

# Generated Test Cases via mixtral-8x7b-32768 LLM Model and Few-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:**

Test Case Creation Template:

Test Case ID: USP-001

Test Case Title: Verify that users can add their name to their profile

Test Case Description: Ensure that users can input their name and save it successfully to their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in
- User is on the profile editing page

Test Data: User name

Test Steps:

1. Enter a name in the name field
2. Click the save button

Postconditions:

- Name is saved in the profile

Expected Result: The system saves the user's name and displays a confirmation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: USP-002

Test Case Title: Verify that users can edit their name on their profile

Test Case Description: Ensure that users can edit their name and save it successfully to their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in
- User has a saved name on their profile
- User is on the profile editing page

Test Data: Changed user name

Test Steps:

1. Change the name in the name field
2. Click the save button

Postconditions:

- Name is updated in the profile

Expected Result: The system saves the updated user name and displays a confirmation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: USP-003

Test Case Title: Verify that users can add a profile picture to their profile

Test Case Description: Ensure that users can upload a profile picture in standard image formats and save it successfully to their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in
- User is on the profile editing page

Test Data: Standard image format file

Test Steps:

1. Click the profile picture upload button
2. Select a standard image format file
3. Click the save button

Postconditions:

- Profile picture is saved in the profile

Expected Result: The system saves the user's profile picture and displays a confirmation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: USP-004

Test Case Title: Verify that users can set privacy settings for each field in their profile

Test Case Description: Ensure that users can set privacy settings for each field in their profile, such as name, bio, profile picture, location, and interests.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in
- User is on the profile editing page

Test Data: Privacy settings

Test Steps:

1. Go to the privacy settings section
2. Set privacy options for the desired fields
3. Save changes

Postconditions:

- Privacy settings are updated in the profile

Expected Result: The system saves the user's privacy settings and displays a confirmation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: USP-005

Test Case Title: Verify that users receive a confirmation upon saving their profile

Test Case Description: Ensure that users receive a confirmation message when successfully saving their profile.

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

- User is logged in
- User has entered or edited profile information

Test Data: No test data needed

Test Steps:

1. Click the save button

Postconditions:

- Profile is saved

Expected Result: The system displays a confirmation message that the profile has been saved successfully.

Severity: Minor

Type of Testing: Functional

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:**

Test Case ID: sfF1eD

Test Case Title: Verify that the news feed displays posts from followed accounts

Test Case Description: This test case verifies that when a user logs in, the news feed displays posts from the accounts they follow.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- User follows at least one account

Test Data: User accounts and followed accounts

Test Steps:

1. Log in as a user
2. Navigate to the news feed
3. Verify that posts from the followed accounts are displayed

Postconditions:

- News feed displays posts from the followed accounts

Expected Result: The news feed displays posts from the accounts the user follows, ensuring that the user sees updated content from connections.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: fgfd3E

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

Test Case Description: This test case checks that the news feed displays content either in chronological order or as determined by a specific algorithm.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- User follows at least one account

Test Data: User accounts, followed accounts, time-stamped posts

Test Steps:

1. Log in as a user
2. Navigate to the news feed
3. Observe the order of posts
4. Verify that the posts are displayed either in chronological order or as determined by an algorithm

Postconditions:

- News feed displays content in a specific order

Expected Result: The news feed shows content either in chronological order based on the time of posting or in an algorithmically determined sequence, providing a tailored user experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: dssg3E

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

Test Case Description: This test case confirms that users can interact with the posts in the news feed using like, comment, or share functionalities.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- User follows at least one account

Test Data: User accounts, followed accounts, posts with interactive elements

Test Steps:



1. Log in as a user
2. Navigate to the news feed
3. Select a post
4. Interact with the post using like, comment, or share
5. Verify that the interaction is recorded and reflected on the post

Postconditions:

- User has successfully interacted with a post

Expected Result: The ability for users to interact with posts ensures engagement and enables the platform to deliver a socially immersive experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: gfrt4G

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

Test Case Description: This test case aims to confirm that the news feed updates in real-time when new posts are added or interactions occur on followed accounts.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- User follows at least one account
- At least one of the followed accounts is active and posting

Test Data: User accounts, followed accounts, real-time posting activities

Test Steps:

1. Log in as a user
2. Navigate to the news feed

3. Observe the display of the feed
4. Perform an action on followed accounts, such as posting or interaction
5. Verify that the action appears instantly on the news feed

Postconditions:

- News feed displays real-time updates

Expected Result: Real-time updates to the news feed ensure that users see up-to-date content from their connections without delay.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: thfr3E

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

Test Case Description: This test case confirms that users can filter displayed content based on specific types or topics from their followed accounts.

Test Suite: News Feed

Test Priority: Medium

Preconditions:

- User is logged in
- User follows at least one account
- There is diverse content available from followed accounts

Test Data: User accounts, followed accounts, diverse content (text, image, video)

Test Steps:

1. Log in as a user
2. Navigate to the news feed
3. Use the filter option to select content by type or topic
4. Verify that the display is limited to the chosen type or topic

Postconditions:

- Users can filter displayed content

Expected Result: Filters enable users to personalize their news feed and organize displayed content based on criteria that matter most to them, ensuring a more tailored experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: SMP\_M1

Test Case Title: Verify that users can send text messages in private conversations

Test Case Description: As a user, I want to send text messages in private conversations to communicate effectively with my friends.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged in
- User has a friend on the platform

Test Data: Text message content

Test Steps:

1. Navigate to the messaging feature
2. Select a friend for a private conversation
3. Type a text message
4. Click the send button
5. Verify the message is delivered

Postconditions:

- Text message is displayed in the conversation

Expected Result: The system allows users to send text messages in private conversations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP\_M2

Test Case Title: Verify that users can send images in private conversations

Test Case Description: As a user, I want to share images in private conversations to enhance my communication with friends.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged in
- User has a friend on the platform

Test Data: Image file

Test Steps:

1. Navigate to the messaging feature
2. Select a friend for a private conversation
3. Click the image attachment button
4. Select an image file
5. Click the send button
6. Verify the image is displayed in the conversation

Postconditions:

- Image is displayed in the conversation

Expected Result: The system allows users to send images in private conversations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP\_M3

Test Case Title: Verify that real-time messages are delivered in group chats

Test Case Description: As a user, I want to participate in group chats and send real-time messages to communicate with multiple friends simultaneously.

Test Suite: Messaging

Test Priority: Medium

Preconditions:

- User is logged in
- User is a member of a group chat

Test Data: Text message content

Test Steps:

1. Navigate to the group chat
2. Type a text message
3. Click the send button
4. Verify the message is displayed in the group chat for all members
5. Monitor the delay between sending and receiving the message

Postconditions:

- Text message is displayed in the group chat
- Message delivery time is acceptable

Expected Result: The system allows users to send real-time messages in group chats.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP\_M4

Test Case Title: Verify that messages show read receipts

Test Case Description: As a user, I want to know if the recipient has read my messages, so I can gauge their engagement in our conversation.

Test Suite: Messaging

Test Priority: Medium

Preconditions:

- User is logged in
- User has a friend on the platform
- A message is sent from the user to the friend

Test Data: No test data needed

Test Steps:

1. Navigate to the messaging feature
2. Select a friend for a private conversation
3. Verify that sent messages show a read receipt once viewed

Postconditions:

- Message read receipts are displayed

Expected Result: The system shows read receipts for messages in private conversations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP\_M5

Test Case Title: Verify that users are notified of new messages

Test Case Description: As a user, I want to be notified of new messages, so I can engage in conversations in a timely manner.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged in
- User has a friend on the platform
- The friend sends a new message

Test Data: No test data needed

Test Steps:

1. Navigate away from the messaging feature
2. Wait for a new message
3. Verify that the user receives a notification for the new message

Postconditions:

- User receives a notification

Expected Result: The system notifies users of new messages in the messaging feature.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: ShrRt1

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

Test Case Description: Ensure users are able to compose, proofread, and publish text posts to their followers or the public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in
- User is on the platform's main page or profile page

Test Data: Text content (minimum 5 words, maximum 280 characters)

Test Steps:

1. Access the platform's main page or profile page
2. Click on the "Create Post" button
3. Select the text post option
4. Input the desired text content
5. Choose whether to share with followers or the public
6. Click "Post"

Postconditions:

- Text post is published
- Text post is visible to the intended audience

Expected Result: Users can create and share text posts with their followers or the public.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ImgTg2

Test Case Title: Verify that users can share photos

Test Case Description: Make sure users can upload, preview, and share photos with their followers or the public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in
- User is on the platform's main page or profile page

Test Data: Photo file (minimum 1 photo, supported formats: JPG, PNG, GIF)

Test Steps:

1. Access the platform's main page or profile page
2. Click on the "Create Post" button
3. Select the photo option
4. Choose the desired photo file
5. Input a caption for the photo (optional)
6. Choose whether to share with followers or the public
7. Click "Post"

Postconditions:

- Photo post is published

- Photo post is visible to the intended audience

Expected Result: Users can upload, preview, and share photos with their followers or the public.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: VidSt3

Test Case Title: Verify that users can share videos

Test Case Description: Confirm that users can upload, set a thumbnail, and share videos with their followers or the public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in
- User is on the platform's main page or profile page

Test Data: Video file (minimum 3-second duration, supported formats: MP4, MOV, AVI)

Test Steps:

1. Access the platform's main page or profile page
2. Click on the "Create Post" button
3. Select the video option
4. Choose the desired video file
5. Input a title and description for the video (optional)
6. Set a custom thumbnail (optional)
7. Choose whether to share with followers or the public
8. Click "Post"

Postconditions:

- Video post is published

- Video post is visible to the intended audience

Expected Result: Users can upload, set a thumbnail, and share videos with their followers or the public.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EdDl4

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

Test Case Description: Validate that users can edit or delete their own posts, without affecting other users' interactions or content.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- User is logged in
- User has previously posted content
- Post is not older than the allowed editing time

Test Data: Previously posted content (text post, photo, or video)

Test Steps:

1. Access the user's profile page
2. Locate the previously posted content
3. For editing: Click on the "Edit" button, modify the content, then click "Save"

For deleting: Click on the "Delete" button

Postconditions:

- Edited post is updated
- Deleted post is removed from the user's profile

Expected Result: Users can edit or delete their own posts without affecting other users' interactions or content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: HshTg5

Test Case Title: Verify that posts support hashtags and tagging

Test Case Description: Ensure users can include hashtags and tag other users in their posts, creating a more connected and engaging social media experience.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in
- User is on the platform's main page or profile page

Test Data: Text content (minimum 5 words, maximum 280 characters) with a hashtag, and another user's handle

Test Steps:

1. Access the platform's main page or profile page
2. Click on the "Create Post" button
3. Select the text post option
4. Input the desired text content, including a hashtag (e.g., #test) and another user's handle (e.g., @username)
5. Choose whether to share with followers or the public
6. Click "Post"

Postconditions:

- Text post is published with the hashtag and tagged user

Expected Result: Posts support hashtags and tagging, allowing users to discover and connect with related content and other users.

Severity: Major

Type of Testing: Functional Testing

**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:**

Test Case Creation Template for Social Media Platform Privacy Settings:

Test Case ID: PRV1

Test Case Title: Verify that users can set their profile to public

Test Case Description: Ensure that the system allows users to set their profile visibility to public, allowing anyone to view their information.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in

Test Data: User profile information

Test Steps:

1. Go to profile settings
2. Select privacy settings
3. Set profile to public
4. Save changes

Postconditions:

- Profile privacy is updated

Expected Result: The system sets the profile to public and notifies the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PRV2

Test Case Title: Verify that users can change privacy settings at any time

Test Case Description: Verify that users can modify their privacy settings regardless of previous configurations.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in
- User has an existing privacy setting (public, private, or restricted)

Test Data: User profile information and updated privacy settings

Test Steps:

1. Go to profile settings
2. Select privacy settings
3. Change the existing privacy setting
4. Save changes

Postconditions:

- Profile privacy is updated

Expected Result: The system updates the privacy settings according to the user's new preferences.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PRV3

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

Test Case Description: Validate that users can change the visibility of individual posts, even when the profile is set to public.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in
- User has created a post

Test Data: User post with visibility options



Test Steps:

1. Go to the specific post
2. Click on the privacy settings for the post
3. Change the post's visibility
4. Save changes

Postconditions:

- Post privacy is updated

Expected Result: The system updates the post's privacy settings according to the user's new preferences, regardless of the profile's general privacy settings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PRV4

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

Test Case Description: Ensure that the system enforces the updated privacy settings without delay.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in
- User has an existing privacy setting (public, private, or restricted)

Test Data: User profile information and user attempting to view the profile

Test Steps:

1. Go to profile settings
2. Select privacy settings
3. Change the existing privacy setting
4. Save changes

5. Log out and log in as another user
6. Attempt to view the first user's profile

Postconditions:

- Profile privacy is enforced

Expected Result: The system prevents the second user from viewing the first user's profile, posts, or other information according to the new privacy settings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PRV5

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

Test Case Description: Validate that the system notifies the user via email, push notification, or in-app message about any changes in their privacy settings.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in
- User has a valid email address or notification preference

Test Data: User profile information and updated privacy settings

Test Steps:

1. Go to profile settings
2. Select privacy settings
3. Change the existing privacy setting
4. Save changes

Postconditions:

- Profile privacy is updated

- User receives a notification of the changes

Expected Result: The system sends a notification to the user regarding the updated privacy settings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

## **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 ID: NtfyC1

Test Case Title: Verify that users receive notifications for likes, comments, and shares

Test Case Description: This test case ensures that users are notified when they receive likes, comments, or shares on their posts.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has posted content

Test Data: User post with embedded media

Test Steps:

1. Log in as a user.
2. Create a new post with text or media.
3. Have another user interact with the post by liking, commenting, or sharing it.
4. Monitor the notifications.

Postconditions:

- Notification for interaction is displayed

Expected Result: User receives a notification for the like, comment, or share on their post.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: NtfyC2

Test Case Title: Verify that users are notified of new friend requests and mentions

Test Case Description: This test case verifies that users are notified when they receive friend requests and mentions from other users.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has not accepted an existing friend request

Test Data: Friend request from another user

Test Steps:

1. Log in as a user.
2. Ensure that there are no existing friend requests.
3. Have another user send a friend request.
4. Monitor the notifications.

Postconditions:

- Notification for friend request is displayed
5. Accept the friend request

Test Steps (continued):

6. Have another user mention the user in a post or comment.
7. Monitor the notifications.

Postconditions (continued):

- Notification for mention is displayed

Expected Result: User receives notifications for the new friend request and mention.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: NtfyC3

Test Case Title: Verify that notifications are displayed in real-time

Test Case Description: This test case confirms that notifications are displayed in real-time as new activities occur.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has posted content

Test Data: User post with embedded media

Test Steps:

1. Log in as a user.
2. Create a new post with text or media.
3. Have another user interact with the post by liking, commenting, or sharing it.
4. Monitor the notifications immediately after the interaction.

Expected Result: Notification for the interaction is displayed instantly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: NtfyC4

Test Case Title: Verify that users can customize notification preferences

Test Case Description: This test case checks if users can configure their notification preferences according to their needs.

Test Suite: Notifications

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Various notification types and delivery options

Test Steps:

1. Log in as a user.
2. Go to account settings.
3. Access notification preferences.
4. Select or deselect various notification types.
5. Choose specific delivery options for each notification type.
6. Save the changes.

Expected Result: User's notification preferences are updated and saved correctly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: NtfyC5

Test Case Title: Verify that notifications include a link to the relevant activity

Test Case Description: This test case confirms that notifications have a link directing users to the relevant activity or content.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has posted content

Test Data: User post with embedded media

Test Steps:

1. Log in as a user.
2. Create a new post with text or media.
3. Have another user interact with the post by liking, commenting, or sharing it.

4. Click on the notification for the interaction.

Expected Result: User is directed to the specific post or content related to the notification.

Severity: Major

Type

-----

## **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:

Test Case Creation Template:

Test Case ID: SMP-SF-001

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

Test Case Description: This test case verifies that the search functionality allows users to search for content using keywords and returns relevant results.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in
- There is content available on the platform

Test Data: Keywords related to available content

Test Steps:

1. Go to the search bar
2. Enter a keyword related to available content
3. Initiate the search
4. Check the search results

Postconditions:

- Relevant content is displayed in the search results

Expected Result: The system returns relevant content based on the entered keyword.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SMP-SF-002

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

Test Case Description: This test case verifies that the search functionality provides search results that include profiles, posts, and topics.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in
- There is content available on the platform, including profiles, posts, and topics

Test Data: No test data needed

Test Steps:

1. Go to the search bar
2. Initiate a search for a generic term related to the platform (e.g., "social media")
3. Check the search results

Postconditions:

- Profiles, posts, and topics are displayed in the search results

Expected Result: The system returns profiles, posts, and topics in the search results for a generic term.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SMP-SF-003

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

Test Case Description: This test case checks that search results are displayed in order of relevance.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in

- There is content available on the platform

Test Data: Keywords related to available content

Test Steps:

1. Go to the search bar
2. Enter a keyword related to available content
3. Initiate the search
4. Check the search results' order

Postconditions:

- Search results are displayed in order of relevance

Expected Result: The system returns search results in order of relevance, with the most relevant results displayed first.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SMP-SF-004

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

Test Case Description: This test case confirms that the search functionality provides users with the option to filter search results by content type.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in
- There is content available on the platform

Test Data: Keywords related to available content

Test Steps:

1. Go to the search bar

2. Enter a keyword related to available content
3. Initiate the search
4. Filter the search results by content type (e.g., posts, profiles, or topics)
5. Check the filtered search results

Postconditions:

- Search results are filtered by content type

Expected Result: The system allows users to filter search results by content type and displays only the selected type of content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SMP-SF-005

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

Test Case Description: This test case checks if the search functionality provides suggestions as the user types.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Keywords related to platform content and irrelevant keywords

Test Steps:

1. Go to the search bar
2. Enter a keyword related to platform content or an irrelevant keyword
3. Monitor the search suggestions

Postconditions:

- Search suggestions are provided

Expected Result: The system offers search suggestions related to the entered keywords.

Severity: Major

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:**

Test Case ID: SMTC-H1

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

Test Case Description: This test case aims to verify that users can add hashtags to their posts, enabling content categorization and discoverability.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in
- User has privileges to create posts

Test Data: New post with hashtags (e.g., #test, #socialmedia)

Test Steps:

1. Navigate to the post creation screen
2. Input text content for the post
3. Add hashtags within the post text
4. Publish the post

Postconditions:

- Post is successfully published
- Hashtags are parsed and clickable

Expected Result: The system accepts and saves the post with hashtags, making them clickable and searchable.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMTC-H2

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

Test Case Description: This test case ensures users can tag other profiles, brands, or locations, expanding the reach of their content.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in
- User has privileges to create posts

Test Data: New post with tags (e.g., @testuser, @brandname, @location)

Test Steps:

1. Navigate to the post creation screen
2. Input text content for the post
3. Add tags for other profiles, brands, or locations within the post text
4. Publish the post

Postconditions:

- Post is successfully published
- Tags are clickable and linked to the referenced page

Expected Result: The system accepts and saves the post with tags, making them clickable and connected to the referenced page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMTC-H3

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

Test Case Description: This test case verifies the functionality of clickable hashtags and tags in posts, ensuring content discoverability.

Test Suite: Hashtags and Tagging

Test Priority: Medium

Preconditions:

- User is logged in
- Post containing hashtags and tags exists

Test Data: Click on a hashtag or tag within an existing post

Test Steps:

1. Navigate to the post containing hashtags and tags
2. Click on a hashtag or tag
3. Observe redirection

Postconditions:

- User redirected to a page displaying related content

Expected Result: The system redirections user to a page displaying related content for the clicked hashtag or tag.

Severity: Normal

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMTC-H4

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

Test Case Description: This test case checks if users can search for content using hashtags, enhancing content discoverability.

Test Suite: Hashtags and Tagging

Test Priority: Medium

Preconditions:

- User is logged in
- Posts containing hashtags exist

Test Data: Search terms with hashtags (e.g., #test)

Test Steps:

1. Navigate to the search bar



2. Input a search term with a hashtag

3. Initiate the search

Postconditions:

- Relevant results with hashtags are displayed

Expected Result: The system returns relevant posts containing the hashtag.

Severity: Normal

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMTC-H5

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

Test Case Description: This test case checks if the system offers suggestions for popular hashtags while users create posts, promoting content discoverability.

Test Suite: Hashtags and Tagging

Test Priority: Low

Preconditions:

- User is logged in
- User has privileges to create posts
- Similar content exists with hashtags

Test Data: New post with hashtag input (e.g., typing "test")

Test Steps:

1. Navigate to the post creation screen
2. Input text content for the post
3. Start typing a hashtag
4. Observe hashtag suggestions

Postconditions:

- Popular or relevant hashtags are suggested during input

Expected Result: The system displays popular hashtags or relevant hashtags based on the input for users to add and categorize their content.

Severity: Normal

Type of Testing: Functional

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:**

Test Case ID: Empj1A

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

Test Case Description: This test case ensures users have the ability to post various emojis as reactions to different posts on the social media platform.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in
- A post exists on the user's news feed

Test Data: A selection of emojis

Test Steps:

1. Navigate to the news feed
2. Identify a post to react to
3. Select an emoji from the list
4. Post the emoji reaction
5. Verify the emoji appears below the post

Postconditions:

- Reaction emoji appears below the post

Expected Result: A variety of emojis can be used to react to posts on the platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: Empj2B

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

Test Case Description: This test case checks if users can add emojis to their comments and messages, enabling richer communication and expressiveness.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

- User is logged in
- A post or message exists

Test Data: A selection of emojis

Test Steps:

1. Navigate to a post or compose a new message
2. Add a comment or write a message
3. Insert an emoji into the text
4. Post the comment or send the message
5. Verify the emoji appears in the comment or message

Postconditions:

- Emoji appears within the comment or message

Expected Result: Emojis can be used effectively in both comments and messages.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: Empj3C

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

Test Case Description: This test case guarantees that the platform displays the total count of emoji reactions

for each post, enhancing user engagement and expression.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

- User is logged in
- A post with emoji reactions exists on the user's news feed

Test Data: None

Test Steps:

1. Navigate to the news feed
2. Identify a post with emoji reactions
3. Verify the number of reactions is displayed next to the post

Postconditions:

- Total reaction count is displayed near the post

Expected Result: The system accurately and visibly displays the reaction count for each post.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: Empj4D

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

Test Case Description: This test case ensures the platform can handle and render a wide variety of Unicode emojis, enabling support for cross-platform consistency and expressiveness.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in

Test Data: A diverse set of emojis (Unicode)

Test Steps:

1. Navigate to the comment or message composer
2. Input a wide range of emojis
3. Verify that the emojis are correctly displayed and convey the intended meaning

Postconditions:

- Wide range of emojis are successfully displayed or delivered

Expected Result: The social media platform supports the display and use of a large variety of Unicode emojis.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: Empj5E

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

Test Case Description: This test case checks that users can remove or modify their emoji reactions on posts as needed, ensuring they can maintain up-to-date and accurate expressiveness.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in
- User has previously reacted to a post

Test Data: A selection of emojis

Test Steps:

1. Navigate to the news feed
2. Identify a post with the user's reaction
3. Remove or change the reaction

4. Verify the reaction is removed or updated

Postconditions:

- User's reaction is successfully removed or updated

Expected Result: The system permits users to remove or alter their emoji reactions on posts.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: smpLv1

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

Test Case Description: Ensure that users have the ability to start and stop a live stream as intended.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Navigate to the live streaming feature
2. Click on the "Start Stream" button
3. Verify that the stream has started successfully
4. Click on the "Stop Stream" button
5. Verify that the stream has stopped and cannot be resumed

Expected Result: The user can start and stop a live stream without encountering any issues.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: smpLv2

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



Test Case Description: Ensure that users have the option to broadcast live streams to the public or limit them to their followers.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection
- User has at least one follower

Test Data: Two user accounts, one with no followers and one with at least one follower

Test Steps:

1. Navigate to the live streaming feature
2. Choose either "Public" or "Followers only" before starting the stream
3. Verify that the selected audience can view the stream
4. Verify that the restricted audience cannot view the stream

Expected Result: Users can choose between making their live streams public or limiting them to their followers.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: smpLv3

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

Test Case Description: Ensure that viewers can engage with the live stream through comments and reactions as intended.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection
- User is following the streamer

Test Data: User accounts, one streaming and one viewing

Test Steps:

1. Start a live stream as the streamer
2. Access the live stream as the viewer
3. Send a comment or reaction from the viewer account
4. Verify that the comment or reaction is visible to both the viewer and the streamer

Expected Result: Viewers can interact with the streamer through comments and reactions, and these interactions are visible to both parties.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: smpLv4

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

Test Case Description: Ensure that the live stream displays the viewer count and live comments in real-time.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection
- User is following the streamer
- Multiple users are viewing the live stream

Test Data: User accounts and live stream

Test Steps:

1. Start a live stream as the streamer
2. Access the live stream as the viewer
3. Verify that the viewer count is displayed on the live stream
4. Verify that live comments appear on the live stream

Expected Result: The live stream displays both viewer count and live comments in real-time.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: smpLv5

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

Test Case Description: Ensure that users have the ability to save a live stream for later viewing.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection
- User is following the streamer

Test Data: User accounts and live stream

Test Steps:

1. Start a live stream as the streamer
2. Access the live stream as the viewer
3. Click the "Save Stream" button
4. Verify that the saved stream is accessible from the user's profile

Expected Result: Users can save a live stream for later viewing.

Severity: Major

Type of Testing: Functional Testing

## **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:**

Test Case ID: SMP-St-001

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

Test Case Description: This test case verifies that users can create stories by adding photos, videos, and text overlays.

Test Suite: Stories

Test Priority: High

Preconditions:

- User is logged in
- User has access to the story creation feature

Test Data: Photo or video files, story text

Test Steps:

1. Log in as a user
2. Navigate to the story creation feature
3. Select the option to add a photo or video
4. Upload a photo or video from the device
5. Enter text for the story overlay
6. Save the story

Postconditions:

- Story is saved with a photo, video, and text overlay

Expected Result: The system allows users to create stories with photos, videos, and text overlays, and saves the story correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP-St-002

Test Case Title: Verify that stories disappear after 24 hours

Test Case Description: This test case ensures that stories are automatically removed from the platform after 24

hours.

Test Suite: Stories

Test Priority: High

Preconditions:

- User has created a story
- At least 24 hours have passed since story creation

Test Data: No test data needed

Test Steps:

1. Log in as a user
2. Verify that the user's story is no longer visible on the profile or news feed
3. Check the story archive for removed stories

Postconditions:

- Story is no longer visible on the profile or news feed

Expected Result: The system automatically removes stories from the platform after 24 hours.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP-St-003

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

Test Case Description: This test case checks if users are able to view the list of followers who have viewed their stories.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- User has created a story
- Users have viewed the story

Test Data: No test data needed

Test Steps:

1. Log in as a user
2. Navigate to the story views section
3. Verify the list of followers who have viewed the story

Postconditions:

- Story view list is available and accurate

Expected Result: The system provides users with a list of followers who have viewed their stories, displaying the information accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP-St-004

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

Test Case Description: This test case confirms that users can choose to share stories with either the public or a specific list of followers.

Test Suite: Stories

Test Priority: High

Preconditions:

- User is logged in
- User has access to the story creation feature

Test Data: Story, list of followers

Test Steps:

1. Log in as a user
2. Navigate to the story creation feature
3. Create a story

4. Choose the sharing option: public or specific followers

5. Save the story

Postconditions:

- Story is shared according to user preference

Expected Result: The system allows users to share stories publicly or with selected followers, and the stories are shared as specified.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMP-St-005

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

Test Case Description: This test case confirms that users can customize their stories with stickers and filters.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to the story creation feature

Test Data: Story, stickers, filters

Test Steps:

1. Log in as a user
2. Navigate to the story creation feature
3. Create a story
4. Add stickers and filters to the story
5. Save the story

Postconditions:

- Story contains added stickers and filters



Expected Result: The system allows users to add stickers and filters to their stories, and the stickers and filters are displayed correctly on the stories.

Severity: Major

Type of Testing: Functional

Test Case Approach:

-----

## **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:**

Test Case ID: tTR5vB

Test Case Title: Verify that users can view data on follower demographics

Test Case Description: This test case ensures that users can access and view the demographic data of their followers, such as age range, gender, location, and engagement patterns.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in
- User has followers

Test Data: No test data needed

Test Steps:

1. Navigate to the analytics and insights section
2. Select the follower demographics tab
3. Verify the age range, gender, and location distribution of followers
4. Check engagement patterns of followers

Postconditions:

- Follower demographic data is displayed

Expected Result: A comprehensive view of follower demographic data is presented in a user-friendly format.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ddM8wR

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

Test Case Description: This test case confirms the availability of post reach and engagement metrics, such as likes, comments, shares, and impressions, for users to evaluate their content's performance.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in
- User has published posts

Test Data: No test data needed

Test Steps:

1. Go to the analytics and insights section
2. Select a post from the list
3. Verify the reach, impressions, and engagement metrics for the post

Postconditions:

- Post reach and engagement metrics are displayed

Expected Result: Accurate data on post reach and engagement metrics are available for each post.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: hpV3qQ

Test Case Title: Verify that users can track audience growth trends

Test Case Description: This test case validates the platform's ability to display audience growth trends, allowing users to monitor their audience's expansion and identify potential areas for improvement.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

- User is logged in
- User has at least two weeks of activity on the platform

Test Data: No test data needed

Test Steps:

1. Access the analytics and insights section
2. Locate the audience growth trends section
3. Analyze audience growth over time

Postconditions:

- Audience growth trends are displayed

Expected Result: Accurate and user-friendly display of audience growth trends.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: uBg7zL

Test Case Title: Verify that analytics are displayed in a user-friendly format

Test Case Description: This test case verifies the presentation of analytics and insights data in a clear, concise, and easily understandable format for users, facilitating informed decision-making.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Navigate to the analytics and insights section
2. Review the displayed data for post reach, engagement metrics, follower demographics, and audience

growth trends

Postconditions:

- Analytics are displayed in a user-friendly manner

Expected Result: The platform presents analytics and insights data clearly for easy interpretation by users.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: yL9fF7

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

Test Case Description: This test case confirms the platform's functionality to filter displayed analytics and insights data based on specific time ranges and content types, optimizing the assessment and comparison process for users.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in
- User has published at least three different posts within two weeks

Test Data: Specific time range and content types

Test Steps:

1. Go to the analytics and insights section
2. Choose a metric from the post reach and engagement, follower demographics, or audience growth trends sections
3. Apply filters for the desired time range and content type
4. Verify that the platform displays the filtered data as requested

Postconditions:

- Data is filtered by time range and content type

Expected Result: The platform accurately displays the filtered analytics and insights data based on user-selected criteria.

Severity: Major

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:**

Test Case Creation Template for Events and Groups Feature:

Test Case ID: SMP-EG-001

Test Case Title: Verify that users can create events

Test Case Description: Ensure the system allows users to create events with valid details.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- User is logged in
- User has access to create events

Test Data: Event details (name, date, time, location, description)

Test Steps:

1. Go to the Events page
2. Click on "Create Event"
3. Enter valid event details
4. Save changes

Postconditions:

- Event is created

Expected Result: The system creates a new event with the provided details and displays it on the Events page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

---

Test Case ID: SMP-EG-002

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

Test Case Description: Ensure the system allows users to join interest-based groups.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- User is logged in
- User is on the Groups page

Test Data: Group name

Test Steps:

1. Browse through the list of groups
2. Select an interest-based group to join
3. Click on the "Join Group" button

Postconditions:

- User is added to the group

Expected Result: The system adds the user to the selected group, enabling access to group discussions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

---

Test Case ID: SMP-EG-003

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

Test Case Description: Ensure the system organizes group discussions by topic, allowing users to filter and engage in relevant conversations.



Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

- User is logged in
- User is part of a group

Test Data: Group discussion topics

Test Steps:

1. Navigate to a group's Discussion page
2. Create new topics or reply to existing ones
3. Search for relevant topics
4. Filter discussions by topic

Postconditions:

- Discussions are organized by topic

Expected Result: The system organizes group discussions by topic, allowing users to filter, search, and engage in relevant conversations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

---

Test Case ID: SMP-EG-004

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

Test Case Description: Ensure the system displays complete event details including date, time, location, and description.

Test Suite: Events and Groups

Test Priority: High

**Preconditions:**

- User is logged in
- Event exists on the platform

**Test Data:** Event details (name, date, time, location, description)

**Test Steps:**

1. Go to the Events page
2. Click on an event
3. Verify event details (date, time, location, description)

**Postconditions:**

- Event details are visible and accurate

**Expected Result:** The system displays the event's details (date, time, location, description) accurately on the event's page.

**Severity:** Major

**Type of Testing:** Functional

**Test Case Approach:** Positive

---

**Test Case ID:** SMP-EG-005

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

**Test Case Description:** Ensure the system allows users to RSVP to events and view the list of attendees.

**Test Suite:** Events and Groups

**Test Priority:** Medium

**Preconditions:**

- User is logged in
- Event exists on the platform

**Test Data:** Event name

Test Steps:

1. Go to the Events page
2. Click on an event
3. Select RSVP option
4. Check the event's attendee list

Postconditions:

- User's RSVP is recorded
- Attendee list is displayed

Expected Result: The system allows users to RSVP to events and displays the updated list of attendees.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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 ID: smpM1

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

Test Case Description: Ensure that the system allows users to apply for various monetization options available on the platform.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in
- User meets eligibility requirements

Test Data: User profile details

Test Steps:

1. Go to account settings
2. Select monetization options
3. Click on apply for monetization
4. Fill out the application form
5. Submit the application

Postconditions:

- Application status is pending review

Expected Result: The system allows the user to submit an application for monetization options and updates the application status accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: smpM2

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

Test Case Description: Ensure that the system provides content creators with advertising revenue and sponsored content partnership opportunities.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in
- User has access to monetization options

Test Data: No test data needed

Test Steps:

1. Go to monetization dashboard
2. Check for advertising revenue and sponsored content options
3. Select an advertising revenue or sponsored content option

Test Postconditions:

- Advertising revenue or sponsored content opportunity is available

Expected Result: The system offers advertising revenue and sponsored content partnership opportunities for content creators.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: smpM3

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

Test Case Description: Ensure that the system facilitates payment processing for content creators.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in
- User has access to monetization options

Test Data: Payment method details

Test Steps:

1. Go to monetization dashboard
2. Set up a payment method
3. Initiate a payment request
4. Monitor payment status

Postconditions:

- Payment request is processed

Expected Result: The system processes payment requests and updates the payment status accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: smpM4

Test Case Title: Verify that detailed earnings reports are available

Test Case Description: Ensure that the system provides detailed earnings reports to content creators.

Test Suite: Monetization Options

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to monetization options
- User has received earnings

Test Data: No test data needed

Test Steps:

1. Go to monetization dashboard
2. Select earnings report
3. Verify detailed earnings information

Postconditions:

- Earnings report is available

Expected Result: The system generates and displays detailed earnings reports with accurate information.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: smpM5

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

Test Case Description: Ensure that the system clearly communicates monetization eligibility requirements to users.

Test Suite: Monetization Options

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Go to monetization options
2. Review the eligibility requirements

Test Postconditions:

- Eligibility requirements are clearly stated

Expected Result: The system clearly states monetization eligibility requirements, providing users with a clear understanding of what they need to do to access monetization options.

Severity: Major

Type of Testing: Usability

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:**

Test Case ID: EXPIC1

Test Case Title: Verify that the explore tab shows trending and popular content

Test Case Description: Ensure that the explore tab displays a collection of popular and trending posts, accounts, and topics from across the platform.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection
- User has not customized any content filters

Test Data: No test data needed

Test Steps:

1. Navigate to the explore tab
2. Scroll through the displayed content
3. Verify that various posts, accounts, and topics are present

Postconditions:

- Trending and popular content is visible on the explore tab

Expected Result: The explore tab presents a variety of popular and trending content, drawing users' attention

to new accounts, topics, and trends.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EXPIC2

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

Test Case Description: Validate that the explore tab allows users to filter content by categories, such as photos, videos, or specific topics, enhancing user control over displayed content.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged in
- User has a stable internet connection

Test Data: Select categories/topics of interest

Test Steps:

1. Navigate to the explore tab
2. Access the filter or category selection feature
3. Choose a category or topic
4. Verify that only content from the chosen category/topic is displayed

Postconditions:

- Filtered content appears on the explore tab

Expected Result: The explore tab allows users to customize their content experience by filtering posts, photos, and videos based on categories or topics, making content discovery more engaging and relevant.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EXPIC3

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

Test Case Description: Check that the explore tab considers user interests, engagement history, and social connections, providing tailored and relevant content recommendations.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection
- User has interacted with various posts, accounts, or topics over time

Test Data: User engagement history, interests, and social connections

Test Steps:

1. Navigate to the explore tab
2. Analyze the recommended content
3. Verify that the recommended content aligns with the user's interests and engagement history

Postconditions:

- Personalized content recommendations are visible on the explore tab

Expected Result: The explore tab intelligently analyzes user data to deliver content that matches the user's preferences, improving their content discovery experience.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EXPIC4

Test Case Title: Verify that users can save or follow content from the explore tab

Test Case Description: Ensure that the explore tab allows users to save or follow accounts, topics, or posts,

promoting user engagement and increasing content familiarization.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection

Test Data: Posts, accounts, or topics of interest

Test Steps:

1. Navigate to the explore tab
2. Select a post, account, or topic to save or follow
3. Access the save/follow feature
4. Verify that the post, account, or topic is properly saved or followed

Postconditions:

- Saved or followed content appears in the user's library or feed

Expected Result: The explore tab provides an easy way for users to save or follow content, allowing for hassle-free content management and increased user interaction.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: EXPIC5

Test Case Title: Verify that the explore tab updates regularly with new content

Test Case Description: Check that the explore tab consistently refreshes and provides users with fresh, updated content, ensuring a dynamic and engaging content discovery experience.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged in
- User has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Navigate to the explore tab
2. Analyze the displayed content
3. Wait for several hours or refresh the page after some time
4. Verify that new content has been added or updated

Postconditions:

- The explore tab displays up-to-date content

Expected Result: The explore tab frequently refreshes and updates its content, ensuring that users continuously discover new posts, accounts, and trends.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

## **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:**

Test Case ID: SMP-ACC-001

Test Case Title: Verify that the system provides alt text for images

Test Case Description: Ensure that the social media platform provides descriptive alt text for all images, enabling users with visual impairments to understand image content using screen readers.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in
- A post containing an image is available

Test Data: Image post

Test Steps:

1. Navigate to the image post
2. Activate screen reader
3. Read the image alt text

Postconditions:

- Screen reader reads the image alt text

Expected Result: The system provides accurate and descriptive alt text for the image, allowing screen reader users to interpret the image content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: SMP-ACC-002

Test Case Title: Verify screen reader compatibility

Test Case Description: Confirm that the social media platform is compatible with popular screen readers, ensuring users with visual impairments can effectively navigate and engage with the platform.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in
- Screen reader is enabled

Test Data: No test data needed

Test Steps:

1. Navigate to various platform sections (homepage, profiles, settings, etc.)
2. Verify screen reader pronunciation and interpretation

Postconditions:

- Screen reader accurately interprets and pronounces platform elements

Expected Result: The social media platform is compatible with the screen reader, enabling users with visual impairments to navigate and engage with the platform using their screen reader of choice.

Severity: Major

Type of Testing: Compatibility Testing

Test Case Approach: Positive

---

Test Case ID: SMP-ACC-003

Test Case Title: Verify that keyboard shortcuts can be used for navigation

Test Case Description: Validate that users can navigate the social media platform using keyboard shortcuts, making it more accessible to users with motor impairments.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Navigate to various platform sections (homepage, profiles, settings, etc.)
2. Use keyboard shortcuts for navigation
3. Interact with content elements (posts, buttons, etc.) using keyboard

Postconditions:

- Keyboard shortcuts enable navigation and content interaction

Expected Result: The social media platform supports keyboard shortcuts, allowing users with motor impairments to navigate and engage with the platform using a keyboard instead of a mouse.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive



---

Test Case ID: SMP-ACC-004

Test Case Title: Verify that color contrast and font size options are provided

Test Case Description: Confirm that the social media platform offers customizable color contrast and font size options, improving readability and accessibility for users with visual impairments.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Access platform settings
2. Locate color contrast and font size settings
3. Adjust color contrast and font size to personal preference

Postconditions:

- Color contrast and font size are adjusted

Expected Result: The social media platform provides color contrast and font size adjustment options, enhancing readability and accessibility for users with visual impairments.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: SMP-ACC-005

Test Case Title: Verify that accessibility features are clearly documented

Test Case Description: Validate that the social media platform's accessibility features are explained and demonstrated in an accessible and comprehensive manner, helping users with disabilities take full advantage of them.

Test Suite: Accessibility Features

Test Priority: Low

Preconditions:

- User has access to platform documentation

Test Data: Accessibility feature documentation

Test Steps:

1. Review accessibility feature documentation
2. Confirm that each feature includes a description and instructions for use

Postconditions:

- Documentation covers all accessibility features

Expected Result: The social media platform documentation clearly explains and demonstrates all accessibility features, ensuring users with disabilities can effectively utilize them.

Severity: Major

Type of Testing: Documentation Review

Test Case Approach: Positive

-----

**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:**

Test Case ID: smpR1

Test Case Title: Verify that content moderation algorithms filter inappropriate content

Test Case Description: Ensure that the system's content moderation algorithms can accurately identify and filter inappropriate content from user posts and comments.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- User attempts to post or comment on content

Test Data: Inappropriate keywords, phrases, and images

#### Test Steps:

1. Log in as a user
2. Create a post or comment with inappropriate content
3. Attempt to submit the post or comment
4. Monitor the system's response

#### Postconditions:

- Inappropriate content is filtered and not displayed
- User receives a notification or warning

Expected Result: The system successfully filters inappropriate content from user posts and comments.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case ID: smpR2

Test Case Title: Verify that users can report harmful or inappropriate content

Test Case Description: Ensure that users can report harmful or inappropriate content, initiating a review process by the platform's moderation team.

Test Suite: Safety and Moderation Tools

Test Priority: Medium

#### Preconditions:

- User is logged in
- Content (post or comment) is available on the platform

Test Data: Harmful or inappropriate content ID or URL

#### Test Steps:

1. Log in as a user
2. Identify harmful or inappropriate content
3. Locate and click the "Report" button

4. Provide additional information in the report form

5. Submit the report

Postconditions:

- Report is submitted successfully
- User receives a confirmation of the submission

Expected Result: The system allows users to report harmful or inappropriate content, triggering the content review process by the platform's moderation team.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: smpR3

Test Case Title: Verify that blocking capabilities are available to users

Test Case Description: Make sure that users can block other users, preventing them from engaging in interactions or sending messages.

Test Suite: Safety and Moderation Tools

Test Priority: Medium

Preconditions:

- User is logged in
- Another user exists on the platform

Test Data: No test data needed

Test Steps:

1. Log in as a user
2. Locate the user to be blocked
3. Access the blocking feature (e.g., from user profile, messages, etc.)
4. Confirm the blocking action

Postconditions:

- Blocked user can no longer initiate interactions or send messages
- User is notified of the block

Expected Result: The system allows users to block other users, preventing them from engaging in interactions or sending messages.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: smpR4

Test Case Title: Verify that comment filters prevent offensive language

Test Case Description: Ensure that comment filters can identify and prevent the display of offensive language in user comments.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- User attempts to comment on content

Test Data: Offensive keywords and phrases

Test Steps:

1. Log in as a user
2. Create a comment with offensive language
3. Attempt to submit the comment
4. Monitor the system's response

Postconditions:

- Offensive language is filtered and not displayed
- User receives a notification or warning

Expected Result: The system successfully filters and prevents offensive language in user comments, ensuring

a safer environment for users.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case ID: smpR5

Test Case Title: Verify that community guidelines are enforced consistently

Test Case Description: Ensure that community guidelines are enforced consistently throughout the platform, resulting in appropriate actions taken for guideline violations.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- Users are active on the platform
- Community guidelines exist

Test Data: Violations of the community guidelines

Test Steps:

1. Identify users who have violated the community guidelines
2. Assess the system's response (e.g., warnings, suspensions, bans)
3. Monitor for additional guideline violations
4. Evaluate the duration and severity of consequences for violations

Postconditions:

- Users face appropriate consequences for community guideline violations
- Consistent enforcement ensures a safer environment for users

Expected Result: The system consistently enforces community guidelines, leading to a safer online environment for all users.

Severity: Major

Type of Testing: Functional Testing

## **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:**



Test Case ID: SM-MLS-1

Test Case Title: Verify that users can select their preferred interface language

Test Case Description: This test case verifies that users can change the platform's interface language to their preferred language.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged in
- User has access to settings

Test Data: List of supported languages (e.g., English, Spanish, French, German)

Test Steps:

1. Log in as a user
2. Access the settings menu
3. Choose the language option
4. Select a preferred language from the list
5. Save changes

Postconditions:

- Interface language is updated to the selected language

Expected Result: The platform's interface displays the correct language, and the user can interact with the platform in their preferred language.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SM-MLS-2

Test Case Title: Verify that posts and messages can be translated into different languages

Test Case Description: This test case ensures that user-generated content, including posts and messages, can

be accurately translated into various languages.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged in
- A post or message in another language exists

Test Data: A post or message in another language (e.g., Arabic, Chinese, Russian)

Test Steps:

1. Access the news feed or messaging system
2. Locate the post or message in a different language
3. Click on the translate button
4. Monitor the translation process

Postconditions:

- The post or message is translated into the user's preferred language

Expected Result: The translated post or message is coherent and accurately conveys the intended information.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SM-MLS-3

Test Case Title: Verify that language preferences are saved and applied consistently

Test Case Description: This test case validates that a user's selected language preferences are saved and applied across multiple sessions and platform sections.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged in

- User has changed the platform language preference

Test Data: No test data needed

Test Steps:

1. Log in as a user
2. Change the platform language preference
3. Log out and close the browser
4. Reopen the browser and log in again as the same user
5. Navigate to various sections of the platform (e.g., news feed, settings, messaging)

Postconditions:

- The platform displays the updated language preference

Expected Result: The user's language preference is maintained throughout multiple sessions and across all sections of the platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SM-MLS-4

Test Case Title: Verify that the platform supports a wide range of languages

Test Case Description: This test case confirms that the platform has multi-language support for popular and less popular languages.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

- Access to the platform

Test Data: List of supported languages (e.g., English, Spanish, French, German, Swedish, Chinese, Arabic, Vietnamese, Hungarian, Gujarati)

Test Steps:

1. Access the platform
2. Check the supported language list

Postconditions:

- The list contains a variety of popular and less common languages

Expected Result: The platform offers comprehensive multi-language support, including less common languages.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SM-MLS-5

Test Case Title: Verify that language selection does not disrupt the user experience

Test Case Description: This test case ensures that changing the platform's interface language does not cause unexpected behavior or disrupt the user's interaction with the platform.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to settings

Test Data: List of supported languages (e.g., English, Spanish, French, German)

Test Steps:

1. Log in as a user
2. Change the platform language preference
3. Interact with the platform by performing common actions (e.g., create a post, like a post, reply to a comment)

Postconditions:

- Core features are accessible, and expected results are achieved

Expected Result: The user experience remains consistent, and the user can interact with the platform as intended through the interface in their preferred language.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: SMI-CP-1

Test Case Title: Verify that users can link their accounts with other social media platforms

Test Case Description: This test case verifies the ability for users to link their social media platform account with external social media platforms through cross-platform integration.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is logged in
- User is on the account settings page

Test Data: Details of external social media accounts

Test Steps:

1. Navigate to the account settings page
2. Select the 'Link Accounts' option
3. Choose an external social media platform from the list
4. Enter the login credentials for the selected platform
5. Authorize the connection between the two platforms

Postconditions:

- The external account is linked

Expected Result: The user's account is successfully linked with the selected external social media platform, and the connection is established securely.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: SMI-CP-2

Test Case Title: Verify that content can be cross-posted to other platforms

Test Case Description: This test case checks the functionality of cross-posting content from the social media platform to external social media platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is logged in
- User has linked at least one external social media account

Test Data: Test post, image, or video

Test Steps:

1. Create a new post with specified test data
2. Select the option to cross-post the content
3. Choose the linked external social media account
4. Confirm the cross-posting action

Postconditions:

- Content is posted on both platforms

Expected Result: The test content is successfully posted on both the social media platform and the selected external social media platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: SMI-CP-3

Test Case Title: Verify that single sign-on (SSO) is available through cross-platform integration

Test Case Description: This test case confirms the single sign-on (SSO) feature, allowing users to sign in using

their external social media platform credentials.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User has an account on the social media platform
- User is logged in to an external social media platform

Test Data: User credentials for the external social media account

Test Steps:

1. Navigate to the login page on the social media platform
2. Select the 'Sign In with [External Platform]' option
3. Enter user credentials for the external social media account
4. Authorize the single sign-on process

Postconditions:

- User is logged in to the social media platform

Expected Result: The user is successfully logged in to the social media platform using their credentials from the external social media account.

Severity: Major

Type of Testing: Security Testing

Test Case Approach: Positive

Test Case ID: SMI-CP-4

Test Case Title: Verify that integration with external media sources is available

Test Case Description: This test case evaluates the cross-platform integration with external media sources, verifying users can access and share media from external sources.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:



- User is logged in

Test Data: External media source URL

Test Steps:

1. Navigate to the post creation page
2. Select the option to attach media from an external source
3. Enter the URL for the external media source
4. Confirm the media attachment

Postconditions:

- Media from the external source is attached to the post

Expected Result: The external media source is successfully attached to the user's post with proper attribution and functionality.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: SMI-CP-5

Test Case Title: Verify that content sharing across platforms is seamless

Test Case Description: This test case checks the functionality of sharing platform content on other social networks without interruption.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

- User is logged in
- User has linked at least one external social media account

Test Data: Test post on the social media platform

Test Steps:

1. Locate the test post on the social media platform

2. Select the option to share the post on an external social media account

3. Confirm the share action

Postconditions:

- The post is shared on the external social media account

Expected Result: The test post is successfully shared on the chosen external social media account without any issues, and all features (e.g., likes, comments) are accessible.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Test Case ID: SocialMediaReward1

Test Case Title: Verify that users can earn points and badges for their activities

Test Case Description: This test case verifies that users can earn points and badges based on their activities on the social media platform.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User is logged in
- User has an active account

Test Data: User activities such as posting content, commenting, and sharing

Test Steps:

1. Perform user activity (post content, comment, share)
2. Check if points have been added to user account
3. Confirm that user has received a badge if applicable

Postconditions:

- User account is updated with points
- User has appropriate badges

Expected Result: Users can earn points and badges as a result of their activities on the platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SocialMediaReward2

Test Case Title: Verify that leaderboards display top users based on points or achievements

Test Case Description: This test case checks if leaderboards correctly display top users based on their accumulated points or achievements.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- Users have performed various activities earning points and badges
- User is logged in

Test Data: Users' points, badges, and achievements

Test Steps:

1. Access the leaderboard feature
2. Review the list of top users
3. Validate the ranking based on points and badges

Postconditions:

- Leaderboard is accurately displayed

Expected Result: The leaderboard displays the top users ranked by their points or achievements.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SocialMediaReward3

Test Case Title: Verify that challenges are available for users to complete and earn rewards

Test Case Description: This test case confirms that challenges are present and users can successfully complete them to earn rewards.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Available challenges

Test Steps:

1. Access the challenges feature
2. Choose a challenge to complete
3. Perform the required actions to finish the challenge
4. Confirm that rewards (points, badges, or virtual goods) are granted

Postconditions:

- User account is updated with rewards

Expected Result: Users can participate in and complete challenges to earn rewards.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SocialMediaReward4

Test Case Title: Verify that rewards can include virtual goods, discounts, or exclusive content

Test Case Description: This test case validates that rewards are correctly assigned based on the specified criteria of virtual goods, discounts, or exclusive content.

Test Suite: Gamification and Rewards

Test Priority: Low

Preconditions:

- Users have earned rewards through activities, achievements, or challenges

- User is logged in

Test Data: User rewards

Test Steps:

1. Access the rewards section of user profile
2. Review available rewards
3. Confirm receipt of virtual goods or discount codes
4. Validate access to exclusive content

Postconditions:

- User has access to rewards

Expected Result: Users can receive and utilize rewards in the form of virtual goods, discounts, or exclusive content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SocialMediaReward5

Test Case Title: Verify that users can view their progress and achievements in a dedicated section

Test Case Description: This test case ensures that users can easily access and review their progress and achievements in a dedicated section of their profile.

Test Suite: Gamification and Rewards

Test Priority: Low

Preconditions:

- User is logged in
- User has performed activities, earned badges, and completed challenges

Test Steps:

1. Navigate to the user profile
2. Access the progress and achievements section

3. Review the accumulated points, badges, and completed challenges

Postconditions:

- User profile displays progress and achievements

Expected Result: Users can view their progress and achievements in a dedicated section of their profile.

Severity: Major

Type of Testing: Functional

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:**

Test Case 1:

Test Case ID: TC001\_TaskCreation

Test Case Title: Verify that users can enter a task title, description, due date, and priority level.

Test Case Description: This test case ensures that the system allows users to input necessary task details during creation.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- User is logged in

Test Data: Task details (title, description, due date, priority level)

Test Steps:

1. Go to the task creation page
2. Enter task title
3. Add task description
4. Set due date
5. Choose a priority level

Postconditions:

- Task details entered

Expected Result: The system saves the entered task details and displays them correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive



## Test Case 2:

Test Case ID: TC002\_TaskAssignment

Test Case Title: Verify that tasks can be assigned to specific team members.

Test Case Description: This test case checks if the system properly assigns a created task to a specific team member.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- User is logged in
- A team member is available in the system

Test Data: Task details (title, description, due date, priority level) and team member information

Test Steps:

1. Go to the task creation page
2. Enter task title, description, due date, priority level
3. Assign task to a team member
4. Save the task

Postconditions:

- Task is assigned to a team member

Expected Result: The system assigns the task to the selected team member.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

## Test Case 3:

Test Case ID: TC003\_TaskDisplay

Test Case Title: Verify that the system displays created tasks.

Test Case Description: This test case confirms that the system successfully displays tasks that have been created.

Test Suite: Task Creation

Test Priority: Medium

Preconditions:

- User is logged in
- A task has been created

Test Data: No test data needed

Test Steps:

1. Go to the task list
2. Review the task list

Postconditions:

- Task appears in the task list

Expected Result: The system correctly displays the task's title, description, due date, and priority level.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC004\_TaskEditing

Test Case Title: Verify that users can edit task details after creation.

Test Case Description: This test case validates if the system enables users to modify the details of already created tasks.

Test Suite: Task Creation

Test Priority: Medium

Preconditions:

- User is logged in
- A task has been created

Test Data: New task details for title, description, due date, and priority level

Test Steps:

1. Go to the task list
2. Click on the task to be edited
3. Update the task details
4. Save the changes

Postconditions:

- Task details are updated

Expected Result: The system updates the task's title, description, due date, or priority level as specified by the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC005\_TaskValidation

Test Case Title: Verify that the system validates required fields for task creation (e.g., task title).

Test Case Description: This test case tests the validation of required fields during task creation.

Test Suite: Task Creation

Test Priority: Low

Preconditions:

- User is logged in

Test Data: Task creation attempt with missing title

Test Steps:

1. Go to the task creation page
2. Leave the task title field blank
3. Try to create the task

Postconditions:

- Task creation fails

Expected Result: The system does not allow task creation without a title and indicates an error.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: TMTAssignTask1

Test Case Title: Verify that users can select team members to assign tasks

Test Case Description: This test case ensures that the system allows users to select team members when assigning a task.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Team member list

Test Steps:

1. Create a new task or select an existing one
2. Click on the assign button
3. Choose a team member from the list
4. Save the changes

Postconditions:

- Team member is assigned to the task

Expected Result: The system assigns the task to the selected team member.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTAssignTask2

Test Case Title: Verify that assignees receive notifications about their tasks

Test Case Description: This test case checks if the assignee receives a notification when they are assigned a task.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- User (assignee) is logged in
- Another user has assigned a task to the assignee

Test Data: Assigned task details

Test Steps:

1. Log in as the assignee
2. Check for notifications
3. View the assigned task

Postconditions:

- Assignee received a notification
- Assignee can view the assigned task

Expected Result: The assignee receives a notification about the new task and can view the assigned task in their task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTAssignTask3

Test Case Title: Verify that assigned tasks display in the assignee's task list

Test Case Description: This test case verifies that the assigned task appears in the assignee's task list.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- User (assignee) is logged in
- Another user has assigned a task to the assignee

Test Data: Assigned task details

Test Steps:

1. Log in as the assignee
2. Go to the task list
3. Check for the assigned task

Postconditions:

- Assigned task is visible in the task list

Expected Result: The assignee can see the assigned task in their task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTAssignTask4

Test Case Title: Verify that users can reassign tasks to different team members

Test Case Description: This test case confirms that users can reassign tasks to different team members.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

- User (current assignee) is logged in

- Another user has assigned a task to the current assignee

Test Data: New team member for reassignment

Test Steps:

1. Log in as the current assignee
2. Open the assigned task
3. Reassign the task to a different team member
4. Save the changes

Postconditions:

- Task is reassigned to a different team member

Expected Result: The system reassigns the task to the newly selected team member.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTAssignTask5

Test Case Title: Verify that task assignment history is tracked

Test Case Description: This test case ensures that the system tracks the history of task assignments.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

- User has assigned a task to a team member

Test Data: No test data needed

Test Steps:

1. Log in as an administrator or with proper permissions
2. View the task details
3. Check the task assignment history

Postconditions:



- Task assignment history is visible

Expected Result: The system displays the task assignment history, including the dates and users involved in the assignments.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

## **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:

Test Case Creation Template:

Test Case ID: TMDT-001

Test Case Title: Verify that users can set reminders for tasks based on deadlines

Test Case Description: Ensure that the system allows users to set task deadlines and corresponding reminders.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged in
- A task is created

Test Data: Task details, Reminder settings

Test Steps:

1. Go to the task details page
2. Click on the "Set Reminder" button
3. Choose a deadline
4. Customize the urgency level
5. Save the reminder settings

Postconditions:

- Reminder is set for the chosen deadline

Expected Result: The system sets the reminder for the specified deadline and urgency level, allowing users to manage tasks effectively.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

## Test Case Creation Template:

Test Case ID: TMDT-002

Test Case Title: Verify that reminders can be customized by urgency level

Test Case Description: Ensure that users can customize reminders based on urgency levels.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged in
- A task is created

Test Data: Task details, Reminder settings, Urgency Levels

Test Steps:

1. Go to the task details page
2. Click on the "Set Reminder" button
3. Choose a deadline
4. Select an urgency level
5. Save the reminder settings

Postconditions:

- Customized reminder is saved

Expected Result: The system saves the customized reminder with the selected urgency level, providing users with prioritized notifications.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: TMDT-003

Test Case Title: Verify that the system sends notifications for upcoming deadlines

Test Case Description: Ensure that the system sends notifications for upcoming task deadlines.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged in
- A task is created with a deadline and reminder

Test Data: Task details, Reminder settings

Test Steps:

1. Go to the task details page
2. Wait for the deadline minus the configured lead time
3. Check if a notification is received

Postconditions:

- Notification is sent

Expected Result: The system sends a notification before the task deadline, providing users with sufficient time to complete tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: TMDT-004

Test Case Title: Verify that users can choose the delivery method for reminders

Test Case Description: Ensure that users can choose the delivery method for task deadline reminders.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged in
- A task is created with a deadline

Test Data: Task details, Reminder settings, Delivery methods

Test Steps:

1. Go to the task details page
2. Click on the "Set Reminder" button
3. Set a deadline
4. Choose a preferred delivery method (e.g., email, in-app)
5. Save the reminder settings

Postconditions:

- Delivery method is set

Expected Result: The system saves the user's preferred delivery method for the reminder, allowing users to receive notifications through their chosen channel.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: TMDT-005

Test Case Title: Verify that reminder settings can be adjusted or disabled

Test Case Description: Ensure that users can adjust or disable task deadline reminders as needed.

Test Suite: Task Deadline Reminders

Test Priority: Low

Preconditions:

- User is logged in
- A task with a reminder is created

Test Data: Task details, Reminder settings

Test Steps:

1. Go to the task details page
2. Click on the "Set Reminder" button
3. Adjust or disable the reminder settings

Postconditions:

- Reminder settings are updated

Expected Result: The system updates the user's reminder settings, allowing them to manage reminders as required.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

-----

## **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:**

Test Case ID: TMT-P01

Test Case Title: Verify that users can set priority levels for tasks

Test Case Description: As a user, I want to set a priority level for a task to manage my workload effectively.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: High, Medium, Low priority levels

Test Steps:

1. Go to the task list
2. Select a task
3. Click on the priority dropdown
4. Choose a priority level (High, Medium, Low)
5. Save changes

Postconditions:

- Task priority is updated

Expected Result: The system allows the user to set a priority level for the task, which is visually distinguished based on the selected priority.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-P02

Test Case Title: Verify that tasks can be sorted based on priority levels

Test Case Description: As a user, I want to view tasks based on their priority so that I can focus on high-priority tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Tasks with various priority levels have been created

Test Data: No test data needed

Test Steps:

1. Go to the task list
2. Set the sorting criteria to priority
3. Verify the tasks are arranged in the correct order (High > Medium > Low)

Postconditions:

- Tasks are sorted according to their priority

Expected Result: The system sorts tasks based on the assigned priority levels, allowing users to efficiently manage their workload.

Severity: Major

Type of Testing: Functional



Test Case Approach: Positive

Test Case ID: TMT-P03

Test Case Title: Verify that users can change the priority level of a task

Test Case Description: As a user, I want to adjust the priority level of a task to match changing project needs.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in
- A task with a specific priority level is created

Test Data: New priority level (High, Medium, Low)

Test Steps:

1. Go to the task list
2. Select a task with a specific priority level
3. Click on the priority dropdown
4. Change the priority level
5. Save changes

Postconditions:

- Task priority is updated

Expected Result: The system allows users to change the priority level of a task, reflecting the updated urgency and importance.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-P04

Test Case Title: Verify that the system visually distinguishes tasks based on priority

Test Case Description: As a user, I want to identify tasks with different priority levels quickly and efficiently.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks with various priority levels are created

Test Data: No test data needed

Test Steps:

1. Go to the task list
2. Verify tasks with High priority are visually distinct
3. Verify tasks with Medium priority are visually distinct
4. Verify tasks with Low priority are visually distinct

Postconditions:

- Tasks are visually distinguished based on their assigned priority

Expected Result: The system visually distinguishes tasks based on their priority levels, enabling users to manage their workload effectively.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-P05

Test Case Title: Verify that tasks without a specified priority default to Medium

Test Case Description: As a user, I want the system to assign a default priority when I don't specify a priority level.

Test Suite: Task Prioritization

Test Priority: Low

Preconditions:

- User is logged in
- A task is created without a priority level

Test Data: No test data needed

Test Steps:

1. Go to the task list
2. Create a new task
3. Skip setting a priority level
4. Save changes
5. Refresh the task list

Postconditions:

- New task is saved
- New task has a default priority

Expected Result: The system automatically assigns a default priority (Medium) to tasks without a specified priority level, ensuring proper task management.

Severity: Major

Type of Testing: Functional

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:**

Test Case 1:

Test Case ID: TMT-001

Test Case Title: Verify that users can create subtasks under main tasks.

Test Case Description: This test case aims to ensure that the system allows users to create subtasks associated with a specific main task.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Main task is created

Test Data: Subtask details (name, description, etc.)

Test Steps:

1. Go to the main task page
2. Click on the "Add Subtask" button
3. Enter the required subtask details
4. Save the new subtask

Postconditions:

- Subtask is created and linked to the main task

Expected Result: The system creates the subtask and displays it as part of the main task, allowing the user to view and manage it.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-002

Test Case Title: Verify that subtasks can have their own due dates and assignees.

Test Case Description: This test case aims to confirm that the system allows users to set unique due dates and assign specific users for each subtask.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Main task is created
- At least one user is registered in the system

Test Data: Subtask details (name, description, due date, assignee)

Test Steps:

1. Go to the main task page

2. Select an existing subtask to edit
3. Set a unique due date for the subtask
4. Assign the subtask to a specific user
5. Save the changes

Postconditions:

- Subtask has an updated due date and assignee

Expected Result: The system saves the changes, displaying the updated due date and assignee for the subtask in the main task page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-003

Test Case Title: Verify that the system tracks the progress of each subtask.

Test Case Description: This test case aims to validate that the system accurately reflects the progress of each subtask based on user interactions.

Test Suite: Subtasks

Test Priority: Medium

Preconditions:

- User is logged in
- Main task is created
- Subtask with due date and assignee is created

Test Steps:

1. Go to the main task page
2. Change the progress status of a subtask

3. Save the updated task status

4. Repeat steps 2-3 for different subtask status (e.g., In Progress, Completed)

Postconditions:

- Subtask status is updated

Expected Result: The system accurately updates the subtask's progress, reflecting real-time interaction and updates.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-004

Test Case Title: Verify that subtasks can be marked as complete independently of the main task.

Test Case Description: This test case aims to confirm that users can mark subtasks as complete even when the main task remains incomplete.

Test Suite: Subtasks

Test Priority: Medium

Preconditions:

- User is logged in
- Main task is created
- Subtask with due date and assignee is created

Test Steps:

1. Go to the main task page
2. Mark a subtask as complete
3. Save the updated task status

Postconditions:

- Subtask is marked as complete

Expected Result: The system saves the subtask's completion status without changing the main task's status.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-005

Test Case Title: Verify that dependencies between subtasks can be set.

Test Case Description: This test case aims to confirm that users can establish dependencies between subtasks, ensuring that the hierarchy and sequence of tasks are maintained.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Main task has multiple subtasks

Test Data: Dependency values (subtask A depends on subtask B)

Test Steps:

1. Go to the main task page
2. Set a dependency value for a subtask
3. Save the changes
4. Verify that the subsequential steps reflect the dependency

Postconditions:

- Subtask dependencies are established

Expected Result: The system reflects any established dependencies between subtasks, thereby ensuring proper task completion order.



Severity: Major

Type of Testing: Functional

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:**

Test Case ID: TMTA01

Test Case Title: Verify that users can set dependencies between tasks

Test Case Description: Ensure that the system allows users to set dependencies between tasks, and these dependencies are saved and displayed correctly.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- Tasks are created

Test Data: Two or more tasks

Test Steps:

1. Go to the task list
2. Select a task
3. Click on the dependencies tab
4. Enter the ID of another task in the dependency field
5. Save the changes

Test Postconditions:

- Dependency is saved and displayed in the task details

Expected Result: The system saves the dependency, and it is displayed in the task details.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTA02

Test Case Title: Verify that tasks cannot be marked as complete if dependent tasks are not completed

Test Case Description: Ensure that the system prevents users from marking a task as complete if its dependent tasks are not completed.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- Tasks with dependencies are created

Test Data: Two tasks with a dependency relationship

Test Steps:

1. Go to the task list
2. Select the task that has a dependent task
3. Try to mark it as complete

Test Postconditions:

- Task should not be marked as complete

Expected Result: The system does not allow users to mark the task as complete and shows a warning or alert.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

Test Case ID: TMTA03

Test Case Title: Verify that users can update or remove dependencies as needed

Test Case Description: Ensure that the system allows users to modify or remove dependencies between tasks.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks with dependencies are created

Test Data: Two tasks with a dependency relationship

Test Steps:

1. Go to the task list
2. Select a task with dependencies
3. Update or remove the dependency in the task details
4. Save the changes

Test Postconditions:

- Dependency is updated or removed

Expected Result: The system updates or removes the dependency, as requested by the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTA04

Test Case Title: Verify that the system provides alerts or warnings if a dependent task is not completed

Test Case Description: Ensure that the system notifies the user when a dependent task is not completed by the expected completion date.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks with dependencies are created with different expected completion dates

Test Data: Two tasks with a dependency relationship and different expected completion dates

Test Steps:

1. Go to the task list
2. Monitor the task that has a dependent task
3. Allow the dependent task's expected completion date to pass

Test Postconditions:

- User is notified about the incomplete dependent task

Expected Result: The system notifies the user about the incomplete dependent task and provides alerts or warnings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

Test Case ID: TMTA05

Test Case Title: Verify that dependencies are displayed in the task details

Test Case Description: Ensure that the system displays dependencies in the task details in a clear and understandable manner.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks with dependencies are created

Test Data: Two tasks with a dependency relationship

Test Steps:

1. Go to the task list
2. Select a task with dependencies
3. Check the task details for displayed dependencies

Test Postconditions:

- Dependencies are displayed

Expected Result: The system displays the dependencies in the task details clearly and understandably.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

---

## **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:**

Test Case ID: COMM123

Test Case Title: Verify that users can add comments to tasks

Test Case Description: Ensure that registered users can post comments on tasks in a project.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in
- User has access to a project with existing tasks

Test Data: Comment text

Test Steps:

1. Navigate to a project
2. Select a task within the project
3. Type a comment in the comment section
4. Click the 'Post' or 'Comment' button

Postconditions:

- Comment is displayed in the task interface

Expected Result: The system displays the user's comment in the task interface in chronological order.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: COMM234

Test Case Title: Verify that comments can include text, attachments, and links

Test Case Description: Ensure that users can create comments with rich content such as text, attachments, and hyperlinks.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in
- User has access to a project with existing tasks

Test Data: Comment text, attachment file, URL

Test Steps:

1. Navigate to a project
2. Select a task within the project
3. Create a comment including text, an attachment, and a hyperlink
4. Post the comment

Postconditions:

- Comment is displayed in the task interface

Expected Result: The system displays the comment with the attached file and hyperlink in the task interface.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: COMM345

Test Case Title: Verify that users can reply to and edit their comments

Test Case Description: Ensure that users can reply to and edit their own comments in a task.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- User has posted a comment on a task

Test Data: Comment text (for editing)

Test Steps:

1. Navigate to a project
2. Select a task within the project
3. Locate the user's comment
4. Reply to or edit the user's own comment
5. Save changes



Postconditions:

- Updated comment is displayed in the task interface

Expected Result: The system saves the user's updates or replies to their comment in the task interface.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: COMM456

Test Case Title: Verify that users receive notifications of new comments

Test Case Description: Ensure that users receive notifications when there are new comments on tasks they created or are assigned to.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- User is assigned to or created a task
- Another user posts a comment on the same task

Test Data: Comment text

Test Steps:

1. Log in as another user
2. Navigate to the same project
3. Post a comment on the task assigned or created by the primary user

Postconditions:

- Primary user receives a notification of the new comment

Expected Result: The primary user receives a notification (email or in-app) of a new comment posted on the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: COMM567

Test Case Title: Verify that comments are displayed in chronological order

Test Case Description: Ensure that comments are sorted and displayed in chronological order with the most recent comments displayed at the top.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to a project with multiple comments on tasks

Test Data: Multiple comments with varying post times

Test Steps:

1. Navigate to a project
2. Select a task with multiple comments
3. Check the sequence of comments

Postconditions:

- Comments are arranged in chronological order

Expected Result: The system displays comments in the task interface in order with the most recent comments at the top.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template:

Test Case ID: TMT-F01-AT01

Test Case Title: Verify that users can attach multiple files to a task

Test Case Description: This test case ensures that users are able to attach more than one file to a task.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in
- A task is created

Test Data: Multiple files with varying formats and sizes

Test Steps:

1. Navigate to the task details page
2. Click on the "Attach File" button
3. Select and upload multiple files
4. Confirm that the files are displayed under the task's attachments section

Postconditions:

- Files are attached successfully

Expected Result: The system allows users to attach multiple files to a task and displays them in the attachments section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: TMT-F01-AT02

Test Case Title: Verify that attachments can include various file types (documents, images, etc.)

Test Case Description: This test case ensures that users can attach various file formats supported by the system.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in

- A task is created

Test Data: A mix of document formats (e.g., .docx, .pdf), image formats (e.g., .jpg, .png), and link URLs

Test Steps:

1. Navigate to the task details page
2. Click on the "Attach File" button
3. Select and upload various file formats
4. Confirm that each file is displayed under the task's attachments section

Postconditions:

- Files are attached successfully

Expected Result: The system supports attaching different file types and displays the attachments in the attachments section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: TMT-F01-AT03

Test Case Title: Verify that users can view and download attachments

Test Case Description: This test case ensures that users can view and download the attached files.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in
- A task with attachments is created

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page
2. Click on the attachment thumbnail or title
3. Verify the file is opened in a new tab or downloaded

Postconditions:

- File is successfully opened or downloaded

Expected Result: The system allows users to view and download attachments easily.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: TMT-F01-AT04

Test Case Title: Verify that the system limits the size of attachments

Test Case Description: This test case ensures that the system limits the size of attachments to prevent data overload.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: File larger than the allowable size

Test Steps:

1. Navigate to the task details page
2. Click on the "Attach File" button
3. Attempt to upload a file larger than the allowable size

Postconditions:

- File is not attached

Expected Result: The system displays an error message and does not allow attachment of a file larger than the specified limit.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

Test Case Creation Template:

Test Case ID: TMT-F01-AT05

Test Case Title: Verify that users can remove attachments

Test Case Description: This test case ensures that users can delete unwanted attached files.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

- User is logged in
- A task with attachments is created

Test Data: No test data needed

Test Steps:

1. Navigate to the task details page
2. Locate the attachment to be removed
3. Click the "Remove" or "Delete" button associated with the attachment
4. Confirm the attachment deletion

Postconditions:

- Attachment is removed

Expected Result: The system allows users to remove attachments as desired.

Severity: Major

Type of Testing: Functional

## **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:**

Test Case ID: TMT-RT-001



Test Case Title: Verify that users can set tasks to recur at specified intervals

Test Case Description: Ensure that the system allows users to create recurring tasks for specific time intervals such as daily, weekly, or monthly.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- User is logged in
- User has access to create tasks

Test Data: New task with a specified recurrence interval (daily, weekly, monthly)

Test Steps:

1. Log in as a user
2. Navigate to the task creation page
3. Enter task details, including a title, description, and deadline
4. Select the recurring task option and set the interval (daily, weekly, monthly)
5. Save the new task

Postconditions:

- Recurring task is created
- System schedules instances of the recurring task according to the specified interval

Expected Result: The system creates a new recurring task and schedules instances based on the selected interval, with the first instance displayed in the user's task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RT-002

Test Case Title: Verify that the system generates instances of recurring tasks automatically

Test Case Description: Check that the system creates instances of recurring tasks in accordance with the

specified interval.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- User is logged in
- A recurring task exists with a specified interval (daily, weekly, monthly)
- Time has passed since the initial creation of the recurring task

Test Data: Recurring task instances

Test Steps:

1. Log in as a user
2. Access the task list
3. Verify that new instances of the recurring task have been generated based on the specified interval

Postconditions:

- Recurring task instances are present in the user's task list

Expected Result: The system has created instances of the recurring task based on the interval, and the user can view these instances in their task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RT-003

Test Case Title: Verify that users can modify the recurrence pattern or cancel the recurrence

Test Case Description: Ensure that users can modify or cancel recurring task patterns.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- User is logged in

- A recurring task exists

Test Data: Modified recurring task settings

Test Steps:

1. Log in as a user
2. Go to the task list
3. Select the recurring task and access the settings
4. Edit the recurrence pattern or click the cancel recurrence button
5. Save the modified task settings

Postconditions:

- Recurring task pattern has been updated or recurrence has been cancelled
- System updates or removes future instances of the recurring task

Expected Result: The user has successfully modified the recurrence pattern or cancelled the recurrence of the task, and the system has updated or removed future instances accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RT-004

Test Case Title: Verify that the system notifies users of upcoming instances of recurring tasks

Test Case Description: Ensure that the system alerts users about the impending instances of recurring tasks based on predefined triggers.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

- User is logged in
- A recurring task with notifications enabled exists

Test Steps:

1. Log in as a user
2. Await the predefined time trigger for the recurring task instance's notification
3. Check for the notification (email, in-app alert, etc.)

Postconditions:

- User has received a notification regarding an instance of the recurring task

Expected Result: The user receives a notification about the impending instance of a recurring task based on the predefined time and notification settings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-RT-005

Test Case Title: Verify that users can view a list of all instances of a recurring task

Test Case Description: Confirm that users can access a comprehensive list of all scheduled instances of a recurring task.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

- User is logged

```diff

- A recurring task exists

```

Test Data: Instances of recurring tasks

Test Steps:

1. Log in as a user
2. Access the task list
3. Locate the recurring task in the task list

4. Click on the recurring task to access the associated instances

Postconditions:

- User can view a list of all instances of the recurring task

Expected Result: The user is able to view a comprehensive list of all instances of the recurring task, including start and end dates.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: TMT-TagCreation

Test Case Title: Verify that users can add tags to tasks

Test Case Description: Ensure that the system allows users to create and add tags to tasks.

Test Suite: Task Tags and Labels

Test Priority: High

Preconditions:

- User is logged in
- A task has been created

Test Data: New tag name

Test Steps:

1. Go to the task details page
2. Click on "Add Tag" button
3. Enter the new tag name
4. Save changes

Postconditions:

- Task has an additional tag

Expected Result: The system saves the new tag and attaches it to the task.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: TMT-TagEditing

Test Case Title: Verify that tags can be customized and edited

Test Case Description: Ensure that the system allows users to edit existing tags.

Test Suite: Task Tags and Labels

Test Priority: Medium

Preconditions:

- User is logged in
- A task with an existing tag is created

Test Data: Modified tag name

Test Steps:

1. Go to the task details page
2. Click on the existing tag
3. Edit the tag name
4. Save changes

Postconditions:

- Task tag is updated

Expected Result: The system saves the updated tag name and displays it in the task list.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: TMT-TagFiltering

Test Case Title: Verify that tasks can be filtered and sorted based on tags and labels

Test Case Description: Ensure that the system allows users to filter and sort tasks based on tags and labels.

Test Suite: Task Tags and Labels

Test Priority: High

Preconditions:

- User is logged in

- Tasks with tags and labels have been created

Test Data: Selected tags or labels

Test Steps:

1. Access the task list
2. Click on the filter or sort button
3. Select desired tags or labels
4. Confirm the filter or sort

Postconditions:

- Tasks are displayed according to the selected tags or labels

Expected Result: The system filters or sorts tasks based on the selected tags or labels.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: TMT-TagProminence

Test Case Title: Verify that the system displays tags and labels prominently in the task list

Test Case Description: Ensure that the system displays tags and labels clearly and visibly in the task list.

Test Suite: Task Tags and Labels

Test Priority: Low

Preconditions:

- User is logged in
- Tasks with tags and labels have been created

Test Data: Tasks with tags and labels

Test Steps:

1. Access the task list
2. Examine the displayed tasks

Postconditions:



- Tags and labels are clearly visible for each task

Expected Result: The system shows tags and labels prominently in the task list, allowing users to easily identify them.

Severity: Major

Type of Testing: Usability

Test Case Behaviour: Positive

Test Case ID: TMT-TagRemoval

Test Case Title: Verify that users can remove tags if needed

Test Case Description: Ensure that the system allows users to delete or remove tags from tasks.

Test Suite: Task Tags and Labels

Test Priority: Medium

Preconditions:

- User is logged in
- A task with an existing tag is created

Test Data: Selected tag for removal

Test Steps:

1. Go to the task details page
2. Click on the existing tag
3. Select the remove option
4. Confirm the removal

Postconditions:

- Task is no longer associated with the removed tag

Expected Result: The system removes the selected tag from the task and displays the updated list of tags.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Negative

---

## **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:**

Test Case ID: TMTHb1

Test Case Title: Verify that users can update the status of tasks

Test Case Description: This test case ensures that users are able to update the status of tasks, including changing them from "To Do" to "In Progress", and finally to "Completed".

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- A task has been created

Test Data: Task details

Test Steps:

1. Go to the task list
2. Select a task
3. Change the status of the task from "To Do" to "In Progress"
4. Save changes
5. Change the status of the task from "In Progress" to "Completed"
6. Save changes

Postconditions:

- Task status is updated

Expected Result: The system updates the task status and displays the updated status prominently in the task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTHb2

Test Case Title: Verify that the system displays task status prominently in the task list

Test Case Description: This test case verifies that the system displays the task status prominently in the task list, making it easy for users to monitor progress.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- A task has been created with a specific status

Test Data: Task with "In Progress" status

Test Steps:

1. Go to the task list
2. Check the status of the task in the list

Test Postconditions:

- Task status is displayed prominently

Expected Result: The system displays the task status clearly in the task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTHb3

Test Case Title: Verify that users can view the progress percentage of tasks

Test Case Description: This test case confirms that users can view the progress percentage of tasks, allowing them to monitor progress and identify bottlenecks.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- A task has been created and updated with progress

Test Data: Task with 50% progress

Test Steps:

1. Go to the task list
2. Select a task with progress

### 3. Check the progress percentage

Test Postconditions:

- Progress percentage is displayed

Expected Result: The system displays the progress percentage of the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTHb4

Test Case Title: Verify that the system provides notifications for status changes

Test Case Description: This test case checks that the system provides notifications for status changes, keeping users informed about task updates.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- A task has been created and assigned to another user
- The assignee updates the task status

Test Data: Task assigned to another user

Test Steps:

1. Go to the task list
2. Check for notifications indicating status changes

Test Postconditions:

- Notifications are provided

Expected Result: The system sends notifications for task status changes.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMTHb5

Test Case Title: Verify that task status can be filtered and sorted

Test Case Description: This test case confirms that users can filter and sort tasks based on their status, simplifying task management and progress tracking.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- Tasks with various statuses have been created

Test Data: Multiple tasks with different statuses

Test Steps:

1. Go to the task list
2. Filter tasks by status
3. Sort tasks by status

Test Postconditions:

- Tasks are filtered and sorted

Expected Result: The system allows users to filter and sort tasks based on status.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template for Task Management Tool - Task Filtering and Sorting Feature:

Test Case ID: TFS1

Test Case Title: Verify that users can filter tasks based on due date

Test Case Description: As a user, ensure that the system allows filtering tasks by their due dates.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in
- Tasks with different due dates exist

Test Data: Tasks with varying due dates

Test Steps:

1. Navigate to the task list
2. Open the filter options
3. Select a due date range
4. Apply the filter

Test Postconditions:

- The task list displays only tasks within the selected due date range

Expected Result: The system filters tasks based on the selected due date range.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case Creation Template for Task Management Tool - Task Filtering and Sorting Feature:

Test Case ID: TFS2

Test Case Title: Verify that users can filter tasks based on priority

Test Case Description: As a user, ensure that the system allows filtering tasks by their priority levels.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in
- Tasks with different priority levels exist

Test Data: Tasks with varying priority levels



Test Steps:

1. Navigate to the task list
2. Open the filter options
3. Select a priority level
4. Apply the filter

Test Postconditions:

- The task list displays only tasks with the selected priority level

Expected Result: The system filters tasks based on the selected priority level.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case Creation Template for Task Management Tool - Task Filtering and Sorting Feature:

Test Case ID: TFS3

Test Case Title: Verify that users can filter tasks based on assignee

Test Case Description: As a user, ensure that the system allows filtering tasks by the assignee.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in
- Multiple users are registered in the system
- Tasks are assigned to different users

Test Data: Tasks assigned to different users

Test Steps:

1. Navigate to the task list
2. Open the filter options

3. Select an assignee

4. Apply the filter

Test Postconditions:

- The task list displays only tasks assigned to the selected user

Expected Result: The system filters tasks based on the selected assignee.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case Creation Template for Task Management Tool - Task Filtering and Sorting Feature:

Test Case ID: TFS4

Test Case Title: Verify that users can sort tasks by multiple attributes

Test Case Description: As a user, ensure that the system allows sorting tasks by various attributes.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in
- Tasks with different attributes (due dates, priorities, assignees) exist

Test Data: Tasks with varying attributes

Test Steps:

1. Navigate to the task list
2. Open the sort options
3. Select a task attribute to sort by
4. Apply the sorting

Test Postconditions:

- The task list displays tasks in sorted order based on the selected attribute

Expected Result: The system sorts tasks based on the selected attribute.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case Creation Template for Task Management Tool - Task Filtering and Sorting Feature:

Test Case ID: TFS5

Test Case Title: Verify that users can save and apply custom filter and sort settings

Test Case Description: As a user, ensure that the system allows saving and applying custom filter and sort settings.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in
- Tasks with different attributes (due dates, priorities, assignees) exist

Test Data: Custom filter and sort settings

Test Steps:

1. Navigate to the task list
2. Open the filter and sort options
3. Customize filter and sort settings
4. Save the custom settings
5. Apply the custom settings

Test Postconditions:

- The task list displays tasks based on the saved and applied custom filter and sort settings

Expected Result: The system saves and applies the custom filter and sort settings.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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:**

Test Case 1:

Test Case ID: TT001

Test Case Title: Verify that users can start and stop a timer for each task

Test Case Description: Ensure that users can start, pause, and stop a timer accurately for individual tasks.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has a task assigned

Test Data: A single task

Test Steps:

1. Go to the task page
2. Click on the 'Start Timer' button
3. Wait for 5 minutes, then click on 'Pause Timer'
4. Wait for another 5 minutes, then click on 'Resume Timer'
5. After 5 more minutes, click on 'Stop Timer'

Postconditions:

- Timer records the elapsed time accurately (15 minutes)

Expected Result: The system starts, pauses, resumes, and stops the timer, displaying a total time of 15 minutes for the task.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TT002

Test Case Title: Verify that the system records and displays total time spent on tasks

Test Case Description: Ensure the system records the time spent on tasks by users accurately and displays the total time on the task page.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has a task assigned

Test Data: A single task

Test Steps:

1. Start the timer for the task and work for 30 minutes
2. Stop the timer
3. Navigate back to the task page

Postconditions:

- Total time spent on the task is displayed (30 minutes)

Expected Result: The system records the time spent and displays the total time (30 minutes) on the task page.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TT003

Test Case Title: Verify that users can manually enter or adjust time entries

Test Case Description: Ensure users can input or modify time entries manually when necessary.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has a task assigned

Test Data: A single task, time data for manual input

Test Steps:

1. Go to the task page
2. Click 'Add Time Manually'
3. Enter a time value (e.g., 20 minutes) and a description
4. Save the time entry
5. Verify the new time entry is displayed on the task page

Postconditions:

- Manual time entry displayed on the task page

Expected Result: The system allows users to enter a time value manually, saving and displaying the time entry on the task page.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TT004

Test Case Title: Verify that time tracking data can be exported

Test Case Description: Ensure users can export time tracking data as a report for analysis and record keeping.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

- User is logged in

- User has a task with time entries

Test Data: Time tracking data for a project or multiple tasks

Test Steps:

1. Go to the time tracking dashboard
2. Click 'Export'
3. Select the desired format (e.g., CSV, PDF)
4. Save the exported file

Postconditions:

- Exported file contains time tracking data

Expected Result: The system exports time tracking data as a report, retaining all relevant information in the selected format.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TT005

Test Case Title: Verify that the system provides reports on time spent per task or project

Test Case Description: Ensure the system generates reports on time spent on individual tasks or entire projects.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has a task or project with time entries

Test Data: Tasks or projects with time tracking data



Test Steps:

1. Go to the time tracking dashboard
2. Select a task or project
3. Generate a report for the selected task or project

Postconditions:

- Report is generated for time spent on the chosen task or project

Expected Result: The system generates a report with detailed time tracking information for the selected task or project.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Test Case ID: TMT-001

Test Case Title: Verify that users can create and save task templates

Test Case Description: Ensure that the system allows users to create and save new task templates with predefined task structures, descriptions, assignees, and due dates.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in
- User has the necessary permissions

Test Data: New task template details (name, structure, description, assignees, due dates)

Test Steps:

1. Go to the task templates section
2. Click on "Create New Template"
3. Enter the template name, structure, description, assignees, and due dates
4. Save the new task template

Postconditions:

- The new task template is saved and displayed in the list of available templates

Expected Result: The system creates and saves the new task template with all the provided information, making it available for future use.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-002

Test Case Title: Verify that templates can include predefined task structures, descriptions, assignees, and due dates

Test Case Description: Ensure that task templates can be created with predefined task structures, descriptions, assignees, and due dates, allowing users to streamline task creation.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in
- User has the necessary permissions

Test Data: Existing task template with predefined task structure, description, assignees, and due dates

Test Steps:

1. Go to the task templates section
2. Select an existing task template with predefined details
3. Use the template to create a new task or project

Postconditions:

- New tasks or projects are created with the predefined task structure, descriptions, assignees, and due dates from the template

Expected Result: The system applies the predefined task structures, descriptions, assignees, and due dates from the template, streamlining task creation and ensuring consistency.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-003

Test Case Title: Verify that templates can be applied to new tasks or projects

Test Case Description: Ensure that users can apply task templates to new tasks or projects, saving time and ensuring consistency.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in
- User has the necessary permissions
- A task template with predefined details exists

Test Data: No test data needed

Test Steps:

1. Go to the task creation page
2. Select a task template from the list of available templates
3. Create a new task with the selected template

Postconditions:

- A new task is created using the predefined structure, descriptions, assignees, and due dates from the template

Expected Result: The system applies the selected task template to the new task, ensuring consistency and streamlining the task creation process.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-004

Test Case Title: Verify that templates can be edited or deleted as needed

Test Case Description: Ensure that users can edit and delete task templates to maintain the relevance and

accuracy of template information.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- User is logged in
- User has the necessary permissions
- An existing task template is available

Test Data: Existing task template with updated details

Test Steps:

1. Go to the task templates section
2. Select an existing task template
3. Edit the template name, structure, description, assignees, or due dates
4. Save the changes

Test Steps for deleting a template:

1. Go to the task templates section
2. Select an existing task template
3. Click on the "Delete Template" button

Postconditions:

- The task template is updated with the provided changes or deleted from the system

Expected Result: The system allows users to edit or delete task templates, maintaining the relevance and accuracy of template information.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-005

Test Case Title: Verify that the system displays a list of available templates for selection

Test Case Description: Ensure that the system displays a list of available task templates for users to select from during task creation.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- User is logged in
- The system contains at least one task template

Test Data: No test data needed

Test Steps:

1. Go to the task creation page
2. Look for the list of available task templates

Postconditions:

- A list of available task templates is displayed

Expected Result: The system displays a list of available task templates for users to choose from during task creation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template:

Test Case ID: CalIntTCS01

Test Case Title: Verify that users can sync tasks with external calendars

Test Case Description: This test case verifies that users can sync their tasks with external calendar applications such as Google Calendar and Outlook.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged in
- Task(s) are created
- Calendar application (e.g., Google Calendar, Outlook) is configured

Test Data: Created tasks with deadlines and descriptions

Test Steps:

1. Go to task management page
2. Select one or more tasks
3. Click on "Sync with Calendar" button
4. Follow the instructions to link the task management tool with the calendar application
5. Verify the tasks are added to the calendar

Postconditions:

- Tasks are added to the calendar
- Tasks are updated in the calendar if changes are made in the task management tool

Expected Result: Users can sync tasks with external calendar applications and view deadlines in their calendar, with updates automatically reflected in both the calendar and the task management tool.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

---

Test Case Creation Template:

Test Case ID: CalIntTCS02

Test Case Title: Verify that the system displays task deadlines in the calendar view

Test Case Description: The test case checks if the task deadlines are shown accurately in the calendar view, allowing users to manage their time effectively.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:



- User is logged in
- Task(s) are created
- Tasks are synced with the calendar

Test Data: Created tasks with deadlines

Test Steps:

1. Go to calendar view
2. Check for presence of task deadlines
3. Verify the date and time of the task deadlines

Postconditions:

- Task deadlines are displayed
- Task deadlines match the calendar events

Expected Result: The task deadlines are visible in the calendar view and are accurate, helping users manage their time efficiently.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case Creation Template:

Test Case ID: CalIntTCS03

Test Case Title: Verify that users can set reminders for tasks via the calendar

Test Case Description: This test case verifies that users can set reminders for tasks in the calendar, ensuring timely completion of tasks.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks are synced with the calendar

Test Data: Created tasks

Test Steps:

1. Go to calendar view
2. Select a task
3. Click on "Add Reminder" button
4. Set the reminder time and date
5. Save the reminder
6. Verify if the reminder is set in the calendar

Postconditions:

- Reminder is set for the task
- Reminder time and date are accurate

Expected Result: Users can set reminders for tasks via the calendar, ensuring timely completion of tasks and improving productivity.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case Creation Template:

Test Case ID: CalIntTCS04

Test Case Title: Verify that calendar events update automatically when tasks are changed

Test Case Description: This test case checks whether calendar events associated with tasks are updated

automatically when changes are made to the tasks in the task management tool.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged in
- Tasks are synced with the calendar
- Task(s) are modified (date, time or description)

Test Data: Modified tasks

Test Steps:

1. Go to task management page
2. Modify tasks (date, time or description)
3. Verify if sync is active
4. Check on calendar events for the same tasks

Postconditions:

- Calendar events are updated
- Updated information is correct

Expected Result: Calendar events associated with tasks are updated automatically when changes are made to the tasks in the task management tool, providing a unified view of tasks and appointments.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

---

Test Case Creation Template:

Test Case ID: CalIntTCS05

Test Case Title: Verify that users can manage calendar sync settings within the app

Test Case Description: This test case verifies that users can manage calendar sync settings within the task management tool easily, allowing customization of the syncing experience.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- User is logged in

Test Steps:

1. Log in as a user
2. Access the "Sync Settings" or related menu
3. Configure sync settings (i.e. direction of sync, supported resolutions)
4. Apply and save changes
5. Verify that calendar sync settings have been updated

Postconditions:

- Calendar sync settings match the applied configuration

Expected Result: Users can manage calendar sync settings within the task management tool easily, allowing them to customize their syncing experience.

Severity: Major

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:**

Test Case ID: DupTsk1

Test Case Title: Verify that users can duplicate tasks

Test Case Description: This test case verifies that users can duplicate an existing task while retaining all original details, including descriptions, assignees, and deadlines.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in
- A task with details is created

Test Data: Existing task with descriptions, assignees, and deadlines

Test Steps:

1. Navigate to the task list
2. Select a task to duplicate
3. Click on the duplicate button
4. Confirm duplication

Test Postconditions:

- A new task is created with identical details as the original task

Expected Result: The system successfully duplicates the task, and the duplicated task is displayed in the task list, indicating that it has been duplicated.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: DupTsk2

Test Case Title: Verify that duplicated tasks retain original details

Test Case Description: This test case confirms that when a task is duplicated, it retains all the original details, descriptions, assignees, and deadlines.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

- User is logged in
- A task with details, assignees, and deadlines is created

Test Data: Existing task with descriptions, assignees, and deadlines

Test Steps:

1. Duplicate the task as per test case DupTsk1
2. Verify the new task's details

Test Postconditions:

- The new task contains identical details as the original task

Expected Result: The duplicated task retains all original details, descriptions, assignees, and deadlines.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: DupTsk3

Test Case Title: Verify that users can modify duplicated tasks

Test Case Description: This test case checks whether users can modify the duplicated task if necessary.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

- User is logged in
- A task with details, assignees, and deadlines is duplicated

Test Data: Duplicated task, user inputs for modifications

Test Steps:

1. Modify the duplicated task details
2. Save the changes

Test Postconditions:

- Duplicated task's details are updated

Expected Result: The system allows users to modify the duplicated task, and the updates are saved.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: DupTsk4

Test Case Title: Verify that the system displays clear duplication indication

Test Case Description: This test case ascertains that the system clearly indicates that a task has been duplicated.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

- User is logged in
- A task is duplicated

Test Data: Original and duplicated task

Test Steps:

1. Compare the original and duplicated tasks

Test Postconditions:

- The system shows notifications or indications that a task is duplicated

Expected Result: The system clearly indicates that a task has been duplicated, either through visual notifications or clear labeling.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case ID: DupTsk5

Test Case Title: Verify that duplicated tasks are saved and displayed

Test Case Description: This test case confirms that the duplicated tasks are saved and displayed within the task list.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in
- A task is duplicated



Test Data: Original and duplicated task

Test Steps:

1. Navigate to the task list

Test Postconditions:

- Duplicated task is present in the task list

Expected Result: The duplicated task is saved in the system and is displayed within the task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: CHnG7Z

Test Case Title: Verify that users can view a detailed history of changes to tasks

Test Case Description: This test case verifies that users can access and view the detailed history of modifications, comments, and other relevant activities related to tasks.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in
- A task is created

Test Data: Existing task with modifications, comments, and attachments

Test Steps:

1. Navigate to the task details page
2. Locate and click on the "Task History" button
3. Review modifications, comments, and other activities

Test Postconditions:

- Task history is displayed

Expected Result: The system displays a detailed task history, including all modifications, comments, and other relevant activities with timestamps.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

---

Test Case ID: JK3G8A

Test Case Title: Verify that the history includes modifications, comments, and other relevant activity

Test Case Description: This test case checks if the task history includes various types of changes and user interactions such as modifications, comments, and attachments.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in
- A task is created

Test Data: Existing task with modifications, comments, and attachments

Test Steps:

1. Create a new task
2. Modify the task description
3. Add a comment to the task
4. Upload an attachment to the task
5. Access the task history
6. Check for the appearance of each action with a timestamp

Test Postconditions:

- Task history is displayed

Expected Result: The system displays a detailed task history, including modifications, comments, and attachments with timestamps.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

---

Test Case ID: LS2F5D

Test Case Title: Verify that users can filter the history by type of change or date

Test Case Description: This test case confirms that users have the ability to filter the task history by specific types of changes or by date.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in
- A task is created
- Task history includes various types of changes and user interactions

Test Data: Existing task history with multiple change types and dates

Test Steps:

1. Navigate to the task details page
2. Locate and click on the "Task History" button
3. Utilize filter options to narrow the history view by type or date

Test Postconditions:

- Filtered task history is displayed

Expected Result: The system allows users to filter the task history by specific types of changes (e.g., modifications, comments, attachments) or by date.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

---

Test Case ID: PL4Q9R

Test Case Title: Verify that the system provides timestamps for all changes

Test Case Description: This test case ensures that the system provides accurate and consistent timestamps for all changes, comments, and user interactions within the task history.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in
- A task is created
- Task history includes various types of changes and user interactions

Test Data: Existing task history with multiple change types and dates

Test Steps:

1. Navigate to the task details page
2. Locate and click on the "Task History" button
3. Verify timestamp presence for each activity in the history

Test Postconditions:

- Task history is displayed with timestamps

Expected Result: The system displays the task history with accurate and consistent timestamps for each activity.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

---

Test Case ID: ZE5K1T

Test Case Title: Verify that task history is accessible from the task details page

Test Case Description: This test case confirms that the task history feature is easily accessible from the task details page.

Test Suite: Task History

Test Priority: Low

Preconditions:

- User is logged in
- A task is created

Test Data: No test data needed

Test Steps:

1. Navigate to a task details page
2. Locate and click on the "Task History" button

Test Postconditions:

- Task history is displayed

Expected Result: The system provides a visible and accessible link to the task history from the task details page.

Severity: Minor

Type of Testing: Usability

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:**

Test Case ID: TMN-TN-001

Test Case Title: Verify that users receive notifications for task updates

Test Case Description: This test case verifies that the system delivers notifications when task updates occur.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- A task assigned to the user exists

Test Data: Task updates

#### Test Steps:

1. Log in as a user
2. Access the assigned task
3. Perform a task update (e.g., change the status, due date, or priority)
4. Monitor for notifications via email, mobile push notifications, or in-app alerts

#### Postconditions:

- Notification is received for the task update

Expected Result: The system sends a notification regarding the task update to the user according to their notification preferences.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMN-TN-002

Test Case Title: Verify customization of notification settings

Test Case Description: This test case verifies that users can customize their notification preferences based on their personal preferences.

Test Suite: Task Notifications

Test Priority: Medium

#### Preconditions:

- User is logged in

Test Data: Notification preferences

#### Test Steps:

1. Log in as a user
2. Access the account settings
3. Navigate to the notification preferences section
4. Enable or disable desired notification types (email, mobile push, or in-app)



## 5. Save the updated settings

### Postconditions:

- Notification settings are updated

Expected Result: The system allows users to customize their notification preferences and saves these settings accordingly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMN-TN-003

Test Case Title: Verify detailed notification information

Test Case Description: This test case verifies that notifications provide detailed information about the task activity.

Test Suite: Task Notifications

Test Priority: High

### Preconditions:

- User is logged in
- A task assigned to the user exists

Test Data: Task updates

### Test Steps:

1. Log in as a user
2. Access the assigned task
3. Perform a task update (e.g., change the status, due date, or priority)
4. Receive the generated notification
5. Review the notification content

### Postconditions:

- Notification is received with detailed information

Expected Result: The notification contains accurate and detailed information regarding the task activity, including the task name, relevant actions, and the person responsible for the update.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMN-TN-004

Test Case Title: Verify notification history logging

Test Case Description: This test case verifies that the system logs a history of user notifications for future reference.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in
- A task assigned to the user exists

Test Data: Task updates

Test Steps:

1. Log in as a user
2. Access the assigned task
3. Perform a task update (e.g., change the status, due date, or priority)
4. Monitor the notification history

Postconditions:

- Notification is added to the notification history

Expected Result: The system logs task update notifications with the appropriate timestamp, task details, and user information, and maintains this record for future reference.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: TMN-TN-005

Test Case Title: Verify that notification delivery handles unstable network conditions

Test Case Description: This test case verifies that notification delivery adapts to varying network conditions while maintaining accuracy and reliability.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in
- A task assigned to the user exists

Test Steps:

1. Log in as a user
2. Access the assigned task
3. Perform a task update (e.g., change the status, due date, or priority)
4. Simulate varying network conditions (stable to unstable)
5. Monitor for notification deliveries

Postconditions:

- Notifications are delivered according to network conditions

Expected Result: The system handles varying network conditions while attempting to deliver notifications, ensuring accurate and reliable delivery when possible, and logging undelivered notifications for later delivery or review.

Severity: Major

Type of Testing: Non-Functional Testing (Performance Testing)

Test Case Approach: Negative

-----

**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:**

Test Case ID: archiveTask01

Test Case Title: Verify that users can archive completed tasks

Test Case Description: As a user, I want to be able to archive a task that has been marked as complete, so that it is moved out of my active task list and into a separate section for easy access.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in
- A task has been created and marked as complete

Test Data: A single completed task

Test Steps:

1. Go to the active task list
2. Select a completed task
3. Click on the archive button
4. Confirm the archiving of the task

Postconditions:

- The task is removed from the active task list
- The task appears in the archived task section

Expected Result: The system moves the selected completed task from the active task list to the archived task section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: archiveTask02

Test Case Title: Verify that archived tasks are moved to a separate section

Test Case Description: As a user, I want to be able to find all my archived tasks in a separate section, so that I can easily access them if needed while keeping my active task list focused on tasks that require attention.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in

- A task has been archived

Test Data: An archived task

Test Steps:

1. Go to the archived task section
2. Verify that the archived task appears in the list

Postconditions:

- The archived task appears in the archived task section

Expected Result: The system displays the archived task in the separate archived task section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: archiveTask03

Test Case Title: Verify that users can search and view archived tasks

Test Case Description: As a user, I want to be able to search for and view specific archived tasks, so that I can quickly access the information or details I need.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in
- A task has been archived

Test Data: An archived task

Test Steps:

1. Go to the archived task section
2. Use the search function to find the archived task
3. Verify that the archived task appears in the search results

Postconditions:

- The archived task appears in the search results

Expected Result: The system allows users to search for and view archived tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: archiveTask04

Test Case Title: Verify that users can restore archived tasks to the active list

Test Case Description: As a user, I want to be able to restore an archived task back into my active task list, so that I can continue working on it if needed.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in
- A task has been archived

Test Data: An archived task

Test Steps:

1. Go to the archived task section
2. Select an archived task
3. Click on the restore button
4. Confirm the restoring of the task

Postconditions:

- The task is removed from the archived task section
- The task appears in the active task list

Expected Result: The system moves the selected archived task back into the active task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: archiveTask05

Test Case Title: Verify that archived tasks retain all original details and history

Test Case Description: As a user, I want to make sure that when a task is archived, all its details, descriptions, attachments, deadlines, comments, and history are preserved, so that I can refer back to the information as needed.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in
- A task has been archived with details, descriptions, attachments, deadlines, comments, etc.

Test Data: An archived task with various details attached

Test Steps:

1. Go to the archived task section
2. Select the archived task with various details attached
3. Verify that all details, descriptions, attachments, deadlines, comments, etc., are visible and accessible

Postconditions:

- All original details and history are visible and accessible

Expected Result: The system preserves all details, descriptions, attachments, deadlines, comments, etc., when a task is archived.

Severity: High

Type of Testing: Functional

Test Case Approach: Positive

-----



**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:**

Test Case ID: GSTC1

Test Case Title: Verify that users can invite external collaborators as guests

Test Case Description: Ensure that the system allows users to invite external collaborators as guests with limited access.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in
- User has a project or task to share

Test Data: Email address of external collaborator, Project or task details

Test Steps:

1. Go to the project or task
2. Click on "Share" or "Invite Guests" option
3. Enter the email address of the external collaborator
4. Set guest permissions (view-only or edit)
5. Send the invitation

Postconditions:

- Invitation is sent to the guest

Expected Result: The system sends an invitation to the external collaborator and grants them the specified level of access to the shared project or task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: GSTC2

Test Case Title: Verify that guests can view and collaborate on specific tasks or projects

Test Case Description: Ensure that guests can view and collaborate on specific tasks or projects according to their assigned permissions.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User has invited a guest

- Guest has accepted the invitation

Test Data: Guest account, Project or task details

Test Steps:

1. Log in as the guest
2. Access the shared project or task
3. Verify access to task information based on assigned permissions
4. Attempt to perform allowed collaboration actions (e.g., comment, upload attachments)
5. Verify successful completion of collaboration actions

Postconditions:

- Guest can view and collaborate on shared tasks or projects as allowed

Expected Result: The guest can view and collaborate on tasks or projects according to the assigned permissions, and all collaboration actions are successful.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: GSTC3

Test Case Title: Verify that users can manage and revoke guest access at any time

Test Case Description: Ensure that users can manage and revoke guest access at any time, without any issues.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User has invited a guest
- Guest has accepted the invitation

Test Data: Guest account, Project or task details

Test Steps:

1. Log in as the user

2. Access the shared project or task
3. Revoke guest access
4. Verify removal of guest from the project or task
5. Verify that the guest cannot access the project or task

Postconditions:

- User has successfully revoked guest access

Expected Result: The user can manage and revoke guest access at any time, and the guest loses access to the project or task once their access has been revoked.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: GSTC4

Test Case Title: Verify that guests cannot access sensitive or restricted areas of the platform

Test Case Description: Ensure that guests cannot access sensitive or restricted areas of the platform even if they possess the direct URLs.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User has invited a guest
- Guest has accepted the invitation

Test Data: Guest account, Direct URLs for restricted areas

Test Steps:

1. Log in as the guest
2. Enter direct URLs for restricted areas (if available)

Postconditions:

- Guest cannot access restricted areas

Expected Result: Guests cannot access sensitive or restricted areas of the platform, even if they possess the direct URLs.

Severity: Major

Type of Testing: Security

Test Case Approach: Negative

Test Case ID: GSTC5

Test Case Title: Verify that the system provides clear distinctions between guest and full access roles

Test Case Description: Ensure that the system clearly distinguishes between guest and full access roles through visual cues or labels.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in
- User has a project or task
- Guest has accepted the invitation to the shared project or task

Test Data: Project or task list, User and guest accounts

Test Steps:

1. Log in as the user
2. Access the project or task list
3. View shared projects or tasks
4. Observe the visual cues or labels distinguishing between user and guest access

Postconditions:

- Clear distinctions are visible

Expected Result: The system provides clear distinctions between user and guest roles in project or task lists through visual cues or labels.

Severity: Major

Type of Testing: Usability

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:**

Test Case ID: impEx001

Test Case Title: Verify that users can import tasks from external sources in CSV format

Test Case Description: This test case aims to ensure that users can import tasks from external sources such as CSV files.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

- User is logged in
- A valid CSV file with task data is prepared

Test Data: Prepared CSV file with task data

Test Steps:

1. Go to the Import Tasks page
2. Choose CSV as the import format
3. Upload the prepared CSV file
4. Map the fields during import
5. Preview the imported data
6. Finalize the import process

Postconditions:

- Tasks are imported
- Imported tasks are visible in the task list

Expected Result: The system successfully imports the tasks from the prepared CSV file, and the imported tasks are displayed in the task list with correct mapping of fields.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: impEx002

Test Case Title: Verify that users can import tasks from external sources in Excel format

Test Case Description: This test case verifies that users can import tasks from external sources such as Excel files.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

- User is logged in
- A valid Excel file with task data is prepared

Test Data: Prepared Excel file with task data

Test Steps:

1. Go to the Import Tasks page
2. Choose Excel as the import format
3. Upload the prepared Excel file
4. Map the fields during import
5. Preview the imported data
6. Finalize the import process

Postconditions:

- Tasks are imported
- Imported tasks are visible in the task list

Expected Result: The system successfully imports the tasks from the prepared Excel file, and the imported tasks are displayed in the task list with correct mapping of fields.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: impEx003

Test Case Title: Verify that users can export tasks to other task management tools or file formats



Test Case Description: This test case checks that users can export tasks to other task management tools or file formats.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

- User is logged in
- Tasks are created in the task management tool

Test Data: Created tasks

Test Steps:

1. Go to the Export Tasks page
2. Choose the desired export format
3. Confirm the export process
4. Save the exported file

Postconditions:

- Tasks are exported
- Exported file can be opened in the chosen tool or format

Expected Result: The system successfully exports the tasks in the chosen format, and the exported file can be opened and viewed in the desired task management tool or application.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: impEx004

Test Case Title: Verify that the system provides clear mapping of fields during import/export

Test Case Description: This test case aims to confirm that the system provides clear mapping of fields during import and export processes.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

- User is logged in
- Task data is prepared for import

Test Data: Prepared task data

Test Steps:

1. Go to the Import or Export Tasks page
2. Begin the import or export process
3. Observe the field mapping options
4. Map or match the fields as necessary
5. Preview the mapping choices

Test Steps:

- Task fields match clearly during import and export

Postconditions:

- Field mapping is accurate

Expected Result: The system offers clear and accurate mapping of task fields during the import and export processes, ensuring proper data transfer.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: impEx005

Test Case Title: Verify that users can preview imported data before finalizing the process

Test Case Description: This test case tests the ability of users to preview imported data before finalizing the import process.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

- User is logged in
- A valid CSV or Excel file with task data is prepared

Test Data: Prepared CSV or Excel file with task data

Test Steps:

1. Go to the Import Tasks page
2. Upload the prepared CSV or Excel file
3. Begin the import process
4. Preview the imported data
5. Confirm or edit the import as necessary

Postconditions:

- Import preview is available
- Data can be edited before finalizing the import

Expected Result: The system allows users to preview imported task data before finalizing the import process, giving them the opportunity to make necessary edits or adjustments.

Severity: Major

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:**

Test Case ID: TMTP01

Test Case Title: Verify that users can set task priorities

Test Case Description: Ensure that the system allows users to set priorities for their tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Task details with various priority levels (e.g., Low, Medium, High)

Test Steps:

1. Go to the task list
2. Select a task

3. Click on the set priority button

4. Choose a priority level

5. Save changes

Postconditions:

- Task priority is updated

Expected Result: The system sets the priority level for the selected task as specified by the user.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: TMTP02

Test Case Title: Verify that tasks can be sorted and filtered based on priority levels

Test Case Description: Ensure that the system allows tasks to be sorted and filtered according to the defined priority levels.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Tasks with various priority levels are created

Test Data: No test data needed

Test Steps:

1. Go to the task list
2. Use the sort or filter options
3. Select priority level as the sorting or filtering criteria

Postconditions:

- Tasks are sorted or filtered by priority

Expected Result: The system sorts or filters tasks based on the selected priority criteria.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: TMTP03

Test Case Title: Verify that the system provides visual indicators of task priority

Test Case Description: Ensure that the system visually represents the priority levels assigned to tasks.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks with various priority levels are created

Test Data: No test data needed

Test Steps:

1. Go to the task list
2. View the tasks

Postconditions:

- Tasks display visual indicators of priority

Expected Result: The system displays visual indicators representative of the assigned priority levels.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: TMTP04

Test Case Title: Verify that users receive notifications for high-priority tasks

Test Case Description: Ensure that the system sends notifications to users when assigned high-priority tasks.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in
- High-priority task is assigned to the user

Test Data: No test data needed

Test Steps:

1. Go to the task notifications
2. Check for new notifications

Postconditions:

- Notification for high-priority task is received

Expected Result: The system sends a notification to the user for the assigned high-priority task.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: TMTP05

Test Case Title: Verify that priority changes are logged and tracked

Test Case Description: Ensure that the system logs and tracks priority changes made to tasks.

Test Suite: Task Prioritization

Test Priority: Low

Preconditions:

- User is logged in
- Task with a priority level is created

Test Data: Task priority change details

Test Steps:

1. Go to the task details
2. Change the task priority

### 3. Save changes

Postconditions:

- Priority change is logged and tracked

Expected Result: The system logs the priority change and tracks the revision history of the task.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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:**

Test Case ID: TMT-CC01

Test Case Title: Verify that users can assign tasks to team members

Test Case Description: This test case verifies that the system allows users to assign tasks to other team members within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- A task has been created

Test Data: Team member information

Test Steps:

1. Go to the task details page
2. Click on the "Assign" button
3. Choose a team member from the list
4. Save the changes

Postconditions:

- The task is assigned to the selected team member

Expected Result: The system successfully assigns the task to the chosen team member, and the assignee receives a notification.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC02

Test Case Title: Verify that users can share updates and communicate through comments or messages

Test Case Description: This test case checks if users can share updates and communicate with their team members using comments or messages within the task management tool.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- A task has been created and assigned to at least one team member

Test Data: Task update and a comment

Test Steps:

1. Go to the task details page
2. Add an update to the task
3. Post a comment related to the task update
4. Verify that the assigned team member can view the update and the comment

Postconditions:

- Task update and comment are visible to the assigned team member

Expected Result: The system allows users to share updates and communicate through comments or messages, and the assigned team member can view them.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC03

Test Case Title: Verify that task updates and comments are visible to assigned team members

Test Case Description: This test case ensures that task updates and comments shared by users are visible to the

assigned team members.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- A task has been created and assigned to at least one team member
- Task update and comment have been added by a user

Test Data: No test data needed

Test Steps:

1. Go to the task details page
2. Verify that the assigned team member can view the task update and the comment

Postconditions:

- Task update and comment are visible to the assigned team member

Expected Result: The system displays task updates and comments to the assigned team members.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC04

Test Case Title: Verify that users receive notifications for task updates and comments

Test Case Description: This test case asserts that users receive notifications for task updates and comments within the task management tool.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- A task has been created and assigned to at least one team member

- Notifications are enabled for the user

Test Data: Task update and a comment

Test Steps:

1. Add an update to the task
2. Post a comment related to the task update
3. Verify that the assigned team member receives a notification for the update and the comment

Postconditions:

- Assigned team member receives a notification for the update and the comment

Expected Result: The system notifies users of task updates and comments through email or in-app notifications, as configured.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC05

Test Case Title: Verify that the system provides a clear view of task collaboration history

Test Case Description: This test case checks whether users can view a clear history of task collaboration, including updates, comments, and assignments.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- A task has been created and assigned to at least one team member
- Task updates, comments, and assignments have been made

Test Data: No test data needed

Test Steps:

1. Go to the task details page

2. Verify that the "Collaboration History" section displays all task updates, comments, and assignments

Postconditions:

- Task collaboration history is visible and clear

Expected Result: The system shows a comprehensive and easy-to-understand history of task collaboration.

Severity: Major

Type of Testing: Functional

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:

Test Case ID: tmtP1

Test Case Title: Verify that users can view task statuses

Test Case Description: This test case verifies that the system allows users to view the status of tasks within a project or category.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in
- Project or category with tasks is created

Test Data: Project or category with tasks

Test Steps:

1. Navigate to the project or category
2. Scroll through the list of tasks
3. Check the status of each task

Postconditions:

- Task statuses are displayed

Expected Result: The system displays the status of each task within the project or category.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case ID: tmtP2

Test Case Title: Verify that users can track task progress and deadlines

Test Case Description: This test case ensures that the system allows users to monitor the progress and deadlines of tasks.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in
- Project or category with tasks is created

Test Data: Project or category with tasks and their deadlines

Test Steps:

1. Navigate to the project or category
2. Select a task to view its details
3. Check the progress and deadline information

Postconditions:

- Task progress and deadline are displayed

Expected Result: The system displays the progress and deadline of the selected task, indicating whether it is on schedule or not.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case ID: tmtP3

Test Case Title: Verify that tasks are updated in real-time

Test Case Description: This test case confirms that the system updates task statuses in real-time as users make progress on tasks.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in
- Project or category with tasks is created

- User has started working on a task

Test Data: Task being worked on

Test Steps:

1. Navigate to the project or category
2. Begin working on a task
3. Save the progress periodically
4. Refresh the task list

Postconditions:

- Task status reflects the latest progress

Expected Result: The system updates the task status to reflect the user's latest progress as they save the task periodically.

Severity: Major

Type of Testing: Functional Testing, Real-time Testing

Test Case Behaviour: Positive

Test Case ID: tmtP4

Test Case Title: Verify that the system provides visual indicators for task progress

Test Case Description: This test case checks that the system provides visual cues, like progress bars, to show users the current state of a task.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Project or category with tasks is created

Test Data: Project or category with tasks

Test Steps:

1. Navigate to the project or category



2. Scroll through the list of tasks

3. Check for visual indicators of task progress

Postconditions:

- Visual indicators represent task progress

Expected Result: The system displays visual indicators, such as progress bars, that accurately show the current progress of each task.

Severity: Major

Type of Testing: Functional Testing, User Interface Testing

Test Case Behaviour: Positive

Test Case ID: tmtP5

Test Case Title: Verify that the system handles task status changes during maintenance windows

Test Case Description: This test case checks that the system handles task status changes during maintenance windows to prevent data loss.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Project or category with tasks is created
- System maintenance window is scheduled

Test Data: Project or category with tasks

Test Steps:

1. Navigate to the project or category
2. Update task statuses at the start of the maintenance window
3. Await the completion of the maintenance window
4. Refresh the task list

Postconditions:

- Task status changes are saved

Expected Result: The system saves task status changes during the maintenance window and reflects the updated states after the maintenance window ends.

Severity: Major

Type of Testing: Functional Testing, Maintenance Testing

Test Case Behaviour: Negative (to test maintenance handling)

-----

## **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:

Test Case ID: TN001

Test Case Title: Verify that users receive notifications for new tasks

Test Case Description: Ensure that the system sends a notification to the user when a new task is assigned to them.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- Another user has access to the current user's email address

Test Data: Task details including the assignee

Test Steps:

1. Log in as a different user
2. Create a new task and assign it to the first user
3. Monitor the first user's email for a notification

Test Postconditions:

- Notification is present in the first user's email

Expected Result: The system sends a notification to the assigned user's email upon task creation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TN002

Test Case Title: Verify that users receive notifications for task updates

Test Case Description: Ensure that the system sends a notification to the user when a task assigned to them is updated.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in
- User has at least one assigned task

Test Data: Updated task details

Test Steps:

1. Log in as the user
2. Update a task assigned to the user
3. Monitor the user's notifications for an update

Test Postconditions:

- Notification is present in the user's notifications

Expected Result: The system sends a notification to the user when a task assigned to them is updated.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TN003

Test Case Title: Verify that users receive notifications for task comments

Test Case Description: Ensure that the system sends a notification to the user when a comment is added to a task assigned to them.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in
- User has at least one assigned task
- Another user has access to the current user's email address

Test Data: Comment details

Test Steps:

1. Log in as another user
2. Add a comment to a task assigned to the first user
3. Monitor the first user's email for a notification

Test Postconditions:

- Notification is present in the first user's email

Expected Result: The system sends a notification to the user when a comment is added to a task assigned to them.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TN004

Test Case Title: Verify that users can customize notification settings

Test Case Description: Ensure that the system allows users to customize their notification preferences.

Test Suite: Task Notifications

Test Priority: Low

Preconditions:

- User is logged in

Test Data: Notification preferences

Test Steps:

1. Log in as the user
2. Access notification settings
3. Modify notification preferences

Test Postconditions:

- Notification settings are updated

Expected Result: The system allows users to customize their notification preferences.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TN005

Test Case Title: Verify that users receive reminders for upcoming deadlines

Test Case Description: Ensure the system sends a reminder notification to the user before a task's deadline.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has at least one task with a deadline

Test Data: Task with a deadline within 24 hours

Test Steps:

1. Log in as the user
2. Monitor notifications for a deadline reminder

Test Postconditions:

- Notification is present in the user's notifications

Expected Result: The system sends a reminder notification to the user before a task's deadline.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template:

Test Case ID: TMT-CC1

Test Case Title: Verify that users can create and assign tasks to different categories or projects

Test Case Description: Ensure that users can create tasks and assign them to specific categories or projects within the task management tool.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in
- User has access to create tasks
- Categories or projects are already created

Test Data: Task details and category/project names

Test Steps:

1. Log in as a user
2. Navigate to the task creation page
3. Enter task details
4. Select a category or project to assign the task
5. Save or submit the task

Postconditions:

- Task appears under the chosen category or project
- User receives a confirmation message

Expected Result: The system creates the task and assigns it to the chosen category or project.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC2



Test Case Title: Verify that users can view tasks grouped by categories or projects

Test Case Description: Ensure that users can view tasks organized by their assigned categories or projects.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in
- Tasks have been previously created and assigned to categories or projects

Test Data: Existing tasks and categories/projects

Test Steps:

1. Log in as a user
2. Navigate to the task management or task list page
3. Select a category or project from the filter or grouping options

Postconditions:

- Tasks are displayed based on the selected category or project

Expected Result: The system displays tasks grouped by the chosen category or project.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC3

Test Case Title: Verify that tasks can be moved between categories or projects

Test Case Description: Ensure that users can move tasks from one category or project to another.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks have been previously created and assigned to categories or projects

Test Data: Existing tasks and categories/projects

Test Steps:

1. Log in as a user
2. Navigate to the task list or task management page
3. Select a task to be moved
4. Use the "Move" or "Change Category" option
5. Select a new category or project
6. Confirm the move

Postconditions:

- Task appears under the new category or project
- Task disappears from the previous category or project

Expected Result: The system moves the task to the newly selected category or project.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC4

Test Case Title: Verify that category or project information is accurately reflected in task views

Test Case Description: Ensure that all relevant information, such as the category or project name, is correctly displayed for each individual task.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks have been previously created and assigned to categories or projects

Test Data: Existing tasks and categories/projects

Test Steps:

1. Log in as a user
2. Navigate to the task list or task management page
3. Select a task

Postconditions:

- Category or project information appears in the task view

Expected Result: The system accurately displays category or project information in each individual task view, including the task details and task summary pages.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TMT-CC5

Test Case Title: Verify that categories or projects are clearly distinguished from one another in task views

Test Case Description: Ensure that categories or projects are easily distinguishable in task views, allowing users to quickly identify the tasks they need to prioritize, complete, or manage.

Test Suite: Task Categorization

Test Priority: Low

Preconditions:

- User is logged in
- Tasks have been previously created and assigned to categories or projects

Test Data: Existing tasks and categories/projects

Test Steps:

1. Log in as a user
2. Navigate to the task list or task management page
3. View the list of tasks grouped by categories or projects

Postconditions:

- Categories or projects are differentiated by color, icons, names, or other visual cues

Expected Result: The system visually differentiates categories or projects in task views, making it easy for users to identify and prioritize tasks.

Severity: Major

Type of Testing: Functional

**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:**

Test Case ID: TC001\_TaskSearch

Test Case Title: Verify that users can search for tasks by name

Test Case Description: This test case verifies that the system allows users to search for tasks by entering a task

name in the search bar.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in
- Tasks are created within the system

Test Data: A specific task name

Test Steps:

1. Navigate to the task list or project board
2. Enter a specific task name in the search bar
3. Click the 'Search' button

Test Postconditions:

- The system displays a list of tasks that match the entered name

Expected Result: The system returns accurate and relevant search results with tasks containing the entered name.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC002\_TaskSearch

Test Case Title: Verify that users can search for tasks by description

Test Case Description: This test case verifies that the system allows users to search for tasks by entering a keyword from a task description in the search bar.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- User is logged in

- Tasks with descriptions are created within the system

Test Data: A specific keyword from a task description

Test Steps:

1. Navigate to the task list or project board
2. Enter a specific keyword from a task description in the search bar
3. Click the 'Search' button

Test Postconditions:

- The system displays a list of tasks that match the entered keyword in their descriptions

Expected Result: The system returns accurate and relevant search results with tasks containing the entered keyword in their descriptions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC003\_TaskSearch

Test Case Title: Verify that users can filter search results by category, priority, or due date

Test Case Description: This test case verifies that the system allows users to filter search results based on category, priority, or due date after conducting a search.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in
- Tasks with various categories, priorities, and due dates are created within the system

Test Data: Category, priority, or due date

Test Steps:

1. Conduct a search for tasks
2. Use the filter options to select a category, priority, or due date

### 3. Apply the filter

Test Postconditions:

- The system displays the filtered search results

Expected Result: The system displays search results based on the selected category, priority, or due date.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC004\_TaskSearch

Test Case Title: Verify that search results are accurate and relevant

Test Case Description: This test case verifies that the system returns accurate and relevant search results when users search for tasks by name, description, or filter by category, priority, or due date.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in
- Tasks with various names, descriptions, categories, priorities, and due dates are created within the system

Test Data: Task name, description, category, priority, and due date

Test Steps:

1. Search for tasks using a name, description, or a combination of category, priority, and due date
2. Review the search results

Test Postconditions:

- The system displays the search results

Expected Result: The system returns accurate and relevant search results containing the entered text or matching the selected filters.

Severity: Critical

Type of Testing: Functional and Validation



Test Case Approach: Positive

Test Case ID: TC005\_TaskSearch

Test Case Title: Verify that the search bar accepts task-related phrases

Test Case Description: This test case verifies that the search bar accepts task-related phrases instead of exact keywords, providing more flexibility in the search.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks with various names, descriptions, categories, priorities, and due dates are created within the system

Test Data: Task-related phrases

Test Steps:

1. Enter a task-related phrase in the search bar
2. Click the 'Search' button

Test Postconditions:

- The system displays search results based on the entered task-related phrase

Expected Result: The system returns accurate and relevant search results containing the entered task-related phrase.

Severity: Major

Type of Testing: Functional and Usability

Test Case Approach: Positive and Ad-hoc

-----

**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:**

Test Case Creation Template for Task Collaboration Feature:

Test Case ID: TC\_TASKCOLLAB\_001

Test Case Title: Verify that users can add comments to tasks

Test Case Description: This test case verifies the ability to post a comment on a task within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in

- Task is created

Test Data: Comment text

Test Steps:

1. Go to the task details page
2. Enter a comment in the comment box
3. Click on the "Post" button

Postconditions:

- Comment is displayed on the task's comment section

Expected Result: The system allows the user to add a comment to the task successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC\_TASKCOLLAB\_002

Test Case Title: Verify that users can attach files to tasks

Test Case Description: This test case verifies the ability to attach a file to a task within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in

- Task is created

Test Data: Sample file (e.g., .docx, .pdf, .jpg)

Test Steps:

1. Go to the task details page
2. Click on the "Attach File" button

3. Select a file from the device

4. Click on the "Open" button to attach the file

Postconditions:

- File is displayed on the task's file section with the correct name

Expected Result: The system allows the user to attach a file to the task successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC\_TASKCOLLAB\_003

Test Case Title: Verify that users can share tasks with other users

Test Case Description: This test case verifies the ability to share a task with other users within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Other user(s) exists in the system
- Task is created

Test Data: Email address(es) of other user(s)

Test Steps:

1. Go to the task details page
2. Click on the "Share" button
3. Enter the email address(es) in the "Add collaborator(s)" field
4. Click on the "Send Invitation" button

Postconditions:

- Other user(s) receives the invitation

- Other user(s) can access the task

Expected Result: The system allows the user to share the task with other users successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC\_TASKCOLLAB\_004

Test Case Title: Verify task collaboration features enhance team communication

Test Case Description: This test case verifies that the task collaboration features improve team communication within the task management tool.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- User A and User B are logged in
- Task is created and shared between User A and User B

Test Data: Comments, attachments, and task updates

Test Steps:

1. User A adds comments or attachments to a shared task
2. User B receives a notification and responds to the comment
3. User A modifies the task description or deadline
4. User B is notified of the changes
5. Both users collaborate and complete the task

Postconditions:

- Effective collaboration has taken place
- Team communication has been visibly improved

Expected Result: The task collaboration features significantly enhance team communication in managing tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC\_TASKCOLLAB\_005

Test Case Title: Verify that users cannot share tasks with invalid email addresses

Test Case Description: This test case verifies that the system does not allow users to share tasks with invalid email addresses within the task management tool.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created

Test Data: Invalid email address (e.g., non-existing domain)

Test Steps:

1. Go to the task details page
2. Click on the "Share" button
3. Enter an invalid email address in the "Add collaborator"

-----

**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:**

Test Case Creation Template for Course Creation Tools:

Test Case ID: CCf1w

Test Case Title: Verify that instructors can create courses using videos, presentations, quizzes, and assignments

Test Case Description: Ensure that the course creation tools support adding various multimedia elements like videos, presentations, quizzes, and assignments.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor navigates to the course creation page

Test Data: Course details, multimedia files, quiz questions, assignment details

Test Steps:

1. Access the course creation page
2. Provide course details
3. Upload videos, presentations, and other multimedia files
4. Add quiz questions
5. Create and configure assignments
6. Save course draft

Postconditions:

- Course is saved as a draft
- All multimedia elements, quizzes, and assignments are properly integrated

Expected Result: The system creates a course with all multimedia elements, quizzes, and assignments intact, ready for preview.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: gvTnR5

Test Case Title: Verify that the course creation tools require minimal technical skills

Test Case Description: Confirm that the course creation tools are user-friendly and intuitive for instructors with minimal technical skills.

Test Suite: Course Creation Tools

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Instructor navigates to the course creation page

Test Data: Course content (text, multimedia), quiz questions, assignment details



Test Steps:

1. Access the course creation page
2. Add course content using text and multimedia
3. Create a quiz
4. Add a basic assignment

Test Postconditions:

- Course content, quiz, and assignment are saved

Expected Result: The system saves the course, quiz, and assignment entries, demonstrating the user-friendly nature of the tools.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case ID: RbMh5d

Test Case Title: Verify that courses can be previewed before publishing

Test Case Description: Ensure that instructors can preview their courses before publishing them.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor navigates to the draft course

Test Data: Course content, multimedia files, quiz questions, assignment details

Test Steps:

1. Access the draft course
2. Open the preview mode
3. Review course content, quizzes, and assignments

Postconditions:

- Course preview is completed

Expected Result: The system displays the course as it would appear to students, allowing the instructor to review the content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: Gr0L4d

Test Case Title: Verify that instructors can save courses as drafts

Test Case Description: Confirm that instructors can save courses as drafts, allowing them to return and complete them later.

Test Suite: Course Creation Tools

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Instructor navigates to the course creation page

Test Data: Course content (text, multimedia), quiz questions, assignment details

Test Steps:

1. Access the course creation page
2. Enter basic course information (title, description)
3. Save course as a draft

Test Postconditions:

- Course is saved as a draft

Expected Result: The system saves the course as a draft, allowing the instructor to resume course creation at a later time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: xjCdJk

Test Case Title: Verify that courses can be edited post-publishing

Test Case Description: Verify that instructors can edit the published courses, allowing them to update and make changes.

Test Suite: Course Creation Tools

Test Priority: Low

Preconditions:

- Instructor is logged in
- Instructor has a previously published course

Test Data: Course updates (text, multimedia, quiz questions, assignment details)

Test Steps:

1. Access the published course
2. Click on the edit button
3. Modify course content, quizzes, or assignments
4. Save changes

Test Postconditions:

- Changes are saved and applied to the course

Expected Result: The system updates the published course with the modifications, ensuring the course information is always current.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: olPf1

Test Case Title: Verify that lessons include quizzes, polls, simulations, and discussions

Test Case Description: Ensure interactive elements such as quizzes, polls, simulations, and discussions are included in the lesson content.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in
- Course is selected

Test Data: Course with interactive lessons

Test Steps:

1. Select a course
2. Access a lesson
3. Verify the presence of quizzes, polls, simulations, and discussions
4. Interact with each element

Postconditions:

- Interactive elements are present and usable

Expected Result: The lesson includes quizzes, polls, simulations, and discussions, which can be interacted with by the learner.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olPf2

Test Case Title: Verify that interactive elements are integrated seamlessly into the lesson content

Test Case Description: Validate that interactive components are integrated seamlessly into lesson content without disrupting the learning flow.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in
- Course with interactive lessons is selected

Test Data: Course with integrated interactive lessons

Test Steps:

1. Select a course
2. Access a lesson with integrated interactive lessons
3. Assess the flow and consistency of the interactive elements

Postconditions:

- Interactive elements blend well with the lesson content

Expected Result: Interactive elements are integrated seamlessly into the lesson content and do not disrupt the learning flow.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: olPf3

Test Case Title: Verify that learners can track their progress through these activities

Test Case Description: Validate that learners can track progress through quizzes, polls, simulations, and discussions.

Test Suite: Interactive Lessons

Test Priority: Medium

Preconditions:

- User is logged in
- Course with interactive lessons is selected

Test Data: No test data needed

Test Steps:

1. Select a course
2. Access a lesson
3. Participate in quizzes, polls, simulations
4. Check progress tracking

Postconditions:

- Progress is accurately updated

Expected Result: Learners can track their progress through quizzes, polls, simulations, and discussions with accurate update of completion status.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olPf4

Test Case Title: Verify that feedback is provided after each interactive component

Test Case Description: Ensure that feedback is provided after each interactive component (quiz, poll, simulation) to help learners understand their performance.

Test Suite: Interactive Lessons

Test Priority: Medium

Preconditions:

- User is logged in
- Course with interactive lessons is selected

Test Data: Course with interactive lessons

Test Steps:

1. Select a course
2. Access a lesson
3. Participate in quizzes, polls, simulations
4. Observe feedback provided

Postconditions:

- Feedback is presented after participation

Expected Result: Feedback is given after each interactive component to guide learners in their learning journey.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olPf5

Test Case Title: Verify that interactive lessons can be accessed on various devices

Test Case Description: Ensure that learners can access interactive lessons on multiple devices, including desktops, laptops, tablets, and mobile phones, for a consistent learning experience.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- User is logged in
- Course with interactive lessons is selected

Test Data: Course with interactive lessons

Test Steps:

1. Select a course
2. Access a lesson
3. Test accessibility on different devices (desktop, laptop, tablet, mobile phone)

Postconditions:

- Interactive lessons are accessible and functional on various devices

Expected Result: Interactive lessons can be accessed and function correctly on various devices for a consistent learning experience.

Severity: Major

Type of Testing: Compatibility Testing

Test Case Approach: Positive

-----



**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:**

Test Case ID: olp-PT01

Test Case Title: Verify that learners can choose courses based on their interests

Test Case Description: As a learner, I want to select courses that match my interests, ensuring the online learning platform supports personalized learning paths.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in
- The system has a variety of courses available

Test Data: List of courses from various subjects and topics

Test Steps:

1. Log in as a learner
2. Browse the course catalog
3. Select a course based on personal interests
4. Verify the selected course is added to the learner's path

Postconditions:

- The chosen course is added to the learning path

Expected Result: The system enables learners to choose courses based on their interests, providing a customized learning path for each learner.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-PT02

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress

Test Case Description: As a learner, I want my learning path to automatically adapt as I progress through various courses and modules, ensuring a continuously personalized educational experience.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has a learning path with multiple courses and modules

Test Data: Learner's progress through courses and modules

Test Steps:

1. Log in as a learner
2. Navigate to the learning path
3. Complete a module within a course
4. Verify that the learning path is updated and adapts to include new recommendations or adjustments

Postconditions:

- Learning path is updated based on learner progress

Expected Result: The platform dynamically adjusts personalized learning paths, providing learners with course and module options that suit their education goals and progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-PT03

Test Case Title: Verify that learners receive recommendations for new courses and modules

Test Case Description: As a learner, I want to receive personalized course and module recommendations that build upon my existing learning path, enriching my educational experience.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has completed one or more courses or modules

Test Data: List of courses and modules tailored to individual learning preferences

Test Steps:

1. Login as a learner
2. Navigate to the learning path

3. Review the recommended courses and modules section

4. Verify that the recommended content aligns with the learner's interests and previous progress

Postconditions:

- Learner has new course and module recommendations

Expected Result: The platform provides learners with tailored course and module recommendations based on their educational goals, interests, and learning path progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-PT04

Test Case Title: Verify that progress tracking is personalized to each learner's path

Test Case Description: As a learner, I want to track my progress on my personalized learning path, ensuring accurate insight into my educational progress and remaining activities.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has a learning path with multiple courses and modules

Test Data: Learner's progress data

Test Steps:

1. Login as a learner
2. Navigate to the learning path
3. Review the progress tracking information
4. Verify personalized progress tracking

Postconditions:

- Progress tracking information is available and accurate for the learner

Expected Result: The system provides accurate and personalized progress tracking, providing learners with insights into their educational progress on their chosen learning paths.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olp-PT05

Test Case Title: Verify that learners can adjust their learning paths at any time

Test Case Description: As a learner, I want the flexibility to adjust my learning path by adding, removing, or reordering courses and modules, allowing me to tailor my educational experience according to my evolving interests and needs.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has a learning path with multiple courses and modules

Test Data: List of course and module options

Test Steps:

1. Login as a learner
2. Navigate to the learning path
3. Choose to modify the learning path
4. Add, remove, or reorder courses or modules
5. Verify changes are saved and applied to the learning path

Postconditions:

- Learner's learning path is updated based on adjustments

Expected Result: The platform provides learners with the flexibility to modify their learning paths at any time, enabling them to tailor their educational experiences based on their evolving interests and needs.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

## **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:**

## Test Case Creation Template:

Test Case ID: OLP-PT-001

Test Case Title: Verify that progress tracking shows completed activities and grades

Test Case Description: Ensure that progress tracking displays the completed activities and grades for learners.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has enrolled in a course

Test Data: Course with multiple modules and activities

Test Steps:

1. Log in as a learner.
2. Enroll in a course.
3. Complete various activities within the course.
4. View the progress tracking feature.
5. Check the completed activities and associated grades.

Postconditions:

- Completed activities and grades are displayed.

Expected Result: The system displays the completed activities and corresponding grades in the progress tracking feature.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: OLP-PT-002

Test Case Title: Verify that learners can view detailed reports on their progress

Test Case Description: Ensure that the progress tracking feature provides learners with detailed reports on their progress through courses, modules, and tasks.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has enrolled in a course

Test Data: Course with multiple modules and activities

Test Steps:

1. Log in as a learner.
2. Enroll in a course.
3. Complete various activities within the course.
4. Access the detailed progress report.
5. Verify the information presented, including the percentage of completion, grades, time spent, and task status.

Postconditions:

- Detailed reports are accessible and accurate.

Expected Result: The system generates and displays detailed reports on learners' progress, including the percentage of completion, grades, time spent, and task status.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case Creation Template:



Test Case ID: OLP-PT-003

Test Case Title: Verify that achievements are highlighted in the progress tracker

Test Case Description: Ensure that the progress tracking feature highlights achievements earned by the learner throughout the course.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner has enrolled in a course
- Learner has earned achievements

Test Data: Course with multiple modules, activities, and achievements

Test Steps:

1. Log in as a learner.
2. Enroll in a course.
3. Complete activities and earn specified achievements.
4. View the progress tracking feature.
5. Check if the earned achievements are highlighted and properly displayed.

Postconditions:

- Achievements are appropriately highlighted and displayed.

Expected Result: The system highlights and properly displays the achievements earned by the learner in the progress tracking feature.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: OLP-PT-004

Test Case Title: Verify that learners receive notifications about their progress

Test Case Description: Ensure that learners receive notifications about their progress updates and milestones.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner has enrolled in a course

Test Data: Course with multiple modules and activities

Test Steps:

1. Log in as a learner.
2. Enroll in a course.
3. Complete activities and make progress.
4. Verify that notifications are received through the preferred method (email, platform notifications, etc.)

Postconditions:

- Learners receive progress-related notifications.

Expected Result: The system sends notifications regarding learners' progress updates, such as completion of modules or achievement unlocking.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case Creation Template:

Test Case ID: OLP-PT-005

Test Case Title: Verify that progress tracking can be exported as reports

Test Case Description: Ensure that learners have the ability to export their progress tracking data as a report.

Test Suite: Progress Tracking

Test Priority: Low

Preconditions:

- Learner is logged in
- Learner has enrolled in a course
- Learner has made progress in the course

Test Data: Course with multiple modules and activities

Test Steps:

1. Log in as a learner.
2. Enroll in a course.
3. Make progress in the course.
4. Access the export function within the progress tracking feature.
5. Export the progress tracking data as a report (e.g., CSV, PDF, Excel).

Postconditions:

- Progress tracking data is successfully exported.

Expected Result: The system exports the learner's progress tracking data into a report, allowing for offline analysis and storage.

Severity: Major

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:**

Test Case ID: OLP-A01

Test Case Title: Verify that instructors can create quizzes, tests, and assignments

Test Case Description: Ensure that the system allows instructors to create different types of assessments, including quizzes, tests, and assignments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- Instructor is logged in

- Course is created

Test Data: Assessment details (quiz, test, or assignment)

Test Steps:

1. Go to the course page
2. Click on the assessments tab
3. Select "Create new assessment"
4. Choose the assessment type (quiz, test, or assignment)
5. Enter assessment details (name, instructions, etc.)
6. Save the assessment

Postconditions:

- Assessment is created

Expected Result: The system creates a new assessment according to the instructor's selection and saves it for use in the course.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-A02

Test Case Title: Verify that assessments can include various question types

Test Case Description: Ensure that the system allows instructors to include multiple question types when creating assessments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- Instructor is logged in
- Course is created
- Assessment is created

Test Data: Assessment with multiple question types (MCQs, short answers, essays)

Test Steps:

1. Go to the course page
2. Select the assessment
3. Click on "Edit assessment"
4. Add multiple question types (MCQs, short answers, essays)
5. Save the changes

Postconditions:

- Assessment includes various question types

Expected Result: The system saves the assessment with the specified question types, enabling instructors to evaluate learners effectively.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-A03

Test Case Title: Verify automatic grading for objective questions

Test Case Description: Ensure that the system automatically grades objective questions within assessments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- Instructor is logged in
- Course is created
- Assessment with objective questions (MCQs) is created

Test Data: Assessment with MCQs, correct answers provided

Test Steps:

1. Go to the course page

2. Select the assessment
3. Allow learners to complete the assessment
4. Automatically grade the objective questions (MCQs)

Postconditions:

- Objective questions are graded

Expected Result: The system automatically calculates scores for the objective questions and updates the learners' grades.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-A04

Test Case Title: Verify manual grading for subjective questions

Test Case Description: Ensure that the system supports instructors in manually grading subjective questions within assessments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- Instructor is logged in
- Course is created
- Assessment with subjective questions (short answers, essays) is created

Test Data: Assessment with subjective questions, grading criteria provided

Test Steps:

1. Go to the course page
2. Select the assessment
3. Allow learners to complete the assessment
4. Manually grade the subjective questions (short answers, essays) using provided grading criteria

Postconditions:

- Subjective questions are graded

Expected Result: The system allows instructors to manually grade the subjective questions and update the learners' grades.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-A05

Test Case Title: Verify that feedback is provided to learners after grading

Test Case Description: Ensure that the system offers a feedback mechanism for instructors to communicate assessment results to learners.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- Instructor is logged in
- Course is created
- Assessment with subjective and objective questions is created
- Learners have completed the assessment

Test Data: Assessment with both objective and subjective questions, graded results

Test Steps:

1. Go to the course page
2. Select the assessment
3. View the learners' submitted assessments and grades
4. Provide feedback on each question (objective and subjective)
5. Save the feedback

Postconditions:



- Feedback is saved

Expected Result: The system saves the feedback and displays it to the learners.

Severity: Major

Type of

-----

## **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:**

Test Case ID: olPf1a

Test Case Title: Verify that learners can post questions and responses in forums

Test Case Description: As a learner, I should be able to create new threads and reply to existing ones in the discussion forums.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- Learner is logged in
- Course is selected

Test Data: Question or response text

Test Steps:

1. Navigate to the course discussion forum
2. Click on "Start a discussion" or reply to an existing thread
3. Enter the question or response text
4. Submit the post

Postconditions:

- New thread or reply is created

Expected Result: The system allows the learner to post questions or responses in the forums without any errors.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: nbTg2s

Test Case Title: Verify that forums support threaded discussions

Test Case Description: Discussion forums should allow learners and instructors to reply to specific posts, creating a threaded conversation.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- Learner or instructor is logged in
- Course is selected
- A thread exists in the discussion forum

Test Data: Reply text

Test Steps:

1. Navigate to the course discussion forum
2. Select an existing thread
3. Click on "Reply" below a specific post
4. Enter the reply text
5. Submit the post

Postconditions:

- A new reply is added to the selected post

Expected Result: The system enables learners and instructors to create threaded discussions by replying to specific posts.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: kjGh3d

Test Case Title: Verify that instructors can moderate discussions

Test Case Description: Instructors should have the ability to edit, delete, and lock posts within the discussion forums.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Course is selected
- A thread or reply exists in the discussion forum

Test Data: Moderation action (e.g., edit, delete, lock)

Test Steps:

1. Navigate to the course discussion forum
2. Select a thread or reply to moderate
3. Perform the desired moderation action

Postconditions:

- Action is applied to the selected thread or reply

Expected Result: The system allows instructors to moderate discussions by editing, deleting, or locking posts as needed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ioLm4e

Test Case Title: Verify that learners receive notifications for new posts

Test Case Description: Learners should be notified of new posts in threads they have participated in or favorited.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- Learner is logged in
- A thread exists in the discussion forum

Test Data: New post in a watched or favorited thread

Test Steps:

1. Navigate to the course discussion forum
2. Participate in a thread or mark it as a favorite
3. Wait for a new post to be added to the thread

Postconditions:

- Learner receives a notification for the new post

Expected Result: The system notifies learners of new posts in threads they have previously engaged with.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: bvCv5f

Test Case Title: Verify that discussion forums are accessible on all devices

Test Case Description: The discussion forums should function and display correctly on various devices, including desktops, tablets, and smartphones.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- Learner or instructor is logged in
- Course is selected

Test Data: No test data needed

Test Steps:

1. Access the online learning platform on different devices
2. Navigate to the course discussion forum
3. Create a new thread or reply
4. View and interact with existing threads

Postconditions:

- Discussion forums are accessible and functional

Expected Result: The system enables users to access and participate in discussion forums using various devices without issues.

Severity: Major

Type of Testing: Compatibility

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:**

Test Case ID: livCls1

Test Case Title: Verify that learners can attend live classes and webinars

Test Case Description: Ensure that learners can join, view, and interact with instructors during live classes and webinars.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Learner is logged in
- Instructor has started a live class or webinar

Test Data: No test data needed

Test Steps:

1. Access the live class or webinar from the platform's schedule
2. Verify the video conferencing tool launches
3. Check audio and video connections
4. Interact with the instructor using live chat or Q&A features
5. Monitor the live session for a predefined time

Postconditions:

- Learner has successfully attended the live class or webinar

Expected Result: Learner can attend, view, and interact with the instructor during a live class or webinar.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: livCls2

Test Case Title: Verify that learners receive notifications before live class starts

Test Case Description: Ensure that learners are notified of upcoming live classes or webinars via email or platform notifications.

Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

- Learner is registered for a live class or webinar
- Learner is logged in or logged out

Test Data: Notification settings and email address

Test Steps:

1. Verify notification settings for the live class or webinar (in-app, email, or both)
2. Schedule a test live class or webinar
3. If logged out, check for email notifications
4. If logged in, check for in-app notifications

Postconditions:

- Learner receives a notification before the live class or webinar starts

Expected Result: Learner receives a timely notification about the upcoming live class or webinar according to notification settings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: livCls3

Test Case Title: Verify that instructors can share screen and resources during live classes

Test Case Description: Ensure that instructors can share their screen and resources (documents, presentations, etc.) during live classes or webinars.



Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Instructor has started a live class or webinar
- Resources to be shared are prepared and uploaded

Test Data: Prepared resources (documents, presentations, etc.)

Test Steps:

1. Start a live class or webinar
2. Prepare to share a resource
3. Share the resource with attendees
4. Confirm attendees can view and interact with the shared resource

Postconditions:

- Instructor has successfully shared a resource during a live class or webinar

Expected Result: Instructors can share resources and attendees can view and interact with them during live classes or webinars.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: livCls4

Test Case Title: Verify that live classes and webinars are recorded and available for later viewing

Test Case Description: Ensure that recorded live classes or webinars are accessible for viewing by registered learners.

Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

- Instructor has started a live class or webinar

- Live class or webinar has been recorded

Test Data: No test data needed

Test Steps:

1. Access the recording from the platform's archive
2. Verify the recording plays without issues
3. Monitor the entire recording for quality

Postconditions:

- Recording can be accessed and viewed by registered learners

Expected Result: Recorded live classes or webinars are available for later viewing in the platform's archive.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: livCls5

Test Case Title: Verify that attending live classes does not allow learners to skip prerequisites

Test Case Description: Ensure that learners must complete prerequisites before accessing live classes or webinars related to those prerequisites.

Test Suite: Live Classes and Webinars

Test Priority:LOW

Preconditions:

- Learner is registered for a live class or webinar with prerequisites
- Some prerequisites have not been completed

Test Data: Incomplete prerequisites

Test Steps:

1. Attempt to access the live class or webinar
2. Check for restriction messages or warnings about uncompleted prerequisites

3. If allowed to enter, check if functionality is disabled or limited for uncompleted prerequisites

Postconditions:

- Learner cannot access the live class or webinar without completing prerequisites

Expected Result: Learners cannot attend live classes or webinars related to incomplete prerequisites.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Negative

-----

**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:**

Test Case ID: olPla1

Test Case Title: Verify that the online learning platform is accessible on smartphones

Test Case Description: Ensure that the online learning platform can be accessed and loaded on various smartphone devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- A stable internet connection is available on the smartphone.

Test Data: No test data needed

Test Steps:

1. Open a web browser on the smartphone.
2. Enter the URL of the online learning platform.
3. Attempt to log in or sign up for a new account.
4. Navigate through the different sections of the platform.

Postconditions:

- The platform is accessible and loads correctly on the smartphone.

Expected Result: The online learning platform can be accessed, logged in, and navigated on the smartphone without issues.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olPla2

Test Case Title: Verify that the mobile UI is optimized for touch navigation

Test Case Description: Validate that the user interface and elements on the online learning platform are responsive and suitable for touch navigation on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- A stable internet connection is available on the smartphone.
- The online learning platform is accessible on the smartphone.

Test Data: No test data needed

Test Steps:

1. Use touch gestures to scroll, swipe, and zoom on the platform.
2. Tap buttons and input fields to ensure they are responsive.
3. Check the readability of text elements at various zoom levels.
4. Verify that menus and dropdowns can be interacted with using touch.

Postconditions:

- The UI elements are responsive and suitable for touch navigation on mobile devices.

Expected Result: The online learning platform's UI elements are optimized for touch navigation and can be easily interacted with on a smartphone.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: olPla3

Test Case Title: Verify that course materials are accessible and readable on mobile devices

Test Case Description: Confirm that course materials, such as videos and documents, can be accessed and viewed on mobile devices without issues.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- A stable internet connection is available on the smartphone.
- The online learning platform is accessible on the smartphone.
- A course with materials is available on the platform.

Test Data: A course with various types of materials (video lectures, downloadable resources, etc.)

Test Steps:

1. Access a course on the online learning platform using a smartphone.
2. Attempt to view and interact with all types of course materials.
3. Check the readability of text materials on the smartphone.
4. Validate the playback functionality of video lectures on the smartphone.

Postconditions:

- Course materials are accessible, readable, and functional on smartphones.

Expected Result: Course materials can be accessed, viewed, and interacted with on a smartphone without issues.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olPla4

Test Case Title: Verify that notifications are sent to mobile devices

Test Case Description: Ensure that the online learning platform can send notifications, such as course updates or messages, to mobile devices.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- A stable internet connection is available on the smartphone.
- The online learning platform is accessible on the smartphone.
- The user is logged in and has allowed notifications.

Test Data: No test data needed

Test Steps:

1. Log in to the online learning platform on the smartphone.
2. Enable notifications for the platform.
3. Trigger a notification, such as a new course update or message.
4. Check for the notification on the smartphone.

Postconditions:

- Notifications are successfully sent to the smartphone.

Expected Result: The online learning platform sends notifications to the smartphone as expected.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olPla5

Test Case Title: Verify that the platform supports offline access to downloaded materials

Test Case Description: Confirm that learners can download course materials for offline access and view them on their mobile devices.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- A stable internet connection is available on the smartphone.
- The online learning platform is accessible on the smartphone.
- A course with downloadable materials is available on the platform.

Test Data: A course with downloadable resources (e.g., PDFs, audio files)

#### Test Steps:

1. Access a course on the online learning platform using a smartphone.
2. Download a resource for offline access.
3. Disconnect from the internet.
4. Open the downloaded resource on the smartphone.

#### Postconditions:

- Downloaded resources can be accessed offline on the smartphone.

Expected Result: The online learning platform allows learners to download course materials for offline access and view them on their mobile devices.

Severity: Major

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:**

Test Case ID: RCe6T7

Test Case Title: Verify that the resource library includes e-books, articles, and reference guides

Test Case Description: Ensure that the online learning platform's resource library contains e-books, articles, and reference guides to support learners' studies.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- Learner is logged in
- The learner is enrolled in at least one course

Test Data: N/A

Test Steps:

1. Log in as a learner
2. Access the resource library
3. Verify that e-books, articles, and reference guides are available

Postconditions:

- Resource library includes e-books, articles, and reference guides

Expected Result: The resource library contains various types of resources to support learners' studies.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: wW3Xm5

Test Case Title: Verify that learners can search for specific materials in the resource library

Test Case Description: Confirm that learners can search for specific resources within the resource library using keywords or filters.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- Learner is logged in
- The learner is enrolled in at least one course

Test Data: Keyword or filter criteria (e.g., "Python programming," "data structures," "intermediate level")

Test Steps:

1. Log in as a learner
2. Access the resource library
3. Enter a keyword or select filters to search for specific resources
4. Review the search results

Postconditions:

- The search results match the entered keywords or filters

Expected Result: Learners can find the desired resources using the search function.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: Yf2Eg1

Test Case Title: Verify that resources are categorized and tagged for easy navigation in the resource library

Test Case Description: Ensure that resources in the resource library are categorized by subject and tagged with relevant keywords for easy navigation and discovery.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- Learner is logged in
- The learner is enrolled in at least one course

Test Data: N/A

Test Steps:

1. Log in as a learner
2. Access the resource library
3. Browse the resources by subject or tags
4. Verify that resources are categorized and tagged appropriately

Postconditions:

- The resource library is organized using subjects and tags

Expected Result: Learners can easily navigate and find resources through categorization and tagging.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: dSsQp6

Test Case Title: Verify that learners can download resources for offline access

Test Case Description: Confirm that learners can download resources from the resource library for offline access.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- Learner is logged in
- The learner is enrolled in at least one course

Test Data: One or more resources from the resource library

Test Steps:

1. Log in as a learner
2. Access the resource library
3. Choose a resource to download
4. Initiate the download process
5. Verify that the downloaded file is accessible offline

Postconditions:

- Downloaded resources are accessible offline

Expected Result: Learners can save resources for offline usage.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ghR9j3

Test Case Title: Verify that instructors can recommend resources to learners in the resource library

Test Case Description: Ensure that instructors can recommend specific resources within the resource library to learners.

Test Suite: Resource Library

Test Priority: Low

Preconditions:

- Instructor is logged in
- The instructor is associated with at least one course

- Learner is logged in and enrolled in the instructor's course

Test Data: A resource from the instructor's course

Test Steps:

1. Log in as an instructor
2. Access the course associated with the learner
3. Select a resource related to the course
4. Choose the option to recommend the resource
5. Log in as the learner
6. Verify that the recommended resource is displayed in the learner's resource library

Postconditions:

- Recommended resource is displayed in the learner's resource library

Expected Result: Instructors can share additional resources to support learners' studies.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template for Peer Review Feature:

Test Case ID: PR-001

Test Case Title: Verify that learners can submit assignments for peer review

Test Case Description: Ensure that learners can submit their coursework for peer review within the online learning platform.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in
- Course is active
- Assignment is available

Test Data: Completed assignment

Test Steps:

1. Login as a learner

2. Access the active course
3. Navigate to the available assignment
4. Upload the completed assignment
5. Submit the assignment for peer review

Postconditions:

- Assignment is submitted for peer review
- Learner can view submission status

Expected Result: The system accepts the assignment, making it available for peer review and displaying the correct submission status.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PR-002

Test Case Title: Verify that peers can provide feedback on submitted assignments

Test Case Description: Ensure that learners can access and provide constructive feedback on their peers' submitted assignments.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in
- Assignment is submitted for peer review

Test Data: Feedback comments and rating

Test Steps:

1. Login as a learner
2. Access the submitted assignment for peer review
3. Read the assignment requirements and content

4. Provide feedback in the form of comments and ratings

5. Submit the feedback

Postconditions:

- Feedback is saved and linked to the assignment
- Learner can view the feedback

Expected Result: The system records and displays peer feedback, connecting it to the correct assignment.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PR-003

Test Case Title: Verify that a structured review process guides the feedback

Test Case Description: Ensure that the peer review process includes a structured format for providing feedback, including comments and ratings.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in
- Assignment is submitted for peer review

Test Data: Feedback comments and rating

Test Steps:

1. Login as a learner
2. Access the submitted assignment for peer review
3. Provide feedback based on the structured format (e.g., required sections, overall rating)
4. Submit the feedback

Postconditions:

- Feedback is saved and linked to the assignment



- Learner can view the feedback in the structured format

Expected Result: The system enforces the structured review process and displays feedback accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PR-004

Test Case Title: Verify that learners can view feedback on their work

Test Case Description: Ensure that learners can view the feedback provided by their peers on their submitted assignments.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in
- Peer feedback is available on the submitted assignment

Test Data: No test data needed

Test Steps:

1. Login as a learner
2. Access the submitted assignment with available peer feedback
3. Review the provided feedback

Postconditions:

- Feedback is displayed for the learner

Expected Result: The system shows the correct feedback for the learner to view.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: PR-005

Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback

Test Case Description: Ensure that learners cannot identify their peer reviewers during the review process.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in
- Assignment is submitted for peer review

Test Data: No test data needed

Test Steps:

1. Login as a learner
2. Access the submitted assignment with available peer feedback
3. Attempt to identify the peer reviewers

Postconditions:

- Peer reviewer identities are not displayed or revealed

Expected Result: The system maintains reviewer anonymity throughout the peer review process.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

-----

**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:**

Test Case ID: OLP1

Test Case Title: Verify that learners earn certificates upon course completion

Test Case Description: This test case ensures that learners receive a certificate upon completing a course on the online learning platform.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has enrolled in a course
- Learner has accessed and completed all the content within the course

Test Data: Course completion data

Test Steps:

1. Log in as a learner
2. Enroll in a course
3. Complete all course modules and lectures
4. Navigate to the learner profile
5. Access the course completion section

Test Postconditions:

- Certificate is visible and accessible in the learner profile

Expected Result: Upon completing a course, the system generates a certificate and displays it in the learner's profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP2

Test Case Title: Verify that badges are awarded for specific achievements and milestones

Test Case Description: This test case guarantees that learners receive badges for specific accomplishments within a course.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has enrolled in a course

Test Data: Milestone completion data

Test Steps:

1. Log in as a learner

2. Enroll in a course that includes milestones
3. Complete a specific milestone (e.g., quiz, project)
4. Navigate to the learner profile
5. Access the badges section

Test Postconditions:

- Badge is visible and accessible in the learner profile

Expected Result: Upon reaching a milestone, the system awards the corresponding badge and displays it in the learner's profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP3

Test Case Title: Verify that learners can access certificates and badges through their profiles

Test Case Description: This test case confirms that learners can access their certificates and badges through their profiles on the learning platform.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

- Learner is logged in

Test Data: No test data needed

Test Steps:

1. Log in as a learner
2. Navigate to the learner profile
3. Access the certificates and badges sections

Test Postconditions:

- Certificates and badges are accessible

Expected Result: The system enables learners to view their certificates and badges from their profiles.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP4

Test Case Title: Verify that learners can share their certificates and badges on social media

Test Case Description: This test case validates that learners have the ability to share their certificates and badges directly on social media platforms.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner has earned at least one certificate or badge

Test Data: Certificates or badges to be shared

Test Steps:

1. Log in as a learner
2. Navigate to the learner profile
3. Access the sharing options for a certificate or badge
4. Choose a social media platform to share
5. Confirm that the certificate or badge is successfully shared

Test Postconditions:

- Certificate or badge is shared on selected social media platform

Expected Result: The system allows learners to share their certificates and badges with their connections on various social media platforms.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP5

Test Case Title: Verify that instructors can issue custom badges for special achievements

Test Case Description: This test case checks that instructors can create custom badges for learners to recognize unique accomplishments within a course.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Instructor has access to course management tools

Test Data: Special achievement and custom badge image

Test Steps:

1. Log in as an instructor
2. Create or access a course
3. Set up a special achievement milestone within the course
4. Design and upload a custom badge for the special achievement
5. Navigate to the course's milestone section

Test Postconditions:

- Custom badge is assigned to learners who complete the special achievement milestone

Expected Result: Upon completing the special achievement milestone, the system awards the custom badge accordingly.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: olePf1

Test Case Title: Verify that learners earn points for completing activities

Test Case Description: Ensure that the system correctly awards points to learners when they complete various activities within the online learning platform.



Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is logged in
- Learner is enrolled in a course

Test Data: Course activity details

Test Steps:

1. Log in as a learner
2. Access an enrolled course
3. Complete an activity (e.g., watch a video, pass a quiz)
4. Check the learner's point balance

Postconditions:

- Points have been added to the learner's balance

Expected Result: The system awards points for completed activities and updates the learner's point balance accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olePf2

Test Case Title: Verify that levels are achieved based on accumulated points

Test Case Description: Ensure that the system accurately assigns levels to learners based on the number of points they have earned.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is logged in

- Learner has earned points

Test Data: Learner point balance

Test Steps:

1. Log in as a learner
2. Access the gamification dashboard
3. Verify the current level based on the point balance

Postconditions:

- Level is updated based on point balance

Expected Result: The system correctly assigns a level to the learner based on their earned points.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olePf3

Test Case Title: Verify that leaderboards display top-performing learners

Test Case Description: Ensure that the system generates accurate and up-to-date leaderboards that display the top learners based on their earned points.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Multiple learners are logged in
- Learners have earned points

Test Data: Learner point balances

Test Steps:

1. Access the leaderboard feature
2. Validate the positions of each learner based on their point balances

Postconditions:

- Leaderboard is ranked according to learner point balances

Expected Result: The system generates an accurate leaderboard based on earned points.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olePf4

Test Case Title: Verify that badges are awarded for reaching milestones

Test Case Description: Ensure that the system awards badges to learners when they reach specific milestones within courses or across the platform.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner is enrolled in a course

Test Data: Course milestones and badges

Test Steps:

1. Log in as a learner
2. Access a course containing milestones
3. Complete milestone activities
4. Check the learner's badge collection

Postconditions:

- Badge is added to the learner's collection

Expected Result: The system awards badges to learners when they reach specific milestones as defined in the course or platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olePf5

Test Case Title: Verify that learners can view their progress in the gamification dashboard

Test Case Description: Ensure that learners can access and view their progress in the gamification dashboard, including earned points, levels, badges, and leaderboard rankings.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Learner is logged in

Test Data: No test data needed

Test Steps:

1. Log in as a learner
2. Access the gamification dashboard
3. Verify information displayed in the dashboard

Postconditions:

- Learner can access and see their progress information

Expected Result: The system allows learners to access, view, and interact with the gamification dashboard, showing their earned points, levels, badges, and leaderboard rankings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: olPla1

Test Case Title: Verify that the platform interface is available in multiple languages

Test Case Description: Ensure that the online learning platform offers an interface in various languages, allowing learners to use the platform comfortably in their preferred language.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- User is logged in
- The learner's preferred language is set in their profile

Test Data: Preferred language settings

Test Steps:

1. Log in to the platform
2. Access the account settings
3. Set the preferred language
4. Navigate through various sections
5. Check for consistent language in menus, labels, prompts, and messages

Postconditions:

- The platform interface is displayed in the preferred language

Expected Result: The platform interface is available in the learner's preferred language as set in their profile, including menus, labels, prompts, and messages.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPla2

Test Case Title: Verify that course materials are available in different languages

Test Case Description: Confirm that the learning platform provides course materials in multiple languages, allowing learners to choose their preferred language for studying content.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- User is logged in

- The learner's preferred language is set in their profile

Test Data: Preferred language settings, courses with multilingual materials

Test Steps:

1. Log in to the platform
2. Browse the available courses
3. Select a course offering multilingual materials
4. Access the course materials (e.g., video lectures, text, or downloadable resources)
5. Check for consistency in the language throughout the materials

Postconditions:

- Course materials are displayed in the preferred language

Expected Result: Course materials are provided in the learner's preferred language as set in their profile, including text, audio, and downloadable resources.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPla3

Test Case Title: Verify that learners can select their preferred language at any time

Test Case Description: Ensure that learners can change their preferred language easily and access the platform in their preferred language without any interruptions.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Preferred language settings

Test Steps:

1. Log in to the platform
2. Access the account settings
3. Switch the preferred language to a different one
4. Verify that the interface is updated in the new preferred language
5. Confirm that no errors occur during the language switch

Postconditions:

- The platform interface is displayed in the learner's newly selected preferred language

Expected Result: Learners can change their preferred language at any time, and the platform interface and course materials are updated accordingly without issuing any errors.

Severity: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPla4

Test Case Title: Verify that multilingual support includes translated help and support resources

Test Case Description: Confirm that learners can access help and support materials (e.g., documentation, FAQs, contact forms) in their preferred language.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- User is logged in
- The learner's preferred language is set in their profile

Test Data: Preferred language settings, help and support resources

Test Steps:



1. Log in to the platform
2. Access the help or support section
3. Verify that materials are available in the preferred language
4. Check for consistency in the language throughout the materials

Postconditions:

- Help and support resources are displayed in the preferred language

Expected Result: Learners have access to help and support resources in their preferred language, allowing them to resolve issues efficiently.

Severity: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPla5

Test Case Title: Verify that instructors can provide content in multiple languages

Test Case Description: Ensure that content creators (instructors) can upload and manage course materials in more than one language.

Test Suite: Multilingual Support

Test Priority: Low

Preconditions:

- Instructor is logged in
- Instructor has course editing permissions
- The content to be translated exists (e.g., video lectures, text, or downloadable resources)

Test Data: Course materials, preferred language settings

Test Steps:

1. Log in as an instructor

2. Access the course editing interface
3. Upload course materials in multiple languages
4. Save and publish the course
5. Verify that materials are available in different languages

Postconditions:

- Course materials are displayed in multiple languages

Expected Result: Instructors can provide course materials in multiple languages, enabling learners with diverse linguistic backgrounds to access content comfortably.

Severity: Low

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:**

Test Case Creation Template for Online Learning Platform Accessibility Features:

Test Case ID: olp-001

Test Case Title: Verify that the platform is compatible with screen readers

Test Case Description: Ensure that popular screen readers can navigate through the platform's pages, courses, and other elements.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- A supported screen reader is installed and configured
- User is logged in

Test Data: Navigation through different pages and courses

Test Steps:

1. Log in to the platform
2. Enable the screen reader
3. Navigate through the homepage, courses list, and individual course pages
4. Use the screen reader to interact with buttons, menus, and form elements

Postconditions:

- Screen reader provides accurate and complete information
- User can navigate and interact with the platform effectively

Expected Result: The screen reader is able to navigate, read, and interact with the platform comprehensively, without any major issues.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olp-002

Test Case Title: Verify that learners can adjust font sizes for better readability

Test Case Description: Ensure that learners can adjust the font size of the text displayed on the platform for improved readability.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in

Test Data: Default font size and adjusted font sizes

Test Steps:

1. Log in to the platform
2. Access any text-heavy page, such as a lecture or a quiz
3. Adjust the font size using platform settings or browser zoom options
4. Observe the readability of the text at various font sizes

Postconditions:

- Text remains readable and visually appealing

Expected Result: The platform allows users to adjust font sizes easily, maintaining readability and visual appeal regardless of the chosen font size.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: olp-003

Test Case Title: Verify that color contrast options are available for improved visibility

Test Case Description: Ensure that learners can customize color contrast settings for better visibility of text and other elements on the platform.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in

Test Data: Default color contrast and adjusted color contrast settings

Test Steps:

1. Log in to the platform
2. Access any text-heavy page, such as a lecture or a quiz
3. Adjust the color contrast settings using platform options
4. Observe the visibility of the text and other elements at various contrast levels

Postconditions:

- Text and other elements remain visible and distinguishable

Expected Result: The platform allows users to customize color contrast settings effectively, ensuring that text and other elements remain visible and distinguishable regardless of the chosen contrast level.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: olp-004

Test Case Title: Verify that accessibility features are easy to activate and customize

Test Case Description: Ensure that learners with disabilities can easily activate and customize accessibility features on the platform.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in

Test Data: Navigation through accessibility settings

Test Steps:

1. Log in to the platform
2. Access the accessibility settings or options
3. Activate or customize various accessibility features, such as screen reader compatibility, adjustable font sizes, and color contrast
4. Observe the ease and intuitiveness of activating and customizing the features

Postconditions:

- Accessibility features can be easily activated and customized

Expected Result: The platform provides a user-friendly interface for activating and configuring accessibility options, allowing learners with disabilities to utilize these features effectively.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case ID: olp-005

Test Case Title: Verify that support resources are available to assist learners with disabilities

Test Case Description: Ensure that the platform provides support resources that cater to the needs of learners with disabilities.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Navigation through support resources

Test Steps:

1. Log in to the platform
2. Access support resources or documentation
3. Verify the availability of resources and guidance related to accessibility features

Postconditions:

- Relevant support resources are provided

Expected Result: The platform offers support resources tailored to learners with disabilities, covering the use and customization of accessibility features.

Severity: Major

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:**

Test Case Creation Template for Discussion Moderation Feature:

Test Case ID: DM-001

Test Case Title: Verify that instructors can edit posts in discussions

Test Case Description: Ensure that instructors can edit posts within discussion forums to maintain a positive and respectful learning environment.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in
- A student has posted a message in a discussion forum

Test Data: Message post in a discussion forum

Test Steps:

1. Access the course as an instructor



2. Go to the discussion forum with the student's post
3. Click on the "Edit" button next to the student's post
4. Modify the post content
5. Save the changes

Postconditions:

- Student's post is updated

Expected Result: The instructor can edit the student's post successfully, and the changes are saved.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: DM-002

Test Case Title: Verify that instructors can delete posts in discussions

Test Case Description: Ensure that instructors can delete inappropriate posts within discussion forums to maintain a positive and respectful learning environment.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in
- A student has posted a message in a discussion forum

Test Data: Message post in a discussion forum

Test Steps:

1. Access the course as an instructor
2. Go to the discussion forum with the student's post
3. Click on the "Delete" button next to the student's post
4. Confirm the deletion

Postconditions:

- Student's post is removed

Expected Result: The instructor can delete the student's post successfully, and the post is no longer visible in the discussion forum.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: DM-003

Test Case Title: Verify that instructors can highlight helpful posts in discussions

Test Case Description: Ensure that instructors can highlight helpful posts within discussion forums to promote positive and constructive interactions.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- Instructor is logged in
- A student has posted a message in a discussion forum

Test Data: Message post in a discussion forum

Test Steps:

1. Access the course as an instructor
2. Go to the discussion forum with the student's post
3. Click on the "Highlight" button next to the student's post

Postconditions:

- Student's post is highlighted

Expected Result: The instructor can highlight the student's post successfully, and it is visually distinguishable from other posts in the discussion forum.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: DM-004

Test Case Title: Verify that instructors can mute or ban disruptive users

Test Case Description: Ensure that instructors can mute or ban disruptive users from discussion forums to maintain a positive and respectful learning environment.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in
- Disruptive user has posted a message in a discussion forum

Test Data: Message post in a discussion forum from the disruptive user

Test Steps:

1. Access the course as an instructor
2. Go to the discussion forum with the disruptive user's post
3. Click on the disruptive user's profile picture
4. Select "Mute" or "Ban" option
5. Confirm the mute or ban action

Postconditions:

- Disruptive user's posts are no longer visible
- Disruptive user is muted or banned from the discussion forum

Expected Result: The disruptive user's posts are not visible anymore, and they are muted or banned from participating in the discussion forum.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: DM-005

Test Case Title: Verify that moderation actions are reported and logged

Test Case Description: Ensure that moderation actions (e.g., post editing, deletion, highlighting) are reported and logged for accountability and auditing.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in

Test Data: No test data needed

Test Steps:

1. Access the course as an instructor
2. Perform moderation actions (e.g., editing or deleting a post)
3. Verify that the action is recorded in the activity log

Postconditions:

- Moderation actions are recorded and logged

Expected Result: Moderation actions (e.g., post editing, deletion, highlighting) are logged and reported.

Severity: Major

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:**

Test Case ID: LAnalytics1

Test Case Title: Verify that learning analytics dashboards display engagement metrics

Test Case Description: As an instructor, ensure that the learning analytics dashboard displays accurate learner engagement metrics.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Course with learners is created

Test Data: No test data needed

Test Steps:

1. Log in as an instructor
2. Navigate to the learning analytics dashboard
3. Verify if engagement metrics like average time spent, last activity, and number of visits per learner are visible

Postconditions:

- Engagement metrics are displayed

Expected Result: The learning analytics dashboard accurately displays engagement metrics for instructors.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: LAnalytics2

Test Case Title: Verify that performance trends are visualized over time

Test Case Description: As an instructor, ensure that performance trends of learners are visualized on the learning analytics dashboard over a specified period.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Course with learners is created

Test Data: No test data needed

Test Steps:

1. Log in as an instructor

2. Navigate to the learning analytics dashboard
3. Set time range for visualizing performance trends
4. Verify if performance trends chart is displayed correctly

Postconditions:

- Performance trends chart is visible

Expected Result: The learning analytics dashboard displays accurate performance trends for instructors over a specified period.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: LAnalytics3

Test Case Title: Verify that instructors can access detailed reports on learner progress

Test Case Description: As an instructor, ensure that detailed reports on learner progress are accessible from the learning analytics dashboard.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Course with learners is created

Test Data: No test data needed

Test Steps:

1. Log in as an instructor
2. Navigate to the learning analytics dashboard
3. Click on the 'Learner Progress' tab

4. Verify if detailed learner progress reports are visible

Postconditions:

- Detailed learner progress reports are accessible

Expected Result: Instructors can access comprehensive reports on learner progress from the learning analytics dashboard.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: LAnalytics4

Test Case Title: Verify that analytics data is updated in real-time

Test Case Description: As an instructor, ensure that data on the learning analytics dashboard is updated in real-time.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Course with learners is created

Test Data: No test data needed

Test Steps:

1. Log in as an instructor
2. Navigate to the learning analytics dashboard
3. Perform an action that should update analytics data (e.g. learners completing activities)
4. Verify if real-time data updates are displayed without refreshing the dashboard

Postconditions:



- Real-time data updates are visible

Expected Result: The learning analytics dashboard updates data in real-time as learners interact with the course.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: LAnalytics5

Test Case Title: Verify that instructors can export analytics reports for further analysis

Test Case Description: As an instructor, ensure that analytics reports can be exported from the learning analytics dashboard for further analysis.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Course with learners is created
- Detailed reports on learner progress are available

Test Data: No test data needed

Test Steps:

1. Log in as an instructor
2. Navigate to the learning analytics dashboard
3. Click on the 'Learner Progress' tab
4. Click on the 'Export' button
5. Verify if the report is successfully exported in the desired format (e.g. CSV or XLS)

Postconditions:

- Analytics report is exported

Expected Result: The learning analytics dashboard allows instructors to export analytics reports in various formats for further analysis.

Severity: Major

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:**

Test Case ID: OLP-SLF-TC1

Test Case Title: Verify that learners can create and join groups for projects

Test Case Description: This test case ensures that learners can successfully create new groups for collaborative projects and join existing groups on the online learning platform.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in
- Learner is enrolled in a course

Test Data: Course information, group name

Test Steps:

1. Log in as a learner
2. Navigate to the course page
3. Click on the "Groups" tab
4. Select "Create Group"
5. Enter group name and description
6. Click on "Create Group" button
7. Alternatively, select an existing group and click "Join Group"

Postconditions:

- Group is created or learner is a member of the group

Expected Result: Learner can create a new group or successfully join an existing group with the correct course association.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-SLF-TC2

Test Case Title: Verify that collaboration tools are available for group communication

Test Case Description: This test case verifies that learners have access to collaboration tools for group communication within social learning features on the online learning platform.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner is a member of a group

Test Data: Group message, group file

Test Steps:

1. Log in as a learner
2. Navigate to the group page
3. Use collaboration tools, such as chat, messaging, or discussion forums
4. Send a group message or upload a group file

Test Steps:

1. Verify that the message or file is visible to other group members

Postconditions:

- Group communication is possible

Expected Result: Learners can communicate effectively using collaboration tools within their groups.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-SLF-TC3

Test Case Title: Verify that learners can share resources and feedback within groups

Test Case Description: This test case validates that learners can share resources and provide feedback within groups as part of the social learning features on the online learning platform.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner is a member of a group

Test Data: Learning resource, feedback

Test Steps:

1. Log in as a learner
2. Navigate to the group page
3. Upload or share a learning resource, such as a document, link, or video
4. Provide feedback or a comment on another learner's resource

Test Steps:

1. Verify that the resource or feedback is visible to other group members

Postconditions:

- Resource sharing and feedback are possible within groups

Expected Result: Learners can share resources and provide feedback, fostering collaboration and interactive learning experiences within groups.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-SLF-TC4

Test Case Title: Verify that instructors can assign group projects and monitor progress

Test Case Description: This test case confirms that instructors can create and assign group projects, as well as monitor learners' progress within social learning features on the online learning platform.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor is associated with a course

Test Data: Course, group, project

Test Steps:

1. Log in as an instructor
2. Navigate to the course page
3. Create a new group project or select an existing group
4. Assign the project to the group
5. Monitor learners' progress and participation within the project

Postconditions:

- Group projects are assigned and progress is tracked

Expected Result: Instructors can effectively manage group projects and monitor learners' progress within their assigned courses.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: OLP-SLF-TC5

Test Case Title: Verify that social sharing features allow learners to share achievements

Test Case Description: This test case checks that learners can share their achievements, such as course completion or quiz scores, through social sharing features on the online learning platform.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner has completed a course or quiz

Test Data: Course completion certificate, quiz score

Test Steps:

1. Log in as a learner
2. Navigate to the course or quiz completion page
3. Use social sharing features to share achievements

Test Steps:

1. Verify that the achievements are shared correctly on external platforms

Postconditions:

- Learners can share achievements through social sharing featuresa

Expected Result: Learners can promote their achievements on external platforms using social sharing features, enhancing the online learning platform's visibility and reputation.

Severity: Major

Type of Testing: Functional

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:**

Test Case Creation Template:

Test Case ID: OLPC1

Test Case Title: Verify that instructors can create multimedia presentations

Test Case Description: Ensure that instructors are able to create multimedia presentations using the content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor is logged in



- Instructor is on the course creation page

Test Data: Course details, multimedia elements (e.g., images, audio, text)

Test Steps:

1. Go to course creation page
2. Choose to add multimedia presentation
3. Upload required multimedia elements
4. Arrange components and customize presentation
5. Save and preview the multimedia presentation

Postconditions:

- Multimedia presentation is created

Expected Result: The system allows instructors to create multimedia presentations with customized layouts, including uploaded multimedia elements, and presents them accurately during the preview.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case ID: OLPC2

Test Case Title: Verify that interactive modules can be developed and integrated

Test Case Description: Ensure that instructors can create and integrate interactive modules into their course content using the content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor is on the course creation page

Test Data: Course content, interactive module details (e.g., quiz, poll, discussion)

Test Steps:

1. Go to course creation page
2. Choose to add interactive module
3. Select the type of interactive module
4. Configure the interactive module settings
5. Save and preview the interactive module

Postconditions:

- Interactive module is created

Expected Result: The system allows instructors to create and integrate interactive modules (e.g., quizzes, polls, discussions) with configured settings into their course content, and presents them accurately during the preview.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case ID: OLPC3

Test Case Title: Verify that rich media resources are supported

Test Case Description: Ensure that rich media resources, such as videos and animations, can be uploaded, processed, and displayed via the content authoring tools.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor is on the course creation page

Test Data: Rich media resources (e.g., videos, animations, GIFs)

Test Steps:

1. Go to course creation page
2. Choose to add rich media resource

3. Upload the rich media resource

4. Save and preview the course content

Postconditions:

- Rich media resource is rendered appropriately

Expected Result: The system supports rich media resources and displays them correctly without loss of quality or functionality during the preview.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case ID: OLPC4

Test Case Title: Verify that content authoring tools are user-friendly and intuitive

Test Case Description: Evaluate the usability and intuitiveness of the content authoring tools, including available options, navigation, and overall user experience.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

- Instructor is logged in

Test Data: No test data needed

Test Steps:

1. Access the content authoring tools during course creation

2. Explore available options for course content creation (e.g., multimedia presentations, interactive modules)

3. Perform basic customization and functional settings

4. Try undoing and redoing actions

5. Navigate through various pages and submenus

Postconditions:

- Instructor can effectively create and edit course content using the content authoring tools

Expected Result: The content authoring tools are user-friendly and intuitive, allowing instructors to create and edit course content quickly and easily without requiring extensive guidance or a steep learning curve.

Severity: Major

Type of Testing: Usability Testing

Test Case Behaviour: Positive

Test Case ID: OLPC5

Test Case Title: Verify that instructors can preview and edit content before publishing

Test Case Description: Ensure that instructors can preview and make necessary modifications to their course content before publishing it for students.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor is logged in
- Instructor has created course content (e.g., multimedia presentations, interactive modules)

Test Data: Course content

Test Steps:

1. Create a draft of course content
2. Choose the preview option for the draft content
3. Identify areas requiring modifications
4. Edit the draft content accordingly
5. Save the updated draft and preview again

Postconditions:

- Draft content reflects modified changes

Expected Result: The content authoring tools allow instructors to preview and edit their draft course content, ensuring the published materials are accurate and meet their expectations, including the display of

modifications during the preview.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: 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:**

Test Case ID: olpL01

Test Case Title: Verify that learning paths include sequential courses and modules

Test Case Description: Ensure that the learning path is designed with a logical sequence of interconnected courses and modules, enabling learners to progress systematically.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- A learning path is created
- Courses and modules are added to the learning path

Test Data: Learning path with courses and modules

Test Steps:

1. Access the learning path
2. Review the sequence of courses and modules
3. Verify the prerequisites and dependencies between courses and modules
4. Check for consistency in the learning objectives
5. Monitor progress tracking features for each course and module

Postconditions:

- Learning path is structured logically

Expected Result: The system demonstrates a sequential arrangement of courses and modules, ensuring learners can follow a systematic pathway to meet their learning objectives.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olpL02

Test Case Title: Verify that curricula are aligned with specific learning objectives

Test Case Description: Confirm that the learning path's curriculum is specifically designed to cater to the defined learning objectives, providing learners with a clear and purposeful educational journey.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Learning objectives have been defined
- Courses, modules, and resources have been added to the learning path

Test Data: Learning path with defined learning objectives

Test Steps:

1. Identify learning objectives
2. Review the courses, modules, and resources
3. Verify that all content contributes to the learning objectives
4. Ensure that all resources are relevant and contextually appropriate
5. Monitor the progression towards achieving the learning objectives

Postconditions:

- Learning path curriculum supports meeting learning objectives

Expected Result: The curriculum delivers the learning objectives effectively, providing learners with a valuable and coherent educational experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olpL03

Test Case Title: Verify that learners can track their progress within a learning path

Test Case Description: Confirm that learners can monitor and evaluate their progress efficiently, encouraging

them to remain motivated and focused throughout the learning path.

Test Suite: Learning Paths and Curriculum

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learning path is created with courses and modules

Test Data: Learner and learning path

Test Steps:

1. Access the learning path as a learner
2. Begin a course or module
3. Complete specified tasks or quizzes
4. Track the completed courses and modules
5. Assess progress percentage and remaining activities

Postconditions:

- Learner can view and understand their progress within the learning path

Expected Result: Learners can track their progress effectively, ensuring they are aware of their achievements and the remaining content required to complete the learning path.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olpL04

Test Case Title: Verify that instructors can customize learning paths for different needs

Test Case Description: Ensure that instructors can modify the learning path according to diverse learning requirements, allowing them to deliver tailored educational experiences.

Test Suite: Learning Paths and Curriculum

Test Priority: Medium



Preconditions:

- Instructor is logged in
- Learning path is created

Test Data: Instructor, learning path, and custom content

Test Steps:

1. Access the learning path as an instructor
2. Edit the learning path
3. Add or remove courses and modules
4. Adjust the course prerequisites and dependencies
5. Assign custom resources to individual modules

Postconditions:

- Instructor can customize learning path

Expected Result: Instructors can modify the learning path to cater to varying learning needs, creating an adaptable and versatile educational experience for learners.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olpL05

Test Case Title: Verify that notifications are sent for upcoming courses or milestones

Test Case Description: Confirm that learners receive timely notifications for upcoming courses, modules, or milestones, ensuring they are well-prepared and informed about their learning journey.

Test Suite: Learning Paths and Curriculum

Test Priority: Low

Preconditions:

- Learner is logged in
- Learning path is created with scheduled courses or milestones

Test Data: Learner, learning path, and scheduled events

Test Steps:

1. Access the learning path as a learner
2. Monitor notifications
3. Observe scheduling of courses and milestones
4. Ensure notifications are received prior to course or milestone start dates
5. Check for consistency in notification frequency

Postconditions:

- Learners receive notifications for scheduled courses or milestones

Expected Result: Learners receive timely notifications for upcoming courses or milestones within the learning path, enhancing their overall learning experience.

Severity: Minor

Type of Testing: Functional

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:**

Test Case ID: olPf1A

Test Case Title: Verify that attendance is automatically recorded during live sessions

Test Case Description: Ensure that the system accurately and automatically records learner attendance during live classes and webinars.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Instructor has scheduled a live session
- Learner is logged in

Test Data: Learner account, live session details

Test Steps:

1. Learner joins a live session
2. Learner remains in the session for the entire duration
3. Learner leaves the session
4. Instructor checks the attendance report
5. System administrator checks the attendance record in the backend

Postconditions:

- Learner is marked as present in the attendance report
- Attendance record is saved and visible in the backend

Expected Result: The system accurately records the learner's attendance during the live session, and this information is available in both the instructor's attendance report and the backend attendance record.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: kjHg2B

Test Case Title: Verify that instructors can view attendance reports for each session

Test Case Description: Ensure that the system generates and displays accurate attendance reports for each live session, allowing instructors to monitor learner participation.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Instructor has scheduled and conducted a live session

Test Data: Instructor account, live session details

Test Steps:

1. Instructor goes to the session management page
2. Instructor selects the completed session
3. Instructor clicks on the 'View Attendance Report' button

Test Postconditions:

- Attendance report appears with accurate attendance data

Expected Result: The system generates and displays an accurate attendance report for the selected live session, with correct attendance data for each learner who participated.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: vcXz3C

Test Case Title: Verify that learners can see their attendance records

Test Case Description: Ensure that the system provides learners with access to their attendance records, allowing them to track their participation in live sessions.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Learner is logged in

Test Data: Learner account, attendance records

Test Steps:

1. Learner goes to their profile page
2. Learner clicks on the 'Attendance Records' tab
3. Learner checks the attendance data for live sessions

Test Postconditions:

- Attendance records are displayed with accurate attendance information

Expected Result: The system shows learners their attendance records for live sessions, displaying accurate attendance data and enabling them to monitor their participation.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: bnMn4D

Test Case Title: Verify that attendance data is integrated with participation metrics

Test Case Description: Ensure that the system correlates attendance data with other participation metrics,

providing instructors a comprehensive view of learner engagement in live sessions.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Instructor has scheduled and conducted a live session
- Learners have attended the session

Test Data: Instructor account, learner accounts, live session details, participation metrics

Test Steps:

1. Instructor goes to the session management page
2. Instructor selects the completed session
3. Instructor clicks on the 'View Participation Metrics' button

Test Postconditions:

- Attendance data is displayed along with other participation metrics (e.g., time spent in session, questions asked, etc.)

Expected Result: The system correlates attendance data with other participation metrics, allowing instructors to assess learner engagement in live sessions more effectively.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: qrSt5E

Test Case Title: Verify that notifications are sent for missed sessions

Test Case Description: Ensure that the system sends notifications to learners when they miss a scheduled live session, promoting accountability and encouraging participation.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Instructor has scheduled a live session
- Learner is enrolled in the session and has notifications enabled

Test Data: Instructor account, learner account, live session details, notification system

Test Steps:

1. Instructor schedules a live session
2. Learner does not join the session

Test Postconditions:

- Learner receives a notification about missing the session (email, app notification, etc.)

Expected Result: The system sends notifications to learners when they miss a scheduled live session, encouraging them to maintain participation and accountability.

Severity: Major

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:**

Test Case ID: olcLM1

Test Case Title: Verify that content licenses are assigned to each resource

Test Case Description: Ensure that the system assigns a content license to each educational resource uploaded by the content creator.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Content creator is logged in
- Content creator uploads a new resource

Test Data: Newly uploaded educational resource

Test Steps:

1. Log in as a content creator
2. Upload a new educational resource
3. Check if the resource has a license assigned

Postconditions:



- Content license is assigned

Expected Result: Each educational resource has a content license assigned, which is visible to the content creator.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olcLM2

Test Case Title: Verify that usage rights and permissions are clearly defined

Test Case Description: Check if the system clearly defines and displays the usage rights and permissions for each content license.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Content creator is logged in
- Educational resource with a content license is uploaded

Test Data: Uploaded educational resource with content license

Test Steps:

1. Log in as a content creator
2. Upload an educational resource
3. Assign a content license
4. Review usage rights and permissions

Postconditions:

- Usage rights and permissions are visible

Expected Result: The system clearly defines and displays the usage rights and permissions for the content license, which is visible to the content creator.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olcLM3

Test Case Title: Verify that content creators can set restrictions on usage and distribution

Test Case Description: Ensure that the content creator can set restrictions on usage and distribution for the content license.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Content creator is logged in
- Educational resource with a content license is uploaded

Test Data: Uploaded educational resource with content license

Test Steps:

1. Log in as a content creator
2. Upload an educational resource
3. Assign a content license
4. Set restrictions on usage and distribution
5. Save changes

Postconditions:

- Restrictions on usage and distribution are set

Expected Result: The content creator can set and save restrictions on usage and distribution for the content license, which should be visible in the system.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olcLM4

Test Case Title: Verify that instructors and learners are informed of content licensing terms

Test Case Description: Check if the system informs instructors and learners about content licensing terms when accessing the educational resource.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

- Content creator is logged in
- Educational resource with a content license is uploaded
- Instructor or learner logs in

Test Data: Uploaded educational resource with content license

Test Steps:

1. Log in as an instructor or learner
2. Access the educational resource
3. Check for notification of content licensing terms

Postconditions:

- Content licensing terms are visible

Expected Result: The system informs instructors and learners about the content licensing terms when they access the educational resource.

Severity: Normal

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olcLM5

Test Case Title: Verify that notifications are sent for license expiration or renewal

Test Case Description: Ensure that the system sends notifications to the content creator for content license expiration or renewal.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

- Content creator is logged in
- Educational resource with a content license is uploaded with an expiration date

Test Data: Uploaded educational resource with content license and expiration date

Test Steps:

1. Log in as a content creator
2. Access the educational resource
3. Check for notification of license expiration or renewal

Postconditions:

- Notification is visible

Expected Result: The system sends a notification to the content creator for content license expiration or renewal, which should be visible in the system.

Severity: Normal

Type of Testing: Functional

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 ID: VCv5Eg

Test Case Title: Verify that instructors can create and save multiple versions of content

Test Case Description: Ensure that the system allows instructors to create and save various versions of course materials.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in
- Course is created

Test Data: New course material for versioning

Test Steps:

1. Go to the course materials section
2. Create new course material
3. Save a version of the material
4. Edit the material and save another version
5. Repeat steps 3 and 4 for more versions

Postconditions:

- Multiple versions of the course material are saved

Expected Result: The system allows instructors to create and save multiple versions of the course material.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: We8Gb

Test Case Title: Verify that version history is maintained and accessible

Test Case Description: Ensure that the system maintains and provides access to the history of content versions.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in
- Course with multiple versions of materials is created

Test Data: Access to course version history

Test Steps:

1. Go to the course materials section
2. Select a material with multiple versions
3. Access the version history
4. Review the details of each version

Postconditions:

- Version history is displayed with all required details

Expected Result: The system maintains a complete version history for each piece of course material, allowing instructors to access and review all versions.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: Rt6Gh

Test Case Title: Verify that instructors can compare different versions of content

Test Case Description: Ensure that the system allows instructors to compare different versions of course materials.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Course with multiple versions of materials is created

Test Data: Two different versions of a course material

Test Steps:

1. Go to the course materials section
2. Select a material with two versions
3. Compare the two versions

Postconditions:

- Differences between the two versions are displayed

Expected Result: The system allows instructors to easily compare different versions of course materials, highlighting any changes made between them.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: Yt9Gf

Test Case Title: Verify that previous versions can be restored if needed

Test Case Description: Ensure that the system allows instructors to restore previous versions of course materials if necessary.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Course with multiple versions of materials is created

Test Data: Previous version of a course material

Test Steps:

1. Go to the course materials section
2. Select a material with previous versions
3. Choose a previous version to restore
4. Confirm the restoration

Postconditions:

- Selected previous version is restored as the active version

Expected Result: The system allows instructors to restore previous versions of course materials as the active version if needed.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: Um7Gv

Test Case Title: Verify that notifications are sent for significant content updates



Test Case Description: Ensure that the system sends notifications for significant updates made to course materials.

Test Suite: Content Versioning

Test Priority: Low

Preconditions:

- Instructor is logged in
- Student is enrolled in the course
- Course notifications are enabled

Test Data: Significant update to a course material

Test Steps:

1. Go to the course materials section
2. Update a material significantly
3. Save the new version

Postconditions:

- Notification about the update is sent to the students

Expected Result: The system sends notifications to enrolled students whenever significant updates are made to course materials.

Severity: Medium

Type of Testing: Functional

Test Case Behaviour: Positive

-----

**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:**

Test Case ID: OLP-API-1

Test Case Title: Verify that APIs are available for key platform functionalities

Test Case Description: Ensure the online learning platform provides APIs for essential functionalities such as course details, user management, and progress tracking.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API documentation is available

Test Data: Course details, user account, progress tracking data

Test Steps:

1. Access the API documentation
2. Identify essential platform functionalities
3. Verify if APIs are available for each functionality
4. Check the API endpoints
5. Inspect the API responses

Postconditions:

- APIs for key functionalities are identified

Expected Result: The system has APIs available for course details, user management, and progress tracking.

Severity: Major

Type of Testing: API Testing

Test Case Approach: Positive

Test Case ID: OLP-API-2

Test Case Title: Verify comprehensive API documentation is provided

Test Case Description: Validate that the API documentation covers all necessary aspects, including endpoints, request parameters, response formats, and security protocols.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API documentation is available

Test Data: N/A

Test Steps:

1. Access the API documentation
2. Review the endpoint definitions (GET, POST, PUT, DELETE)
3. Analyze request parameters, headers, and body requirements
4. Inspect response formats (structured data, error codes, etc.)

5. Verify security protocols (API keys, authentication, rate limiting, etc.)

Postconditions:

- Comprehensive API documentation is reviewed

Expected Result: The API documentation is comprehensive, covering endpoints, request parameters, response formats, and security protocols.

Severity: Major

Type of Testing: Documentation Review

Test Case Approach: Positive

Test Case ID: OLP-API-3

Test Case Title: Verify that third-party applications can authenticate and access APIs

Test Case Description: Ensure third-party applications can successfully authenticate and use the platform's APIs.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API documentation is available
- A test third-party application is developed

Test Data: API keys, authentication details, endpoint data

Test Steps:

1. Register a test third-party application
2. Obtain API keys and authentication credentials
3. Implement third-party application to access the platform's API
4. Test various endpoints using the third-party application
5. Verify successful API calls and responses

Postconditions:

- Third-party applications authenticate and access APIs

Expected Result: The test third-party application successfully authenticates and accesses the platform's API endpoints.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: OLP-API-4

Test Case Title: Verify rate limiting and monitoring features are in place

Test Case Description: Validate that the platform implements rate limiting and monitoring features to ensure API performance and security.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

- API documentation is available

Test Data: Multiple API calls, various endpoints

Test Steps:

1. Perform API calls at various endpoints
2. Monitor the number and frequency of API calls
3. Check for error messages or limitations
4. Analyze logs for API performance, usage, and security

Postconditions:

- Rate limiting and monitoring features are verified

Expected Result: The platform implements rate limiting and monitoring features, with accurate error messages and log analysis.

Severity: Major

Type of Testing: Non-Functional Testing

Test Case Approach: Positive

Test Case ID: OLP-API-5

Test Case Title: Verify that secure access protocols are implemented

Test Case Description: Ensure that secure access protocols, such as OAuth, are enforced for API authentication.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API documentation is available

Test Data: API keys, authentication details, endpoint data

Test Steps:

1. Access the API documentation
2. Check the authentication protocol(s) required
3. Implement OAuth authentication in a test third-party application
4. Test API calls with valid and invalid credentials
5. Analyze response errors for invalid credentials

Postconditions:

- Secure access protocols are implemented

Expected Result: The platform enforces secure access protocols (e.g., OAuth) for API authentication, blocking unauthorized access attempts.

Severity: Major

Type of Testing: Security Testing

Test Case Approach: Negative

-----

**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:**

Test Case ID: OLP-DA-01

Test Case Title: Verify that data visualization tools are available for analysis

Test Case Description: As an administrator, I want to ensure that data visualization tools are accessible and functional so that I can effectively analyze learner performance.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in
- Data is available for analysis

Test Data: Prepopulated data set for learner performance and engagement

Test Steps:

1. Log in as an administrator
2. Navigate to the data analytics dashboard
3. Verify that at least one data visualization tool (e.g., chart, graph, or diagram) is present
4. Interact with the data visualization tool (e.g., hover over data points or select a section) to ensure functionality
5. Verify that the data displayed is related to learner performance and engagement

Postconditions:

- Data visualization tools are accessible and functional

Expected Result: Data visualization tools are available, interactive, and display relevant data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: OLP-DA-02

Test Case Title: Verify that reports can be generated for learner performance metrics

Test Case Description: As an administrator, I want to generate reports to access detailed learner performance metrics for further analysis.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in
- Data is available for report generation



Test Data: Prepopulated data set for learner performance and engagement

Test Steps:

1. Log in as an administrator
2. Navigate to the data analytics dashboard
3. Select the option for report generation
4. Choose a predefined report or create a custom report
5. Confirm that the generated report includes relevant data such as completion rates, quiz scores, and time spent on courses

Postconditions:

- Reports containing learner performance metrics are generated

Expected Result: The system generates detailed, accurate reports on learner performance metrics.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: OLP-DA-03

Test Case Title: Verify that real-time analytics provide insights into engagement and outcomes

Test Case Description: As an administrator, I want real-time analytics capabilities to monitor learner engagement and outcomes proactively.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in
- Data is being actively collected from learners

Test Data: Prepopulated data set for ongoing learner activity

Test Steps:

1. Log in as an administrator

2. Navigate to the real-time analytics dashboard
3. Verify that data is currently being displayed and updated
4. Observe the data for at least five minutes to ensure that it reflects current learner engagement and outcomes
5. Check that data includes relevant metrics such as concurrent logins, course progress, and quiz results

Postconditions:

- Real-time analytics display up-to-date information

Expected Result: Real-time analytics provide accurate, current insights into learner engagement and outcomes.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: OLP-DA-04

Test Case Title: Verify that data can be exported for further analysis

Test Case Description: As an administrator, I want to export data for external analysis using third-party tools or sharing with stakeholders.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator is logged in
- Data is available for export

Test Data: Prepopulated data set for learner performance and engagement

Test Steps:

1. Log in as an administrator
2. Navigate to the data analytics dashboard
3. Select the export data option
4. Choose the desired data format (e.g., CSV, XLSX, or JSON)
5. Confirm the successful export of data

Postconditions:

- Data is exported in a usable format

Expected Result: Data is successfully exported in the desired format for further analysis.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: OLP-DA-05

Test Case Title: Verify that notifications are sent for significant data trends or anomalies

Test Case Description: As an administrator, I want to receive notifications for significant data trends or anomalies to stay informed and take appropriate action.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator is logged in
- Data is being actively collected from learners

Test Data: Prepopulated data set with a significant trend or anomaly

Test Steps:

1. Log in as an administrator
2. Navigate to notification settings
3. Ensure that notification options for data trends or anomalies are enabled
4. Trigger a predefined significant trend or anomaly in the data
5. Verify that a notification is received through the configured channel (e.g., email or platform notification)

Postconditions:

- Notifications are sent for significant data trends or anomalies

Expected Result: The system sends notifications for predefined significant data trends or anomalies.

Severity: Major

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:**

Test Case ID: olPL1

Test Case Title: Verify that administrators can create and assign user roles

Test Case Description: Ensure that the system allows administrators to create new user roles and assign them to users.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- Administrator is logged in
- Navigation to the user management section is possible

Test Data: New role name, user for assignment

Test Steps:

1. Navigate to user management
2. Click on "Create New Role"
3. Enter the new role name
4. Save the new role
5. Select a user
6. Assign the new role to the user

Test Postconditions:

- New role is created
- User is assigned the new role

Expected Result: The system creates the new user role and assigns it to the selected user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olPL2

Test Case Title: Verify that permissions are customizable for each role

Test Case Description: Ensure that the system provides the ability to customize permissions for each user role.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- Administrator is logged in
- An existing user role is available

Test Data: Customized permission sets

Test Steps:

1. Navigate to user management
2. Select an existing user role
3. Edit the role permissions
4. Save the changes

Test Postconditions:

- Permission changes are saved and applied to the role

Expected Result: The system saves and applies the customized permission sets for the selected user role.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olPL3

Test Case Title: Verify that users are restricted to functionalities based on their roles

Test Case Description: Ensure that users can only access features and functionalities based on their assigned roles.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- Multiple users with different roles are created

Test Steps:

1. Log in as a user with a specific role
2. Attempt to access functionalities available only for other roles

Test Postconditions:

- User can only access functionalities allowed for their role

Expected Result: Users can only access the features and functionalities corresponding to their assigned roles.

Severity: Major

Type of Testing: Security

Test Case Approach: Negative

Test Case ID: olPL4

Test Case Title: Verify that changes to roles and permissions are logged

Test Case Description: Ensure that the system records changes to user roles and permissions in a log.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- Administrator is logged in
- A user role and permission change is performed

Test Data: Logs

Test Steps:

1. Navigate to user management
2. Perform a role or permission change

Test Postconditions:

- A log entry is created

Expected Result: The system records the role or permission change in a log.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: olPL5

Test Case Title: Verify that users are notified of their assigned roles and permissions

Test Case Description: Ensure that the system notifies users of their assigned roles and permissions.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- A new user is created
- The user is assigned a role and permissions

Test Data: User notification

Test Steps:

1. Create a new user account
2. Assign a user role and permissions

Test Postconditions:

- User receives a notification about their assigned role and permissions

Expected Result: The system notifies the new user about their assigned role and permissions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: olPfR1

Test Case Title: Verify that learners can rate courses and instructors

Test Case Description: Ensure that learners can provide star ratings for courses and instructors on a scale of 1 to 5.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- Learner is logged in
- Course or instructor details page is open

Test Data: Course or instructor details

Test Steps:

1. Navigate to the course or instructor details page.
2. Locate the rating section.
3. Select a rating between 1 and 5.
4. Confirm the rating submission.

Postconditions:

- Rating is saved and displayed
- System triggers any notification or updates aggregated ratings

Expected Result: The system accepts the rating, saves it in the database, and displays the rating on the course or instructor details page.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPfR2

Test Case Title: Verify that feedback forms are available for detailed comments

Test Case Description: Ensure that learners can access feedback forms to provide detailed comments regarding courses and instructors.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- Learner is logged in
- Course or instructor details page is open

Test Data: Course or instructor details

#### Test Steps:

1. Navigate to the course or instructor details page.
2. Locate the feedback form.
3. Fill out the feedback form with valid input.
4. Submit the feedback form.

#### Postconditions:

- Feedback is saved

Expected Result: The system saves the feedback and associates it with the correct course or instructor.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPfR3

Test Case Title: Verify that instructors can view and respond to feedback

Test Case Description: Ensure that instructors can access and reply to learner feedback on their courses or personal profiles.

Test Suite: User Feedback and Rating System

Test Priority: High

#### Preconditions:

- Instructor is logged in
- Feedback from a learner exists

Test Data: Learner feedback

#### Test Steps:

1. Navigate to the instructor's profile or course page.
2. Locate the learner feedback.

3. Read the learner feedback.
4. Write a response to the learner feedback.
5. Submit the response.

Postconditions:

- Response is saved and displayed

Expected Result: The system saves the instructor's response and associates it with the original learner feedback.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPfR4

Test Case Title: Verify that aggregated ratings are displayed publicly

Test Case Description: Ensure that the platform calculates and displays the aggregated ratings for courses and instructors based on individual learner ratings.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

- Multiple learners have rated courses or instructors

Test Data: Multiple ratings for a course or instructor

Test Steps:

1. Navigate to the course or instructor details page.
2. Locate the aggregated rating.
3. Verify that the aggregated rating is accurate based on the available ratings data.

Expected Result: The system calculates the aggregated rating correctly and displays it on the course or

instructor details page.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: olPfR5

Test Case Title: Verify that anonymous feedback options are available

Test Case Description: Ensure that learners have the option to provide feedback anonymously, hiding their identities from instructors.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

- Learner is logged in
- Feedback form is available

Test Data: No test data needed

Test Steps:

1. Navigate to the feedback form.
2. Fill out the feedback form.
3. Before submitting, select the "Anonymous" option.
4. Submit the feedback form.

Postconditions:

- Feedback is saved anonymously

Expected Result: The system saves the feedback without associating it with the learner's account, preserving their anonymity.

Severity: Major

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:**

Test Case ID: olpSMI1

Test Case Title: Verify that learners can connect their social media accounts to the platform

Test Case Description: Ensure that learners have the ability to link their social media profiles to their account on the online learning platform.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- Learner is logged into the online learning platform
- Learner has access to their social media account

Test Data: Learner's social media account credentials

Test Steps:

1. Log in as a learner
2. Navigate to the account settings page
3. Click on the "Connect Social Media" button
4. Select the desired social media platform from the list
5. Enter the login credentials for the chosen social media platform
6. Authorize the connection between the online learning platform and the social media account

Postconditions:

- Social media account is successfully linked to the learner's account

Expected Result: The system establishes a connection between the learner's account and their chosen social media platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olpSMI2

Test Case Title: Verify that learners can share achievements on social media

Test Case Description: Ensure that learners can post their course completions, certificates, and other achievements on their linked social media accounts.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- Learner is logged into the online learning platform
- Learner has at least one social media account linked
- Learner has completed at least one course and earned an achievement

Test Data: Course completion certificate or badge

Test Steps:

1. Log in as a learner
2. Navigate to the "My Achievements" page
3. Select the achievement to be shared
4. Click on the "Share" button
5. Choose the desired social media platform from the list
6. Edit the post as desired and post it

Postconditions:

- Achievement post appears on the chosen social media platform

Expected Result: The system shares the learner's achievement on the selected social media platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olpSMI3

Test Case Title: Verify that learners can control the privacy settings of shared content

Test Case Description: Ensure that learners can manage the visibility of their shared achievements by configuring privacy settings.



Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- Learner is logged into the online learning platform
- Learner has at least one social media account linked
- Learner has completed at least one course and earned an achievement

Test Data: Course completion certificate or badge

Test Steps:

1. Log in as a learner
2. Navigate to the "My Achievements" page
3. Select the achievement to be shared
4. Click on the "Share" button
5. Choose the desired social media platform from the list
6. Edit the post to include privacy settings and post it
7. Verify the post's visibility by logging into the social media account of another user

Postconditions:

- Achievement post appears on the chosen social media platform with the configured privacy settings

Expected Result: The system shares the learner's achievement on the chosen social media platform with the configured privacy settings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olpSMI4

Test Case Title: Verify that instructors can promote courses on social media

Test Case Description: Ensure that instructors can broadcast course updates, announcements, and promotions through their linked social media accounts.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- Instructor is logged into the online learning platform
- Instructor has at least one social media account linked

Test Data: Course update, announcement, or promotional material

Test Steps:

1. Log in as an instructor
2. Navigate to the course management page
3. Click on the "Promote via Social Media" button
4. Choose the desired social media platform from the list
5. Edit the post as desired and post it

Postconditions:

- Course update post appears on the chosen social media platform

Expected Result: The system shares the instructor's course update on the selected social media platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: olpSMI5

Test Case Title: Verify that analytics track the reach and engagement of shared content

Test Case Description: Ensure that social media engagement metrics, such as likes, shares, and comments, are tracked and displayed within the online learning platform.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- A learner or instructor shares a piece of content

- The shared content receives likes, shares, and/or comments on the social media platform

Test Data: Social media engagement metrics (e.g., likes, shares, comments)

Test Steps:

1. Log in as a user with permissions to view analytics (admin or instructor)
2. Navigate to the social media integration analytics page
3. Verify that the shared content is listed along with its engagement metrics

Postconditions:

- Social media engagement metrics are correctly displayed within the online learning platform

Expected Result: The system accurately presents social media engagement metrics for shared content within the online learning platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Test Case ID: At1jgC

Test Case Title: Verify that the app records data on distance, duration, pace, and calories burned

Test Case Description: Ensure that the Fitness Tracker App accurately captures and displays data related to distance, duration, pace, and calories burned during various activities.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

- User has logged in
- User has started tracking an activity (e.g., running, walking, cycling, swimming)
- Wearable device or fitness equipment is connected and functioning properly

Test Data: N/A

Test Steps:

1. Begin tracking an activity using the Fitness Tracker App
2. Perform the activity for a specified duration
3. Stop tracking the activity
4. Review the captured data for accuracy and completeness

5. Compare the captured data with the data from the wearable device or fitness equipment

Postconditions:

- Data is accurately recorded and displayed in the app

Expected Result: The Fitness Tracker App records and displays data on distance, duration, pace, and calories burned accurately.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: Nw2sD8

Test Case Title: Verify that users can view activity summaries

Test Case Description: Ensure that the Fitness Tracker App allows users to access and review summaries of their completed activities, providing an overview of their performance and progress towards fitness goals.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User has logged in
- User has completed at least one activity (e.g., running, walking, cycling, swimming)

Test Data: N/A

Test Steps:

1. Go to the "Activities" section of the Fitness Tracker App
2. Select a completed activity in the list
3. Review the summary details for that activity (e.g., distance, duration, pace, calories burned)

Postconditions:

- Activity summary is displayed accurately

Expected Result: The Fitness Tracker App displays a clear, concise, and accurate summary of the user's completed activity, including details on distance, duration, pace, and calories burned.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: zsJh6g

Test Case Title: Verify that data is stored for historical analysis

Test Case Description: Ensure that the Fitness Tracker App securely stores activity data in a database or cloud storage, allowing users to access their historical data for analysis, comparison, and trend identification.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User has logged in
- User has completed at least one activity (e.g., running, walking, cycling, swimming)

Test Data: N/A

Test Steps:

1. Go to the "Activities" section of the Fitness Tracker App
2. Select the "History" or "Analysis" option
3. Review the list of completed activities, filtered by date or category
4. Select a specific activity from the list
5. Review the data and insights for that activity

Postconditions:

- Historical activity data is accessible and displayed accurately

Expected Result: The Fitness Tracker App securely stores and retrieves user activity data, providing a historical record for comparison and analysis.

Severity: Major

Type of Testing: Data-Driven Testing

Test Case Approach: Positive

Test Case ID: wk5cXp

Test Case Title: Verify that activity tracking can be paused and resumed

Test Case Description: Ensure that the Fitness Tracker App allows users to temporarily pause activity tracking, and then resume where they left off, enabling smooth continuity during interruptions or transitions (e.g., unexpected breaks, switching activities).

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User has logged in
- User has started tracking an activity (e.g., running, walking, cycling, swimming)

Test Data: N/A

Test Steps:

1. Begin tracking an activity using the Fitness Tracker App
2. Pause the activity (e.g., press the "pause" or "stop" button)
3. Perform an unrelated task for a specified duration (e.g., answering a phone call)
4. Resume the activity (e.g., press the "start" or "resume" button)
5. Complete the activity and stop tracking

Postconditions:

- Data is accurately recorded and displayed for the entire activity duration (including the paused portion)

Expected Result: The Fitness Tracker App allows users to pause and resume activity tracking without losing data, ensuring accurate and continuous records of user activities.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: Hr8vtL

Test Case Title: Verify that users receive notifications for milestones

Test Case Description: Ensure that the Fitness Tracker App notifies users when they reach predefined milestones (e.g., distance, duration, or calories burned), encouraging motivation and progress towards fitness goals.

Test Suite: Activity Tracking

Test Priority: Low

Preconditions:

- User has logged in
- User has set specific fitness or activity goals
- User has started tracking an activity (e.g., running, walking, cycling, swimming)

Test Data: N/A

Test Steps:

1. Set activity goals in the Fitness Tracker App
2. Begin tracking an activity using the Fitness Tracker App
3. Perform the activity until a milestone is reached (e.g., a specified distance or duration)

Postconditions:

- Notification is displayed (e.g., on-screen alert, sound, vibration)

Expected Result: The Fitness Tracker App provides notifications when users reach predefined milestones during their activities, promoting motivation and engagement towards fitness goals.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template for Fitness Tracker App's GPS Tracking Feature:

Test Case ID: GPS001

Test Case Title: Verify that GPS tracking accurately records route data

Test Case Description: Ensure the Fitness Tracker App's GPS tracking feature accurately records various outdoor activity routes.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- App is installed and opened
- User is outdoors with clear GPS signal
- User has granted location permissions

Test Data: Pre-defined routes, such as a park trail or a running track

Test Steps:

1. Start recording the workout
2. Follow the pre-defined route
3. Stop recording after completing the route
4. Check the recorded route on the map
5. Compare the recorded data with the pre-defined route

Postconditions:

- Route data matches the pre-defined route

Expected Result: The GPS tracking feature accurately records and displays the user's outdoor activity route.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: GPS002

Test Case Title: Verify that users can view detailed maps and elevation profiles

Test Case Description: Ensure users can visualize the workout maps and elevation profiles for detailed analysis.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- App is installed and opened

- User has completed a workout with GPS tracking

Test Data: Recorded workout data

Test Steps:

1. Navigate to the workout history
2. Select a recorded workout
3. View the workout route on the map
4. Analyze the elevation profile

Postconditions:

- Workout map is displayed accurately
- Elevation profile is displayed correctly

Expected Result: The Fitness Tracker App displays workout maps and elevation profiles for detailed analysis.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: GPS003

Test Case Title: Verify that performance data includes location-based metrics

Test Case Description: Validate that user performance data includes location-based workout metrics.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- App is installed and opened
- User has completed a workout with GPS tracking

Test Data: Recorded workout data

Test Steps:

1. Navigate to the workout history
2. Select a recorded workout

3. Check the location-based metrics, such as distance traveled and average speed

Postconditions:

- Location-based metrics are displayed

Expected Result: The Fitness Tracker App includes location-based metrics in the user performance data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: GPS004

Test Case Title: Verify that routes can be saved and revisited

Test Case Description: Ensure users can save and revisit their workout routes.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- App is installed and opened
- User has completed a workout with GPS tracking

Test Data: Recorded workout data

Test Steps:

1. Navigate to the workout history
2. Select a recorded workout
3. Save the workout route
4. Navigate to the saved routes
5. Select the saved route and check if it displays correctly

Postconditions:

- Workout route is saved
- Saved routes are displayed accurately

Expected Result: The Fitness Tracker App allows users to save and revisit their workout routes.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: GPS005

Test Case Title: Verify that notifications are provided for deviations from planned routes

Test Case Description: Ensure users receive notifications when they deviate from their planned workout routes.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- App is installed and opened
- User has a pre-defined route saved
- User has granted location permissions

Test Data: Pre-defined route

Test Steps:

1. Start recording a workout
2. Follow the pre-defined route
3. Deviate from the route

Postconditions:

- Notification is displayed

Expected Result: The Fitness Tracker App notifies users when they deviate from their pre-defined workout routes.

Severity: Minor

Type of Testing: Functional 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:**

Test Case ID: hrTRk1

Test Case Title: Verify that heart rate data is recorded in real-time

Test Case Description: Ensure that the Fitness Tracker App accurately captures and displays heart rate data in real-time.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in
- Wearable device is connected and monitoring heart rate

Test Data: Real-time heart rate data from a wearable device

Test Steps:

1. Connect a compatible wearable device
2. Start monitoring heart rate on the device
3. Open the Fitness Tracker App
4. Navigate to the Heart Rate Monitoring feature
5. Check that the heart rate data matches the data from the wearable device

Postconditions:

- Real-time heart rate data is accurately displayed on the app

Expected Result: The Fitness Tracker App displays real-time heart rate data provided by a connected wearable device.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: hrTRk2

Test Case Title: Verify that users can set and monitor target heart rate zones

Test Case Description: Validate that users can set target heart rate zones, monitor their current heart rate relative to these zones, and receive appropriate feedback.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in
- User has previously recorded heart rate data

Test Data: Previously recorded heart rate data for setting target zones

Test Steps:

1. Open the Fitness Tracker App
2. Navigate to the Heart Rate Monitoring feature
3. Set a target heart rate zone (e.g., fat burn, cardio)
4. Perform an exercise that increases the user's heart rate
5. Verify that the current heart rate is accurately displayed relative to the target zone
6. Confirm that appropriate feedback is given (e.g., "In the zone," "Above zone")

Postconditions:

- Target heart rate zones are set and displayed
- Current heart rate is displayed and accurately monitored relative to target zone

Expected Result: The Fitness Tracker App allows users to set and monitor target heart rate zones and provides appropriate feedback.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: hrTRk3

Test Case Title: Verify that historical heart rate data is stored and viewable

Test Case Description: Confirm that the Fitness Tracker App saves and displays historical heart rate data for users.

Test Suite: Heart Rate Monitoring

Test Priority: Medium



Preconditions:

- User is logged in
- User has previously recorded heart rate data

Test Data: Previously recorded heart rate data

Test Steps:

1. Open the Fitness Tracker App
2. Navigate to the Heart Rate Monitoring feature
3. Access the historical data menu
4. Verify that the app displays previously recorded heart rate data
5. Select a specific date and time range
6. Confirm that the selected data is accurately displayed

Postconditions:

- Historical heart rate data is saved and accessible

Expected Result: The Fitness Tracker App stores and displays historical heart rate data for users.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: hrTRk4

Test Case Title: Verify that alerts are provided for abnormal heart rate readings

Test Case Description: Validate that the Fitness Tracker App notifies users when heart rate readings are outside of the normal range.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

- User is logged in
- Wearable device is connected and monitoring heart rate

- User has set an upper and lower heart rate threshold for abnormal readings

Test Data: Abnormal heart rate data from a wearable device

Test Steps:

1. Connect a compatible wearable device
2. Start monitoring heart rate on the device
3. Open the Fitness Tracker App
4. Navigate to the Heart Rate Monitoring feature
5. Record abnormal heart rate data (e.g., too high or too low)

Postconditions:

- An alert is displayed on the Fitness Tracker App

Expected Result: The Fitness Tracker App provides an alert for abnormal heart rate readings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case ID: hrTRk5

Test Case Title: Verify that data is compatible with various wearable devices

Test Case Description: Ensure that the Fitness Tracker App can connect to and receive data from multiple types of wearable devices.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- Different wearable devices (e.g., smartwatches, fitness trackers, heart rate monitors) are available for testing

Test Data: Data from various wearable devices

Test Steps:

1. Connect a compatible wearable device
2. Start monitoring heart rate on the device

3. Open the Fitness Tracker App
4. Navigate to the Heart Rate Monitoring feature
5. Verify that the heart rate data matches the data from the wearable device

Postconditions:

- Data from various wearable devices is accurately displayed on the app

Expected Result: The Fitness Tracker App can connect and receive data from multiple types of wearable devices.

Severity: Major

Type of Testing: Compatibility 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:**

Test Case ID: FC101

Test Case Title: Verify that users can log daily food intake and calorie consumption

Test Case Description: Ensure that the Fitness Tracker App allows users to add food items consumed during the day, along with their calorie values, to accurately track calorie intake.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged in
- App has access to calorie database

Test Data: List of consumed food items with their respective calorie values

Test Steps:

1. Tap on the "Add Food Intake" button
2. Select food items from the calorie database
3. Enter the quantity of each food item consumed
4. Save the food intake data

Postconditions:

- Food intake is logged in the user's diary
- Calorie intake is calculated and displayed

Expected Result: The system accurately logs the food intake, calculates calorie consumption, and displays the total in a readable format.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: FC102

Test Case Title: Verify that the app calculates and displays total calories consumed and burned

Test Case Description: Ensure that the Fitness Tracker App calculates total calories consumed and burned, and displays the values in an easily understandable format.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has food intake logged for the day
- User has physical activity logged for the day

Test Data: Food intake, physical activity data

Test Steps:

1. Navigate to the "Calorie Summary" screen
2. Observe the displayed calorie values

Postconditions:

- Total calories consumed are displayed
- Total calories burned are displayed

Expected Result: The system accurately calculates and displays both total calories consumed and burned on the "Calorie Summary" screen.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: FC103

Test Case Title: Verify that users can set daily calorie goals and track progress

Test Case Description: Ensure that the Fitness Tracker App allows users to create daily calorie goals and monitor their progress towards achieving those goals.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Calorie goal value

Test Steps:

1. Go to the "Settings" menu
2. Set a daily calorie goal
3. Navigate to the "Calorie Summary" screen
4. Observe the progress towards the calorie goal

Postconditions:

- Daily calorie goal is set
- Progress towards the calorie goal is displayed

Expected Result: The system enables users to create a daily calorie goal and displays their progress toward it in real-time.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: FC104

Test Case Title: Verify that food items and calorie information are searchable in the database

Test Case Description: Ensure that users can easily find food items and their associated calorie values in the app's built-in database.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- App has access to calorie database

Test Data: Search keywords for food items

Test Steps:

1. Access the calorie database
2. Enter search keywords for specific food items
3. Observe the search results

Postconditions:

- Relevant food items are displayed

Expected Result: The system provides users with search functionality to quickly find and select food items from the calorie database and display their associated calorie values.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: FC105

Test Case Title: Verify that users receive alerts for exceeding calorie goals

Test Case Description: Ensure that the Fitness Tracker App sends alerts to users when they have exceeded their daily calorie goals.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has a daily calorie goal set
- User has logged food intake for the day
- User has exceeded their daily calorie goal

Test Data: Daily calorie goal, exceeded calorie value

Test Steps:

1. Navigate to the "Calorie Summary" screen
2. Observe if any alert is displayed about exceeding the daily calorie goal

Postconditions:

- Alert displayed for exceeding daily calorie goal

Expected Result: The system generates alerts or notifications when users exceed their daily calorie goals to help them maintain awareness of their energy intake.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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:**

Test Case Creation Template:

Test Case ID: ftS1

Test Case Title: Verify that sleep data is recorded including duration and quality

Test Case Description: Ensure the app accurately records sleep duration, stages, and quality metrics for a user.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has wearable device or fitness equipment connected
- User has enabled sleep tracking

Test Data: N/A

Test Steps:

1. Wear the connected device while sleeping
2. Sync the device with the app
3. Open the sleep tracking feature
4. Check sleep duration, stages, and quality metrics

Postconditions:

- Sleep data is displayed in the app

Expected Result: The app records sleep data including duration, stages, and quality metrics.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftS2

Test Case Title: Verify that users can view detailed sleep stages and patterns

Test Case Description: Confirm the user can access and understand their sleep stage data.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has sleep data recorded

Test Data: N/A

Test Steps:

1. Open the sleep tracking feature
2. Select a recorded sleep session
3. View sleep stage data (deep, REM, light)

Test Postconditions:

- Sleep stage data is displayed

Expected Result: The user can view detailed sleep stage data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftS3

Test Case Title: Verify that the app provides insights into sleep disturbances

Test Case Description: Validate the app's ability to detect and notify users of sleep disturbances.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has sleep data recorded with disturbances

Test Data: N/A

Test Steps:

1. Open the sleep tracking feature
2. Select a recorded sleep session with disturbances
3. Review disturbance notifications

Test Postconditions:

- Sleep disturbance notifications are displayed

Expected Result: The app provides insights into sleep disturbances, notifying users when detected.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftS4

Test Case Title: Verify that users receive recommendations for improving sleep quality

Test Case Description: Confirm the app offers personalized recommendations based on sleep data.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- User has sleep data recorded
- Sleep data includes poor sleep quality or disturbances

Test Data: N/A

Test Steps:

1. Open the sleep tracking feature
2. Select a recorded sleep session with poor quality or disturbances
3. Review personalized sleep improvement recommendations

Test Postconditions:

- Sleep improvement recommendations are displayed

Expected Result: The app generates personalized sleep improvement recommendations based on sleep data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftS5

Test Case Title: Verify that sleep data can be viewed in daily, weekly, and monthly summaries

Test Case Description: Ensure the user can view sleep data in various summary formats.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- User has sleep data recorded for at least a week

Test Data: N/A

Test Steps:

1. Open the sleep tracking feature
2. Switch to summary view
3. Select daily, weekly, or monthly summary

Test Postconditions:

- Sleep data summary is displayed as selected

Expected Result: The user can view sleep data in daily, weekly, and monthly summaries.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: ft1wkZ

Test Case Title: Verify that users can log various workout types with details

Test Case Description: Ensure the system allows users to log different workout types, including cardio, strength training, and flexibility exercises, along with necessary details such as duration and intensity.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User is logged in
- User is on the workout logging page

Test Data: Cardio workout (e.g., running), strength training workout (e.g., weightlifting), and flexibility workout (e.g., yoga) details

Test Steps:

1. Go to the workout logging page
2. Select a workout type (cardio, strength training, or flexibility)
3. Enter workout details including duration and intensity
4. Save the workout entry
5. Repeat steps 2-4 for other workout types

Postconditions:

- Workout entries for each type are saved in the system

Expected Result: The system saves workout entries for each selected type along with provided details.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: fgA8b2

Test Case Title: Verify that workout logs include duration, intensity, and notes

Test Case Description: Ensure the system records duration, intensity, and any additional notes or comments provided by the user for each logged workout.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User is logged in
- User has previously logged workouts

Test Data: Previously logged workouts with varying durations, intensities, and notes

Test Steps:

1. Go to the workout log page
2. Select a logged workout
3. Verify the workout's duration, intensity, and notes are displayed

Postconditions:

- Workout details are accurately displayed

Expected Result: The system displays the workout's duration, intensity, and any additional notes as provided by the user.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ghC4j4

Test Case Title: Verify that users can edit and delete workout entries

Test Case Description: Ensure users can modify or remove logged workout entries.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged in
- User has previously logged workouts

Test Data: Previously logged workouts requiring edits or deletion

Test Steps:

1. Go to the workout log page
2. Select a logged workout
3. Edit the workout details or delete the entry as needed
4. Save changes or confirm deletion

Postconditions:

- Workout entries are updated or removed

Expected Result: The system allows users to edit workout details or delete workout entries as requested.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: rfH9v1

Test Case Title: Verify that data is stored and viewable for tracking progress

Test Case Description: Ensure the system stores logged workout data for future viewing and progress tracking.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged in
- User has previously logged workouts

Test Data: Previously logged workouts

Test Steps:



1. Go to the workout log page
2. Filter logs by date or workout type
3. Verify the displayed log entries match the filter criteria

Postconditions:

- Log entries are correctly displayed based on filter criteria

Expected Result: The system displays the appropriate subset of logged workout entries according to filter selection.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: uiOp5

Test Case Title: Verify that users can filter logs by date and workout type

Test Case Description: Ensure users can sort and organize their workout logs using date and workout type filters for better data analysis.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged in
- User has previously logged workouts

Test Data: Previously logged workouts spanning various dates and workout types

Test Steps:

1. Go to the workout log page
2. Use date and workout type filters to narrow down the displayed logs
3. Verify the displayed log entries match the filter criteria

Postconditions:

- Log entries are correctly displayed based on filter criteria

Expected Result: The system displays the appropriate subset of logged workout entries according to filter selection.

Severity: Major

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:**

Test Case ID: ftG1a

Test Case Title: Verify that users can set goals for various fitness metrics

Test Case Description: Ensure that the system allows users to set personalized fitness goals for different metrics such as steps, calories, distance, and heart rate.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User is logged in
- App has access to necessary health metrics (e.g., steps, calories, distance, heart rate)

Test Data: Goal settings for steps (10,000 steps), calories (2,000 calories), distance (5 miles), and heart rate (120 bpm)

Test Steps:

1. Navigate to the goal setting page
2. Select a fitness metric (e.g., steps)
3. Set the goal value (e.g., 10,000 steps)
4. Save the goal
5. Repeat steps 2-4 for other metrics

Postconditions:

- Goals are saved and visible in the app

Expected Result: The system saves and displays the goals for each selected fitness metric.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: ftG2b

Test Case Title: Verify that progress tracking is available for each goal

Test Case Description: Ensure that users can view their progress towards their set fitness goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- User is logged in
- Fitness goals are set

Test Data: Previously set fitness goals

Test Steps:

1. Navigate to the goal tracking page
2. Select a fitness goal (e.g., steps)
3. Observe the displayed progress for the goal (e.g., number of steps taken today)
4. Repeat steps 2-3 for other fitness goals

Postconditions:

- Progress is displayed accurately

Expected Result: The system displays the user's progress towards each set fitness goal with appropriate tracking information.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: ftG3c

Test Case Title: Verify that users receive motivational feedback and reminders

Test Case Description: Ensure that the system provides motivational feedback and reminders to users based on their fitness goals and progress.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- User is logged in
- Fitness goals are set

Test Data: Previously set fitness goals

Test Steps:

1. Navigate to the goal tracking page
2. Observe the motivational feedback or reminders (e.g., "Great job! You've reached 75% of your daily step goal.")

Postconditions:

- Motivational feedback is provided

Expected Result: The system provides motivational feedback or reminders based on the user's progress towards their fitness goals.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: ftG4d

Test Case Title: Verify that goals can be adjusted or updated as needed

Test Case Description: Ensure that users can edit or update their fitness goals when needed.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- User is logged in
- Fitness goals are set

Test Data: Previously set fitness goals

Test Steps:

1. Navigate to the goal setting page
2. Select a fitness goal (e.g., steps)
3. Edit the goal value (e.g., change from 10,000 to 12,000 steps)
4. Save the edited goal

Postconditions:

- The goal is updated

Expected Result: The system allows users to edit their fitness goals, saving the updated values.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

---

Test Case ID: ftG5e

Test Case Title: Verify that the app provides a summary of goal attainment

Test Case Description: Ensure that users can view a summary of their progress towards their set fitness goals.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User is logged in
- Fitness goals are set

- Users have tracked their progress

Test Data: Previously set fitness goals and progress

Test Steps:

1. Navigate to the goal tracking page
2. Observe the summary view of the fitness goals (e.g., percentage progress)
3. Confirm that the system displays the summary for all fitness goals

Postconditions:

- Goal attainment summary is displayed

Expected Result: The system displays the user's progress towards each set fitness goal as a summary, with appropriate visualizations.

Severity: Major

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:**

Test Case ID: ft1A23

Test Case Title: Verify that users can select workout plans based on fitness levels

Test Case Description: Ensure that the fitness tracker app allows users to choose from a variety of workout plans designed for different fitness levels.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in
- User has access to the workout plan library

Test Data: Pre-designed workout plans for various fitness levels (beginner, intermediate, advanced)

Test Steps:

1. Open the workout plan library
2. View the list of available workout plans
3. Filter the plans by fitness level
4. Select a plan for a specific fitness level



## 5. Review the details of the selected plan

### Postconditions:

- Selected workout plan matches the chosen fitness level

Expected Result: The system displays appropriate workout plans based on the user-selected fitness level.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ft2B45

Test Case Title: Verify that plans include detailed exercise routines and schedules

Test Case Description: Verify that the selected workout plan contains complete exercise routines with clearly defined schedules.

Test Suite: Workout Plans

Test Priority: High

### Preconditions:

- User is logged in
- User has selected a workout plan

Test Data: Selected workout plan with exercise routines and schedules

### Test Steps:

1. Navigate to the selected workout plan
2. View the list of exercises included in the plan
3. Check each exercise for accuracy of instructions
4. Validate the schedule for each exercise routine (day of the week, time of day)

### Postconditions:

- Exercise routines have accurate instructions
- Schedules are appropriately set

Expected Result: The app displays detailed exercise routines with accurate instructions and corresponding

schedules for each workout in the selected plan.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ft3C67

Test Case Title: Verify that users can track adherence and progress within the plan

Test Case Description: Check that users can monitor their adherence to the selected workout plan and track their progress over time.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in
- User has selected a workout plan
- User has started the workout plan

Test Data: Workout plan adherence and progress data

Test Steps:

1. Access the workout plan dashboard
2. Review the adherence and progress statistics
3. Validate the percentages or values displayed

Postconditions:

- Adherence percentage/tracking is available
- Progress is accurately represented

Expected Result: The system displays accurate adherence and progress tracking data for the selected workout plan.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ft4D89

Test Case Title: Verify that the app provides notifications and reminders for workouts

Test Case Description: Ensure that the fitness tracker app sends timely notifications and reminders for users to follow their workout plans.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is logged in
- User has selected a workout plan
- User has enabled notifications and reminders

Test Data: Workout plan schedule data

Test Steps:

1. Wait for the scheduled time of the first workout
2. Verify that the app sends a notification or reminder
3. Check the accuracy of the notification (workout name, time, etc.)

Postconditions:

- Notification or reminder received at the scheduled time

Expected Result: The app sends a timely notification or reminder for the scheduled workout, displaying accurate workout details.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ft5E01

Test Case Title: Verify that users can adjust plans based on progress and preferences

Test Case Description: Confirm that users can modify their selected workout plans to accommodate their progress and changing preferences.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is logged in
- User has selected a workout plan

Test Data: Workout plan, user progress, and preferences

Test Steps:

1. Navigate to the workout plan settings
2. Adjust the plan based on user progress (increase/decrease difficulty)
3. Modify the plan according to user preferences (change exercises or schedule)
4. Save the updated plan
5. Review the modified plan to ensure accuracy

Postconditions:

- Workout plan has been updated to match user progress and preferences

Expected Result: The system allows users to modify their selected workout plans to accommodate their progress and changing preferences, displaying the updated plan accurately.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Test Case ID: ftCmR1

Test Case Title: Verify that users can join and participate in community groups

Test Case Description: Ensure that users have the ability to join community groups and participate by viewing, creating, and responding to posts.

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is logged in

- App has internet connection

Test Data: User account, community group name

Test Steps:

1. Navigate to the community section
2. Select 'Explore Groups' and choose a group to join
3. Tap 'Join Group' and confirm action
4. Wait for group moderator approval (if applicable)
5. Access the joined group
6. Create a new post and add relevant hashtags
7. Respond to another user's post

Postconditions:

- User is a member of the group
- User can view, create, and respond to posts

Expected Result: Users are able to join and participate in community groups, fostering engagement and interaction with other users.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ftCmR2

Test Case Title: Verify that users can share progress updates and achievements

Test Case Description: Ensure users have the capability to share their progress updates and achievements with their community.

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is logged in

- User has tracked some physical activity data (e.g., steps, distance, calories burned)

Test Data: User account, tracked physical activity data

Test Steps:

1. Navigate to the user's profile page
2. Select 'Share Progress'
3. Choose which metrics to share (steps, distance, calories, etc.)
4. Write a short description of the achievement or progress update
5. Tap 'Post' to share the update

Postconditions:

- Progress update is visible on the user's profile
- The update is shared with the user's followers or the public (depending on user settings)

Expected Result: The fitness tracker app enables users to share their progress updates and achievements, helping to keep them accountable and motivated.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ftCmR3

Test Case Title: Verify that the app facilitates participation in community challenges

Test Case Description: Ensure that users can participate in community challenges created by other users or moderators, view challenge rankings, and track their own progress.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to community groups
- Community challenge is available

Test Data: User account, community challenge

Test Steps:

1. Navigate to the community section
2. Select an active community challenge
3. Tap 'Join Challenge' and confirm action
4. Track physical activity data related to the challenge
5. Monitor challenge rankings to view progress

Postconditions:

- User is participating in the community challenge
- User's progress is being tracked and displayed

Expected Result: Users can join, view, and participate in community challenges, increasing engagement and fostering a strong sense of community.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ftCmR4

Test Case Title: Verify that users can offer and receive support and encouragement

Test Case Description: Ensure users are able to leave comments, reactions, and messages on other users' posts for support and encouragement within the community.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to community groups
- Another user's progress update has been shared

Test Data: User account, another user's progress update



#### Test Steps:

1. Navigate to the community section
2. Access another user's shared progress update
3. Leave a supportive comment, or provide encouragement through app-specific reactions (e.g., a thumbs-up or heart icon)
4. Check for notifications to determine if the other user responds to the comment or reaction

#### Postconditions:

- Supportive comments or reactions are visible on the shared progress update
- User receives a notification if the other user responds to the interaction (comment or reaction)

Expected Result: Users can provide support and encouragement to one another, building a supportive community environment.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ftCmR5

Test Case Title: Verify that community activity and engagement are tracked

Test Case Description: Ensure user activity and engagement within the community are properly tracked and displayed within the app and/or on the user's profile.

Test Suite: Community Support

Test Priority: Low

#### Preconditions:

- User is logged in
- User has access to community groups
- User has participated in community activities (joining groups, sharing progress updates, commenting, etc.)

Test Data: User account, community activities

#### Test Steps:

1. Navigate to the user's profile page
2. Check for 'Community' tab or similar
3. Review the community activities and engagement tracking information (joined groups, progress updates, comments, etc.)
4. Filter the community activities by specific time periods (e.g., today, this week, this month, etc.)

Postconditions:

- User's community activities and engagement are tracked and displayed

Expected Result: The fitness tracker app accurately tracks and showcases users' community activities and engagement, allowing them to monitor their involvement and maintain a strong connection with their community.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template for Fitness Tracker App:

Test Case ID: FTW-1

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: Ensure that the Fitness Tracker App allows users to connect and sync data from multiple wearable devices.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- Wearable devices are charged and in range
- User is logged in to the Fitness Tracker App

Test Data: Multiple wearable devices with active Bluetooth connections

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Devices menu
3. Enable Bluetooth and search for wearable devices

4. Connect and authorize each wearable device

5. Sync data from connected devices

Postconditions:

- Devices are connected
- Data is synced from each connected wearable device

Expected Result: The Fitness Tracker App connects to and syncs data from multiple wearable devices without issues.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: FTW-2

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: Ensure that users are able to manually sync data from their wearable devices and view the synced data in the Fitness Tracker App.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions:

- User is logged in to the Fitness Tracker App
- Wearable device is connected and in range

Test Data: Wearable device with activity data

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Devices menu
3. Select the connected wearable device
4. Initiate manual sync
5. View the synced activity data

Postconditions:

- Data is synced and displayed in the Fitness Tracker App

Expected Result: The Fitness Tracker App displays the synced activity data from the wearable device.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: FTW-3

Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: Ensure that the Fitness Tracker App updates user data in real-time as they perform activities while wearing their connected devices.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User is logged in to the Fitness Tracker App
- Wearable device is connected and in range
- User starts an activity (e.g., walking, running, cycling) with the wearable device

Test Data: User activity data from the wearable device

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Dashboard menu
3. Observe real-time data updates, checking steps, heart rate, and calories burned
4. Perform an activity while wearing the connected device for a period of time
5. Check for continuous real-time data updates on the Fitness Tracker App

Postconditions:

- Real-time data is displayed and updated in the Fitness Tracker App

Expected Result: The Fitness Tracker App displays real-time activity data from the connected wearable device

without delays or interruptions.

Severity: Major

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case ID: FTW-4

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: Ensure that the Fitness Tracker App displays correct wearable data for steps taken, heart rate, and calories burned in the user dashboard.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions:

- User is logged in to the Fitness Tracker App
- Wearable device is connected and in range
- User has performed an activity (e.g., walking, running, cycling) with the wearable device

Test Data: User activity data from the wearable device

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Dashboard menu
3. Verify that the user dashboard displays data for steps taken, heart rate, and calories burned
4. Observe the accuracy and consistency of the displayed wearable data

Postconditions:

- Wearable data is displayed in the user dashboard

Expected Result: The Fitness Tracker App displays accurate wearable data for steps taken, heart rate, and calories burned in the user dashboard.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: FTW-5

Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: Ensure that users are able to resolve connectivity issues with wearable devices through the Fitness Tracker App.

Test Suite: Integration with Wearable Devices

Test Priority: Low

Preconditions:

- User is logged in to the Fitness Tracker App

Test Data: No test data needed

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Devices menu
3. Attempt to connect a wearable device
4. Disconnect or disable the wearable device
5. Go to the Bluetooth settings in the Fitness Tracker App
6. Attempt to reconnect the wearable device
7. Follow in-app prompts for troubleshooting connectivity issues

Postconditions:

- Wearable device is connected

Expected Result: Users are able to resolve connectivity issues with their wearable devices through the Fitness Tracker App.

Severity: Medium

Type of Testing: Usability 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:**

Test Case ID: WI001

Test Case Title: Verify that users can log their daily water intake

Test Case Description: As a user, I want to record the amount of water I consume daily to track my hydration



levels.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged in
- App is running

Test Data: Water intake amount (e.g., 8 oz, 500 mL, 1 glass)

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Water Intake Tracking feature
3. Enter the water intake amount
4. Save the entry

Postconditions:

- Water intake is recorded in the app

Expected Result: The system saves the water intake entry and displays it in the user's hydration log.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: WI002

Test Case Title: Verify that the app calculates total water consumption for the day

Test Case Description: As a user, I want the app to automatically calculate my total water intake for a given day.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged in

- App is running
- User has logged at least one water intake entry

Test Data: N/A

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Water Intake Tracking feature
3. View the daily water intake summary
4. Observe the total water consumption for the day

Postconditions:

- Total water consumption for the day is displayed accurately

Expected Result: The system calculates and displays the user's total water consumption for the day, including all entries.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: WI003

Test Case Title: Verify that users can set daily hydration goals

Test Case Description: As a user, I want to set custom hydration goals based on my personal needs.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- App is running

Test Data: Desired daily hydration goal (e.g., 8 cups, 2 liters)

Test Steps:

1. Open the Fitness Tracker App

2. Go to the Water Intake Tracking feature
3. Access hydration goal settings
4. Input the desired daily hydration goal
5. Save the new goal

Postconditions:

- Daily hydration goal is stored

Expected Result: The system stores the user's hydration goal and displays it in the Water Intake Tracking feature.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: WI004

Test Case Title: Verify that reminders are provided to encourage regular water intake

Test Case Description: As a user, I want the app to provide reminders throughout the day to help me maintain consistent hydration.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- App is running
- User has not met or exceeded their daily hydration goal

Test Data: N/A

Test Steps:

1. Open the Fitness Tracker App
2. Ensure notifications are enabled
3. Set a daily hydration goal

4. Wait for the scheduled reminder times

5. Observe the reminder notifications

Postconditions:

- Reminder notifications are sent at the scheduled times

Expected Result: The system provides hydration reminders at scheduled intervals, guiding the user to stay on track with their hydration goal.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: WI005

Test Case Title: Verify that users can view hydration progress and trends

Test Case Description: As a user, I want to see my hydration progress over time to assess my overall hydration status.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- App is running
- User has logged water intake data for at least two days

Test Data: N/A

Test Steps:

1. Open the Fitness Tracker App
2. Navigate to the Water Intake Tracking feature
3. Access hydration progress view
4. Observe hydration trends and progress visualizations

Postconditions:

- Hydration progress and trends are displayed accurately based on logged data

Expected Result: The system displays the user's hydration progress and trends in a visual representation, enabling the user to monitor their hydration status over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: flJK3H

Test Case Title: Verify that users can log their daily meals and snacks

Test Case Description: Ensure that the Fitness Tracker App allows users to record the meals and snacks they consume daily.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in

Test Data: Meal and snack details (e.g., food name, quantity, time)

Test Steps:

1. Go to the nutrition tracking section
2. Click on the "Log meal" or "Log snack" button
3. Enter meal or snack details
4. Save the entry

Postconditions:

- Meal or snack is logged in the app

Expected Result: The system records the meal or snack details accurately and displays them in the user's nutritional history.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: weS3d

Test Case Title: Verify that the app calculates macronutrient intake (carbs, fats, proteins)

Test Case Description: Validate that the Fitness Tracker App accurately calculates the macronutrient intake of the user based on the meals and snacks logged.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has logged at least one meal or snack with nutritional information

Test Data: Meal or snack details (e.g., food name, quantity, time, and nutritional values)

Test Steps:

1. Go to the nutrition tracking section
2. Review the macronutrient intake summary
3. Check the individual macronutrient values (carbohydrates, fats, proteins)

Postconditions:

- Macronutrient intake is calculated

Expected Result: The system accurately calculates and displays the macronutrient intake based on the logged meals and snacks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: hkT4d

Test Case Title: Verify that users can set nutritional goals and track progress

Test Case Description: Confirm that users can create custom nutritional goals and monitor their progress towards those goals in the Fitness Tracker App.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has access to set nutritional goals

Test Data: Specific nutritional goals (e.g., daily caloric intake, daily protein intake, etc.)

Test Steps:

1. Go to the nutrition tracking settings
2. Set custom nutritional goals
3. Log meals and snacks for a specified duration
4. Review the progress towards the set goals

Postconditions:

- Nutritional goals are set
- Progress is tracked

Expected Result: The system calculates and displays progress towards the user's nutritional goals, allowing users to monitor their progress over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: kJW3m

Test Case Title: Verify that the app provides nutritional information and suggestions

Test Case Description: Ensure that the Fitness Tracker App offers nutritional information and personalized recommendations based on the user's nutritional history, goals, and preferences.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has logged meals and snacks



Test Data: Nutritional history data, user goals, and preferences

Test Steps:

1. Go to the nutrition tracking section
2. Check the nutritional information and recommendations provided by the app

Postconditions:

- Nutritional information is displayed
- Personalized recommendations are provided

Expected Result: The system provides accurate and relevant nutritional information based on the user's nutritional history, goals, and preferences, making appropriate suggestions for improvements.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: npL4f

Test Case Title: Verify that users can view their nutritional history and trends

Test Case Description: Validate that the Fitness Tracker App allows users to view their nutritional history and any emerging trends related to their eating habits.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has logged meals and snacks for a certain period

Test Steps:

1. Go to the nutrition tracking history
2. Review the nutritional data for a specified period (e.g., daily, weekly, monthly)
3. Identify trends in the nutritional history

Postconditions:

- Nutritional history is displayed
- Nutritional trends can be observed

Expected Result: The system clearly displays the user's nutritional history and identifies potential trends in their eating habits, allowing users to make informed decisions regarding their nutrition and diet.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: ftR1J2

Test Case Title: Verify that users receive weekly and monthly progress reports

Test Case Description: Ensure that the system generates and delivers weekly and monthly progress reports to users as per their preferences.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

- User is logged in
- User has set a fitness goal
- User has tracked activity or workouts

Test Data: User activity data for a week or month

Test Steps:

1. Log in as a user
2. Set a fitness goal
3. Track activity or workouts for at least a week or month
4. Check progress report settings (weekly, monthly, or both)
5. Wait for the report generation period to elapse
6. Monitor email inbox or in-app notifications

Postconditions:

- Progress reports are generated and delivered

Expected Result: The system generates and delivers weekly and/or monthly progress reports to the user based

on their settings, including activity levels, workout performance, and goal attainment.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: hGk2B7

Test Case Title: Verify that progress reports include activity levels, workout performance, and goal attainment

Test Case Description: Ensure that the progress reports contain detailed information about users' activity levels, workout performance, and goal attainment.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

- User is logged in
- User has set a fitness goal
- User has tracked activity or workouts
- User has received a progress report

Test Data: User activity data and fitness goals

Test Steps:

1. Log in as a user
2. Set a fitness goal
3. Track activity or workouts for a sufficient period
4. Access the latest progress report
5. Review the report content

Postconditions:

- Progress report contains activity levels, workout performance, and goal attainment data

Expected Result: The progress report includes detailed breakdowns of users' activity levels, workout performance, and goal attainment, offering a comprehensive overview of their fitness progress.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: kLq4G2

Test Case Title: Verify that users can view detailed breakdowns of achievements and trends

Test Case Description: Ensure that users can access detailed breakdowns of their achievements and trends within progress reports.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in
- User has set a fitness goal
- User has tracked activity or workouts
- User has received a progress report

Test Data: User activity data and fitness goals

Test Steps:

1. Log in as a user
2. Set a fitness goal
3. Track activity or workouts for a sufficient period
4. Access the latest progress report
5. Navigate to the achievements and trends section of the report

Postconditions:

- Detailed breakdowns are available for achievements and trends

Expected Result: The progress report features a section dedicated to achievements and trends, allowing users to analyze their progress and identify patterns, areas for improvement, and milestones.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: qNp5A4

Test Case Title: Verify that the app provides recommendations based on report findings

Test Case Description: Ensure that the system analyzes progress report data and generates personalized fitness recommendations for users.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in
- User has set a fitness goal
- User has tracked activity or workouts
- User has received a progress report

Test Data: User activity data and fitness goals

Test Steps:

1. Log in as a user
2. Set a fitness goal
3. Track activity or workouts for a sufficient period
4. Access the latest progress report
5. Review the personalized fitness recommendations

Postconditions:

- Recommendations are available based on report findings

Expected Result: The system generates personalized fitness recommendations based on users' progress report data, guiding them towards achieving their fitness goals more effectively.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: jMb8H9

Test Case Title: Verify that users can customize the frequency and content of reports

Test Case Description: Ensure that users can customize the frequency and content of their progress reports.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in

Test Steps:

1. Log in as a user
2. Go to progress report settings
3. Modify the report frequency (weekly, monthly, or both)
4. Enable or disable content categories (activity levels, workout performance, and/or goal attainment)

Postconditions:

- Settings are saved and applied to future progress reports

Expected Result: The system allows users to control the frequency and content of their progress reports, enabling them to tailor the reports to their preferences and needs.

Severity: Major

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:**

Test Case ID: ftC111

Test Case Title: Verify that users can join various in-app challenges

Test Case Description: Ensure that the app allows users to join and participate in different fitness challenges.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged in
- User has a stable internet connection



Test Data: List of available in-app challenges

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the "Challenges" section.
3. Select an available in-app challenge.
4. Choose to join the challenge.
5. Confirm the challenge joining process.

Postconditions:

- User is added to the challenge participant list.

Expected Result: The system successfully adds the user to the chosen challenge and displays the updated challenge information.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftC2r2

Test Case Title: Verify that challenges have clear objectives and rules

Test Case Description: Check that the app provides clear and concise challenge objectives, rules, and guidelines.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged in
- User has joined at least one in-app challenge

Test Data: Selected challenge with objectives and rules

Test Steps:

1. Open the Fitness Tracker App.

2. Go to the "Challenges" section.
3. Select a joined challenge.
4. Review the challenge objectives, rules, and guidelines.

Postconditions:

- User understands the challenge objectives and requirements.

Expected Result: The system displays clear, comprehensive, and easy-to-understand challenge objectives, rules, and guidelines.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftC3a3

Test Case Title: Verify that users can track their progress within the challenge

Test Case Description: Validate that the app accurately tracks user progress within a challenge.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged in
- User has joined at least one in-app challenge with a progress tracking feature

Test Data: Selected challenge with a progress tracking feature

Test Steps:

1. Open the Fitness Tracker App.
2. Go to the "Challenges" section.
3. Select a joined challenge.
4. Perform activities related to the challenge (such as steps taken, calories burned, or distance traveled).
5. Monitor and review the challenge progress.

Postconditions:

- User can view updated challenge progress.

Expected Result: The system tracks user progress accurately and updates the challenge information in real-time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftC4b4

Test Case Title: Verify that rewards are provided for achieving challenge milestones

Test Case Description: Confirm that the app grants rewards for reaching challenge milestones.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

- User is logged in
- User has joined at least one in-app challenge with milestones and rewards

Test Data: Selected challenge with milestones and rewards

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the "Challenges" section.
3. Select a joined challenge.
4. Achieve one or more challenge milestones.
5. Check the user's reward(s) for reaching the milestone(s).

Postconditions:

- User receives the reward(s) for achieving milestone(s).

Expected Result: The system grants rewards, such as badges, points, or virtual currency, for reaching challenge milestones.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftC5c5

Test Case Title: Verify that users can compete with friends and view leaderboards

Test Case Description: Check that the app supports user competition and displays leaderboards.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

- User is logged in
- User is connected with friends who are also using the Fitness Tracker App
- User has joined at least one challenge with friend participation and leaderboard features

Test Data: Selected challenge with friend participation and leaderboard features

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the "Challenges" section.
3. Select a joined challenge with friend participation and leaderboard.
4. Perform activities related to the challenge.
5. View the challenge leaderboard to see friend rankings.

Postconditions:

- User can view live leaderboard data.

Expected Result: The system displays an accurate, real-time leaderboard showing user rankings compared to friends participating in the same challenge.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: fta1C2

Test Case Title: Verify that users can select which metrics to display on their dashboard

Test Case Description: Ensure customizable dashboards allow users to choose the fitness metrics they want to view on their app dashboard.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in
- User has access to dashboard customization settings

Test Data: List of available fitness metrics (steps, calories burned, distance, heart rate, etc.)

Test Steps:

1. Go to dashboard customization settings
2. Select the "Add Metric" option
3. Choose a fitness metric from the list
4. Save changes

Postconditions:

- Selected fitness metric is displayed on the dashboard

Expected Result: The system displays the chosen fitness metric and allows users to add more.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ftb2E4

Test Case Title: Verify that app allows customization of dashboard layout and theme

Test Case Description: Ensure that users can customize their dashboard appearance, including layout and color scheme.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in
- User has access to dashboard customization settings

Test Data: Example dashboard layouts and themes

Test Steps:

1. Go to dashboard customization settings
2. Choose the "Layout" tab
3. Select a new layout from the available options
4. Choose the "Theme" tab
5. Select a new theme from the available options
6. Save changes

Postconditions:

- Selected layout and theme are applied to the dashboard

Expected Result: The system updates the dashboard with the chosen layout and theme.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: ftg3A5

Test Case Title: Verify that users can save and update their dashboard settings

Test Case Description: Ensure that users can save, update, and revert their dashboard customization settings.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in
- User has access to dashboard customization settings

Test Data: Example dashboard customization settings

Test Steps:

1. Go to dashboard customization settings
2. Apply new customization settings
3. Save changes
4. Make additional updates to the settings
5. Save changes again
6. Revert to the original settings
7. Save changes

Postconditions:

- Original customization settings are restored

Expected Result: The system allows users to save, update, and revert their dashboard settings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: fth4D6

Test Case Title: Verify that dashboard displays are dynamic and update in real-time

Test Case Description: Ensure that dashboard customizations and fitness metrics update instantaneously.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in
- User has access to real-time fitness metric updates

Test Data: Active fitness tracking

Test Steps:

1. Start tracking a workout
2. Ensure that fitness metrics are updating in real-time
3. Customize the dashboard during the workout



4. Verify that the dashboard updates dynamically

Postconditions:

- Dashboard updates dynamically with both fitness metrics and customization settings

Expected Result: The system updates the dashboard dynamically in real-time, displaying both active fitness metrics and customization settings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: fti5F7

Test Case Title: Verify that users can reset the dashboard to default settings

Test Case Description: Ensure that users can restore the dashboard layout, theme, and fitness metrics to the standard settings.

Test Suite: Customizable Dashboards

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to dashboard customization settings

Test Data: No test data needed

Test Steps:

1. Go to dashboard customization settings
2. Select the "Reset Dashboard" option
3. Confirm deletion

Postconditions:

- Dashboard is restored to default settings

Expected Result: The system restores the dashboard layout, theme, and fitness metrics to the original default settings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Negative

-----

## **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:

Test Case ID: SMFit1

Test Case Title: Verify that users can share workout summaries and achievements

Test Case Description: Ensure that users can share their workout summaries and achievements on social media platforms.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- User is logged in
- User has completed a workout and has a summary available

Test Data: Workout summary

Test Steps:

1. Go to the workout summary page
2. Click on the share button
3. Select a social media platform
4. Customize the content (if applicable)
5. Post the update

Postconditions:

- Workout summary is shared

Expected Result: The app shares the workout summary on the selected social media platform with the specified content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMFit2

Test Case Title: Verify that the app supports sharing on multiple social media platforms

Test Case Description: Ensure that users can choose from various social media platforms for sharing workout summaries and achievements.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: List of supported social media platforms

Test Steps:

1. Go to the workout summary page
2. Click on the share button
3. Verify the list of available social media platforms

Postconditions:

- Multiple social media platforms are supported

Expected Result: The app supports sharing on at least three social media platforms (e.g., Facebook, Twitter, and Instagram).

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMFit3

Test Case Title: Verify that users can customize the content and format of shared posts

Test Case Description: Ensure that users can modify the content and format of the shared workout summaries and achievements.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- User is logged in
- User has completed a workout and has a summary available

Test Data: Workout summary and customization options

Test Steps:

1. Go to the workout summary page
2. Click on the share button
3. Customize the content and format of the update (e.g., text, images, privacy settings)
4. Post the update

Postconditions:

- Customized post is shared

Expected Result: The app shares the customized workout summary on the selected social media platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMFit4

Test Case Title: Verify that sharing features are integrated seamlessly with social media

Test Case Description: Ensure that the sharing process is smooth and does not encounter any issues while interacting with social media platforms.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- User is logged in
- User has completed a workout and has a summary available
- User has a valid account on the selected social media platform

Test Data: Workout summary and social media account credentials

Test Steps:

1. Go to the workout summary page
2. Click on the share button
3. Select a social media platform
4. Log in to the social media account (if prompted)
5. Customize the content (if applicable)
6. Post the update

Postconditions:

- Workout summary is shared
- User is logged in to the social media platform

Expected Result: The app shares the workout summary without any issues, and the user remains logged in to the social media platform after sharing.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: SMFit5

Test Case Title: Verify that users can track the engagement of their shared posts

Test Case Description: Ensure that users can view the likes, comments, and other engagement metrics of their shared workout summaries and achievements.

Test Suite: Social Sharing

Test Priority: Low

Preconditions:

- User is logged in
- User has completed a workout and has shared the summary

Test Data: Workout summary and engagement metrics

Test Steps:

1. Go to the shared workout summary on the social media platform

2. Observe the engagement metrics (e.g., likes, comments, shares)

Postconditions:

- Engagement metrics are visible

Expected Result: The app displays the engagement metrics for the shared workout summary on the selected social media platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: ftR1a

Test Case Title: Verify that users can log rest days and recovery activities

Test Case Description: Ensure that the Fitness Tracker App allows users to record rest days and recovery activities, such as yoga or meditation, within the app.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has access to the tracking feature

Test Data: Rest day details and recovery activity details

Test Steps:

1. Tap on the tracking feature
2. Select "Log Rest Day" or "Log Recovery Activity"
3. Enter rest day details or recovery activity details
4. Save the log entry

Postconditions:

- Rest day or recovery activity is recorded

Expected Result: The system saves the rest day or recovery activity, displaying the new entry in the user's log history.

Severity: Major



Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftR2b

Test Case Title: Verify that the app tracks metrics such as resting heart rate and sleep quality

Test Case Description: Ensure that the Fitness Tracker App monitors and records resting heart rate and sleep quality as part of the recovery tracking feature.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has access to the tracking feature
- Heart rate monitor and sleep tracker are integrated or paired with the app

Test Data: Resting heart rate data and sleep quality data

Test Steps:

1. Open the tracking feature
2. Select the resting heart rate or sleep quality data tile
3. Verify the data updates throughout the day for resting heart rate and over the course of sleep for sleep quality

Postconditions:

- Resting heart rate and sleep quality data are accurately recorded and displayed

Expected Result: The system displays accurate resting heart rate and sleep quality data, which can be accessed by users for their records.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftR3c

Test Case Title: Verify that users receive recommendations for recovery based on their activities

Test Case Description: Ensure that the Fitness Tracker App provides users with tailored recovery recommendations based on their workout history and other activities.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has a workout log history
- App has a recommendation engine integrated

Test Data: Workout log history and user profile

Test Steps:

1. Navigate to the recovery section
2. Review the recovery recommendations provided
3. Verify that recommendations respect the user's workout history and personal preferences

Postconditions:

- Recovery recommendations are based on user workout history and personal preferences

Expected Result: The Fitness Tracker App offers recovery suggestions that are relevant to the user's workout regimen and other activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftR4d

Test Case Title: Verify that the app provides alerts for insufficient recovery time

Test Case Description: Confirm that the Fitness Tracker App alerts users if they have insufficient recovery time based on their activities.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has a workout log history
- App has an alert system integrated

Test Data: Workout log history and user profile

Test Steps:

1. Navigate to the recovery section
2. Review the recovery reminders and alerts
3. Verify that alerts communicate the need for additional rest or recovery based on the user's activities

Postconditions:

- Alerts are triggered when insufficient recovery time is detected

Expected Result: The Fitness Tracker App generates alerts for users when they have not allowed enough recovery time, promoting optimal performance and health.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftR5e

Test Case Title: Verify that users can view trends and history of their recovery data

Test Case Description: Confirm that the Fitness Tracker App enables users to check trends and historical data related to their rest and recovery.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- User is logged in

- User has a workout log history and recovery logs

- App supports trend and historical data display

Test Data: Recovery log history and user profile

Test Steps:

1. Navigate to the recovery section

2. Select the trend and historical data display option

3. Verify that the data is presented in an easy-to-understand format

Postconditions:

- Users can access and interpret their rest and recovery trends and historical data

Expected Result: The Fitness Tracker App offers users the ability to examine trends and historical data related to their rest and recovery progress, enabling them to evaluate their fitness performance and recovery habits.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

-----

**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:**

Test Case ID: FTA-AC1

Test Case Title: Verify that users can access a variety of audio-guided workouts

Test Case Description: Ensure that the Fitness Tracker App provides users with a diverse selection of audio-guided workouts catering to different fitness levels and workout types.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in
- Network connection is stable

Test Data: List of audio-guided workouts (e.g., Yoga, Running, Weightlifting, HIIT)

Test Steps:

1. Open the Fitness Tracker App
2. Go to the Audio Coaching feature
3. View the available audio-guided workouts

4. Check for diversity in fitness levels (beginner, intermediate, advanced)

Test Postconditions:

- A variety of audio-guided workouts are accessible

Expected Result: The Fitness Tracker App offers users a wide range of audio-guided workouts that cater to different fitness levels and workout types.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FTA-AC2

Test Case Title: Verify that audio coaching includes motivational messages and coaching cues

Test Case Description: Ensure that the Fitness Tracker App's audio coaching feature provides relevant motivational messages and coaching cues throughout the workout.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in
- A workout is selected

Test Data: Selected workout (e.g., Running)

Test Steps:

1. Start an audio-guided workout
2. Listen for motivational messages during the workout
3. Monitor coaching cues (e.g., pace, form, technique)

Test Postconditions:

- Motivational messages and coaching cues are present

Expected Result: The Fitness Tracker App's audio coaching provides motivational messages and coaching cues throughout the selected workout, enhancing user engagement and experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FTA-AC3

Test Case Title: Verify that users can choose coaching based on workout type and intensity

Test Case Description: Ensure that users can select the appropriate audio coaching based on their desired workout type and intensity.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in

Test Data: Workout type (e.g., Yoga, Running) and intensity (e.g., low, moderate, high)

Test Steps:

1. Go to the Audio Coaching feature
2. Choose a workout type (e.g., Yoga)
3. Select the preferred intensity (e.g., low)
4. Start the audio-guided workout

Test Postconditions:

- Workout type and intensity match the selected options

Expected Result: The Fitness Tracker App's audio coaching adapts to the user's selected workout type and intensity, delivering personalized guidance and support.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FTA-AC4

Test Case Title: Verify that audio coaching integrates seamlessly with other app features

Test Case Description: Ensure that the Fitness Tracker App's audio coaching feature integrates properly with other features such as workout logging, goal setting, and progress tracking.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in
- Workout is started

Test Data: Active workout

Test Steps:

1. Initiate an audio-guided workout
2. Perform exercises and track progress via the Fitness Tracker App
3. Complete the workout
4. View the workout log and progress tracking details

Test Postconditions:

- Workout data is accurate and integrated

Expected Result: The Fitness Tracker App's audio coaching feature seamlessly integrates with workout logging, goal setting, progress tracking, and other app features, providing a cohesive user experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FTA-AC5

Test Case Title: Verify that users can control audio settings and preferences

Test Case Description: Ensure that users can easily manage audio settings and preferences within the Fitness Tracker App's audio coaching feature.

Test Suite: Audio Coaching



Test Priority: Medium

Preconditions

- User is logged in

Test Data: Audio settings and preferences (e.g., volume, language)

Test Steps:

1. Open the Fitness Tracker App
2. Navigate to the Audio Coaching feature
3. Access audio settings and preferences
4. Modify settings (e.g., adjust volume)

Test Postconditions:

- Audio settings and preferences are updated

Expected Result: The Fitness Tracker App's audio coaching feature enables users to control audio settings and preferences, enhancing user experience and personalization.

Severity: Major

Type of Testing: Functional

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:**

Test Case ID: FT-IntHealthApp-001

Test Case Title: Verify that the fitness tracker app integrates with popular health and wellness apps

Test Case Description: Ensure that the fitness tracker app can connect and synchronize data with popular health and wellness apps such as calorie counters, meditation apps, and mental health trackers.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User has created an account on the fitness tracker app
- User has internet connectivity
- Health apps are installed and configured

Test Data: Details of health apps for integration

#### Test Steps:

1. Navigate to the integration settings
2. Select the health apps to connect
3. Follow the authentication process for each selected app
4. Authorize data access

#### Postconditions:

- Fitness tracker app is connected to the chosen health apps

Expected Result: The fitness tracker app successfully connects to the selected health apps and can import/export data.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: FT-IntHealthApp-002

Test Case Title: Verify that users can sync data from connected health apps

Test Case Description: Ensure that users can import health data from external connected apps into the fitness tracker app and export data from the fitness tracker app to external connected apps.

Test Suite: Integration with Health Apps

Test Priority: High

#### Preconditions:

- User has connected one or more health apps

Test Data: Health data from connected apps

#### Test Steps:

1. Navigate to the data sync settings
2. Initiate data sync for the connected health apps
3. Monitor the data import/export process

#### Postconditions:

- User's health data is successfully imported/exported

Expected Result: The fitness tracker app imports/exports data successfully from/to external connected health apps.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case ID: FT-IntHealthApp-003

Test Case Title: Verify that the app provides a consolidated view of health metrics

Test Case Description: Ensure the dashboard in the fitness tracker app presents a summary of health data from all connected apps in a single view.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User has connected one or more health apps
- User has health data in at least one connected app

Test Data: Health data from connected apps

Test Steps:

1. Navigate to the dashboard
2. Check for the presence of summary metrics from each connected app

Postconditions:

- Health data from various apps is displayed in the dashboard

Expected Result: The fitness tracker app dashboard shows a clear and actionable summary of health data from all connected apps.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: FT-IntHealthApp-004

Test Case Title: Verify that users receive insights based on combined data from multiple apps

Test Case Description: Ensure the system generates insights and recommendations tailored to the user's health, drawing from data points across all connected health and wellness apps.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User has connected one or more health apps
- User has health data in at least one connected app

Test Data: Health data from connected apps

Test Steps:

1. Navigate to the insights or recommendations section
2. Verify the content is personalized based on health data

Postconditions:

- Personalized insights and recommendations are generated

Expected Result: The fitness tracker app presents personalized insights and recommendations based on data from all connected apps.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: FT-IntHealthApp-005

Test Case Title: Verify that integration settings are easy to manage and configure

Test Case Description: Ensure the integration settings are user-friendly and intuitive in ordering, enabling, disabling, or removing app integrations.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- User has access to integration settings

Test Data: Details of health apps for integration

Test Steps:

1. Navigate to the integration settings
2. Add, modify, order, enable, disable, or remove app integrations
3. Verify the feedback for each action

Postconditions:

- Changes are saved and successfully applied

Expected Result: Integration settings are easily managed and configured by the user.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

-----

**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:**

Test Case ID: BCTrackW1

Test Case Title: Verify that users can log body composition metrics such as weight

Test Case Description: Ensure the system allows users to enter and save their body weight as a body composition metric.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has access to the body composition tracking feature

Test Data: User's current weight

Test Steps:

1. Navigate to the body composition tracking feature
2. Select the "Weight" option

3. Input the user's current weight

4. Save the entered data

Postconditions:

- Weight is saved in the user's body composition data

Expected Result: The system displays a confirmation that the weight has been successfully logged and saved.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BCTrackBFP2

Test Case Title: Verify that the app calculates and displays BMI based on logged data

Test Case Description: Confirm the system calculates BMI correctly based on the user's logged weight and height.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has access to the body composition tracking feature
- User has logged weight and height data

Test Data: User's height and weight

Test Steps:

1. Navigate to the body composition tracking feature
2. View the calculated BMI value
3. Compare the calculated BMI value with the manual calculation (weight (kg) / (height (m))^2)

Postconditions:

- The calculated BMI is displayed

Expected Result: The calculated BMI value matches the manual calculation.



Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BCTrends3

Test Case Title: Verify that users can view trends and history of body composition changes

Test Case Description: Confirm the system displays a historical trend for body composition metrics.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to the body composition tracking feature
- User has logged at least two sets of body composition data (e.g., weight, body fat percentage)

Test Data: User's body composition data

Test Steps:

1. Navigate to the body composition tracking feature
2. Select a body composition metric (e.g., weight)
3. View the historical trend graph
4. Verify that data points are plotted correctly based on the date

Postconditions:

- A historical trend graph is displayed

Expected Result: The historical data trend is accurately displayed, visually indicating changes over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BCInsights4

Test Case Title: Verify that the app provides insights and recommendations based on body composition data

Test Case Description: Confirm the system generates relevant insights and recommendations based on the user's body composition data.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to the body composition tracking feature
- User has logged sufficient body composition data (e.g., weight, body fat percentage, muscle mass, BMI)

Test Data: User's body composition data

Test Steps:

1. Navigate to the body composition tracking feature
2. View the generated insights and recommendations
3. Evaluate the relevance of insights and recommendations based on the body composition data

Postconditions:

- Relevant insights and recommendations are generated

Expected Result: The generated insights and recommendations reflect the user's body composition data, prompting appropriate actions related to training or nutrition plans.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: BCTGoals5

Test Case Title: Verify that users can set goals for body composition metrics

Test Case Description: Ensure the system allows users to create goals and track their progress for body composition metrics.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has access to the body composition tracking feature

Test Data: User's body composition metric (e.g., weight) and the desired goal value

Test Steps:

1. Navigate to the body composition tracking feature
2. Select a body composition metric (e.g., weight)
3. Define a goal value for selected metric
4. Set a target date for reaching the goal
5. Save the created goal

Postconditions:

- A new goal is recorded for the selected

-----

**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:**

Test Case ID: FTWkJ7

Test Case Title: Verify that users can view detailed performance metrics for each workout

Test Case Description: This test case verifies that the Fitness Tracker App displays detailed performance metrics for each workout performed by the user.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User is logged in
- User has completed at least one workout

Test Data: Previous workout data

Test Steps:

1. Launch the Fitness Tracker App
2. Log in as a user
3. Navigate to the workout history
4. Select a completed workout

5. Check the performance metrics such as duration, distance, calories, and heart rate

Postconditions:

- Workout metrics are displayed accurately

Expected Result: The system displays detailed and accurate performance metrics for each workout, allowing users to assess their performance.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: JO4Kr3

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration

Test Case Description: This test case ensures that the Fitness Tracker App offers trends and insights into users' workout intensity and duration over a period of time.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User is logged in
- User has completed multiple workouts

Test Data: Previous workout data

Test Steps:

1. Launch the Fitness Tracker App
2. Log in as a user
3. Navigate to workout insights or trends
4. Analyze the displayed data related to workout intensity and duration

Postconditions:

- Workout trends and insights are correctly displayed

Expected Result: The system analyzes and presents trends and insights into users' workout intensity and

duration, helping them identify patterns and assess progress.

Severity: Major

Type of Testing: Analytical Testing

Test Case Approach: Positive

Test Case ID: Gr8Kg1

Test Case Title: Verify that users can track calorie burn and other health metrics

Test Case Description: This test case checks that the Fitness Tracker App allows users to monitor their health metrics, including calories burned, during and after workouts.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- User is logged in
- User has completed at least one workout

Test Data: Previous workout data

Test Steps:

1. Launch the Fitness Tracker App
2. Log in as a user
3. Navigate to the workout history
4. Select a completed workout
5. Check the health metrics, such as calories burned

Postconditions:

- Health metrics are displayed accurately

Expected Result: The system displays health metrics, including calories burned, distance traveled, and heart rate, for individual workouts and overall progress.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: Nm8An4

Test Case Title: Verify that the app offers recommendations based on workout analysis

Test Case Description: This test case tests that the Fitness Tracker App provides personalized recommendations based on users' workout analysis and health metrics.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User is logged in
- User has completed multiple workouts

Test Data: Previous workout data

Test Steps:

1. Launch the Fitness Tracker App
2. Log in as a user
3. Navigate to the workout insights or recommendations
4. Analyze the recommended exercises or routines

Postconditions:

- Recommendations are based on user's performance

Expected Result: The system analyzes users' workout data and provides recommendations for optimizing training strategies and achieving fitness goals.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: Jf5Ss2

Test Case Title: Verify that users can compare current workouts with past performance

Test Case Description: This test case verifies that the Fitness Tracker App allows users to compare their current workouts with their past performance to assess their progress.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- User is logged in
- User has completed at least two workouts

Test Data: Previous workout data

Test Steps:

1. Launch the Fitness Tracker App
2. Log in as a user
3. Navigate to the workout history
4. Select a current workout
5. Compare the current workout metrics with past performance

Postconditions:

- Comparison is available and accurate

Expected Result: The system allows users to compare their workout metrics from recent sessions with their historical performance, enabling assessment of progress and goal achievement.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----

**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:**

Test Case Creation Template for Fitness Tracker App Weather Integration Feature:

Test Case ID: WI001

Test Case Title: Verify that the app provides current weather conditions and forecasts

Test Case Description: Ensure the Fitness Tracker App displays accurate current weather conditions and forecasts for the user's location.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- User has enabled location services

Test Data: No test data needed

Test Steps:

1. Open the Fitness Tracker App
2. Access the weather feature
3. Check the current weather conditions
4. Verify the weather forecast for the next 24 hours

Postconditions:

- Current weather conditions are displayed accurately
- Weather forecast is available

Expected Result: The system displays the user's current weather conditions and an accurate forecast for the next 24 hours.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: WI002

Test Case Title: Verify that users can view weather details for different locations

Test Case Description: Ensure users can switch locations to view weather data for their desired areas.

Test Suite: Weather Integration

Test Priority: Medium

Preconditions:

- User is logged in
- User has location services enabled or has manually added a location

Test Data: A different location (e.g., city, state, or postal code)

#### Test Steps:

1. Open the Fitness Tracker App
2. Access the weather feature
3. Change the location to the desired area
4. Verify the weather conditions and forecast for that location

#### Postconditions:

- Weather data for the new location is accurate

Expected Result: The system displays accurate weather conditions and forecasts for the selected location.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: WI003

Test Case Title: Verify that weather data is updated in real-time

Test Case Description: Ensure the Fitness Tracker App updates weather data regularly without user intervention.

Test Suite: Weather Integration

Test Priority: Medium

#### Preconditions:

- User is logged in
- User has location services enabled or has manually added a location

Test Data: Weather conditions for the user's location

#### Test Steps:

1. Open the Fitness Tracker App
2. Access the weather feature and note the current weather data
3. Wait for a specified period (e.g., 10 minutes)
4. Refresh the page and check the current weather data

Postconditions:

- Weather data has been updated

Expected Result: The system updates weather data in real-time for the user's location.

Severity: Major

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case ID: WI004

Test Case Title: Verify that the app offers recommendations based on weather conditions

Test Case Description: Validate that the Fitness Tracker App provides exercise suggestions based on current and forecasted weather conditions.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- User has location services enabled or has manually added a location

Test Data: Weather conditions for the user's location

Test Steps:

1. Open the Fitness Tracker App
2. Access the weather feature
3. Verify the exercise suggestions based on the displayed current and forecasted weather conditions

Postconditions:

- Recommendations are relevant to the weather conditions

Expected Result: The system suggests appropriate exercises based on the user's current and forecasted weather conditions.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: WI005

Test Case Title: Verify that users can receive weather alerts for extreme conditions

Test Case Description: Ensure the Fitness Tracker App notifies users when extreme weather conditions occur.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- User has enabled location services and opted-in for weather alerts

Test Data: Extreme weather conditions (e.g., thunderstorm, high wind, etc.)

Test Steps:

1. Open the Fitness Tracker App
2. Access the weather feature and check alert settings
3. Trigger a test using a weather condition generator or a server-side switch
4. Verify the user receives a notification for the extreme condition

-----

**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:**

Test Case ID: FTAMT1

Test Case Title: Verify that users can log their mood, energy levels, and motivation

Test Case Description: Ensure that the Fitness Tracker App allows users to log their mood, energy levels, and motivation before and after workouts.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has accessed the workout logging feature

Test Data: Mood (happy, sad, neutral), energy levels (high, medium, low), motivation (high, medium, low)

Test Steps:

1. Access the workout logging page
2. Select a workout to log
3. Enter the mood, energy levels, and motivation values for before the workout
4. Complete the workout and save it
5. Enter the mood, energy levels, and motivation values for after the workout
6. Save the workout log

Postconditions:

- Workout log is saved with mood and energy data

Expected Result: The system saves the workout log, including the entered mood, energy levels, and motivation values.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FTAMT2

Test Case Title: Verify that the app provides insights into mood and energy trends over time

Test Case Description: Ensure that the Fitness Tracker App provides insights and visualizations for users to understand their mood and energy trends over time.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has logged multiple workouts with mood and energy data

Test Data: Multiple workout logs with mood, energy levels, and motivation data

Test Steps:

1. Access the progress tracking feature
2. Navigate to the mood and energy data visualization

### 3. Observe trends in the mood and energy data

#### Postconditions:

- Mood and energy data visualizations are displayed

Expected Result: The system generates visualizations displaying the user's mood and energy trends over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FTAMT3

Test Case Title: Verify that users receive recommendations based on their mood and energy data

Test Case Description: Ensure that the Fitness Tracker App provides personalized recommendations based on the user's mood and energy data.

Test Suite: Mood Tracking

Test Priority: High

#### Preconditions:

- User is logged in
- User has logged multiple workouts with mood and energy data

Test Data: Multiple workout logs with mood, energy levels, and motivation data

#### Test Steps:

1. Access the personalized recommendation feature
2. Review the recommendations provided by the app

#### Postconditions:

- Personalized recommendations are displayed

Expected Result: The system generates personalized recommendations based on the user's mood and energy data from their workout logs.

Severity: Major

Type of Testing: Functional



Test Case Approach: Positive

Test Case ID: FTAMT4

Test Case Title: Verify that the app supports journaling for personal reflections on mood and workouts

Test Case Description: Ensure that the Fitness Tracker App allows users to add personal reflections and notes on their mood and workout experiences.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has accessed the workout logging feature

Test Data: Text input for journaling

Test Steps:

1. Access the workout logging page
2. Select a workout to log
3. Add a personal reflection or note about the workout and mood
4. Save the workout log

Postconditions:

- Workout log is saved with user's reflection

Expected Result: The system saves the workout log, including the user's personal reflection or note about the workout and mood.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: FTAMT5

Test Case Title: Verify that users can set goals for improving mood and energy levels

Test Case Description: Ensure that the Fitness Tracker App allows users to set goals for improving their mood and energy levels over time.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User is logged in

Test Steps:

1. Access the goal setting feature
2. Choose to set a mood and energy goal
3. Enter the desired mood, energy levels, and motivation improvements
4. Set a deadline for achieving the goal
5. Save the goal

Postconditions:

- Goal is saved in the user's profile

Expected Result: The system saves the user's mood and energy goals and associates them with the user's profile.

Severity: Minor

Type of Testing: Functional

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:**

Test Case ID: RC1

Test Case Title: Verify that users can input and save ingredients and cooking instructions.

Test Case Description: This test case verifies if the system allows users to input ingredients and cooking instructions and saves them correctly.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in
- User is on the recipe creation page

Test Data: Recipe details including ingredients and cooking instructions

Test Steps:

1. Go to the recipe creation page
2. Input recipe name
3. Enter ingredients in the provided field
4. Input cooking instructions in the provided field
5. Save the recipe

Test Postconditions:

- Recipe is saved in the user's profile

Expected Result: Recipe is saved with correct ingredients and cooking instructions.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: RC2

Test Case Title: Verify that users can upload photos for their recipes.

Test Case Description: This test case ascertains if the system permits users to upload photos for their recipes.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in
- User is on the recipe creation page

Test Data: Recipe photo

Test Steps:

1. Go to the recipe creation page
2. Input recipe name
3. Enter ingredients in the provided field

4. Input cooking instructions in the provided field
5. Choose a photo to upload
6. Save the recipe

Test Postconditions:

- Recipe is saved in the user's profile with uploaded photo

Expected Result: Recipe is saved with the uploaded photo.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: RC3

Test Case Title: Verify that recipes are saved in the user's profile.

Test Case Description: This test case checks if the system properly saves recipes in the user's profile.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in
- User has created a recipe

Test Data: Recipe details

Test Steps:

1. Go to the user's profile page
2. View saved recipes

Test Postconditions:

- Recipe is visible in the user's profile

Expected Result: Recipe is saved and visible in the user's profile.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: RC4

Test Case Title: Verify that users receive a confirmation upon successful recipe submission.

Test Case Description: This test case confirms if users receive a confirmation message upon successful recipe submission.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the recipe creation page

Test Data: Recipe details

Test Steps:

1. Go to the recipe creation page
2. Input recipe name
3. Enter ingredients in the provided field
4. Input cooking instructions in the provided field
5. Choose a photo to upload (optional)
6. Save the recipe

Test Postconditions:

- Recipe is saved in the user's profile
- Confirmation message is displayed

Expected Result: User receives a confirmation message upon successful recipe submission.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case ID: RC5

Test Case Title: Verify that recipes are accessible to other users on the platform.

Test Case Description: This test case ensures that recipes saved by users are accessible to other users on the platform.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User A is logged in
- User A has created a recipe and saved it
- User B is logged in

Test Data: Recipe details

Test Steps:

1. Go to User A's profile as User B
2. View User A's recipes

Test Postconditions:

- User B can view User A's recipes

Expected Result: Recipes created by users are accessible to other users on the platform.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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:**

Test Case ID: rspKw1

Test Case Title: Verify that users can search for recipes using keywords and ingredients

Test Case Description: This test case ensures that users are able to search for recipes by entering specific keywords and ingredients into the search bar.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User is logged in



- User is on the recipe search page

Test Data: Keywords: "Chocolate Chip", Ingredients: "Flour, Sugar, Chocolate Chips"

Test Steps:

1. Enter the specified keywords in the search bar
2. Enter the specified ingredients in the search bar
3. Click the "Search" button
4. View the displayed search results

Postconditions:

- Relevant recipes are displayed

Expected Result: The system accurately matches the search criteria and displays recipes containing the specified keywords and ingredients.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: rspKw2

Test Case Title: Verify that search results are relevant and accurately match the search criteria

Test Case Description: This test case verifies that the search results displayed are relevant to the entered search criteria and accurately match the user's query.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User is logged in
- User is on the recipe search page

Test Data: Keywords: "Vegan Italian Pasta"

Test Steps:

1. Enter the specified keywords in the search bar

2. Click the "Search" button

3. Analyze the relevance of the displayed search results

Postconditions:

- Relevant recipes are displayed

Expected Result: The system displays recipes that are vegan, Italian, and pasta-based, accurately matching the search criteria.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: rspKw3

Test Case Title: Verify that users can filter search results by cuisine type, dietary preferences, and other criteria

Test Case Description: This test case checks that users are able to filter search results based on cuisine type, dietary preferences, and other available filters.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the recipe search page

Test Data: Filters: Cuisine type - Italian, Dietary Preferences - Vegan

Test Steps:

1. Perform a recipe search (e.g., enter a keyword or use the default search)
2. Click on the "Filters" button
3. Select the desired filters (e.g., cuisine type, dietary preferences)
4. Apply the filters to the search results

Postconditions:

- Filtered search results are displayed

Expected Result: The system accurately filters the search results based on the selected cuisine type, dietary preferences, and other criteria, displaying relevant recipes.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: rspKw4

Test Case Title: Verify that recipes in the search results are accessible and detailed

Test Case Description: This test case confirms that the recipes displayed in the search results are accessible and provide detailed information about the recipe.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

- User is logged in
- User has performed a recipe search

Test Steps:

1. View the search results
2. Click on a recipe from the search results
3. Analyze the recipe details (e.g., ingredients, cooking instructions, photos, videos, user ratings, and reviews)

Postconditions:

- Recipe details are displayed

Expected Result: The system provides detailed information about each recipe in the search results, including ingredients, cooking instructions, photos, videos, user ratings, and reviews, allowing users to access and evaluate the recipe.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case ID: rspKw5

Test Case Title: Verify that users can save search results for future reference

Test Case Description: This test case checks that users are able to save search results for future reference and access the saved results from their personal profile page.

Test Suite: Recipe Discovery

Test Priority: Low

Preconditions:

- User is logged in
- User has performed a recipe search with relevant results

Test Steps:

1. Perform a recipe search
2. Click on the "Save Search" button
3. Access the user's personal profile page
4. Navigate to the "Saved Searches" section

Postconditions:

- Search results are saved and accessible from the user's profile page

Expected Result: The system allows users to save search results and access the saved results from their personal profile page, enabling users to revisit their favorite searches for future recipe exploration.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

-----