload Photo" button

 4. Select the image file to upload

 5. Verify that the image is uploaded successfully

* **Postconditions:**

 + Image is uploaded to the user's recipe

 + Image is accessible to other users on the platform

* **Expected Result:** The image is uploaded successfully, and the user can access it from their recipe.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 3: Verify that recipes are saved in the user's profile**

* **Test Case ID:** RC-003

* **Test Case Title:** Verify that recipes are saved in the user's profile

* **Test Case Description:** This test case ensures that recipes are successfully saved in the user's profile. It covers the acceptance criteria of recipes being saved in the user's profile.

* **Test Suite:** Recipe Creation

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the platform

 + User has a valid profile

 + Recipe is saved by the user

* **Test Data:** No test data needed

* **Test Steps:**

 1. Log in to the platform as a valid user

 2. Navigate to the user's profile

 3. Verify that the saved recipe is listed in the user's profile

* **Postconditions:**
 + Recipe is listed in the user's profile
 + Recipe is accessible to other users on the platform
* **Expected Result:** The recipe is saved successfully in the user's profile.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 4: Verify that recipes are accessible to other users on the platform**

* **Test Case ID:** RC-004
* **Test Case Title:** Verify that recipes are accessible to other users on the platform
* **Test Case Description:** This test case ensures that recipes are successfully accessible to other users on the platform. It covers the acceptance criteria of recipes being accessible to other users on the platform.
* **Test Suite:** Recipe Creation
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the platform
 + User has a valid profile
 + Recipe is saved by another user
* **Test Data:** No test data needed
* **Test Steps:**
 1. Log in to the platform as a valid user
 2. Navigate to another user's profile
 3. Verify that the saved recipe is listed in the other user's profile
* **Postconditions:**

  + Recipe is listed in the other user's profile
  + Recipe is accessible to all users on the platform
* **Expected Result:** The recipe is accessible to other users on the platform.
* **Severity:** Critical
* **Priority:** High
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that users receive a confirmation upon successful recipe submission**

* **Test Case ID:** RC-005
* **Test Case Title:** Verify that users receive a confirmation upon successful recipe submission
* **Test Case Description:** This test case ensures that users receive a confirmation message upon successful recipe submission. It covers the acceptance criteria of users receiving a confirmation upon successful recipe submission.
* **Test Suite:** Recipe Creation
* **Test Priority:** High
* **Preconditions:**
  + User is logged in to the platform
  + User has a valid profile
  + Recipe is saved by the user
* **Test Data:** No test data needed
* **Test Steps:**
  1. Log in to the platform as a valid user
  2. Navigate to the recipe creation page
  3. Input ingredients and cooking instructions for a new recipe
  4. Click the "Save" button to save the recipe

5. Verify that a confirmation message is displayed

* **Postconditions:**

 + Confirmation message is displayed to the user

 + Recipe is saved successfully

* **Expected Result:** The user receives a confirmation message upon successful recipe submission.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---

**Data - 100**

**Project name -**

Recipe Sharing Platform

**Project description -**

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

**Feature name -**

Recipe Discovery

**Feature description -**

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

**Userstory title -**

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

**Acceptance criteria -**

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

**Generated Test Cases:**

Here are five detailed test cases to thoroughly cover the acceptance criteria for the Recipe Discovery feature:

**Test Case 1: Verify that users can search for recipes using keywords and ingredients**

* **Test Case ID:** RC-001
* **Test Case Title:** Verify that users can search for recipes using keywords and ingredients
* **Test Suite:** Recipe Discovery Module
* **Test Priority:** High
* **Preconditions:**
 + User is logged in to the platform
 + User has selected the desired cuisine type
* **Test Data:** Test keyword or ingredient data (e.g., "chicken", "pasta")
* **Test Steps:**
 1. Log in to the platform as a valid user
 2. Navigate to the search page and enter a keyword or ingredient in the search bar
 3. Click the "Search" button
 4. Verify that search results are displayed on the page
 5. Check the relevance of search results by examining the recipe titles, images, and descriptions
* **Postconditions:** Search results are displayed on the page, and the user can navigate to individual recipes

for more information.

* **Expected Result:** Search results are displayed on the page, and they accurately match the search criteria.

* **Severity:** Major

* **Priority:** High

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 2: Verify that search results are relevant and accurately match the search criteria**

* **Test Case ID:** RC-002

* **Test Case Title:** Verify that search results are relevant and accurately match the search criteria

* **Test Suite:** Recipe Discovery Module

* **Test Priority:** High

* **Preconditions:**

 + User is logged in to the platform

 + User has selected the desired cuisine type

* **Test Data:** Test keyword or ingredient data (e.g., "healthy", "vegan")

* **Test Steps:**

 1. Log in to the platform as a valid user

 2. Navigate to the search page and enter a keyword or ingredient in the search bar

 3. Click the "Search" button

 4. Check the relevance of search results by examining the recipe titles, images, and descriptions

 5. Verify that search results include only recipes that match the search criteria

* **Postconditions:** Search results are accurately matched to the search criteria.

* **Expected Result:** Search results accurately match the search criteria.

* **Severity:** Critical

* **Priority:** High

* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 3: Verify that users can filter search results by cuisine type, dietary preferences, and other criteria**

* **Test Case ID:** RC-003
* **Test Case Title:** Verify that users can filter search results by cuisine type, dietary preferences, and other criteria
* **Test Suite:** Recipe Discovery Module
* **Test Priority:** Medium
* **Preconditions:**
 + User is logged in to the platform
 + User has selected the desired cuisine type
* **Test Data:** Various cuisine types (e.g., Italian, Chinese, vegetarian)
* **Test Steps:**
 1. Log in to the platform as a valid user
 2. Navigate to the search page and enter a keyword or ingredient in the search bar
 3. Click the "Search" button
 4. Select a cuisine type, dietary preference, or other filter criteria from the available options
 5. Click the "Apply" button
 6. Verify that search results are updated to reflect the selected criteria
* **Postconditions:** Search results are filtered by the selected criteria.
* **Expected Result:** Search results are updated to reflect the selected criteria.
* **Severity:** Major
* **Priority:** Medium
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

**Test Case 4: Verify that recipes in search results are accessible and detailed**

* **Test Case ID:** RC-004
* **Test Case Title:** Verify that recipes in search results are accessible and detailed
* **Test Suite:** Recipe Discovery Module
* **Test Priority:** Medium
* **Preconditions:**
 + User is logged in to the platform
 + User has selected the desired cuisine type
* **Test Data:** Test recipe data (e.g., images, descriptions, ingredients)
* **Test Steps:**
 1. Log in to the platform as a valid user
 2. Navigate to the search page and enter a keyword or ingredient in the search bar
 3. Click the "Search" button
 4. Select a recipe from the search results
 5. Verify that the recipe details page displays the expected information, including ingredients, instructions, and images
* **Postconditions:** Recipe details page displays expected information.
* **Expected Result:** Recipe details page displays accurate and detailed information.
* **Severity:** Minor
* **Priority:** Medium
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

**Test Case 5: Verify that users can save search results for future reference**

* **Test Case ID:** RC-005

* **Test Case Title:** Verify that users can save search results for future reference

* **Test Suite:** Recipe Discovery Module

* **Test Priority:** Low

* **Preconditions:**

 + User is logged in to the platform

 + User has selected the desired cuisine type

* **Test Data:** Test search results data (e.g., recipe titles, images)

* **Test Steps:**

 1. Log in to the platform as a valid user

 2. Navigate to the search page and enter a keyword or ingredient in the search bar

 3. Click the "Search" button

 4. Select a recipe from the search results

 5. Click the "Save for Later" button

 6. Verify that the saved recipe is listed under the user's account preferences

* **Postconditions:** Saved recipe is listed under the user's account preferences.

* **Expected Result:** Recipe is saved for future reference.

* **Severity:** Trivial

* **Priority:** Low

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

---