Generated Test Cases via gemini-1.5-pro 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 Cases for Social Media Platform - User Profiles

Test Case 1:

Test Case ID: SMP-UP-001

Test Case Title: Verify that users can add and edit profile information.

Test Case Description: This test case verifies that users can successfully add and edit their name, bio, profile picture, location, and interests on their profile.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in.
- User is on the profile creation/editing page.

Test Data:

- Valid name, bio, profile picture (within platform size and format limits), location, and interests.

Test Steps:

1. Input valid data into all profile fields (name, bio, picture, location, interests).
2. Save the profile.
3. Verify that the information is displayed correctly on the profile page.
4. Edit the information in all fields.
5. Save the changes.
6. Verify that the updated information is displayed correctly on the profile page.

Postconditions:

- Profile information is updated successfully.

Expected Result: The user should be able to add and edit their profile information, and the changes should be reflected accurately on their profile page.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMP-UP-002

Test Case Title: Verify that the profile picture supports standard image formats.

Test Case Description: This test case verifies that the platform supports uploading profile pictures in standard image formats (e.g., JPG, PNG, GIF) and rejects unsupported formats.

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

  - User is logged in.
  - User is on the profile picture upload section.

Test Data:

  - Images in supported formats (e.g., JPG, PNG, GIF).
  - Images in unsupported formats (e.g., PDF, DOCX).

Test Steps:

  1. Attempt to upload images in supported formats.
  2. Verify that the images are uploaded successfully and displayed correctly.
  3. Attempt to upload images in unsupported formats.

Postconditions: None

Expected Result: Images in supported formats should be uploaded and displayed correctly. The system should display an error message when attempting to upload unsupported formats.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive, Negative

Test Case 3:

Test Case ID: SMP-UP-003

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

Test Case Description: This test case ensures users can control the visibility of individual profile fields (name, bio, location, interests) using privacy settings.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in.
- User is on the profile privacy settings page.

Test Data: No test data needed

Test Steps:

1. Navigate to each profile field's privacy settings.
2. Select different privacy options (e.g., Public, Friends only, Private).
3. Save the privacy settings.
4. Log in as another user or switch to an incognito window to simulate different user views.
5. Verify that the visibility of each field reflects the selected privacy settings.

Postconditions: None

Expected Result: Each profile field's visibility should accurately reflect the chosen privacy setting.

Severity: Critical

Type of Testing: Security Testing, Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMP-UP-004

Test Case Title: Verify that the profile is saved successfully.

Test Case Description: This test case checks whether the user profile is saved successfully after adding or editing information and whether data persists across sessions.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in.
- User is on the profile creation/editing page.

Test Data:

- Valid name, bio, profile picture, location, and interests.

Test Steps:

1. Input valid data into profile fields.
2. Save the profile.
3. Log out of the platform.
4. Log back in and navigate to the user's profile page.

Postconditions: None

Expected Result: The profile information should be saved, and the entered data should be displayed correctly after the user logs back in.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMP-UP-005

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

Test Case Description: This test case ensures that a confirmation message is displayed to the user after they successfully save their profile.

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

- User is logged in.
- User is on the profile creation/editing page.

Test Data:

- Valid name, bio, profile picture, location, and interests.

Test Steps:

1. Input valid data into profile fields.
2. Save the profile.

Postconditions: None

Expected Result: A confirmation message (e.g., "Profile saved successfully!") should be displayed to the user after saving the profile.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 2

Project name -

Social Media Platform

Project description -

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

Feature name -

News Feed

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - News Feed

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Type of Testing	Test Case Approach
NF-001	Verify that the news feed displays posts from followed accounts	This test case verifies that the news feed only displays posts from accounts that the user has chosen to follow.	News Feed	High	- User is logged in. - User follows at least one other account.	No test data needed.	1. Log in as a user. 2. Navigate to the news feed. 3. Observe the posts displayed.	None	Only posts from followed accounts are displayed.

Major | Functional Testing | Positive |

| NF-002 | Verify that content is displayed in chronological or algorithmic order | This test case checks if the news feed displays content based on the chosen sorting method (chronological or algorithmic). | News Feed | High | - User is logged in. - User's news feed has at least five posts. | No test data needed. | 1. Log in as a user. 2. Navigate to the news feed. 3. Observe the order of posts displayed. 4. If applicable, change the sorting method in settings. 5. Observe the updated order of posts. | None | Posts are displayed according to the selected sorting method (chronological or algorithmic). | Major | Functional Testing | Positive |

| NF-003 | Verify that users can interact with posts (like, comment, share) | This test case ensures that users can interact with posts in their news feed by liking, commenting, and sharing. | News Feed | High | - User is logged in. - User's news feed has at least one post. | Sample comment: "Great post!" | 1. Log in as a user. 2. Navigate to the news feed. 3. Like a post. 4. Comment on a post. 5. Share a post. | None | User can successfully like, comment on, and share posts. The interactions are reflected accurately. | Major | Functional Testing | Positive |

| NF-004 | Verify that the feed updates in real-time | This test case validates that new posts from followed accounts appear in the news feed without manual refresh. | News Feed | High | - User is logged in. - User has a friend online who can make a new post. | No test data needed. | 1. Log in as a user. 2. Navigate to the news feed. 3. Have a friend publish a new post. 4. Observe the news feed for the new post. | None | The new post from the friend appears at the top of the news feed in real-time without manual refresh. | Major | Functional Testing | Positive |

| NF-005 | Verify that users can filter content based on type or topic | This test case ensures users can filter their news feed to display specific types of content or posts related to specific topics. | News Feed | Medium | - User is logged in. - User's news feed contains various post types and topics. | No test data needed. | 1. Log in as a user. 2. Navigate to the news feed. 3. Access the filter options. 4. Apply a filter based on post type (e.g., photos, videos). 5. Apply a filter based on topic (if available). | None | The news feed displays only the content matching the selected filters. | Minor | Functional Testing | 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 Cases for Social Media Platform - Messaging Feature:

Test Case 1:

Test Case ID: SMP-MSG-001

Test Case Title: Verify that users can send text messages.

Test Case Description: This test case verifies that a user can successfully send a text message to another user in a private conversation.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged in.
- User has at least one friend added.

Test Data: No test data needed

Test Steps:

1. Navigate to the messaging section.
2. Select an existing friend to start a private conversation.
3. Type a text message in the message input field.
4. Click the send button.

Postconditions:

- The sent message should appear in the conversation window.

Expected Result: The text message is sent successfully and displayed in the conversation with the recipient.

Severity: Major

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMP-MSG-002

Test Case Title: Verify that users can send images and videos in messages.

Test Case Description: This test case validates the functionality of sending multimedia content (images & videos) within private messages.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged in.
- User has an active conversation with another user.

Test Data: Sample image file, Sample video file.

Test Steps:

1. Open an existing private conversation.
2. Click the image attachment icon to upload an image.
3. Select and upload a sample image file.
4. Click the video attachment icon to upload a video.
5. Select and upload a sample video file.

Postconditions:

- The uploaded image and video should appear in the conversation window.

Expected Result: The system successfully sends and displays the image and video within the conversation.

Severity: Major

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMP-MSG-003

Test Case Title: Verify that messages are sent in real-time.

Test Case Description: This test case checks if the messages sent appear instantly in the recipient's chat window without significant delay.

Test Suite: Messaging

Test Priority: High

Preconditions:

- Two different user accounts are logged in simultaneously.
- Both users have an active private conversation with each other.

Test Data: No test data needed

Test Steps:

1. Using the first user account, send a text message.
2. Observe the conversation window of the second user account.

Postconditions: None

Expected Result: The message should appear immediately on the second user's chat window without any noticeable lag.

Severity: Major

Type of Testing: Performance, System Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMP-MSG-004

Test Case Title: Verify that messages show read receipts.

Test Case Description: This test ensures that the sender receives a visual indication when their message has been read by the recipient.

Test Suite: Messaging

Test Priority: Medium

Preconditions:

- Two different user accounts are logged in simultaneously.

- Both users have an active private conversation with each other.

Test Data: No test data needed

Test Steps:

1. Using the first user account, send a text message.

2. The second user opens the conversation and reads the message.

3. Observe the message status on the first user's account.

Postconditions: None

Expected Result: A visual indicator (e.g., "Seen" or double tick) appears next to the sent message in the first user's chat window, confirming the message has been read.

Severity: Minor

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMP-MSG-005

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

Test Case Description: This test case validates the notification system for new messages received, both in-app and through system notifications (if applicable).

Test Suite: Messaging, Notifications

Test Priority: High

Preconditions:

- Two different user accounts are logged in simultaneously.

- User has enabled notifications for new messages.

Test Data: No test data needed

Test Steps:

1. Send a message from the second user account to the first user account.

2. Observe both the in-app notification area and the device's notification center (if applicable) for the first user account.

Postconditions: None

Expected Result: The first user receives a notification (visual or audible, depending on settings) about the new message, both within the application and potentially as a system notification.

Severity: Major

Type of Testing: Functional, System Testing

Test Case Approach: Positive

--

Data - 4

Project name -

Social Media Platform

Project description -

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

Feature name -

Content Sharing

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Content Sharing

Test Case 1:

Test Case ID: SMPC-001

Test Case Title: Verify that users can create and share text posts, photos, and videos.

Test Case Description: This test case verifies that users can successfully create and share different types of content, including text posts, photos, and videos, to their followers.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in.

- User has followers.

Test Data:

- Text post content

- Sample photo (jpg, png)

- Sample video (mp4)

Test Steps:

1. Navigate to the content creation area.

2. Create a text post, add content, and share it with followers.

3. Create a post with the sample photo and share it with followers.

4. Create a post with the sample video and share it with followers.

Postconditions:

- All posts are visible on the user's profile.

- Followers can see the shared posts on their feed.

Expected Result: The system should allow users to create and share text posts, photos, and videos without errors. Shared content should be visible to the intended audience.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMPC-002

Test Case Title: Verify that content can be shared with followers or publicly.

Test Case Description: This test case verifies the functionality for users to control the visibility of their shared content by choosing to share with only their followers or make it public.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in.

Test Data:

- Text post content

Test Steps:

1. Navigate to the content creation area.

2. Create a text post.

3. Choose to share the post with "Followers Only."

4. Create another text post.

5. Choose to share the post "Publicly."

Postconditions:

- Posts shared with "Followers Only" should be visible only to followers.

- Posts shared "Publicly" should be visible to all users on the platform.

Expected Result: The system should correctly restrict content visibility based on the user's chosen sharing preference.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMPC-003

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

Test Case Description: This test case ensures users have the ability to edit the content of their posts or delete them entirely after they have been shared.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in.

- User has existing posts.

Test Data:

- Edited text content

Test Steps:

1. Navigate to an existing text post created by the user.

2. Click the "Edit" option for the post.

3. Modify the content of the post.

4. Save the changes.

5. Click the "Delete" option for the post.

6. Confirm the deletion.

Postconditions:

- Edited posts should reflect the changes made.

- Deleted posts should be removed from the user's profile and followers' feeds.

Expected Result: The system should allow users to edit or delete their posts successfully.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMPC-004

Test Case Title: Verify that posts support hashtags and tagging.

Test Case Description: This test case ensures users can add hashtags and tag other users in their posts, enhancing content discoverability and interaction.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data:

- Text post content

- Relevant hashtags (e.g., #travel, #photography)

- Existing usernames for tagging

Test Steps:

1. Navigate to the content creation area.

2. Create a text post and include relevant hashtags within the text.

3. Tag existing users in the post.

4. Share the post.

Postconditions:

- Hashtags and tagged usernames should be clickable and lead to respective feeds/profiles.

Expected Result: The system should correctly process and display hashtags and tagged usernames in posts.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMPC-005

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

Test Case Description: This test case ensures users receive feedback on their shared content through likes, comments, and shares from other users.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- User is logged in.

- User has shared posts.

Test Data: No test data needed

Test Steps:

1. As another user, navigate to the posts shared by the first user.

2. Like, comment on, and share different posts.

Postconditions:

- The original user should see notifications for likes, comments, and shares on their posts.

Expected Result: The system should accurately record and display likes, comments, and shares on posts, providing feedback to the content creator.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

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 Cases for Social Media Platform - Privacy Settings:

Test Case 1:

Test Case ID: Privacy_001

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

Test Case Description: This test case verifies that users can successfully set their profile privacy to "Public" and that their profile information is visible to all users on the platform.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

* User is logged in.

* User has not set any privacy restrictions.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Settings" page.

2. Select the "Privacy" section.

3. Choose the "Public" option for profile visibility.

4. Save the changes.

5. Log out of the current user account.

6. Access the user's profile from a different account or incognito mode.

Postconditions:

* None.

Expected Result:

* The user's profile and posts are visible to all users, including those not connected to the user.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: Privacy_002

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

Test Case Description: This test case ensures users can successfully set their profiles to "Private" and that only confirmed connections can view their profile information.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

* User is logged in.

* User has at least one connection.

Test Data: No test data needed.

Test Steps:

1. Navigate to "Settings."

2. Go to the "Privacy" section.

3. Choose "Private" for profile visibility.

4. Save the changes.

5. Log out of the current user account.

6. Attempt to access the user's profile from a non-connected account.

Postconditions:

* None.

Expected Result:

* The user's profile is only visible to confirmed connections. Non-connected users are shown a message indicating the profile is private.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: Privacy_003

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

Test Case Description: This test case checks if users can utilize the "Restricted" setting to customize which connections can view specific posts or profile information.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

* User is logged in.

* User has created a list of restricted connections.

Test Data: A pre-defined list of restricted connections.

Test Steps:

1. Go to the "Settings" page.

2. Select the "Privacy" section.

3. Choose the "Restricted" option for profile visibility.

4. Add or manage the list of restricted connections.

5. Create a new post with the "Restricted" audience setting.

6. Log in as a restricted user.

7. Attempt to view the restricted post.

Postconditions:

* None.

Expected Result:

* Restricted users cannot see the post, while other connections can.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: Privacy_004

Test Case Title: Verify that privacy settings changes take effect immediately.

Test Case Description: This test case aims to confirm that any changes made to privacy settings are applied instantly across the platform.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

* User is logged in.

* The user has made a recent change to their privacy settings.

Test Data: No test data needed.

Test Steps:

1. Change a privacy setting (e.g., profile visibility, post visibility).

2. Immediately attempt to access a restricted area of the platform (e.g., the user's profile, a specific post) from another account.

Postconditions:

* None.

Expected Result: The new privacy setting takes effect immediately, preventing unauthorized access.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: Privacy_005

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

Test Case Description: This test case validates that the system successfully sends notifications to users whenever changes are made to their privacy settings.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

* User is logged in.

* User has enabled notifications for privacy changes (if applicable).

Test Data: No test data needed.

Test Steps:

1. Make a change to any privacy setting.

2. Check for a notification within the platform.

3. Check the user's registered email for a notification (if enabled).

Postconditions:

* None.

Expected Result: The user receives a notification within the platform and/or by email informing them of the changes made to their privacy settings.

Severity: Minor

Type of Testing: Functional Testing

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 Cases for Social Media Platform - Notifications Feature

Test Case 1:

Test Case ID: SMN-001

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

Test Case Description: This test case verifies that users are notified in real-time when another user likes, comments on, or shares their posts. It ensures that the notification system accurately captures these interactions and promptly informs the user.

Test Suite: Notifications

Test Priority: High

Preconditions:

  - User A and User B are logged in.
  - User A has posted content.

Test Data: No test data needed.

Test Steps:

  1. User B likes User A's post.
  2. User B comments on User A's post.
  3. User B shares User A's post.

Postconditions: None.

Expected Result: User A receives real-time notifications for each interaction (like, comment, share) on their post.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMN-002

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

Test Case Description: This test case ensures that users receive timely notifications when they receive new friend requests and when they are mentioned by other users in posts or comments.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User A and User B are logged in.

Test Data: No test data needed.

Test Steps:

1. User B sends a friend request to User A.
2. User B creates a post and mentions User A in the post.
3. User B comments on a post and mentions User A in the comment.

Postconditions: None.

Expected Result: User A receives real-time notifications for the friend request and for each mention.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMN-003

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

Test Case Description: This test case checks the real-time functionality of the notification system, ensuring that users are immediately alerted about new activity without any significant delays.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User A and User B are logged in.
- User A is on the platform's main feed or their profile page.

Test Data: No test data needed.

Test Steps:

1. User B performs an action that should trigger a notification for User A (e.g., like, comment, mention).
2. Observe the time taken for the notification to appear on User A's screen.

Postconditions: None.

Expected Result: The notification should appear instantaneously or with minimal delay, confirming the real-time delivery of notifications.

Severity: Major

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMN-004

Test Case Title: Verify that users can customize notification preferences.

Test Case Description: This test case ensures users have control over the types of notifications they receive. It verifies that the notification settings allow users to enable/disable specific notification types and save their preferences.

Test Suite: Notifications

Test Priority: Medium

Preconditions:

- User A is logged in.

Test Data: No test data needed.

Test Steps:

1. Navigate to notification settings.
2. Toggle different notification types (likes, comments, shares, mentions, friend requests) on or off.
3. Save the notification preferences.
4. Have User B perform actions that trigger the enabled and disabled notification types.

Postconditions: None.

Expected Result: User A should only receive notifications for the types they have enabled in their settings and should not receive notifications for disabled types.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMN-005

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

Test Case Description: This test case ensures that each notification directs the user to the correct location associated with the activity (e.g., the specific post, the new friend request, the comment mentioning them).

Test Suite: Notifications

Test Priority: High

Preconditions:

- User A is logged in.
- User A has received various types of notifications.

Test Data: No test data needed.

Test Steps:

1. Select a notification related to a post (like, comment, share).
2. Click on the notification.

Postconditions: None.

Expected Result: The notification link should redirect the user to the specific post where the activity occurred. Repeat for other notification types (friend requests, mentions) to confirm accurate redirection.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

--

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 Cases for Social Media Platform Search Functionality:

Test Case 1:

Test Case ID: SMPSF-001

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

Test Case Description: This test case aims to confirm that the search functionality effectively retrieves relevant content when users input keywords. It verifies that the search results accurately reflect the entered keywords and display a variety of relevant content types.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

   - User is logged in.
   - The platform has existing content (posts, profiles, etc.).

Test Data: A list of relevant keywords related to existing content on the platform.

Test Steps:

   1. Navigate to the search bar.
   2. Enter a keyword from the test data.
   3. Observe the displayed search results.

Postconditions: None

Expected Result: The search results should display a list of profiles, posts, and topics containing the entered keyword, indicating a successful keyword-based search.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMPSF-002

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

Test Case Description: This test case ensures that the search functionality covers a comprehensive range of content types, including profiles, posts, and topics. It verifies that users can discover various content formats through their searches.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in.
- The platform has existing profiles, posts, and topics.

Test Data: A search term that guarantees results across profiles, posts, and topics.

Test Steps:

1. Navigate to the search bar.
2. Enter the predetermined search term.
3. Analyze the displayed search results.

Postconditions: None

Expected Result: The search results page should display a mix of relevant profiles, posts, and topics, confirming the search functionality's coverage of different content types.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMPSF-003

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

Test Case Description: This test case assesses the accuracy and efficiency of the search algorithm by verifying that the most relevant results appear at the top of the search results page. It ensures users can quickly find the most pertinent content.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

  - User is logged in.
  - The platform has a substantial amount of varied content.

Test Data: A search term with varying degrees of relevance to existing content.

Test Steps:

  1. Go to the search bar.
  2. Input the chosen search term.
  3. Review the order of displayed search results.

Postconditions: None

Expected Result: The search results should prioritize content with the strongest relevance to the search term, confirming the effectiveness of the ranking algorithm.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMPSF-004

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

Test Case Description: This test case validates the functionality of search filters, allowing users to refine their searches by specific content types. It confirms that filters accurately narrow down results based on user preferences.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in.
- Search results are displayed for a broad search term.

Test Data: No test data needed.

Test Steps:

1. Perform a search using a general term.
2. Locate and apply a content filter (e.g., "Posts," "People," "Topics").
3. Observe the change in displayed search results.

Postconditions: None

Expected Result: The search results should be dynamically updated to display only the content type selected in the filter, demonstrating accurate filter functionality.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMPSF-005

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

Test Case Description: This test case confirms the presence and accuracy of search suggestions that appear dynamically as the user inputs characters into the search bar. It ensures a helpful and efficient search experience by guiding users with relevant suggestions.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: No test data needed.

Test Steps:

1. Navigate to the search bar.
2. Begin typing a word or phrase.
3. Observe the appearance and relevance of suggestions.

Postconditions: None

Expected Result: As the user types, the search bar should display a list of relevant suggestions, including keywords, hashtags, accounts, or topics related to the inputted characters, enhancing the search experience.

Severity: Minor

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 Cases for Social Media Platform - Hashtags and Tagging:

Test Case 1:

Test Case ID: SMPT-HT-001

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

Test Case Description: This test case ensures that users can successfully include hashtags within the body of their posts. It will cover various hashtag formats and placements within the post content.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

* User is logged in.

* User is creating a new post.

Test Data:

* Post content with hashtags: "Excited to share my latest blog post about #technology and its impact on #society. Check it out! #newblogpost"

Test Steps:

1. Navigate to the post creation area.

2. Enter the test data post content containing hashtags.

3. Publish the post.

4. Verify that the hashtags are displayed correctly in the published post.

Postconditions:

* The post is successfully published.

Expected Result: Hashtags are displayed correctly in the published post and are clickable, leading to a search results page for that specific hashtag.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMPT-HT-002

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

Test Case Description: This test verifies that users can tag other user profiles, brand pages, or locations in their posts. It will also check the tagging mechanism and notification system.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

* User is logged in.

* User is creating a new post.

Test Data:

* Post content: "Attending the [Brand Name] conference today! #[Hashtag] @[Username]"

Test Steps:

1. Start creating a new post.

2. Type "@" followed by the username, brand name, or location.

3. Select the correct profile/page from the suggestions.

4. Publish the post.

Postconditions:

* The post is published with the tag displayed correctly.

Expected Result: The system successfully tags the intended profile, brand, or location and the tagged entity receives a notification.

Severity: Major

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMPT-HT-003

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

Test Case Description: This test case ensures that both hashtags and tags are clickable links that redirect users to the relevant content pages.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

* User is logged in.

* A post with hashtags and tags is available.

Test Data: No test data needed

Test Steps:

1. Navigate to a post containing hashtags and tags.

2. Click on a hashtag.

3. Verify redirection to a page displaying other posts with the same hashtag.

4. Return to the original post.

5. Click on a tag.

6. Verify redirection to the tagged user/brand/location's profile/page.

Postconditions: None

Expected Result: Clicking on a hashtag or tag correctly redirects the user to the relevant content page.

Severity: Major

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMPT-HT-004

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

Test Case Description: This test verifies the functionality of searching for content based on specific hashtags.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

* User is logged in.

Test Data:

* A specific hashtag, e.g., #Travel.

Test Steps:

1. Go to the search bar.

2. Enter the chosen hashtag in the search bar.

3. Initiate the search.

4. Verify that the search results display posts containing the specified hashtag.

Postconditions: None

Expected Result: The system accurately displays a list of posts containing the searched hashtag.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMPT-HT-005

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

Test Case Description: This test ensures that the system provides relevant hashtag suggestions as users begin typing in the post creation area, improving content discoverability.

Test Suite: Hashtags and Tagging

Test Priority: Medium

Preconditions:

* User is logged in.

* User is creating a new post.

Test Data: No test data needed

Test Steps:

1. Start typing a post and include the "#" symbol.

2. Observe if the system provides hashtag suggestions based on the entered characters.

3. Partially type a popular hashtag.

4. Observe if the system accurately suggests the complete and relevant hashtags.

Postconditions: None

Expected Result: The system dynamically suggests popular and relevant hashtags as the user types, aiding in content categorization and discoverability.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

--

Data - 9

Project name -

Social Media Platform

Project description -

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

Feature name -

Emojis and Reactions

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Emojis and Reactions Feature:

Test Case 1:

Test Case ID: ER-001

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

Test Case Description: This test case ensures users can react to different posts using various emojis from the platform's selection. It verifies the functionality of applying emojis as reactions and their visibility on the post.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in.
- A post is available for interaction.

Test Data: No test data needed

Test Steps:

1. Navigate to a post.
2. Hover over the reaction button (e.g., thumbs up icon).
3. Select an emoji from the displayed options.
4. Verify the selected emoji appears below the post as a reaction.
5. Repeat steps 3-4 with different emojis.

Postconditions:

- Reactions are displayed correctly.

Expected Result: The system should allow the user to react to a post with various emojis, displaying the chosen reactions under the post.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: ER-002

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

Test Case Description: This test case verifies the functionality of adding emojis within comments on posts and within private messages. It ensures the emojis are displayed correctly and enhance the communication experience.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in.
- A post with a comment section or an active private chat is accessible.

Test Data: No test data needed

Test Steps:

1. Navigate to a post's comment section or a private message window.
2. Tap the comment or message input field.
3. Select the emoji picker button within the input field.
4. Choose an emoji from the available options.
5. Verify the selected emoji appears in the comment or message input field.
6. Send the comment or message.

    7. Verify the emoji is displayed correctly within the sent comment or message.

Postconditions:

    - Messages and comments display emojis correctly.

Expected Result: The system should allow users to insert emojis into comments and messages, displaying them correctly upon sending.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: ER-003

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

Test Case Description: This test case ensures that users can see the total number of reactions a post receives, providing feedback on post engagement and popularity.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

- User is logged in.
- A post with existing reactions is available.

Test Data: No test data needed

Test Steps:

1. Navigate to a post with multiple reactions.
2. Observe the area below the post where reactions are displayed.
3. Verify a numerical count representing the total number of reactions is visible.

Postconditions: No specific postconditions.

Expected Result: The system should display a clear count of total reactions beneath the post, allowing users to gauge engagement.

Severity: Minor

Type of Testing: UI Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: ER-004

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

Test Case Description: This test case ensures the platform provides a diverse selection of emojis, encompassing various categories like emotions, objects, and symbols, to cater to diverse user preferences.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

  - User is logged in.
  - Access to emoji picker in either comments, messages or reactions.

Test Data: No test data needed

Test Steps:

  1. Access the emoji picker within any relevant feature (comments, messages, reactions).
  2. Scroll through the available emoji categories.
  3. Verify the presence of emojis representing various emotions (happy, sad, angry), objects (food, animals, places), and symbols.

Postconditions: No specific postconditions.

Expected Result: The platform should offer a comprehensive and diverse selection of emojis categorized appropriately for easy navigation.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: ER-005

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

Test Case Description: This test case validates that users can undo or modify their reactions to posts, ensuring flexibility and control over their interactions.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

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

Test Data: No test data needed

Test Steps:

1. Navigate to a post where the user has previously reacted.
2. Click on the existing reaction.
3. Verify options to either remove the reaction or choose a different emoji are presented.
4. Select the option to remove the reaction or choose a different emoji.
5. Verify the chosen action is reflected correctly on the post (reaction removed or replaced).

Postconditions:

- Reactions are updated correctly on the post.

Expected Result: The system should allow users to remove their existing reactions or change them to a different emoji, updating the displayed reactions accordingly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 10

Project name -

Social Media Platform

Project description -

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

Feature name -

Live Streaming

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Live Streaming Feature

Test Case 1

Test Case ID: LST-001

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

Test Case Description: This test case verifies that users can successfully initiate and terminate live video broadcasts on the platform. It ensures the basic functionality of starting and stopping a live stream.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

* User is logged in.

* User has necessary permissions to live stream.

Test Data: No test data needed

Test Steps:

1. Navigate to the live stream feature.

2. Initiate a new live stream.

3. Verify that the stream starts successfully and is visible.

4. Stop the live stream.

5. Verify that the stream terminates successfully and is no longer broadcasting.

Postconditions:

* The live stream is stopped.

Expected Result: The user should be able to start and stop a live stream without encountering any errors.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: LST-002

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

Test Case Description: This test case checks the privacy settings of live streams, ensuring users can choose to broadcast publicly or exclusively to their followers.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

* User is logged in.

* User has initiated a new live stream.

Test Data: No test data needed

Test Steps:

1. Before starting the live stream, access privacy settings.

2. Select the "Public" option and start the live stream.

3. Verify that the live stream is accessible to any user on the platform.

4. Stop the live stream.

5. Select the "Followers Only" option and start a new live stream.

6. Verify that the live stream is only accessible to confirmed followers of the user.

Postconditions:

* Live stream privacy settings are reset to default.

Expected Result: The system should correctly apply the chosen privacy setting, making the live stream visible to either the public or only the user's followers.

Severity: Major

Type of Testing: Security Testing, Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: LST-003

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

Test Case Description: This test case ensures viewers can engage with live content by posting comments and using platform-specific reactions during a live stream.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

* User is logged in.

* User is viewing an active live stream.

Test Data: Sample comment text, available reactions (e.g., like, love, wow).

Test Steps:

1. While viewing the live stream, enter a comment in the comment section.

2. Verify that the comment is posted and visible to other viewers.

3. Select a reaction from the available options.

4. Verify that the selected reaction is displayed in the live stream.

Postconditions: None

Expected Result: The viewer's comment and reaction should be successfully submitted and displayed on the live stream in real-time.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: LST-004

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

Test Case Description: This test case checks the real-time display of viewer count and live comments within the live stream interface.

Test Suite: Live Streaming

Test Priority: Medium

Preconditions:

* A live stream is active and running.

* Viewers are present in the live stream.

Test Data: No test data needed

Test Steps:

1. Access an ongoing live stream.

2. Verify that the viewer count accurately reflects the number of current viewers.

3. As viewers join or leave the stream, observe if the count updates in real time.

4. Observe the comment section and confirm that new comments appear dynamically as viewers submit them.

Postconditions: None

Expected Result: The live stream interface should accurately display the real-time viewer count and dynamically update as viewers join and leave the stream. The comment section should display new comments as they are submitted by viewers.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: LST-005

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

Test Case Description: This test case ensures users can save completed live stream recordings to their profiles for later access and viewing.

Test Suite: Live Streaming

Test Priority: Medium

Preconditions:

* User is logged in.

* User has completed a live stream.

Test Data: No test data needed

Test Steps:

1. After ending the live stream, locate the option to save the recording.

2. Select the save option.

3. Verify that the live stream is saved and accessible from the user's profile.

Postconditions: Saved live stream is available for viewing.

Expected Result: The system should successfully save a copy of the completed live stream, making it accessible for later viewing on the user's profile.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 11

Project name -

Social Media Platform

Project description -

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

Feature name -

Stories

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Stories Feature

Test Case 1

Test Case ID: SMPS-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 successfully create and publish stories containing photos, videos, and text overlays, ensuring the core functionality of story creation is working as intended.

Test Suite: Stories

Test Priority: High

Preconditions:

  - User is logged in.
  - User has granted camera and storage permissions (if applicable).

Test Data:

- A photo from the user's device.

- A short video from the user's device.

- Sample text for overlay: "Having a great day!"

Test Steps:

1. Navigate to the story creation screen.

2. Select the option to add a photo and choose the prepared photo.

3. Select the option to add a video and choose the prepared video.

4. Add the sample text overlay using the provided text tool.

5. Publish the story.

Postconditions:

- The story should be visible in the user's story feed.

Expected Result: The story is successfully created and displays the added photo, video, and text overlay correctly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: SMPS-002

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

Test Case Description: This test case aims to verify the ephemeral nature of stories by confirming they automatically disappear from user views and the platform after the designated 24-hour period.

Test Suite: Stories

Test Priority: High

Preconditions:

- User has created and published a story.

Test Data: No test data needed.

Test Steps:

1. Note the time of story publication.
2. Wait for 24 hours.
3. Check the user's story feed and profile.

Postconditions: None

Expected Result: The published story is no longer visible in the user's story feed or profile after 24 hours.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: SMPS-003

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

Test Case Description: This test case ensures users can view a list of users who have viewed their stories, validating the story viewing tracking and display functionality.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- User has created and published a story.
- At least one other user has viewed the story.

Test Data: No test data needed.

Test Steps:

1. Navigate to the published story.
2. Open the viewer list, usually indicated by an icon or counter on the story.

Postconditions: None

Expected Result: The system displays a list of users who viewed the story, including their usernames and profile pictures.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: SMPS-004

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

Test Case Description: This test case validates the story privacy settings, ensuring users can control who sees their stories, either making them public or restricting viewership to selected followers.

Test Suite: Stories

Test Priority: High

Preconditions:

- User is logged in.
- User is about to create a new story.

Test Data: No test data needed.

Test Steps:

1. Initiate the story creation process.
2. Locate the story privacy settings.
3. Choose the option to share publicly.
4. Create and publish the story.
5. Repeat steps 1-3, but this time choose the option to share with selected followers.
6. Select specific followers from the list and publish the story.

Postconditions: None

Expected Result:

- The story published with public settings is visible to all followers and non-followers.
- The story published with selected followers is visible only to those chosen users.

Severity: Major

Type of Testing: Security Testing, Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: SMPS-005

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

Test Case Description: This test case aims to confirm the functionality of adding visual enhancements, specifically stickers and filters, to stories, improving user engagement and creative expression.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- User is logged in.
- User is creating a new story.

Test Data: No test data needed.

Test Steps:

1. Begin creating a new story.
2. Access the sticker library within the story creation interface.
3. Select and apply a sticker to the story.
4. Access the filter options within the story creation interface.
5. Choose and apply a filter to the story.
6. Publish the story.

Postconditions: None

Expected Result:

- The selected sticker is successfully added and displayed on the story.
- The chosen filter is applied to the story, modifying its visual appearance.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 12

Project name -

Social Media Platform

Project description -

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

Feature name -

Analytics and Insights

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Social Media Platform - Analytics and Insights

Test Case 1:

Test Case ID: SMPA-001

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

Test Case Description: This test case aims to validate the functionality of the platform to display accurate and detailed follower demographic information to the user. This includes age range, gender distribution, location, and other relevant demographics.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in.
- User has at least one follower.

Test Data: No test data needed

Test Steps:

1. Navigate to the Analytics and Insights section.
2. Locate the Follower Demographics section.
3. Verify that the demographic data is displayed.
4. Verify that the data displayed is consistent with the user's actual followers.

Postconditions: None

Expected Result: The system accurately displays follower demographics data, including age range, gender, location, and other relevant information.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMPA-002

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

Test Case Description: This test case checks that the system accurately tracks and displays post reach and engagement metrics for individual posts. These metrics should include likes, comments, shares, views, and other relevant engagement data.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in.
- User has published at least one post.

Test Data: No test data needed

Test Steps:

1. Go to the Analytics and Insights section.
2. Select a specific post to view its analytics.
3. Verify that the reach metrics (e.g., views, impressions) are displayed.
4. Verify that the engagement metrics (e.g., likes, comments, shares) are displayed.

Postconditions: None

Expected Result: The system accurately displays reach and engagement metrics for the selected post, including views, impressions, likes, comments, shares, and other relevant data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMPA-003

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

Test Case Description: This test aims to ensure that the platform provides users with the ability to track their follower growth over time. This should include visualizations and data points showing changes in follower count over different timeframes.

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. Locate the Audience Growth section.
3. Verify that a graph or chart displaying follower count over time is present.
4. Verify the user can adjust the timeframe for tracking growth (e.g., past week, month, year).

Postconditions: None

Expected Result: The system presents a clear visualization and data points showing follower growth over selectable timeframes.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMPA-004

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

Test Case Description: This test case aims to ensure the user interface for presenting analytics data is intuitive and easy to understand. This includes visual elements, layout, labeling, and overall presentation of the data.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

   - User is logged in.

Test Data: No test data needed

Test Steps:

   1. Navigate to the Analytics and Insights section.

   2. Review the overall layout, design, and organization of the analytics dashboard.

   3. Assess the clarity and readability of data visualizations (graphs, charts).

   4. Evaluate the use of labels, headings, and tooltips for easy understanding of data points.

Postconditions: None

Expected Result: The analytics dashboard is visually appealing, well-organized, and easy to navigate. Data is presented clearly using intuitive visualizations and informative labels.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMPA-005

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

Test Case Description: This test case checks that the platform allows users to filter analytics data by specific time ranges (e.g., last week, last month) and by content type (e.g., posts, stories, videos).

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

- User is logged in.
- User has different types of content posted.

Test Data: No test data needed

Test Steps:

1. Go to the Analytics and Insights section.
2. Locate the filter options for time range and content type.
3. Apply filters for different time ranges and content types.
4. Verify that the displayed data reflects the applied filters.

Postconditions: None

Expected Result: The system correctly filters and displays analytics data based on the selected time range and content type.

Severity: Minor

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 Cases for Social Media Platform - Events and Groups Feature

Test Case 1:

Test Case ID: SMPEvents-001

Test Case Title: Verify that users can create and join public events.

Test Case Description: This test case verifies the functionality of creating and joining public events on the platform. It ensures that users can successfully create events with the required details and that other users can find and join these events.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* User is logged in.

* User navigates to the Events section.

Test Data: Event Name: "Test Event", Date: Future date, Time: 18:00, Location: "Online", Description: "This is a test event."

Test Steps:

1. Click on the "Create Event" button.

2. Fill in the event details (Name, Date, Time, Location, Description).

3. Set the event privacy to Public.

4. Click on the "Create Event" button.

5. Search for the created event using the search bar.

6. Click on the event from the search results.

7. Click on the "Join Event" button.

Postconditions:

* The event is displayed in the user's list of joined events.

* The event is visible in the platform's event listings.

Expected Result: The system should allow the user to create a public event with the provided details and allow other users to find and join the event successfully.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMPEvents-002

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

Test Case Description: This test case verifies the functionality of users joining interest-based groups on the platform. It ensures users can discover groups relevant to their interests and successfully become members.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* User is logged in.

* User navigates to the Groups section.

Test Data: No test data needed.

Test Steps:

1. Use the search bar to find groups related to a specific interest (e.g., "Photography").

2. Select a group from the search results.

3. Click on the "Join Group" button.

Postconditions:

* The group is added to the user's list of joined groups.

* The user can access the group's content and interact with other members.

Expected Result: The system should successfully add the user to the selected interest-based group.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMPEvents-003

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

Test Case Description: This test case verifies if group administrators can create and manage discussions within their groups, organized by specific topics to ensure streamlined communication.

Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

* User is logged in.

* User is the administrator of a group.

Test Data: Group Topic: "Upcoming Events", Discussion Title: "Suggestions for the next meeting"

Test Steps:

1. Navigate to the desired group.

2. Click on the "Discussions" tab.

3. Click on "Create New Topic".

4. Enter the topic name ("Upcoming Events").

5. Click "Create".

6. Click on "New Discussion" under the created topic.

7. Enter the discussion title ("Suggestions for the next meeting").

8. Create the discussion.

Postconditions:

* The new discussion is visible under the chosen topic.

* Group members can participate in the discussion.

Expected Result: The system should allow the creation of a new discussion topic and a new discussion under that topic, both visible to group members.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMPEvents-004

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

Test Case Description: This test case verifies that created events accurately display all necessary details including date, time, location and description to provide attendees with comprehensive information.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* User is logged in.

* User is on the "Events" page.

Test Data: No test data needed

Test Steps:

1. Click on an event from the list of available events.

2. Verify that the event details page displays the date of the event.

3. Verify that the event details page displays the time of the event.

4. Verify that the event details page displays the location of the event.

5. Verify that the event details page displays the description of the event.

Postconditions: N/A

Expected Result: The system should accurately display the date, time, location, and description of the event.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMPEvents-005

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

Test Case Description: This test case verifies the RSVP functionality for events, ensuring users can express their attendance and view other attendees.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

* User is logged in.

* User is on the event details page.

Test Data: No test data needed.

Test Steps:

1. Click on the "RSVP" button.

2. Verify that the RSVP status updates to "Attending".

3. Navigate to the "Attendees" section of the event.

4. Verify that the user's profile is listed under the "Attending" section.

Postconditions:

* The user is added to the list of attendees.

* The event organizer can view the updated attendee list.

Expected Result: The system should successfully record the user's RSVP and display their profile on the event's attendee list.

Severity: Major

Type of Testing: Functional Testing

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 Cases for Social Media Platform - Monetization Options

Test Case 1:

Test Case ID: SMP-MO-001

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

Test Case Description: This test case verifies that content creators can access and submit applications for available monetization options on the platform. It checks the application process, including the presence of necessary forms, information fields, and submission functionality.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in as a content creator.

- User has met the minimum eligibility criteria (if any).

Test Data: No test data needed

Test Steps:

1. Navigate to the user profile or settings section.

2. Locate and access the "Monetization" or "Earnings" section.

3. Verify the presence of available monetization options (e.g., advertising revenue, sponsored content).

4. Select a monetization option and proceed with the application.

5. Fill in the required information in the application form.

6. Submit the application.

Postconditions:

- Application is successfully submitted.

- User receives confirmation of application submission.

Expected Result: The system should allow eligible content creators to apply for monetization options through a clear and functional application process.

Severity: Major

Type of Testing: Functional, User Interface (UI) Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMP-MO-002

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

Test Case Description: This test case ensures that the platform offers advertising revenue and sponsored content as monetization options for eligible content creators. It validates the presence and functionality of these options within the monetization settings.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in as a content creator.
- User has successfully applied for monetization and been approved.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Monetization" or "Earnings" section of the user account.
2. Verify that "Advertising Revenue" is listed as an available monetization option.
3. Check if there are options to configure ad preferences (e.g., ad types, frequency).
4. Verify that "Sponsored Content" is listed as an available monetization option.
5. Check for functionalities like partnership proposals or brand collaboration tools.

Postconditions: None

Expected Result: The system should clearly display "Advertising Revenue" and "Sponsored Content" as active monetization options with accessible configuration settings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMP-MO-003

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

Test Case Description: This test case aims to confirm that content creators can successfully receive payments earned through the platform's monetization options. It checks the payment settings configuration, payment processing functionality, and the accuracy of payment amounts.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in as a content creator.

- User has accumulated earnings through monetization.

- User has configured their payment settings.

Test Data: Valid payment details (e.g., PayPal account, bank account information)

Test Steps:

1. Navigate to the "Payments" or "Earnings" section of the user account.

2. Verify that the user can add and manage their preferred payment methods.

3. Initiate a payment request for the accumulated earnings.

4. Confirm that the system processes the payment request successfully.

5. Check that the user receives the correct payment amount in their chosen payment method.

Postconditions:

- Earnings balance is updated after payment processing.

- Payment transaction history is accurately recorded.

Expected Result: The system should allow users to configure payment settings, process payment requests efficiently, and ensure accurate and timely disbursement of earned funds.

Severity: Critical

Type of Testing: Functional, Integration Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMP-MO-004

Test Case Title: Verify that detailed earnings reports are available

Test Case Description: This test case verifies that the platform provides content creators with detailed and comprehensive reports on their earnings. This includes information like revenue breakdown per monetization type, date-wise earnings, and transaction history, allowing users to track and analyze their earnings effectively.

Test Suite: Monetization Options

Test Priority: Medium

Preconditions:

- User is logged in as a content creator.
- User has earned revenue through monetization options.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Earnings" or "Reports" section of the user account.
2. Check if the platform generates earnings reports for different time periods (e.g., daily, weekly, monthly).
3. Verify that the reports provide a detailed breakdown of earnings per monetization type (e.g., advertising, sponsored content).
4. Ensure that the reports display accurate information on impressions, clicks, conversions (if applicable), and other relevant metrics.
5. Check for functionalities to download or export the earnings report in different formats.

Postconditions: None

Expected Result: The system should offer comprehensive and easily accessible earnings reports that provide detailed insights into the user's revenue streams and transaction history.

Severity: Minor

Type of Testing: Functional, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMP-MO-005

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

Test Case Description: This test case ensures that the platform transparently communicates the eligibility criteria for accessing monetization options. It checks for clear and accessible documentation outlining requirements like follower count, content engagement metrics, or account standing.

Test Suite: Monetization Options

Test Priority: Medium

Preconditions:

- User is browsing the platform (logged in or not logged in).

Test Data: No test data needed

Test Steps:

1. Navigate to the "Monetization" or "Earnings" section of the platform.
2. Look for a dedicated section or link that explains "Eligibility Requirements" or "How to Qualify".
3. Verify that the eligibility criteria are clearly defined, specifying any required follower counts, content views, engagement rates, or account standing.
4. Check for clarity in language and avoid ambiguous or misleading information.
5. Ensure that the information is easily accessible to all users, regardless of their login status.

Postconditions: None

Expected Result: The system should provide clear and accessible information on monetization eligibility requirements to manage user expectations and ensure transparency.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Data - 15

Project name -

Social Media Platform

Project description -

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

Feature name -

Explore or Discover Tab

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

## Test Cases for Explore Tab - Social Media Platform

Test Case 1:

Test Case ID: EXPLORE-001

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

Test Case Description: This test case aims to validate if the explore tab successfully displays a curated selection of trending and popular content from across the platform.

Test Suite: Explore Tab Functionality

Test Priority: High

Preconditions:

- User is logged in.
- User navigates to the explore tab.

Test Data: No test data needed.

Test Steps:

1. Observe the content displayed in the explore tab.
2. Verify that the content aligns with the platform's definition of "trending" and "popular". This might involve checking for:
    - Posts with a high number of likes, comments, and shares.
    - Topics that are currently being widely discussed.
    - Accounts that are experiencing a surge in followers or engagement.

Postconditions: None

Expected Result: The explore tab should prominently feature a diverse range of content that has been identified as trending or popular based on predefined algorithms or metrics.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: EXPLORE-002

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

Test Case Description: This test case aims to verify if the content displayed in the explore tab is personalized to each user based on their past activity, interactions, and declared interests.

Test Suite: Explore Tab Functionality

Test Priority: High

Preconditions:

- User is logged in.
- User has engaged with content on the platform (e.g., liked posts, followed accounts, joined communities).

Test Data: User's past activity and engagement data.

Test Steps:

1. Analyze the user's past activity and identify their areas of interest.
2. Navigate to the explore tab and observe the recommended content.
3. Verify that the recommended content aligns with the user's identified interests.

Postconditions: None

Expected Result: The explore tab should prioritize content that is relevant to the user's interests, increasing the likelihood of engagement and discovery.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: EXPLORE-003

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

Test Case Description: This test case checks the functionality of filtering content within the explore tab by specific categories or topics.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

- User is logged in.
- User is on the explore tab.

Test Data: Predefined list of categories or topics available for filtering.

Test Steps:

1. Locate and interact with the filtering mechanism within the explore tab.
2. Select a specific category or topic from the available options.
3. Verify that the displayed content is filtered accordingly, showing only content relevant to the selected category or topic.

Postconditions: None

Expected Result: The explore tab should allow users to refine their content discovery by selecting specific categories or topics they are interested in, improving the user experience by providing more targeted content.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: EXPLORE-004

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

Test Case Description: This test ensures that the content displayed in the explore tab is dynamic and updates periodically to reflect new trends and popular posts.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: No test data needed.

Test Steps:

1. Note the content currently displayed on the explore tab.
2. Wait for a predefined refresh interval (e.g., 1 hour, 1 day).
3. Return to the explore tab and compare the content with the previous observation.

Postconditions: None

Expected Result: The explore tab should display new and updated content after the refresh interval, ensuring that users are presented with fresh and engaging content upon each visit.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: EXPLORE-005

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

Test Case Description: This test case aims to ensure users can interact with content in the explore tab by saving or following posts, accounts, or topics they find interesting.

Test Suite: Explore Tab Functionality

Test Priority: Medium

Preconditions:

- User is logged in.
- User is on the explore tab.

Test Data: No test data needed.

Test Steps:

1. Identify a piece of content (post, account, topic) on the explore tab.
2. Locate and interact with the "save" or "follow" option associated with the content.

3. Verify that the content is successfully saved or followed, either through visual confirmation or by checking the user's saved items or following list.

Postconditions: None

Expected Result: Users should be able to easily save or follow content from the explore tab, allowing them to revisit or stay updated on content they find valuable.

Severity: Minor

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 Cases for Social Media Platform Accessibility Features:

Test Case 1:

Test Case ID: SMP-A11Y-001

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

Test Case Description: This test case ensures that all images on the platform have appropriate alternative text (alt text) to convey their meaning to users who are visually impaired and using screen readers.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

  - User is logged in.
  - User navigates to a page with images (e.g., newsfeed, profile page).

Test Data: No test data needed.

Test Steps:

  1. Navigate to different sections of the platform containing images.
  2. Use a screen reader or browser developer tools to inspect the alt text attribute of each image.
  3. Verify that each image has a descriptive and accurate alt text that conveys the image's content or purpose.

Postconditions: No specific postconditions.

Expected Result: All images should have meaningful and descriptive alt text, accurately representing the visual content.

Severity: Major

Type of Testing: Accessibility Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMP-A11Y-002

Test Case Title: Verify that screen reader compatibility is available.

Test Case Description: This test case verifies that the platform is compatible with popular screen reader software, allowing users with visual impairments to navigate and interact with the platform's content effectively.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- A screen reader software (e.g., NVDA, JAWS) is installed and running.

Test Data: No test data needed.

Test Steps:

1. Navigate through the platform's core functionalities (e.g., login, newsfeed, posting, messaging) using a screen reader.

2. Verify that the screen reader accurately announces all elements, including text, buttons, links, and form fields.

3. Ensure that interactive elements are identifiable and operable using the screen reader's commands.

Postconditions: No specific postconditions.

Expected Result: The screen reader should seamlessly navigate the platform, providing auditory feedback that accurately reflects the visual layout and functionality.

Severity: Critical

Type of Testing: Accessibility Testing, Compatibility Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMP-A11Y-003

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

Test Case Description: This test confirms that users can navigate the platform and access its functionalities entirely through keyboard input, catering to users who cannot use a mouse.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in.

Test Data: No test data needed.

Test Steps:

1. Navigate through the platform solely using keyboard commands (e.g., Tab, Enter, arrow keys).
2. Attempt to access all functionalities, including posting, commenting, liking, messaging, and navigating menus, using only the keyboard.
3. Verify that all interactive elements are reachable and operable without mouse interaction.

Postconditions: No specific postconditions.

Expected Result: The platform should be fully navigable and functional using only keyboard input, ensuring accessibility for users who cannot use a mouse.

Severity: Major

Type of Testing: Accessibility Testing, Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMP-A11Y-004

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

Test Case Description: This test ensures that the platform allows users to adjust color contrast and font sizes to their preference, making it accessible for users with visual impairments or reading difficulties.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: No test data needed.

Test Steps:

1. Locate the platform's accessibility settings or display options.
2. Verify that options are available to adjust color contrast levels (e.g., light mode, dark mode, high contrast mode).
3. Confirm that options exist to increase or decrease font sizes across the platform.
4. Test the functionality of these options to ensure they effectively modify the display as intended.

Postconditions:

- Reset any changes made to default settings after testing.

Expected Result: The platform should provide user-adjustable settings for color contrast and font sizes, and these settings should effectively modify the display to improve readability and accessibility.

Severity: Major

Type of Testing: Accessibility Testing, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMP-A11Y-005

Test Case Title: Verify that accessibility features are clearly documented.

Test Case Description: This test ensures that the platform's accessibility features are thoroughly documented, allowing users to learn about and utilize these features effectively.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions: No specific preconditions.

Test Data: No test data needed.

Test Steps:

1. Locate the platform's help documentation or accessibility section.
2. Verify that the documentation includes clear and concise information about the available accessibility features.
3. Ensure that instructions on how to activate and utilize each feature are provided.
4. Confirm that the documentation is easily accessible and understandable for all users.

Postconditions: No specific postconditions.

Expected Result: The platform should provide comprehensive documentation on its accessibility features, clearly explaining their functionality and how users can activate and utilize them.

Severity: Minor

Type of Testing: Documentation Testing

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 Cases for Social Media Platform - Safety and Moderation Tools

Test Case 1

Test Case ID: SMST-001

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

Test Case Description: This test case aims to validate the effectiveness of the platform's content moderation algorithms in detecting and filtering various types of inappropriate content, including text, images, and videos that violate the community guidelines.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

- A user account is created.

- The community guidelines are defined, outlining inappropriate content.

Test Data:

- A set of text posts, images, and videos containing various degrees of inappropriate content (e.g., hate speech, nudity, violence).

Test Steps:

1. Log in to the platform as the test user.

2. Attempt to post the prepared content (text, images, and videos).

3. Observe the system's response to each post attempt.

Postconditions:

- Inappropriate content is successfully flagged and blocked from being posted.

Expected Result: The content moderation algorithms should successfully identify and block the posting of inappropriate content based on predefined rules and community guidelines.

Severity: Blocker

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Negative

Test Case 2

Test Case ID: SMST-002

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

Test Case Description: This test case will ensure users can easily report instances of harmful or inappropriate content they encounter on the platform, including posts, comments, and user profiles.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

- Two user accounts are created (one to post content, one to report).
- A post containing potentially harmful content is available on the platform.

Test Data: No test data needed

Test Steps:

1. Log in as the user who will report the content.
2. Navigate to the post containing the potentially harmful content.
3. Locate and utilize the "Report" functionality associated with the post.
4. Select the appropriate reason for reporting from the provided options.
5. Submit the report.

Postconditions:

- The reported content is flagged for review by the moderation team.

Expected Result: The platform should provide an intuitive and accessible reporting mechanism, allowing users to flag content effectively.

Severity: Major

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: SMST-003

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

Test Case Description: This test aims to confirm users can block other users they wish to avoid interacting with, preventing unwanted contact via messages, comments, or content visibility.

Test Suite: Safety and Moderation

Test Priority: Medium

Preconditions:

- Two user accounts are created (one to be blocked, one to initiate the block).

Test Data: No test data needed

Test Steps:

1. Log in as the user who wants to block another user.

2. Navigate to the profile of the user they want to block.

3. Locate and utilize the "Block User" functionality on the user's profile.

4. Confirm the block action.

Postconditions:

- The blocked user can no longer interact with the user who blocked them.

Expected Result: The platform should successfully prevent any further interaction attempts from the blocked user, ensuring the user's safety and control over their experience.

Severity: Major

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: SMST-004

Test Case Title: Verify that comment filters prevent offensive language.

Test Case Description: This test aims to confirm the functionality of the comment filters in identifying and preventing the posting of comments containing offensive language, based on a predefined list of inappropriate words and phrases.

Test Suite: Safety and Moderation

Test Priority: Medium

Preconditions:

- A user account is created.

- A list of offensive words and phrases is defined and integrated into the comment filter system.

Test Data:

- A set of comments containing various offensive words and phrases.

Test Steps:

1. Log in to the platform as the test user.
2. Navigate to any post that allows comments.
3. Attempt to post each prepared comment containing offensive language.
4. Observe the system's response to each comment attempt.

Postconditions:

- Comments containing offensive language are successfully blocked.

Expected Result: The comment filters should successfully detect and prevent the posting of comments containing offensive language, contributing to a more respectful online environment.

Severity: Major

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Negative

Test Case 5

Test Case ID: SMST-005

Test Case Title: Verify that community guidelines are enforced consistently.

Test Case Description: This test case aims to assess the consistency of community guideline enforcement by analyzing the platform's response to reported content and user behavior.

Test Suite: Safety and Moderation

Test Priority: High

Preconditions:

- Multiple user accounts are created.
- Various pieces of content violating community guidelines are posted.

Test Data:

- A set of predefined scenarios involving different violations of community guidelines (e.g., hate speech, harassment, spam).

Test Steps:

1. Have different test users post content that violates community guidelines in various ways.
2. Report the violating content using different user accounts.
3. Monitor the platform's response to each reported instance (e.g., content removal, account suspension).
4. Analyze the consistency of the platform's actions concerning the severity of the violation and the time taken for action.

Postconditions:

- A report documenting the consistency (or inconsistency) of the platform's response is generated.

Expected Result: The platform should demonstrate a consistent and timely approach in addressing violations of its community guidelines, regardless of the user involved.

Severity: Critical

Type of Testing: Functional Testing, Regression Testing, Exploratory Testing

Test Case Approach: Positive, Negative

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 Cases for Multi-Language Support on Social Media Platform

Test Case 1:

Test Case ID: ML-001

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

Test Case Description: This test case aims to validate that users can successfully navigate to the language settings and select their preferred language for the platform's interface.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

* User is logged in.

* User has access to account settings.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user account settings.

2. Locate the "Language" or "Language Preferences" section.

3. Select a different language from the available options.

4. Save the changes made to language preferences.

Postconditions:

* The user interface should refresh and display in the selected language.

Expected Result: The user interface should successfully switch to the chosen language, confirming user preference is applied.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: ML-002

Test Case Title: Verify that the platform offers content translation features.

Test Case Description: This test case verifies that users are presented with an option to translate posts and messages written in languages different from their selected interface language.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

* User is logged in.

* A post or message in a different language is displayed.

Test Data: A post or message in a language different from the user's selected language.

Test Steps:

1. View a post or message written in a different language.

2. Check for the presence of a "Translate" button or option.

3. Click on the "Translate" button.

Postconditions: N/A

Expected Result: The platform should display the translated content in the user's preferred language.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: ML-003

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

Test Case Description: This test case ensures that the translation feature accurately translates posts and messages into various supported languages, maintaining context and meaning.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

* User is logged in.

* User has encountered content in a different language.

Test Data: A sample of posts and messages in different languages.

Test Steps:

1. Select various posts and messages in different languages.

2. Utilize the "Translate" feature for each selected item.

3. Compare the translated content with the original for accuracy and clarity.

Postconditions: N/A

Expected Result: The translated content should accurately reflect the meaning of the original posts and messages in the chosen languages.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: ML-004

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

Test Case Description: This test case validates that the platform remembers and applies the user's chosen language across different sessions and devices, ensuring a seamless multilingual experience.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

* User is logged in.

* User has selected their preferred language.

Test Data: No test data needed.

Test Steps:

1. Select and save the preferred language.

2. Log out of the platform.

3. Log back into the platform.

4. Navigate to different sections of the platform.

5. Access the platform from a different device or browser.

Postconditions: N/A

Expected Result: The chosen language should be retained and applied consistently across all sessions, devices, and sections of the platform.

Severity: Major

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: ML-005

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

Test Case Description: This test case aims to confirm that the platform offers support for a diverse set of languages, catering to a global user base and promoting inclusivity.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

* User has access to language settings.

Test Data: A list of expected supported languages.

Test Steps:

1. Navigate to the language selection section within the platform.
2. Check the list of available language options.
3. Compare the available language options with the documented list of supported languages.

Postconditions: N/A

Expected Result: The platform should provide a wide array of language options, meeting the needs of a diverse user base.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 19

Project name -

Social Media Platform

Project description -

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

Feature name -

Cross-Platform Integration

Feature description -

Cross-platform integration allows users to connect and share content across different social media platforms and external services. This includes features like single sign-on (SSO) capabilities, cross-posting of content, integration with external media sources, and the ability to share platform content on other social networks, enhancing user convenience and content reach.

Userstory title -

As a user, I want cross-platform integration so that I can connect and share content across multiple platforms.

Acceptance criteria -

['Users can link their accounts with other social media platforms.', 'Content can be cross-posted to other platforms.', 'Users can sign in using single sign-on (SSO).', 'Integration with external media sources is available.', 'Content sharing across platforms is seamless.']

Generated Test Cases:

## Test Cases for Social Media Platform - Cross-Platform Integration:

Test Case 1:

Test Case ID: SM-CPI-001

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

Test Case Description: This test case verifies the functionality of linking a user's account on the platform with their accounts on other social media platforms like Facebook, Twitter, and Instagram.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User is logged in to their account on the platform.

* User has active accounts on other social media platforms.

Test Data: Valid credentials for other social media platforms.

Test Steps:

1. Navigate to the user profile settings.

2. Go to the "Linked Accounts" or "Connections" section.

3. Select the desired social media platform to link.

4. Enter the credentials for the chosen platform.

5. Confirm the linking process.

Postconditions:

* The user's account on the platform is successfully linked to their chosen platform.

* The linked platform is visible in the "Linked Accounts" section.

Expected Result: The system should successfully link the user's account with the other social media platform.

Severity: Major

Type of Testing: Functional, Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SM-CPI-002

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

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

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User is logged in to their account.

* User has successfully linked their account with at least one other social media platform.

Test Data: A sample post (text, image, or video) for cross-posting.

Test Steps:

1. Create a new post on the platform.

2. During post creation, select the option for cross-posting.

3. Choose the linked platform(s) where the content should be shared.

4. Publish the post.

Postconditions:

* The post is successfully published on the platform.

* The post is automatically shared on the selected linked platforms.

Expected Result: The system should successfully cross-post the content to the chosen linked platforms.

Severity: Major

Type of Testing: Functional, Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SM-CPI-003

Test Case Title: Verify that users can sign in using single sign-on (SSO).

Test Case Description: This test case validates the functionality of Single Sign-On (SSO) using existing credentials from supported platforms (e.g., Google, Facebook) to access the platform.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User has an existing account with a supported SSO provider.

Test Data: Valid credentials for the supported SSO provider (e.g., Google, Facebook).

Test Steps:

1. Navigate to the platform's login page.

2. Click on the SSO button (e.g., "Sign in with Google").

3. Enter credentials for the chosen SSO provider.

4. Grant permission to the platform to access necessary information from the SSO provider.

Postconditions:

* User is successfully logged in to the platform.

Expected Result: The system should allow users to sign in seamlessly using their existing credentials from supported SSO providers.

Severity: Major

Type of Testing: Functional, Security Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SM-CPI-004

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

Test Case Description: This test case examines the integration with external media sources, allowing users to embed content from platforms like YouTube, Spotify, or SoundCloud into their posts.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

* User is logged in to their account.

Test Data: A valid URL from a supported external media source (e.g., YouTube video link).

Test Steps:

1. Create a new post on the platform.

2. Paste the URL of the external media content into the post body.

3. Publish the post.

Postconditions:

* The external media content should be embedded within the post.

Expected Result: The system should successfully embed the external media content into the post, allowing users to view and interact with it directly on the platform.

Severity: Minor

Type of Testing: Functional, Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SM-CPI-005

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

Test Case Description: This test case evaluates the overall user experience for sharing content across different platforms, ensuring it is smooth and user-friendly.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

* User is logged in and has linked their account with at least one other platform.

Test Data: No test data needed.

Test Steps:

1. Attempt to link accounts, cross-post content, and share content using various features.

2. Observe the user interface, navigation, and any error messages or prompts.

3. Evaluate the ease of use, clarity of instructions, and overall intuitiveness of the process.

Postconditions: N/A

Expected Result: The user should experience a seamless and intuitive flow while connecting accounts, cross-posting, and sharing content across platforms. Error messages and prompts should be clear, helpful, and guide the user towards a solution.

Severity: Minor

Type of Testing: Usability Testing, User Experience (UX) 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 Cases for Social Media Platform - Gamification and Rewards

Test Case 1:

Test Case ID: SMGR-001

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

Test Case Description: This test case verifies that users earn points and badges for various activities on the platform, such as creating a profile, sharing a post, commenting, liking content, and inviting friends.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User is logged in.

- Gamification features are enabled.

Test Data: No test data needed.

Test Steps:

1. Log in as a registered user.

2. Complete various activities like creating a profile, sharing a post, commenting, liking content, and inviting

friends.

3. Check the user's profile for awarded points and badges.

Postconditions:

- User's points and badges should be updated accordingly.

Expected Result: The user's points and badge count should accurately reflect the completed activities.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMGR-002

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

Test Case Description: This test case verifies that the leaderboard accurately displays the top-ranking users based on their accumulated points or specific achievements.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- Users have earned points and badges.

- The Leaderboard feature is accessible.

Test Data: No test data needed.

Test Steps:

1. Access the Leaderboard section of the platform.

2. Observe the ranking of users displayed.

3. Verify that the ranking accurately reflects the users' points or achievements.

Postconditions: No specific postconditions.

Expected Result: The Leaderboard should accurately display users ranked by points or achievements.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMGR-003

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

Test Case Description: This test case verifies that the platform provides various challenges for users to participate in, complete tasks within a given timeframe, and earn rewards upon successful completion.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- User is logged in.
- Challenges feature is active.

Test Data: List of available challenges and their requirements.

Test Steps:

1. Navigate to the Challenges section.
2. Review the list of available challenges and their completion requirements.
3. Participate and complete a challenge according to its rules.
4. Check for the awarded rewards upon successful completion.

Postconditions:

- User's profile should reflect earned rewards from the challenge.

Expected Result: The platform provides engaging challenges, and rewards are correctly given upon successful completion.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMGR-004

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

Test Case Description: This test case validates that the platform offers a variety of rewards, including virtual goods, discounts on partner services, or access to exclusive content, based on user achievements and participation in the gamified system.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- Users have earned rewards.

Test Data: No test data needed.

Test Steps:

1. Earn different types of rewards by completing challenges or accumulating points.

2. Check the reward inventory for the type and availability of rewards earned (virtual goods, discounts, exclusive content).

Postconditions:

- Reward inventory accurately reflects earned rewards.

Expected Result: Rewards system should offer a diverse range of incentives, including virtual goods, discounts, and exclusive content, that users can readily access and utilize.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMGR-005

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 have a dedicated section (e.g., "Achievements" or "My Progress") where they can easily view their accumulated points, earned badges, completed challenges, and overall progress within the platform's gamification system.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- User is logged in.

- User has earned points, badges, or completed challenges.

Test Data: No test data needed.

Test Steps:

1. Locate and navigate to the dedicated progress section in the user profile.

2. Verify that the section displays the user's accumulated points.

3. Confirm the display of earned badges and their respective levels.

4. Check for the accurate listing of completed challenges and associated rewards.

Postconditions: No specific postconditions.

Expected Result: The user interface should provide a clear and concise overview of the user's progress and achievements within the gamification system, fostering a sense of accomplishment and encouraging continued participation.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 21

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Creation

Feature description -

Users can create tasks by providing detailed information such as the task title, description, due date, priority level, and assigning it to specific team members. This feature serves as the foundation for organizing and managing tasks within the tool.

Userstory title -

As a user, I want to create tasks so that I can organize and manage my work effectively.

Acceptance criteria -

['Users can enter a task title, description, due date, and priority level.', 'Tasks can be assigned to specific team members.', 'The system should save and display created tasks.', 'Users can edit task details after creation.', 'The system should validate required fields (e.g., task title).']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Creation Feature

Test Case 1:

Test Case ID: TMT-TC-001

Test Case Title: Verify that users can create tasks with all required fields.

Test Case Description: This test case ensures that a user can successfully create a new task by filling in all the required fields, including task title, description, due date, and priority level.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- User is logged in.
- User is on the "Create Task" page.

Test Data:

- Task Title: Test Task 1
- Description: This is a test task for verification.
- Due Date: One week from the current date.
- Priority: High

Test Steps:

1. Enter "Test Task 1" in the task title field.
2. Enter "This is a test task for verification." in the description field.
3. Select a due date one week from the current date.
4. Choose "High" from the priority dropdown.
5. Click the "Create Task" button.

Postconditions:

- The created task should be displayed on the user's task list.
- The task details should match the entered data.

Expected Result: The system should successfully create a new task with the provided details and display it on the user's task list.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TC-002

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

Test Case Description: This test case validates the functionality of assigning a newly created task to a specific team member.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- User is logged in.
- User is on the "Create Task" page.
- At least one other team member exists in the system.

Test Data: No test data needed.

Test Steps:

1. Fill in the required task details (title, description, due date, priority).
2. Click on the "Assign to" field.
3. Select a team member from the list of available users.
4. Click the "Create Task" button.

Postconditions:

- The created task should be visible on the assigned team member's task list.

Expected Result: The system should assign the created task to the selected team member.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TC-003

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

Test Case Description: This test case ensures that the system correctly saves all the details of a newly created task and displays it accurately in the task list view.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- User is logged in.

Test Data: No test data needed.

Test Steps:

1. Create a new task with all required details (title, description, due date, priority, assigned team member).
2. Navigate to the task list view.
3. Verify that the newly created task is displayed on the list.
4. Click on the created task to view its details.
5. Verify that all the saved details match the information entered during task creation.

Postconditions: N/A

Expected Result: The created task should be visible in the task list with accurate and complete details.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-TC-004

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

Test Case Description: This test case ensures that users can edit the details of an existing task, including title, description, due date, priority, and assigned team member.

Test Suite: Task Creation

Test Priority: Medium

Preconditions:

- User is logged in.

- A task already exists in the system.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list view.

2. Click on an existing task to open its details.

3. Click the "Edit" button.

4. Modify one or more of the task details.

5. Click the "Save Changes" button.

Postconditions:

- The edited task details should be updated and displayed correctly in the task list.

Expected Result: The system should save the edited task details successfully, and the changes should reflect in the task list.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TC-005

Test Case Title: Verify that the system validates required fields during task creation.

Test Case Description: This test case aims to ensure that the system enforces data validation by checking for required fields like the task title before allowing task creation.

Test Suite: Task Creation

Test Priority: Medium

Preconditions:

- User is logged in.
- User is on the "Create Task" page.

Test Data: No test data needed.

Test Steps:

1. Leave the task title field empty.
2. Fill in other optional fields (description, due date, priority, assigned team member).
3. Click the "Create Task" button.

Postconditions: N/A

Expected Result: The system should prevent task creation and display an error message indicating that the task title is a required field.

Severity: Minor

Type of Testing: Functional Testing, Negative Testing

Test Case Approach: Negative

Data - 22

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Assignment

Feature description -

Tasks can be assigned to specific team members, delegating responsibility and ensuring clarity about who is accountable for each task. Assignees receive notifications about their assigned tasks, helping them stay informed and focused on their responsibilities.

Userstory title -

As a user, I want to assign tasks to specific team members so that responsibilities are clear and tasks are managed effectively.

Acceptance criteria -

['Users can select team members to assign tasks.', 'Assignees receive notifications about their tasks.', "The system should display assigned tasks in the assignee's task list.", 'Users can reassign tasks to different team members.', 'Task assignment history should be tracked.']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Assignment Feature:

Test Case 1:

Test Case ID: TMT-TA-001

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

Test Case Description: This test case verifies that users can successfully assign tasks to specific team members from the list of available team members within the task management tool.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- User is logged in.

- A project with at least one team member exists.

- A task has been created within the project.

Test Data: No test data needed.

Test Steps:

1. Navigate to the desired project.

2. Open the task designated for assignment.

3. Locate the "Assign To" field.

4. Select a team member from the dropdown list.

5. Save the task.

Postconditions:

- The task should display the assigned team member.

Expected Result: The system allows the user to select a team member and successfully assigns the task.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TA-002

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

Test Case Description: This test case checks that upon task assignment, the designated team member receives a notification informing them about the newly assigned task.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- User is logged in.

- A project with at least one team member exists.

- The assignee has enabled notifications for task assignments.

Test Data: No test data needed.

Test Steps:

1. Assign a task to a team member (following steps from Test Case 1).

2. Log in as the assigned team member.

3. Check for a notification indicating the new task assignment.

Postconditions: None.

Expected Result: The assignee receives a clear and timely notification about the assigned task.

Severity: Major

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TA-003

Test Case Title: Verify that the system displays assigned tasks in the assignee's task list.

Test Case Description: This test case ensures that once a task is assigned, it appears correctly within the assigned team member's personal task list view.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- A task has been assigned to a specific team member.

Test Data: No test data needed.

Test Steps:

1. Log in as the team member to whom the task was assigned.

2. Navigate to the user's task list view.

Postconditions: None.

Expected Result: The assigned task is visible and accessible within the team member's task list.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-TA-004

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

Test Case Description: This test case validates the functionality of reassigning tasks from one team member to another.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

- A task is currently assigned to a team member.

Test Data: No test data needed.

Test Steps:

1. Navigate to the assigned task.

2. Locate the "Assign To" field.

3. Select a different team member from the dropdown list.

4. Save the changes.

Postconditions:

- The task should be removed from the initial assignee's list.

- The reassigned team member should receive a notification.

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

Severity: Minor

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TA-005

Test Case Title: Verify that task assignment history is tracked.

Test Case Description: This test case checks whether the system maintains a record of all task assignment and reassignment actions performed.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

- A task has been assigned and subsequently reassigned at least once.

Test Data: No test data needed.

Test Steps:

1. Open the task for which the assignment history needs to be reviewed.

2. Locate and access the "Task History" or "Activity Log."

Postconditions: None.

Expected Result: The system displays a clear and accurate log of all past assignments and reassignments for the task, including timestamps and user details.

Severity: Minor

Type of Testing: Functional Testing, Audit Testing

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 Cases for Task Management Tool - Task Deadline Reminders

Test Case 1

Test Case ID: TMT-TDR-001

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

Test Case Description: This test case verifies the functionality for users to set deadline-based reminders for individual tasks within the task management tool. It ensures users can input a specific date and time for the reminder, linked to the task's deadline.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged into the application.
- A task has been created.

Test Data:

- Task deadline: 2023-12-10, 10:00 AM
- Reminder setting: 2023-12-09, 9:00 AM

Test Steps:

1. Navigate to the created task.
2. Access the task details or settings.
3. Locate the "Set Reminder" or similar option.
4. Input the desired reminder date and time, prior to the task deadline.
5. Save the reminder settings.

Postconditions:

- The reminder is successfully set and associated with the task.

Expected Result: The system should store and display the set reminder within the task details, reflecting the user-defined date and time.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TMT-TDR-002

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

Test Case Description: This test case checks if users can customize the urgency level of task deadline reminders. This includes verifying the availability of different urgency levels and their impact on reminder notifications.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged into the application.
- A task with a deadline is created.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task settings.
2. Locate the reminder settings for the task.
3. Verify the presence of options to customize urgency level (e.g., High, Medium, Low).
4. Select different urgency levels and observe changes in reminder notification style (e.g., notification tone, pop-up appearance).

Postconditions: None

Expected Result: The system should provide options for customizing reminder urgency levels and visually differentiate notifications based on the selected urgency.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TMT-TDR-003

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

Test Case Description: This test case validates that the system effectively delivers notifications to users for upcoming task deadlines based on their set reminder preferences.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged into the application.
- A task with a deadline and reminder is set.

Test Data: System clock set to trigger the reminder notification.

Test Steps:

1. Wait for the system clock to reach the set reminder time.
2. Observe if the notification is delivered through the chosen delivery method.
3. Verify that the notification content accurately reflects the task details and deadline.

Postconditions: None

Expected Result: The system should successfully deliver a notification to the user at the specified reminder time, containing accurate and clear information about the approaching task deadline.

Severity: Critical

Type of Testing: Functional Testing, End-to-End Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TMT-TDR-004

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

Test Case Description: This test case ensures users have the flexibility to select their preferred method of receiving task deadline reminders. It validates the availability and functionality of different delivery options.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged into the application.
- A task is created with a deadline.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user settings or preferences section.

2. Locate the "Notifications" or "Reminders" settings.

3. Verify that options for different delivery methods are present (e.g., email, in-app notifications, mobile push notifications).

4. Select and save the preferred delivery method.

5. Set a reminder for a task and confirm the notification is received via the selected method.

Postconditions: None

Expected Result: The system should allow users to choose from various reminder delivery methods and successfully deliver notifications through the chosen method.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TMT-TDR-005

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

Test Case Description: This test case checks the flexibility provided to users in managing their reminder settings. It ensures that users can modify existing reminder settings for a task or completely disable reminders if they choose.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged into the application.
- A task with a set deadline and reminder exists.

Test Data:

- New reminder time: 2023-12-09, 12:00 PM

Test Steps:

1. Navigate to the task with the pre-set reminder.
2. Access the reminder settings for the task.
3. Attempt to adjust the existing reminder time.
4. Verify that the system saves the changes to the reminder time.
5. Locate the option to disable reminders for the task.
6. Disable reminders and confirm the change is reflected in the task settings.

Postconditions:

- Reminder is disabled for the test task.

Expected Result: The system should allow users to edit the date and time of existing reminders and provide an option to disable reminders entirely.

Severity: Minor

Type of Testing: Functional Testing

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 Cases for Task Prioritization:

Test Case 1:

Test Case ID: TP-001

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

Test Case Description: This test case verifies that users can successfully set the priority level (High, Medium, Low) for newly created tasks within the task management tool.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.

- User is on the "Create New Task" page.

Test Data: No test data needed.

Test Steps:

1. Access the task creation form.

2. Locate the priority level selection field.

3. Select each priority level (High, Medium, Low) individually.

4. Observe if the selection is reflected accurately in the field.

Postconditions:

- The selected priority level is displayed with the task details.

Expected Result: The system should allow the user to select a priority level for the task from the available options (High, Medium, Low) without any errors.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TP-002

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

Test Case Description: This test case ensures that the task management tool correctly sorts and displays tasks based on their assigned priority levels (High, Medium, Low).

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.

- Multiple tasks with different priority levels exist.

Test Data: Tasks with pre-defined High, Medium, and Low priority levels.

Test Steps:

1. Navigate to the task list view.

2. Locate the sorting function.

3. Select "Priority" as the sorting criteria.

4. Observe the order of displayed tasks.

Postconditions: None

Expected Result: The system should display the tasks sorted correctly by priority, with High priority tasks at the top, followed by Medium, and then Low priority tasks.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TP-003

Test Case Title: Verify that users can change the priority level of an existing task.

Test Case Description: This test case validates that users can modify the priority level of a pre-existing task within the task management tool.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.

- A task exists with an assigned priority level.

Test Data: A task with a pre-set priority level (e.g., Medium).

Test Steps:

1. Open the details view of the existing task.

2. Locate the priority level setting.

3. Change the priority level to a different value (e.g., from Medium to High).

4. Save the changes made to the task.

Postconditions:

- The updated priority level is displayed with the task details.

Expected Result: The system should successfully update the task's priority level, reflecting the change both in the task details and within the task list view.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TP-004

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

Test Case Description: This test case aims to confirm that the system utilizes visual cues (e.g., color coding, icons) to differentiate tasks based on their assigned priority levels.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in.
- Tasks with different priority levels are present.

Test Data: Tasks with pre-defined High, Medium, and Low priority levels.

Test Steps:

1. Navigate to the task list view.
2. Observe the visual representation of tasks.
3. Identify if distinct visual cues are used for each priority level.

Postconditions: None

Expected Result: The system should employ clear and distinguishable visual cues, such as color coding or icons, to represent different task priority levels effectively. For instance, High priority tasks might be highlighted in red, Medium in yellow, and Low in green.

Severity: Minor

Type of Testing: UI Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TP-005

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

Test Case Description: This test case ensures that when a new task is created without an explicit priority level selection, the system automatically assigns a default priority level of Medium.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in.

- User is on the "Create New Task" page.

Test Data: No test data needed.

Test Steps:

1. Access the task creation form.

2. Do not select any priority level.

3. Create the task without setting the priority.

4. Open the created task details view.

5. Check the assigned priority level of the newly created task.

Postconditions:

- The created task should display "Medium" as the assigned priority level.

Expected Result: The system should automatically assign a Medium priority level to any newly created task where the user has not explicitly chosen a priority level.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 25

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Subtasks

Feature description -

Tasks can be broken down into smaller subtasks, enabling users to divide complex tasks into manageable components. Subtasks can have their own due dates, assignees, and dependencies, providing granularity and clarity in task execution.

Userstory title -

As a user, I want to create subtasks so that I can break down complex tasks into manageable parts.

Acceptance criteria -

['Users can create subtasks under main tasks.', 'Subtasks can have their own due dates and assignees.', 'The system tracks the progress of each subtask.', 'Subtasks can be marked as complete independently of the main task.', 'Dependencies between subtasks can be set.']

Generated Test Cases:

## Test Cases for Subtasks Feature:

Test Case 1:

Test Case ID: TMT-ST-001

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

Test Case Description: This test case ensures that users can successfully create subtasks associated with existing main tasks within the task management tool. It verifies the functionality of adding subtasks, ensuring a clear parent-child relationship between tasks.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in.
- A main task is already created.

Test Data: No test data needed.

Test Steps:

1. Navigate to the main task.
2. Locate and select the option to add a subtask.
3. Input a title/description for the subtask.
4. Save the subtask.

Postconditions:

- The subtask is displayed under the main task.

Expected Result: The system should successfully create and display the subtask under the parent main task.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-ST-002

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

Test Case Description: This test case validates the ability to set individual due dates and assign different team members to specific subtasks, ensuring flexibility and clarity in task allocation.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in.
- A main task with at least one subtask exists.

Test Data:

- Valid future date for the due date.
- Username of an existing team member.

Test Steps:

1. Navigate to the subtask.
2. Open the subtask details.
3. Input/select a due date for the subtask.
4. Select an assignee from the team.
5. Save the subtask changes.

Postconditions:

- The subtask should reflect the selected due date and assigned team member.

Expected Result: The system should successfully update the subtask with the specified due date and assignee.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-ST-003

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

Test Case Description: This test case ensures that the task management tool accurately reflects the completion status of individual subtasks, providing a clear overview of progress on the main task.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in.
- A main task with at least one subtask exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the main task view.
2. Mark a subtask as complete.
3. Observe the progress indicator for the main task.

Postconditions:

- The main task progress should update to reflect the completed subtask.

Expected Result: The system should accurately update and display the progress of the main task based on the completion status of its subtasks.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-ST-004

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

Test Case Description: This test case confirms that subtasks can be marked as complete even if the main task itself is not yet finished, allowing for accurate progress tracking within complex tasks.

Test Suite: Subtasks

Test Priority: Medium

Preconditions:

- User is logged in.
- A main task with at least one incomplete subtask exists.

Test Data: No test data needed

Test Steps:

1. Navigate to the main task view.
2. Mark a subtask as complete, while the main task remains incomplete.

Postconditions:

- The subtask should be displayed as complete.
- The main task should remain in its current (incomplete) state.

Expected Result: The system should allow marking a subtask as complete without affecting the main task's status.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-ST-005

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

Test Case Description: This test case validates the functionality of establishing dependencies between subtasks, ensuring that users can define sequential relationships and workflow within a main task.

Test Suite: Subtasks

Test Priority: Medium

Preconditions:

- User is logged in.
- A main task with at least two subtasks exists.

Test Data: No test data needed

Test Steps:

1. Navigate to the subtask that should be completed first.
2. Access the dependency settings for this subtask.
3. Select the other subtask as dependent on the completion of the first.
4. Attempt to mark the dependent subtask as complete before the first subtask.

Postconditions:

- The system should prevent marking the dependent subtask as complete until the first subtask is marked complete.

Expected Result: The system should successfully establish and enforce the dependency relationship between the subtasks.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 26

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users

should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Dependencies

Feature description -

Users can define dependencies between tasks, specifying the order in which tasks should be completed to ensure smooth workflow and project progression. Task dependencies help in identifying critical paths and managing interdependencies between tasks.

Userstory title -

As a user, I want to set task dependencies so that tasks are completed in the correct sequence.

Acceptance criteria -

['Users can set dependencies between tasks.', 'The system displays dependencies in the task details.', 'Tasks cannot be marked as complete if dependent tasks are not completed.', 'Users can update or remove dependencies as needed.', 'The system provides alerts or warnings if a dependent task is not completed.']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Dependencies Feature

Test Case 1:

Test Case ID: TMT-TD-001

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

Test Case Description: This test case verifies that users can successfully establish dependencies between different tasks within a project.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in.
- A project with at least two tasks exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the project containing the tasks.
2. Open the details of a task (Task A).
3. Locate the "Dependencies" section or option.
4. Select another task (Task B) from the list of available tasks to set as a dependency for Task A.
5. Save the dependency.

Postconditions:

- Task A now shows Task B as a dependency.

Expected Result: The system should successfully link the two tasks, indicating that Task A is dependent on Task B.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TD-002

Test Case Title: Verify that the system displays dependencies in the task details

Test Case Description: This test case ensures that any established dependencies are clearly displayed within the task details view.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in.

- Two tasks with a defined dependency exist (Task A dependent on Task B).

Test Data: No test data needed.

Test Steps:

1. Navigate to the project containing the dependent tasks.
2. Open the details view of Task A (the dependent task).

Postconditions: N/A

Expected Result: The system should clearly display that Task A is dependent on Task B within its details view. This could be through a visual indicator, a list of dependencies, or a clear statement.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TD-003

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

Test Case Description: This test case ensures that a task cannot be marked as "complete" if it has any uncompleted dependent tasks.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in.
- Two tasks with a defined dependency exist (Task A dependent on Task B).
- Task B is not marked as complete.

Test Data: No test data needed.

Test Steps:

1. Navigate to the project containing the dependent tasks.

2. Attempt to mark Task A (the dependent task) as complete.

Postconditions: N/A

Expected Result: The system should prevent the user from marking Task A as complete. An error message or a notification should be displayed, clearly stating that the action cannot be completed because Task A is dependent on the incomplete Task B.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case 4:

Test Case ID: TMT-TD-004

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

Test Case Description: This test case ensures users have the flexibility to modify or remove existing task dependencies.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

- User is logged in.
- Two tasks with a defined dependency exist (Task A dependent on Task B).

Test Data: No test data needed.

Test Steps:

1. Navigate to the project containing the dependent tasks.
2. Open the details of Task A.
3. Locate the "Dependencies" section.
4. Modify the existing dependency (e.g., change Task B to another task) or remove the dependency entirely.
5. Save the changes.

Postconditions:

- Task A's dependencies are updated.

Expected Result: The system should successfully update the task dependency information, reflecting the changes made by the user. This includes adding new dependencies, changing existing ones, or removing them entirely.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TD-005

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

Test Case Description: This test case checks if the system generates timely alerts or warnings to the user if a task with approaching deadlines has an uncompleted dependent task.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

- User is logged in.
- Two tasks with a defined dependency exist (Task A dependent on Task B).
- Task A has an approaching deadline.
- Task B is not marked as complete.

Test Data: No test data needed.

Test Steps:

1. Wait until the system is within the notification timeframe for Task A's deadline.

Postconditions: N/A

Expected Result: The system should generate an alert or warning, notifying the user that Task A has an approaching deadline and is dependent on the uncompleted Task B. The notification should be clear, informative, and delivered through a defined channel like email, in-app notifications, or both.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

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 Cases for Task Comments and Collaboration:

Test Case 1:

Test Case ID: TC_TCC_001

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

Test Case Description: This test case verifies that users can successfully add comments to existing tasks within the task management tool. It will cover adding simple text comments to ensure basic functionality is working as expected.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- A task exists for the user.

Test Data: No test data needed.

Test Steps:

1. Navigate to an existing task.
2. Locate the comments section.
3. Enter a text comment in the comment box.
4. Click the "Add Comment" or equivalent button.

Postconditions:

- The comment is displayed under the task's comment section.

Expected Result: The system should successfully add the comment to the task, making it visible to the user and other collaborators.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_TCC_002

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

Test Case Description: This test case ensures users can include various content types in their comments, including text, file attachments, and hyperlinks.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- A task exists for the user.

Test Data:

- Sample text comment.
- A file to attach (e.g., .docx, .pdf, .png).
- A valid hyperlink.

Test Steps:

1. Navigate to an existing task.
2. In the comment box, enter a text comment.
3. Click the "Attach File" button and upload the prepared file.
4. Paste the valid hyperlink into the comment box.
5. Click the "Add Comment" button.

Postconditions:

- The comment displays the text, a downloadable link for the attached file, and a clickable hyperlink.

Expected Result: The system should successfully include all three content types (text, attachment, link) within the added comment.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_TCC_003

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

Test Case Description: This test case verifies the comment threading functionality and the ability for users to modify their own comments after posting.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is logged in.
- A task exists with at least one comment from the user.

Test Data: No test data needed.

Test Steps:

1. Navigate to a task with an existing comment from the user.
2. Click the "Reply" button under the user's comment.
3. Enter a reply comment and click "Add Comment."
4. Click the "Edit" button on the user's original comment.
5. Modify the comment text and click "Save" or "Update."

Postconditions:

- The reply comment is displayed as a threaded response under the original comment.
- The original comment reflects the edited changes.

Expected Result: The system should successfully post the reply comment and save the edits made to the original comment.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_TCC_004

Test Case Title: Verify that the system notifies users of new comments.

Test Case Description: This test case checks that appropriate notifications are sent to users when new comments are added to tasks they are collaborating on.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- Two users are logged in (User A and User B).
- Both users are collaborators on the same task.

Test Data: No test data needed.

Test Steps:

1. User A navigates to the shared task.
2. User A adds a new comment to the task.
3. Observe if User B receives a notification about the new comment.

Postconditions: None.

Expected Result: User B should receive a notification (in-app or via email, depending on system settings) indicating that a new comment has been added to the shared task.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_TCC_005

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

Test Case Description: This test case ensures that all comments within a task are displayed chronologically, with the oldest comment at the top and the newest at the bottom.

Test Suite: Task Comments and Collaboration

Test Priority: Low

Preconditions:

- User is logged in.
- A task exists with multiple comments from various users and timestamps.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task with multiple comments.
2. Observe the order in which the comments are displayed.

Postconditions: None.

Expected Result: Comments should be displayed from oldest to newest, maintaining chronological order based on their timestamps.

Severity: Minor

Type of Testing: UI Testing

Test Case Approach: Positive

Data - 28

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

File Attachments

Feature description -

Users can attach relevant files, documents, and resources to tasks, ensuring that all necessary information is accessible in one place. File attachments can include documents, spreadsheets, images, links, and other resources needed for task execution.

Userstory title -

As a user, I want to attach files to tasks so that all relevant information is easily accessible.

Acceptance criteria -

['Users can attach multiple files to a task.', 'Attachments can include various file types (documents, images, etc.).', 'Users can view and download attachments.', 'The system limits the size of attachments to prevent data overload.', 'Users can remove attachments if needed.']

Generated Test Cases:

## Test Cases for Task Management Tool - File Attachments

Test Case 1:

Test Case ID: TMT-FA-001

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

Test Case Description: This test case verifies that users can successfully attach more than one file to a single task, ensuring all relevant information can be stored together.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in.
- A task has been created.

Test Data:

- Multiple files of varying types (e.g., document, image, spreadsheet).

Test Steps:

1. Navigate to the created task.
2. Access the file attachment option.
3. Select and attach the first file.
4. Repeat steps 2-3 for each additional file.
5. Verify all files are listed as attachments to the task.

Postconditions:

- All selected files are successfully attached to the task.

Expected Result: The system allows attaching multiple files to a single task without error.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-FA-002

Test Case Title: Verify that attachments can include various file types.

Test Case Description: This test case aims to confirm the system's ability to handle various file types as attachments, ensuring flexibility for users.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in.

- A task has been created.

Test Data:

- A selection of files including, but not limited to: .doc, .docx, .xls, .xlsx, .pdf, .jpg, .png, .gif, .zip

Test Steps:

1. Navigate to the created task.

2. Access the file attachment option.

3. Attempt to attach each file type from the test data.

4. Verify the attachment is successful for each file type.

Postconditions:

- All tested file types are successfully attached to the task.

Expected Result: The system accepts and attaches all tested file types without error, confirming compatibility.

Severity: Major

Type of Testing: Functional Testing, Compatibility Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-FA-003

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

Test Case Description: This test ensures users can access and download attached files, validating the core functionality of the feature.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in.
- A task with attached files exists.

Test Data:

- No test data needed.

Test Steps:

1. Navigate to the task with attached files.
2. Locate the list of attached files.
3. Click to view a selected file.
4. Verify the file opens correctly within the application or in a new tab.
5. Download the same file.
6. Verify the downloaded file is accessible and undamaged.

Postconditions:

- The selected file is successfully viewed and downloaded.

Expected Result: The system allows seamless viewing and downloading of attached files without corruption or errors.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-FA-004

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

Test Case Description: This test case checks the system's ability to restrict oversized attachments, preventing potential data overload and performance issues.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in.
- A task has been created.
- The maximum attachment size limit is known.

Test Data:

- One file exceeding the defined size limit.

Test Steps:

1. Navigate to the created task.
2. Access the file attachment option.
3. Attempt to attach the oversized file.

Postconditions:

- The file attachment is blocked.

Expected Result: The system prevents the upload of the oversized file and displays a clear error message indicating the size limit.

Severity: Major

Type of Testing: Functional Testing, Performance Testing

Test Case Approach: Negative

Test Case 5:

Test Case ID: TMT-FA-005

Test Case Title: Verify that users can remove attachments.

Test Case Description: This test case ensures users can delete unwanted or outdated files, maintaining relevance and order within the task's information.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

- User is logged in.
- A task with attached files exists.

Test Data:

- No test data needed.

Test Steps:

1. Navigate to the task with attached files.
2. Locate the list of attached files.
3. Select the option to remove a specific file.
4. Confirm the removal action if prompted.
5. Verify the file is no longer listed as an attachment.

Postconditions:

- The selected file is successfully removed from the task.

Expected Result: The system allows users to remove unwanted attachments without impacting other files or task data.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 29

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Recurring Tasks

Feature description -

Users can set tasks to repeat at specified intervals, automating the creation of recurring tasks such as weekly meetings, monthly reports, or quarterly reviews. Recurring tasks save time and effort by eliminating the need to create them manually for each occurrence.

Userstory title -

As a user, I want to set recurring tasks so that regular activities are automated.

Acceptance criteria -

['Users can set tasks to recur at specified intervals (daily, weekly, monthly).', 'The system generates instances of recurring tasks automatically.', 'Users can modify the recurrence pattern or cancel the recurrence.', 'The system notifies users of upcoming instances of recurring tasks.', 'Users can view a list of all instances of a recurring task.']

Generated Test Cases:

## Test Cases for Recurring Tasks Feature:

Test Case 1:

Test Case ID: RecurringTask-001

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

Test Case Description: This test case verifies that users can successfully set a task to recur daily, weekly, or monthly.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- User is logged in.

- User has access to create tasks.

Test Data:

- Task Name: "Team Meeting"

- Task Description: "Weekly team meeting to discuss progress and blockers."

Test Steps:

1. Navigate to the task creation page.

2. Fill in the task name and description.

3. Locate the "Recurring Task" option.

4. Select a recurrence interval (Daily, Weekly, Monthly).

5. Set the desired start date for the recurrence.

6. Save the task.

Postconditions:

- The recurring task is created and saved successfully.

Expected Result: The system should save the task and schedule it to recur at the specified interval (daily, weekly, or monthly).

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: RecurringTask-002

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

Test Case Description: This test case ensures that the system automatically generates new instances of recurring tasks based on the defined recurrence pattern.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- A recurring task is set up with a future start date.

Test Data: No test data needed.

Test Steps:

1. Wait until the recurring task's start date is reached.

2. Navigate to the task list view.

Postconditions:

- None.

Expected Result: A new instance of the recurring task should appear on the task list on the specified start date and subsequently based on the recurrence pattern.

Severity: Critical

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: RecurringTask-003

Test Case Title: Verify that users can modify the recurrence pattern or cancel the recurrence of a task.

Test Case Description: This test case ensures users can edit an existing recurring task to modify its recurrence pattern or cancel the recurrence altogether.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- A recurring task is already created.

Test Data:

- Modified recurrence interval: Change from "Weekly" to "Monthly."

Test Steps:

1. Navigate to the existing recurring task.

2. Access the task's edit options.

3. Locate the "Recurring Task" settings.

4. Modify the recurrence interval or select the option to cancel recurrence.

5. Save the changes to the task.

Postconditions:

- The recurring task should be updated with the modified settings.

Expected Result: The system should successfully save the changes to the recurring task, either updating the recurrence pattern or canceling it, and the future instances should reflect these changes.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: RecurringTask-004

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

Test Case Description: This test case verifies that the system generates and sends appropriate notifications to users, reminding them of upcoming instances of their recurring tasks.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

- A recurring task is scheduled with notifications enabled.

- The user has enabled notifications for new tasks.

Test Data: No test data needed.

Test Steps:

1. Wait for the notification trigger time based on the task's recurrence and notification settings.

2. Verify the user receives a notification.

Postconditions:

- None.

Expected Result: The system should send a timely notification to the user, reminding them of the upcoming recurring task instance. The notification should contain relevant task details.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: RecurringTask-005

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

Test Case Description: This test case ensures that users can access a comprehensive list of all past, current, and future instances of a specific recurring task, allowing them to track its history and plan accordingly.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

- A recurring task with multiple past and future instances exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the details page of the recurring task.

2. Look for a section or tab labeled "Instances," "History," or a similar term.

Postconditions:

- None.

Expected Result: The system should display a list of all instances of the recurring task, including their respective due dates, completion status, and potentially links to access each instance individually.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 30

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Tags and Labels

Feature description -

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

Userstory title -

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

Acceptance criteria -

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Tags and Labels:

Test Case 1:

* **Test Case ID:** TMTL-001
* **Test Case Title:** Verify that users can add tags and labels to tasks.
* **Test Case Description:** This test case verifies that users can successfully add multiple tags and labels to a newly created task within the Task Management Tool.
* **Test Suite:** Task Tags and Labels
* **Test Priority:** High
* **Preconditions:**
    * User is logged in.
    * User has created a new task.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Navigate to the newly created task.
    2. Locate the "Add Tags/Labels" field.
    3. Enter multiple tags and labels, separated by commas or a designated delimiter.
    4. Save the task.
* **Postconditions:**
    * Tags and labels are saved and associated with the task.
* **Expected Result:** The system should successfully save all entered tags and labels to the task, displaying them clearly within the task view.
* **Severity:** Major
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TMTL-002
* **Test Case Title:** Verify that tags can be customized and edited.
* **Test Case Description:** This test case verifies the functionality of customizing and editing existing tags associated with a task, including renaming, deleting, or modifying the tag's appearance.
* **Test Suite:** Task Tags and Labels
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged in.
    * A task with at least one tag exists.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Navigate to a task with an existing tag.
    2. Attempt to edit the tag, including:
        * Renaming the tag.
        * Deleting the tag.
        * Changing the tag's color or appearance (if applicable).
    3. Save the changes to the task.
* **Postconditions:**
    * The tag is updated accordingly, or removed, within the task view.
* **Expected Result:** The system should allow for editing and customizing existing tags. Changes should be reflected accurately in the task view.
* **Severity:** Minor
* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TMTL-003
* **Test Case Title:** Verify that tasks can be filtered and sorted based on tags and labels.
* **Test Case Description:** This test verifies the filtering and sorting functionality of the system, ensuring users can locate tasks based on specific tags or labels and arrange them in a desired order.
* **Test Suite:** Task Tags and Labels
* **Test Priority:** High
* **Preconditions:**
    * User is logged in.
    * Multiple tasks with different tags and labels exist.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Apply a filter using a specific tag or label.
    2. Verify that only tasks associated with that tag are displayed.
    3. Clear the filter and apply sorting based on tags or labels.
    4. Verify that tasks are sorted correctly according to the selected criteria.
* **Postconditions:**
    * The task list accurately reflects the applied filters and sorting criteria.
* **Expected Result:** The system should correctly filter and sort tasks based on the selected tags and labels, providing an efficient way to organize and locate specific tasks.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TMTL-004

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

* **Test Case Description:** This test case ensures that tags and labels are easily visible within the task list view, facilitating quick identification and categorization of tasks.

* **Test Suite:** Task Tags and Labels

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in.

    * Tasks with various tags and labels are present.

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

* **Test Steps:**

    1. Navigate to the main task list view.

    2. Observe the visual presentation of tags and labels associated with each task.

* **Postconditions:** N/A

* **Expected Result:** Tags and labels should be displayed prominently within the task list, using clear visual cues like distinct colors, fonts, or placement to ensure easy readability and identification.

* **Severity:** Minor

* **Type of Testing:** Usability Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TMTL-005

* **Test Case Title:** Verify that users can remove tags and labels if needed.

* **Test Case Description:** This test case ensures the functionality of removing tags and labels from tasks is

working correctly without impacting other task details.

* **Test Suite:** Task Tags and Labels

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in.

    * A task with at least one tag or label exists.

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

* **Test Steps:**

    1. Navigate to a task with an existing tag or label.

    2. Attempt to remove the tag/label using the provided interface (e.g., clicking an 'x' button, selecting 'remove' from a menu).

    3. Save the changes to the task.

* **Postconditions:**

    * The tag/label is removed from the task view.

* **Expected Result:** The system should successfully remove the designated tag or label from the task without affecting other task information. The change should be immediately reflected in the task view.

* **Severity:** Minor

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 31

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users

should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Status Tracking

Feature description -

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task status updates provide visibility into task lifecycle and project progression, helping users monitor task progress and identify bottlenecks.

Userstory title -

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

Acceptance criteria -

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides notifications for status changes.', 'Task status can be filtered and sorted.']

Generated Test Cases:

## Test Cases for Task Management Tool: Task Status Tracking

Test Case 1:

Test Case ID: TMT-TST-001

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

Test Case Description: This test case aims to ensure users can change the status of their tasks to "To Do," "In Progress," or "Completed." It will validate the functionality of the status update mechanism and its impact on task display.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in.
- A task is created.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.
2. Select a task.
3. Locate the task status dropdown or edit option.
4. Change the task status to "To Do," "In Progress," and "Completed" sequentially, verifying the change after each step.

Postconditions:

- Task status is updated in the database.

Expected Result: The system should successfully update the task status and reflect the change in the task list view.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TST-002

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

Test Case Description: This test case checks the visual representation of the task status within the task list. It ensures the status is displayed clearly and is easily identifiable for users to quickly assess task progress.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in.

- Tasks with different statuses ("To Do," "In Progress," "Completed") exist.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.

2. Observe the display of tasks.

3. Verify that each task clearly shows its current status.

Postconditions: None

Expected Result: The task status should be visually prominent, using clear indicators like color-coding, labels, or dedicated columns within the task list.

Severity: Major

Type of Testing: UI Testing, Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TST-003

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

Test Case Description: This test aims to confirm users can see a visual representation of task progress. It will involve checking for a progress bar or percentage display reflecting the completion status of individual tasks.

Test Suite: Task Status Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

- Tasks with defined progress percentages exist (e.g., 25%, 50%, 75%).

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.

2. Locate a task with a defined progress percentage.

3. Verify that the progress percentage is visually displayed alongside the task.

Postconditions: None

Expected Result: The system should accurately display the progress percentage for each task, potentially using a progress bar or a numerical percentage value.

Severity: Minor

Type of Testing: UI Testing, Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-TST-004

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

Test Case Description: This test case ensures users receive notifications when a task's status is modified. It will involve triggering a status change and confirming the generation and delivery of the notification to the relevant user.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in.

- Notifications are enabled for the user.

Test Data: No test data needed.

Test Steps:

1. Navigate to a task.

2. Change the task status.

3. Check for a notification indicating the status change.

4. Verify the notification content includes relevant details about the task and the status update.

Postconditions: None

Expected Result: The system should promptly send a notification to the user upon a task status change. The notification should be clear, informative, and delivered through the user's preferred notification channel.

Severity: Major

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TST-005

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

Test Case Description: This test confirms users can efficiently manage their task view by filtering and sorting tasks based on their status. It involves testing the filtering and sorting mechanisms within the task list.

Test Suite: Task Status Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

- Tasks with different statuses exist.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.

2. Locate the filtering options.

3. Filter tasks by different statuses (e.g., "To Do," "In Progress," "Completed") and verify correct display.

4. Locate the sorting options.

5. Sort tasks based on status and verify the correct order.

Postconditions: None

Expected Result: The system should accurately filter and sort the task list based on the selected status, providing users with a customized view that aids in task management.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

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 Cases for Task Management Tool - Task Filtering and Sorting

Test Case 1

Test Case ID: TMS-FFS-001

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

Test Case Description: This test case verifies that the task filtering functionality correctly filters tasks based on the selected due date range.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

* User is logged in.
* At least three tasks with different due dates exist.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list view.
2. Select the 'Due Date' filter option.
3. Choose a specific date range (e.g., "Today", "Next week", "Custom range").
4. Apply the filter.

Postconditions: None

Expected Result: The system displays only tasks whose due dates fall within the selected range.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TMS-FFS-002

Test Case Title: Verify that tasks can be sorted by priority.

Test Case Description: This test case validates that the task sorting functionality accurately sorts tasks based on their assigned priority levels.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

* User is logged in.

* At least three tasks with different priority levels (e.g., High, Medium, Low) exist.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list view.

2. Select the 'Priority' sorting option.

3. Choose either ascending or descending order.

Postconditions: None

Expected Result: The system displays the tasks sorted according to their priority level in the selected order.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TMS-FFS-003

Test Case Title: Verify that the system provides multiple filter and sort options in the task list.

Test Case Description: This test case ensures the availability of a comprehensive set of filter and sort options within the task list interface.

Test Suite: Task Filtering and Sorting

Test Priority: Medium

Preconditions:

* User is logged in.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list view.

2. Open the filter options.

3. Verify that filter options for 'Due Date', 'Priority', 'Assignee', and 'Project' are present.

4. Open the sorting options.

5. Verify that sorting options for 'Due Date', 'Priority', 'Assignee', 'Project', and 'Creation Date' are present.

Postconditions: None

Expected Result: The system displays the expected filter and sort options, indicating a comprehensive set of tools for managing task views.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TMS-FFS-004

Test Case Title: Verify that users can save and apply custom filter and sort settings.

Test Case Description: This test case ensures that users can create, save, and subsequently apply personalized filter and sort configurations for customized task views.

Test Suite: Task Filtering and Sorting

Test Priority: Medium

Preconditions:

* User is logged in.

* At least three tasks exist with varying attributes (due dates, priority, assignee, project).

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list view.

2. Apply a combination of filters and sorting (e.g., filter by "High" priority and sort by "Due Date" ascending).

3. Save the current filter and sort configuration with a custom name (e.g., "High Priority Tasks").

4. Clear the existing filters and sorting.

5. Apply the saved custom filter and sort configuration "High Priority Tasks".

Postconditions: None

Expected Result: The task list displays the tasks reflecting the saved custom filter and sort settings, demonstrating successful saving and application of personalized configurations.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TMS-FFS-005

Test Case Title: Verify that filters and sorts do not affect the actual task data.

Test Case Description: This test case confirms that applying filters and sorting only modifies the presentation of tasks and doesn't alter the underlying task information.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

* User is logged in.

* At least three tasks exist with varying attributes.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list view.

2. Note the details (due date, priority, assignee, project) of a specific task.

3. Apply a combination of filters and sorting.

4. Locate the same task observed in step 2.

5. Compare the task details from step 2 and step 4.

6. Clear the applied filters and sorting.

7. Again, locate the same task and verify its details.

Postconditions: None

Expected Result: The details of the specific task should remain consistent throughout the test, regardless of the applied filters and sorting, confirming that only the task view is altered, not the actual data.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 33

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users

should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Time Tracking

Feature description -

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

Userstory title -

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Acceptance criteria -

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

Generated Test Cases:

## Test Cases for Task Management Tool - Time Tracking Feature

Here are 5 test cases designed to cover the acceptance criteria for the Time Tracking feature:

Test Case 1:

* **Test Case ID:** TMT-TT-001
* **Test Case Title:** Verify that users can start and stop a timer for each task.
* **Test Case Description:** This test case verifies the core functionality of the time tracking feature by ensuring users can accurately start and stop a timer for individual tasks.
* **Test Suite:** Time Tracking

* **Test Priority:** High
* **Preconditions:**
    * User is logged in.
    * A task has been created.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Navigate to the task list.
    2. Select a task.
    3. Start the timer for the selected task.
    4. Perform some work on the task.
    5. Stop the timer.
* **Postconditions:**
    * The timer should stop.
* **Expected Result:** The system should accurately record the elapsed time between starting and stopping the timer for the selected task.
* **Severity:** Critical
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TMT-TT-002
* **Test Case Title:** Verify that the system records and displays the total time spent on tasks.
* **Test Case Description:** This test case verifies that the system accurately tracks and displays the cumulative time spent on individual tasks, even across multiple timer sessions.
* **Test Suite:** Time Tracking
* **Test Priority:** High

* **Preconditions:**

    * User is logged in.

    * A task exists with recorded time (either via timer or manual entry).

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

* **Test Steps:**

    1. Navigate to the task list.

    2. Select a task with recorded time.

    3. View the task details.

* **Postconditions:** None

* **Expected Result:** The system should clearly display the total time spent on the selected task, including time from all previous timer sessions or manual entries.

* **Severity:** Critical

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TMT-TT-003

* **Test Case Title:** Verify that users can manually enter or adjust time entries.

* **Test Case Description:** This test case confirms that users can manually add or modify time entries for tasks, providing flexibility for situations where the timer wasn't used or needs correction.

* **Test Suite:** Time Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in.

    * A task has been created.

* **Test Data:** Time value (e.g., 1.5 hours)

* **Test Steps:**

    1. Navigate to the task list.

    2. Select a task.

    3. Manually enter a time value for the task (e.g., add 1.5 hours).

    4. Save the changes.

    5. Optionally, edit the previously entered time value.

* **Postconditions:**

    * The manually entered time should be saved.

* **Expected Result:** The system should accurately record and display the manually entered or adjusted time for the selected task.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TMT-TT-004

* **Test Case Title:** Verify that time tracking data is stored and can be exported.

* **Test Case Description:** This test case ensures that the system retains accurate time tracking data and allows users to export this information, potentially for reporting or integration with other systems.

* **Test Suite:** Time Tracking

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in.

    * Tasks exist with recorded time.

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

* **Test Steps:**

    1. Navigate to the Reports or Time Tracking section.

    2. Locate the export functionality.

    3. Choose a format for export (e.g., CSV, Excel).

    4. Initiate the export.

* **Postconditions:** None.

* **Expected Result:** The system should generate an export file in the chosen format containing accurate time tracking data for the selected tasks and/or projects.

* **Severity:** Major

* **Type of Testing:** Functional Testing, Data Validation

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** TMT-TT-005

* **Test Case Title:** Verify that the system provides reports on time spent per task or project.

* **Test Case Description:** This test case checks the reporting capability of the time tracking feature, ensuring that users can generate informative reports on time allocation across tasks and projects.

* **Test Suite:** Time Tracking, Reporting

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in.

    * Tasks and projects exist with recorded time.

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

* **Test Steps:**

    1. Navigate to the Reports section.

    2. Choose a report type related to time tracking (e.g., "Time Spent per Task", "Time Spent per Project").

    3. Select filters or criteria for the report (e.g., date range, specific project).

4. Generate the report.

* **Postconditions:** None.

* **Expected Result:** The system should generate a report displaying accurate and organized information about time spent on tasks or projects, based on the selected criteria. The report should include relevant details like task names, project names, total time, and potentially user information if applicable.

* **Severity:** Major

* **Type of Testing:** Functional Testing, Reporting

* **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 Cases for Task Management Tool - Task Templates:

Test Case 1:

Test Case ID: TMT-TT-001

Test Case Title: Verify that users can create and save task templates.

Test Case Description: This test case verifies that users can successfully create and save new task templates within the Task Management Tool. It will cover different scenarios of inputting data into the task template creation form.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in.

Test Data:

- Template Name: "Website Launch Checklist"
- Description: "A comprehensive checklist for launching a new website."
- Tasks:
  - [ ] Design Approval
  - [ ] Content Upload
  - [ ] SEO Optimization
  - [ ] Domain Connection
- Assigned to: Marketing Team

- Due Date: + 14 days from creation date

Test Steps:

1. Navigate to the "Task Templates" section.
2. Click on the "Create New Template" button.
3. Input the Test Data into the respective fields.
4. Click on the "Save Template" button.

Postconditions:

- The newly created task template should appear in the list of available templates.

Expected Result: The system should successfully create and save the new task template with the provided data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TT-002

Test Case Title: Verify that templates can include predefined task structures, descriptions, assignees, and due dates.

Test Case Description: This test case verifies that the created task template accurately reflects all the predefined data set during the template creation process.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in.
- A task template has been created using the data from Test Case 1.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Task Templates" section.

2. Select the previously created template ("Website Launch Checklist").

Postconditions: N/A

Expected Result: The system should display the template with all predefined data accurately populated: Name, Description, Tasks, Assigned Team, and Due Date.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TT-003

Test Case Title: Verify that users can apply templates to new tasks or projects.

Test Case Description: This test case verifies the functionality of applying an existing task template to a new task, checking if the predefined template data is correctly transferred.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in.

- A task template has been created.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Tasks" or "Projects" section.

2. Click on the "Create New Task/Project" button.

3. Select the "Use Template" option.

4. Choose the desired template from the list.

5. Confirm template application.

Postconditions: N/A

Expected Result: The system should create a new task/project with all predefined data from the selected template applied.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-TT-004

Test Case Title: Verify that templates can be edited or deleted as needed.

Test Case Description: This test case verifies that users have the ability to edit or delete existing task templates, ensuring flexibility in managing templates.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- User is logged in.
- At least one task template exists.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Task Templates" section.
2. Locate an existing template.
3. Attempt to edit the template's name and description.
4. Save the changes.
5. Attempt to delete the template.
6. Confirm the deletion.

Postconditions: N/A

Expected Result:

- Edits to the template should be saved correctly.
- The system should permanently delete the template upon confirmation.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TT-005

Test Case Title: Verify that the system displays a list of available templates for selection.

Test Case Description: This test case ensures that the system correctly displays a comprehensive list of all available task templates when a user attempts to apply a template to a new task or project.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- User is logged in.
- Multiple task templates have been created.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Tasks" or "Projects" section.
2. Click on the "Create New Task/Project" button.
3. Select the "Use Template" option.

Postconditions: N/A

Expected Result: The system should display a list containing all available task templates, allowing users to easily choose and apply the desired template.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

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 Cases for Task Management Tool - Calendar Integration:

Test Case 1:

Test Case ID: TMT-CI-001

Test Case Title: Verify that users can sync tasks with external calendars.

Test Case Description: This test case verifies that users can successfully connect their Task Management Tool account with their external calendar accounts (e.g., Google Calendar, Outlook) and synchronize their tasks.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.
- User has an active external calendar account (Google Calendar, Outlook, etc.).

Test Data: Valid external calendar credentials.

Test Steps:

1. Navigate to the user settings page within the Task Management Tool.
2. Locate the "Calendar Integration" section.
3. Choose the desired external calendar provider (e.g., Google Calendar).
4. Enter the external calendar account credentials and authorize the connection.
5. Verify that the connection is successful.

Postconditions:

- The user's external calendar is linked to the Task Management Tool.

Expected Result: The system successfully connects to the user?s chosen external calendar account, allowing for task synchronization.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-CI-002

Test Case Title: Verify that the system displays task deadlines in the calendar view.

Test Case Description: This test case ensures that once a user synchronizes their tasks, the deadlines for those tasks are accurately reflected within the integrated calendar view alongside other scheduled events.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.
- User has successfully synced their external calendar.
- User has tasks with set deadlines in the Task Management Tool.

Test Data: No test data needed.

Test Steps:

1. Navigate to the calendar view within the Task Management Tool.
2. Verify that tasks with deadlines are displayed on the calendar on their respective due dates.
3. Check that the task details (title, time) are visible on the calendar entry.

Postconditions: None

Expected Result: The system accurately displays all task deadlines on the calendar view, visually distinguishing them from other calendar events.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-CI-003

Test Case Title: Verify that users can set reminders for tasks via the calendar.

Test Case Description: This test case confirms that users can set up reminders for their tasks directly from the integrated calendar view, ensuring timely notifications.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- User is logged into the Task Management Tool.
- User has successfully synced their external calendar.
- A task with a deadline is visible on the calendar view.

Test Data: No test data needed.

Test Steps:

1. On the calendar view, select a task with a set deadline.
2. Access the task details or options.
3. Choose to set a reminder for the task.
4. Configure the reminder time and frequency (e.g., 1 hour before, 1 day before).
5. Save the reminder settings.

Postconditions: The selected task should now have a configured reminder within the Task Management Tool and, if supported by the external calendar, within the external calendar application as well.

Expected Result: The system allows users to set, modify, and remove reminders for tasks directly from the calendar interface, sending notifications at the specified times.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-CI-004

Test Case Title: Verify that calendar events update automatically when tasks are changed.

Test Case Description: This test case ensures that any changes made to a task's details (deadline, title, etc.) within the Task Management Tool are automatically reflected in the integrated calendar view and the external calendar application.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged into the Task Management Tool.
- User has successfully synced their external calendar.
- A task with a deadline is present in both the Task Management Tool and the synced calendar view.

Test Data: Modified task details (e.g., new deadline, updated title).

Test Steps:

1. Within the Task Management Tool, locate and select a task that is synced with the calendar.
2. Modify the task details, such as changing the deadline or updating the task's title.
3. Save the changes to the task.
4. Navigate to the calendar view within the Task Management Tool.
5. Verify that the calendar entry for the modified task reflects the updated information.

Postconditions: None.

Expected Result: The system automatically updates the calendar entry in both the Task Management Tool and the external calendar to reflect the changes made to the task, ensuring consistency.

Severity: Critical

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-CI-005

Test Case Title: Verify that users can manage calendar sync settings within the app.

Test Case Description: This test case verifies that users have the ability to control and modify various settings related to calendar synchronization directly within the Task Management Tool's settings.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- User is logged into the Task Management Tool.
- User has successfully synced their external calendar.

Test Data: No test data needed.

Test Steps:

1. Go to the user settings page within the Task Management Tool.
2. Locate the "Calendar Integration" section.
3. Verify that users can enable or disable the calendar sync feature.
4. Check if users can select which calendar events (e.g., tasks only, all events) are synchronized.
5. Confirm users can disconnect or relink their external calendar account.

Postconditions: None.

Expected Result: The Task Management Tool provides users with a dedicated section within the settings to manage their calendar sync preferences, allowing for customization and control over the integration.

Severity: Minor

Type of Testing: Usability 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 Cases for Task Duplication Feature:

Test Case 1:

Test Case ID: TD-001

Test Case Title: Verify that users can duplicate tasks.

Test Case Description: This test case verifies that a user can successfully duplicate an existing task within the task management tool.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in.
- At least one task exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.
2. Select a task to duplicate.
3. Click the "Duplicate Task" option (specific implementation may vary).
4. Confirm duplication if prompted.

Postconditions:

- A new task, identical to the original, appears in the task list.

Expected Result: The system successfully creates a duplicate of the selected task.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TD-002

Test Case Title: Verify that duplicated tasks retain all original details.

Test Case Description: This test case ensures that all details from the original task, including description, assignees, and deadlines, are accurately copied to the duplicated task.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in.

- A task with a description, assignee(s), and a deadline exists.

Test Data: No test data needed.

Test Steps:

1. Duplicate the pre-existing task (following steps from TD-001).

2. Open the duplicated task.

3. Verify that the description, assignee(s), and deadline match the original task.

Postconditions: None

Expected Result: All details from the original task are accurately replicated in the duplicated task.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TD-003

Test Case Title: Verify that users can modify duplicated tasks.

Test Case Description: This test case confirms that users can edit and modify the details of a duplicated task independently of the original task.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in.

- A duplicated task exists.

Test Data: Modified task details (e.g., new description, different assignee, adjusted deadline).

Test Steps:

1. Open the duplicated task.
2. Modify the task details (description, assignee, deadline, etc.).
3. Save the changes to the duplicated task.
4. Open the original task.

Postconditions: None

Expected Result: The duplicated task is updated with the modifications, while the original task remains unchanged.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TD-004

Test Case Title: Verify that the system indicates duplicated tasks.

Test Case Description: This test case validates that the system provides a clear visual indicator to differentiate duplicated tasks from original tasks.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

- User is logged in.
- A task has been duplicated.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.
2. Locate the duplicated task.
3. Observe if there is a visual indicator (e.g., "(Copy)" in the task title, a specific icon) that distinguishes the

duplicated task.

Postconditions: None

Expected Result: The system clearly indicates which tasks have been duplicated, using a distinct visual cue.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TD-005

Test Case Title: Verify that duplicated tasks are saved and displayed in the task list.

Test Case Description: This test case confirms that duplicated tasks are properly saved by the system and consistently displayed within the user's task list, even after logging out and back in.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in.

Test Data: No test data needed.

Test Steps:

1. Duplicate a task (following the steps from TD-001).
2. Log out of the system.
3. Log back into the system.
4. Navigate to the task list.

Postconditions: None

Expected Result: The duplicated task is persistent and visible in the user's task list.

Severity: Critical

Type of Testing: Functional Testing

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 Cases for Task Management Tool - Task History Feature

Test Case 1:

Test Case ID: TMT-TH-001

Test Case Title: Verify that users can view a detailed history of changes to tasks.

Test Case Description: This test case ensures that users can access and view a comprehensive history of changes made to a specific task, including details of each modification.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in.
- A task exists with at least one modification made to it (e.g., title change, deadline update, status change).

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list.
2. Select a task that has been previously modified.
3. Access the "Task History" section or tab within the task details view.
4. Verify that the task history is displayed, showing a chronological list of changes made to the task.
5. Verify that each history entry includes details such as:
    - Type of modification (e.g., title changed, deadline updated, status updated, comment added).
    - Date and timestamp of the modification.
    - User who made the modification.

Postconditions: None

Expected Result: The task history should be displayed correctly, detailing all modifications made to the task.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TH-002

Test Case Title: Verify that the task history includes modifications, comments, and other relevant activity.

Test Case Description: This test case validates that the task history captures and displays a comprehensive range of activities, including edits, comments, and status changes, ensuring a complete audit trail.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in.
- A task exists with various modifications, including comments and status changes.

Test Data:

- A comment added to the task.
- A change in the task status (e.g., "To Do" to "In Progress").

Test Steps:

1. Navigate to the task with the pre-added modifications.
2. Access the "Task History" section.
3. Verify that the history includes entries for each of the following activities:
    - Original task creation.
    - Modification to the task title or description.
    - Addition of the pre-added comment.
    - Change in the task status.

Postconditions: None

Expected Result: The task history should accurately reflect all modifications, comments, and status changes made to the task.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TH-003

Test Case Title: Verify that users can filter the history by type of change or date.

Test Case Description: This test case checks the filtering functionality of the task history, allowing users to easily locate specific changes by type or date range.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in.
- A task exists with a history containing various types of changes made on different dates.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task with a diverse modification history.
2. Access the "Task History" section.
3. Locate and utilize the filter options for the task history.
4. Apply a filter to display only changes of a specific type (e.g., "Status Updates").
5. Verify that the history display is correctly filtered, showing only the selected type of change.
6. Apply a filter to display changes within a specific date range.
7. Verify that the history display is correctly filtered, showing only changes within the selected timeframe.

Postconditions: None

Expected Result: The task history should be filterable by the type of change and date, displaying only the relevant entries.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-TH-004

Test Case Title: Verify that the system provides timestamps for all changes.

Test Case Description: This test case ensures that each change recorded in the task history is associated with an accurate and readable timestamp, aiding in tracking the chronology of modifications.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in.
- A task exists with a history of modifications.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task with a modification history.
2. Access the "Task History" section.
3. For each entry in the task history, verify that a timestamp is displayed.
4. Verify that the timestamp format is consistent and readable (e.g., "YYYY-MM-DD HH:MM:SS").

Postconditions: None

Expected Result: Each modification entry in the task history should have a corresponding timestamp, clearly indicating the time of change.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TH-005

Test Case Title: Verify that task history is accessible from the task details page.

Test Case Description: This test case confirms that users can easily access the task history directly from the task details page, ensuring convenient access to the change log.

Test Suite: Task History

Test Priority: High

Preconditions:

  - User is logged in.
  - A task exists.

Test Data: No test data needed.

Test Steps:

  1. Navigate to the task list.
  2. Open a task to view its details.
  3. Locate and click the option to view the "Task History" (this could be a button, tab, or link).
  4. Verify that the task history section is successfully loaded and displayed.

Postconditions: None

Expected Result: The user should be able to access the complete task history directly from the task details page without any errors.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 38

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Notifications

Feature description -

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

Userstory title -

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

Acceptance criteria -

['Users receive notifications for task updates, comments, mentions, and other relevant activities.', 'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.', 'The system logs notification history for future reference.']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Notifications

Test Case 1:

Test Case ID: TMTN-001

Test Case Title: Verify that users receive notifications for task updates.

Test Case Description: This test case verifies that users receive notifications across all enabled channels (email, mobile, in-app) when a task they are assigned to is updated.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* User A is logged in.

* User B is logged in.

* User A and User B are part of the same project.

Test Data: None

Test Steps:

1. User A creates a task and assigns it to User B.

2. User B accepts the task.

3. User A updates the task description and marks it as "In Progress".

Postconditions: None

Expected Result: User B receives a notification on all enabled channels (email, mobile, in-app) about the task update.

Severity: Major

Type of Testing: Functional Testing, System Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMTN-002

Test Case Title: Verify that users can customize their notification settings.

Test Case Description: This test case verifies that users can personalize their notification preferences, choosing their preferred delivery channels (email, mobile, in-app) and the types of notifications they wish to receive.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

* User is logged in.

Test Data: None

Test Steps:

1. User navigates to the notification settings page.

2. User selects their preferred notification channels (e.g., email only, mobile and in-app).

3. User selects the types of task activities they want to be notified about (e.g., task assignments, comments, status changes).

4. User saves the changes to their notification settings.

Postconditions: None

Expected Result: The system successfully saves the user's customized notification preferences.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMTN-003

Test Case Title: Verify that notifications provide detailed information about the activity.

Test Case Description: This test case ensures that notifications related to task updates, comments, mentions, and other activities provide comprehensive information, including the task title, project name, user involved, and the specific action taken.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* User A is logged in.

* User B is logged in.

* User A and User B are part of the same project.

Test Data: None

Test Steps:

1. User A adds a comment to a shared task.

2. User A mentions User B in the comment.

Postconditions: None

Expected Result: User B receives a notification detailing the activity. It includes:

   * The project name.

   * The task title.

   * The fact that User A added a comment.

   * The content of the comment, including User B's mention.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMTN-004

Test Case Title: Verify that the system logs notification history for future reference.

Test Case Description: This test ensures that the system maintains a log of all notifications sent to users, allowing users to review past notifications and track task activity history.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

* User is logged in.

* User has received previous task-related notifications.

Test Data: None

Test Steps:

1. User navigates to the notification history section within the application.

Postconditions: None

Expected Result: The system displays a chronological log of previous notifications, including:

  * The date and time each notification was sent.

  * The type of notification (task update, comment, mention).

  * The relevant task and project details.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMTN-005

Test Case Title: Verify that users do not receive notifications for activities they have opted out of.

Test Case Description: This test ensures that users can successfully disable specific notification types and that they will not receive notifications for those disabled activities.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

* User A is logged in.

* User B is logged in.

* User A and User B are part of the same project.

Test Data: None

Test Steps:

1. User B goes to notification settings.

2. User B disables notifications for "Task Comments".

3. User A adds a comment to a task shared with User B.

Postconditions: None

Expected Result: User B does not receive a notification about the new comment on the shared task.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Negative

Data - 39

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Archiving

Feature description -

Completed or inactive tasks can be archived or moved to a separate section, decluttering the task list and maintaining focus on active tasks. Task archiving helps users keep their task lists organized and prioritized, without losing access to important historical data.

Userstory title -

As a user, I want to archive completed tasks so that I can keep my task list focused on active tasks.

Acceptance criteria -

['Users can archive completed or inactive tasks.', 'Archived tasks are moved to a separate section for easy access.', 'Users can search and view archived tasks.', 'The system provides an option to restore archived tasks to the active list.', 'Archived tasks retain all original details and history.']

Generated Test Cases:

## Test Cases for Task Archiving Feature:

Test Case 1:

Test Case ID: TA-001

Test Case Title: Verify that users can archive completed tasks.

Test Case Description: This test case verifies that users can successfully archive completed tasks, removing them from the active task list.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

  - User is logged in.
  - A task exists and is marked as "Completed".

Test Data: No test data needed.

Test Steps:

  1. Navigate to the task list.
  2. Locate the completed task.
  3. Select the task and choose the "Archive" option.

Postconditions:

- The task is removed from the active task list.

Expected Result: The completed task is successfully archived and no longer appears in the active task list.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TA-002

Test Case Title: Verify that archived tasks are moved to a separate section.

Test Case Description: This test case verifies that archived tasks are accessible in a designated "Archived Tasks" section, separate from the active task list.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in.
- At least one task has been archived.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task management tool.
2. Locate and access the "Archived Tasks" section.

Postconditions: None

Expected Result: The system displays a separate section for archived tasks, allowing users to easily access them.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TA-003

Test Case Title: Verify that users can search and view archived tasks.

Test Case Description: This test case ensures users can effectively search for and view details of specific archived tasks.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:

  - User is logged in.
  - Multiple tasks are archived.

Test Data:

  - Search keywords related to archived tasks (e.g., task title, description).

Test Steps:

  1. Navigate to the "Archived Tasks" section.
  2. Use the search bar to enter relevant keywords.
  3. Select a search result to view the archived task details.

Postconditions: None

Expected Result: The system displays relevant archived tasks based on the search query, allowing users to view details and history of archived tasks.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TA-004

Test Case Title: Verify that the system provides an option to restore archived tasks.

Test Case Description: This test case ensures users can restore archived tasks back to their active task list if needed.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in.
- At least one task is archived.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Archived Tasks" section.
2. Select the archived task to be restored.
3. Choose the "Restore" or "Move to Active" option.

Postconditions:

- The task is removed from the "Archived Tasks" section.

Expected Result: The archived task is restored to the active task list, and its status is updated accordingly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TA-005

Test Case Title: Verify that archived tasks retain all original details and history.

Test Case Description: This test case validates that all original task details (description, attachments, comments, deadlines, etc.) are preserved when a task is archived and retrieved later.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:

- User is logged in.
- A task with details (description, attachments, etc.) is archived.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Archived Tasks" section.
2. Open the archived task and review its details and history.

Postconditions: None

Expected Result: All original task information, including descriptions, attachments, comments, deadlines, and history, remains intact and accessible after archiving.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 40

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Guest Access

Feature description -

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task information and involving external stakeholders in project discussions and updates.

Userstory title -

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on specific tasks.

Acceptance criteria -

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the platform.']

Generated Test Cases:

## Test Cases for Guest Access Feature in Task Management Tool:

Test Case 1:

Test Case ID: GuestAccess-001

Test Case Title: Verify that users can invite external collaborators as guests with limited access.

Test Case Description: This test case verifies the user's ability to invite external collaborators as guests and ensures they receive limited access to the platform.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in.
- User has an active project.

Test Data: Valid email address of the external collaborator.

Test Steps:

1. Navigate to the project settings.
2. Locate the "Guest Access" or "Collaborators" section.
3. Enter the email address of the external collaborator.
4. Select the appropriate access level (view-only or collaborative).
5. Send the invitation.

Postconditions:

- An invitation email is sent to the external collaborator.
- The invited guest is listed in the project's guest list.

Expected Result: The system should successfully send an invitation to the external collaborator with the specified access level.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: GuestAccess-002

Test Case Title: Verify that guests can view and collaborate on specific tasks or projects.

Test Case Description: This test case checks if the guest can view the designated project and interact with specific tasks based on their granted permissions.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- A guest account has been invited and activated.

- The guest is logged into the platform.

Test Data: No test data needed

Test Steps:

1. Log in as the invited guest.

2. Navigate to the shared project.

3. Verify access to view task details, descriptions, and attachments.

4. If applicable, attempt to add comments, update task status, or collaborate within permitted features.

Postconditions: None

Expected Result: The guest should only be able to access and interact with the specific tasks or projects they were granted access to. Viewing sensitive information or accessing restricted functionalities should be prohibited.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: GuestAccess-003

Test Case Title: Verify that the system provides clear distinctions between guest and full access roles.

Test Case Description: This test case ensures a clear visual and functional distinction between guest and full access user interfaces to avoid confusion and unauthorized actions.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- Both a guest account and a full access account are logged in.

Test Data: No test data needed

Test Steps:

1. Compare the user interface for both guest and full access accounts.

2. Verify that guest accounts have limited menu options, functionalities, and access compared to full access accounts.

3. Check for clear visual cues or labels indicating the guest's limited access level.

Postconditions: None

Expected Result: The platform should clearly distinguish guest users from full access users through visible and functional differences within the user interface. This includes limited menu options, restricted functionalities, and clear visual indicators of their guest access level.

Severity: Major

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: GuestAccess-004

Test Case Title: Verify that users can manage and revoke guest access at any time.

Test Case Description: This test case confirms the user's ability to manage guest permissions or revoke access entirely.

Test Suite: Guest Access

Test Priority: Medium

Preconditions:

- A guest account has been invited and added to a project.

- The user with administrative privileges is logged in.

Test Data: No test data needed

Test Steps:

1. Navigate to the project settings or guest management section.

2. Locate the guest list and select the target guest account.

3. Attempt to modify the guest's permissions (e.g., from 'collaborative' to 'view-only').

4. Save changes and confirm the updated permissions are reflected in the guest's access.

5. Attempt to revoke guest access completely.

6. Confirm the guest is removed from the project and no longer has access.

Postconditions:

- The guest's access level is updated.

- The guest is removed from the project.

Expected Result: The system should allow the user to modify or revoke guest access successfully. Guests should no longer have access to the project upon revocation.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: GuestAccess-005

Test Case Title: Verify that guests cannot access sensitive or restricted areas of the platform.

Test Case Description: This test case aims to identify and prevent unauthorized access by guests to sensitive information or functionalities beyond their permitted scope.

Test Suite: Guest Access, Security

Test Priority: High

Preconditions:

- A guest account is logged in.

Test Data: No test data needed

Test Steps:

1. Attempt to navigate to various platform sections outside the invited project.

2. Try accessing administrative functions, user management, billing details, or other sensitive areas.

3. Attempt actions beyond the granted permissions, like deleting tasks, inviting new users, or modifying project settings.

Postconditions: None

Expected Result: The system should prevent guest accounts from accessing any sensitive information or functionalities outside their permitted scope. The platform should display an error message or redirect the guest to the permitted areas.

Severity: Critical

Type of Testing: Security Testing, Functional Testing

Test Case Approach: Negative

--

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 Cases for Task Management Tool - Task Import/Export:

Test Case 1:

* **Test Case ID:** TMTool-Import-001
* **Test Case Title:** Verify that users can import tasks from a CSV file.
* **Test Case Description:** This test case aims to verify the functionality of importing tasks from a CSV file, including proper field mapping and data integrity.
* **Test Suite:** Task Import/Export
* **Test Priority:** High
* **Preconditions:**
    * User is logged in.
    * A valid CSV file with task data is available.
* **Test Data:** Sample CSV file with tasks including title, description, due date, and priority.
* **Test Steps:**
    1. Navigate to the "Import Tasks" section.
    2. Select "CSV" as the import format.
    3. Upload the prepared CSV file.

   4. Verify the system correctly maps CSV columns to task fields.

   5. Proceed with the import.

* **Postconditions:**

   * Imported tasks are listed in the user's task list.

   * Task details match the data in the CSV file.

* **Expected Result:** The system successfully imports all tasks from the CSV file, maintaining data integrity and field mapping.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** TMTool-Import-002

* **Test Case Title:** Verify that users can preview imported data before finalizing the import process.

* **Test Case Description:** This test case ensures users can preview and validate imported task data before it's integrated into the system, allowing for error detection and correction.

* **Test Suite:** Task Import/Export

* **Test Priority:** High

* **Preconditions:**

   * User is logged in.

   * A file (CSV or Excel) is ready for import.

* **Test Data:** Sample Excel file with a mix of valid and invalid task data (e.g., missing due dates).

* **Test Steps:**

   1. Initiate the import process and select the file.

   2. Choose the "Preview" option.

   3. Verify the preview accurately reflects the file data.

   4. Check for any errors or inconsistencies highlighted by the system.

* **Postconditions:**

   * No data is imported at this stage.

* **Expected Result:** The system displays a clear and accurate preview of the imported data, highlighting any potential errors or inconsistencies.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** TMTool-Export-001

* **Test Case Title:** Verify that users can export tasks to an Excel file.

* **Test Case Description:** This test case aims to confirm the functionality of exporting tasks to an Excel file, ensuring data accuracy and format consistency.

* **Test Suite:** Task Import/Export

* **Test Priority:** High

* **Preconditions:**

   * User is logged in.

   * User has existing tasks in the system.

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

* **Test Steps:**

   1. Go to the task list view.

   2. Select tasks to be exported.

   3. Choose "Export" and select "Excel" as the format.

   4. Download the generated Excel file.

   5. Verify the file contains all selected tasks and their details.

* **Postconditions:**
    * Users have a downloaded Excel file containing the exported tasks.
* **Expected Result:** The system successfully exports the selected tasks to an Excel file with correct formatting and data integrity.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** TMTool-Import-003
* **Test Case Title:** Verify that the system provides clear error messages for invalid data during import.
* **Test Case Description:** This test assesses the system's error handling during import, ensuring clear and informative messages are provided for invalid data, guiding users on necessary corrections.
* **Test Suite:** Task Import/Export
* **Test Priority:** Medium
* **Preconditions:**
    * User is logged in.
* **Test Data:** A CSV file containing tasks with missing titles, invalid due date formats, and other incorrect data.
* **Test Steps:**
    1. Attempt to import the prepared CSV file.
    2. Observe the system response for each invalid data entry.
    3. Verify the system displays specific error messages for each invalid field.
* **Postconditions:**
    * No tasks with invalid data are imported into the system.
* **Expected Result:** The system identifies all invalid data entries, prevents import, and displays clear,

specific error messages for each issue, guiding the user on necessary corrections.

* **Severity:** Major

* **Type of Testing:** Functional Testing, Negative Testing

* **Test Case Approach:** Negative

Test Case 5:

* **Test Case ID:** TMTool-Log-001

* **Test Case Title:** Verify that the system logs import/export activities.

* **Test Case Description:** This test ensures that all import/export actions are logged for auditing and troubleshooting purposes.

* **Test Suite:** Task Import/Export

* **Test Priority:** Medium

* **Preconditions:**

    * User is logged in.

    * User has performed at least one import or export action.

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

* **Test Steps:**

    1. Navigate to the "Import/Export Logs" section (if such a section is accessible to the user).

    2. Verify that the log displays entries for each import and export action.

    3. Verify the log includes timestamps, user IDs, file types, and import/export status (success/failure).

* **Postconditions:**

    * The log accurately reflects import/export activity.

* **Expected Result:** The system maintains a detailed log of all import and export actions, including timestamps, user information, file details, and the outcome of each operation.

* **Severity:** Minor

* **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 Cases for Task Management Tool - Task Prioritization Feature:

Test Case 1:

Test Case ID: TMTP-001

Test Case Title: Verify that users can set and adjust task priorities.

Test Case Description: This test case verifies that users can successfully set a priority level for new tasks and modify the priority of existing tasks within the task management tool. It will also check that the chosen priority is accurately reflected in the task view.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.
- User has created a task.

Test Data: No test data needed

Test Steps:

1. Navigate to the task view.
2. Locate the priority setting for the task.
3. Select a priority level (e.g., Low, Medium, High).
4. Save the task.
5. Verify the selected priority level is displayed correctly.
6. Change the task priority to a different level.
7. Save the task again.
8. Verify the updated priority level is displayed correctly.

Postconditions:

- The task should display the selected priority level.
- The task priority can be modified, and the changes are reflected.

Expected Result: The system should allow setting and adjusting task priorities, with changes accurately reflected in the task view.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMTP-002

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

Test Case Description: This test case ensures the task management tool allows users to sort and filter their task lists based on different priority levels. This will confirm the system's ability to organize tasks effectively for better management.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.
- User has created multiple tasks with different priority levels.

Test Data: No test data needed

Test Steps:

1. Navigate to the main task list.
2. Locate the sorting or filtering options.
3. Choose to sort/filter by "Priority".
4. Verify that tasks are displayed in the correct order (e.g., High to Low, or vice versa).
5. Select a specific priority level filter (e.g., "High").
6. Verify that only tasks with the selected priority level are displayed.

Postconditions: None

Expected Result: The system should correctly sort and filter tasks according to the chosen priority level, allowing users to focus on urgent tasks.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMTP-003

Test Case Title: Verify that the system provides visual indicators of task priority.

Test Case Description: This test case validates the existence and clarity of visual indicators (e.g., color coding, icons, labels) used to represent different task priority levels. This ensures users can quickly identify the urgency of tasks at a glance.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in.
- User has access to a list of tasks with varying priority levels.

Test Data: No test data needed

Test Steps:

1. Navigate to the task list view.
2. Observe the visual representation of tasks with different priority levels.
3. Verify that each priority level has a distinct and easily identifiable visual indicator.
4. Check the clarity and intuitiveness of the visual indicators (e.g., red for high priority, yellow for medium).

Postconditions: None

Expected Result: Each task should have a clear and distinct visual indicator corresponding to its priority level, making it easy for users to assess task urgency visually.

Severity: Minor

Type of Testing: UI Testing, Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMTP-004

Test Case Title: Verify that users receive notifications for high-priority tasks.

Test Case Description: This test case checks the system's ability to send timely and relevant notifications to users regarding approaching deadlines or updates on high-priority tasks. This ensures users are kept informed about critical tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in.
- User has enabled notifications for tasks.
- A high-priority task with an approaching deadline exists.

Test Data: A high-priority task nearing its deadline.

Test Steps:

1. Set a high-priority task with an approaching deadline.
2. Wait for the notification trigger time.
3. Verify that the user receives a notification about the high-priority task.
4. Check the content of the notification to ensure it includes relevant information (task name, deadline, priority).

Postconditions: None

Expected Result: The system should successfully deliver a notification to the user, alerting them about the high-priority task and its approaching deadline.

Severity: Critical

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMTP-005

Test Case Title: Verify that priority changes are logged and tracked.

Test Case Description: This test case aims to ensure that any changes made to a task's priority are recorded and accessible in a log. This log helps track task progress and identify any potential issues related to priority changes.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in.
- A task exists with a defined priority.

Test Data: No test data needed

Test Steps:

1. Access an existing task.
2. Change the task's priority level.
3. Save the changes.
4. Navigate to the task's activity log or history.
5. Verify that the priority change is logged with details like the timestamp, user who made the change, and the previous and new priority levels.

Postconditions: None

Expected Result: The system should accurately log all changes made to a task's priority, including details of the change for auditing and tracking purposes.

Severity: Minor

Type of Testing: Auditing, Functional Testing

Test Case Approach: Positive

Data - 43

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

Userstory title -

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay updated on task progress.

Acceptance criteria -

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

Generated Test Cases:

## Test Cases for Task Collaboration Feature:

Test Case 1:

Test Case ID: TC-001

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

Test Case Description: This test case verifies that a user can successfully assign a created task to another team member within the task management tool.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- User has created a task.
- The team member to be assigned exists in the system.

Test Data: No test data needed

Test Steps:

1. Navigate to the created task.
2. Locate the "Assign to" option within the task details.
3. Select a team member from the list of available users.
4. Confirm the assignment.

Postconditions:

- The assigned team member is displayed as the assignee for the task.

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

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC-002

Test Case Title: Verify that users can share updates and communicate through comments.

Test Case Description: This test case ensures that users can add comments to a task, providing updates or engaging in discussions with team members.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in.
- User has access to a task.

Test Data: Sample comment text: "This is a test comment for the task."

Test Steps:

1. Open a specific task.
2. Locate the "Comments" section.
3. Enter the test comment text in the comment box.
4. Click the "Add Comment" or equivalent button.

Postconditions:

- The added comment is displayed within the "Comments" section of the task.

Expected Result: The system should successfully add the comment to the task, making it visible to other collaborators.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC-003

Test Case Title: Verify that task updates and comments are visible to assigned team members.

Test Case Description: This test case validates that any changes made to a task, including updates or added comments, are visible to all assigned team members to ensure transparency and collaboration.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User 1 (assigner) is logged in and has assigned a task to User 2 (assignee).
- User 2 (assignee) is logged in.

Test Data:

- Update to Task Description: "Task description updated."
- Comment text: "This is a test comment on the assigned task."

Test Steps:

1. As User 1, open the assigned task and make an update to the task description.
2. As User 1, add a test comment to the task.
3. Log in as User 2 (assignee).
4. Navigate to the assigned task.

Postconditions: None

Expected Result: User 2 should be able to see the updated task description and the new comment added by User 1.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC-004

Test Case Title: Verify that users receive notifications for task updates and comments.

Test Case Description: This test case ensures the notification system functions correctly, notifying users about relevant updates and comments related to their assigned tasks.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User 1 (assigner) is logged in and has assigned a task to User 2 (assignee).
- Notification settings for task updates and comments are enabled for both users.

Test Data:

- Comment text: "This is a test comment requiring notification."

Test Steps:

1. As User 1, add a new comment to the assigned task.
2. Log in as User 2 (assignee).

Postconditions: None

Expected Result: User 2 should receive a notification alerting them about the new comment added to the assigned task by User 1.

Severity: Major

Type of Testing: Functional Testing, Notification Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC-005

Test Case Title: Verify that the system provides a clear view of task collaboration history.

Test Case Description: This test case checks if the system maintains and displays a comprehensive history of all actions and interactions related to a task's collaboration, including assignments, updates, and comments, in a clear and understandable format.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- A task exists with a history of collaboration (assignments, updates, comments).

Test Data: No test data needed

Test Steps:

1. Navigate to the task with collaboration history.

2. Locate the section displaying the task's activity log or history.

Postconditions: None

Expected Result: The system should display a clear and chronological history of all actions related to the task, including:

- Initial task creation.

- Assignments to team members.

- Updates made to the task description, deadlines, etc.

- All comments added to the task.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 44

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Tracking

Feature description -

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

Userstory title -

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

Acceptance criteria -

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.', 'System provides visual indicators for task progress.', '']

Generated Test Cases:

## Test Cases for Task Tracking Feature:

Test Case 1:

Test Case ID: TMT-TT-001

Test Case Title: Verify that users can view task statuses.

Test Case Description: This test case verifies that users are able to view the status of their tasks within the task management tool. The test will cover different task statuses such as "To Do," "In Progress," "Completed," and potentially custom statuses defined by the user.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in.

- User has created at least one task.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task list view.

2. Observe the status displayed for each task.

3. Verify that each task displays a clear and accurate status.

Postconditions: None

Expected Result: The system should display the correct status for each task, allowing users to easily understand the current state of their tasks.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TT-002

Test Case Title: Verify that users can track task progress and deadlines.

Test Case Description: This test case ensures users can track the progress of a task and view its deadline. This includes checking progress bars, percentage completion indicators, and clear deadline displays.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in.

- User has created at least one task with a set deadline.

Test Data: No test data needed.

Test Steps:

1. Navigate to a task's details view.

2. Observe the displayed deadline for the task.

3. Verify that a visual indicator of task progress is present (e.g., progress bar).

4. Update the task's progress (e.g., mark subtasks as complete).

5. Verify that the progress indicator updates accordingly.

Postconditions: None

Expected Result: The system should clearly display the task deadline and provide a visual representation of the task's progress, allowing users to monitor its advancement.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TT-003

Test Case Title: Verify that tasks are updated in real-time.

Test Case Description: This test case ensures that any changes made to a task, such as status updates, progress updates, or deadline modifications, are reflected in real-time for all users collaborating on that task.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- Two users are logged in.

- Both users have access to the same task.

Test Data: No test data needed.

Test Steps:

1. User 1 makes a change to the task (e.g., updates the status, progresses the task, or modifies the deadline).

2. User 2 refreshes their view of the task.

3. Verify that User 2 immediately sees the changes made by User 1.

Postconditions: None

Expected Result: Any changes made to a task by one user should be reflected in real-time for all other users viewing that task, ensuring everyone has access to the most up-to-date information.

Severity: Critical

Type of Testing: Functional Testing, Real-Time Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-TT-004

Test Case Title: Verify that the system provides visual indicators for task progress.

Test Case Description: This test case checks the presence and clarity of visual indicators used to represent task progress. This can include progress bars, percentage completion, color-coding, or any other visual aids used to communicate task status at a glance.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has accessed the task list view.

Test Data: No test data needed.

Test Steps:

1. Observe the task list and individual task views for visual progress indicators.
2. Verify that the indicators are clear, easy to understand, and visually distinct for different progress levels.
3. Verify that the chosen visual indicators align with accessibility standards (e.g., colorblind-friendly).

Postconditions: None

Expected Result: The system should utilize clear and effective visual indicators to communicate task progress, making it easy for users to quickly assess the status of their tasks.

Severity: Minor

Type of Testing: UI Testing, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TT-005

Test Case Title: Verify that the task tracking view handles a large number of tasks efficiently.

Test Case Description: This test case evaluates the performance of the task tracking feature when handling a large volume of tasks. This ensures that the system remains responsive and usable even under heavy load.

Test Suite: Task Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: A large number of tasks (e.g., 1000+).

Test Steps:

1. Populate the system with a large number of tasks.
2. Attempt to scroll through the task list, filter tasks, and access individual task details.
3. Monitor the system for any lag, slow loading times, or performance issues.

Postconditions:

- Delete the large number of test tasks.

Expected Result: The system should remain responsive and maintain acceptable performance levels even when handling a large volume of tasks, ensuring a smooth user experience.

Severity: Major

Type of Testing: Performance Testing, Load Testing

Test Case Approach: Positive

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 Cases for Task Management Tool - Task Notifications:

Test Case 1:

Test Case ID: TMT-TN-001

Test Case Title: Verify that users receive notifications for new tasks.

Test Case Description: This test case verifies that users receive notifications when a new task is assigned to them within the task management tool.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

  - User A and User B are logged in.

  - User A has permission to assign tasks.

Test Data: No test data needed

Test Steps:

  1. User A creates a new task.

  2. User A assigns the task to User B.

  3. Observe User B's notifications.

Postconditions:

  - The new task is visible in User B's task list.

Expected Result: User B receives a notification indicating a new task has been assigned to them.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT-TN-002

Test Case Title: Verify that users receive notifications for task updates and comments.

Test Case Description: This test verifies that users are notified when an existing task they are assigned to is updated or receives a new comment.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User A and User B are logged in.
- User A and User B are assigned to the same task.

Test Data: Text for an update and a comment.

Test Steps:

1. User A opens the shared task.
2. User A adds an update to the task description.
3. User A adds a comment to the task.
4. Observe User B's notifications.

Postconditions:

- The task reflects the update and comment.

Expected Result: User B receives two separate notifications: one for the task update and one for the new comment.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT-TN-003

Test Case Title: Verify that users receive reminders for upcoming deadlines.

Test Case Description: This test ensures the system sends timely reminders to users for tasks with approaching deadlines.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User A is logged in.

- User A has a task with a future deadline.

Test Data: No test data needed

Test Steps:

1. Set the system date/time to a period before the task deadline (e.g., one day before).
2. Observe User A's notifications.

Postconditions: None

Expected Result: User A receives a notification reminding them of the approaching task deadline.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT-TN-004

Test Case Title: Verify that notification settings can be customized by users.

Test Case Description: This test case ensures users have the functionality to personalize their notification preferences within the task management tool.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User A is logged in.

Test Data: No test data needed

Test Steps:

1. Navigate to the user settings.
2. Locate the notification settings section.
3. Verify options exist to customize notification delivery methods (e.g., email, in-app), frequency, and specific event triggers.

  4. Modify notification preferences.

  5. Save changes.

Postconditions: Notification settings are updated.

Expected Result: The system successfully saves the user's customized notification settings.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT-TN-005

Test Case Title: Verify that users do not receive notifications for tasks they are not assigned to.

Test Case Description: This test ensures that users are not overwhelmed with irrelevant notifications and only receive updates for their assigned tasks.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

  - User A and User B are logged in.

  - User A creates a new task but does not assign it to anyone.

Test Data: No test data needed

Test Steps:

  1. User A creates a new task.

  2. Observe both User A and User B's notifications.

Postconditions: The task is visible in User A's task list.

Expected Result: Neither User A nor User B receives a notification for the unassigned task.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Negative

Data - 46

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Categorization

Feature description -

Users can categorize tasks into different groups or projects for better organization and management. Task categorization aims to help users organize tasks by type, priority, or project.

Userstory title -

As a user, I want to categorize tasks into different groups or projects so that I can organize and manage my tasks more effectively.

Acceptance criteria -

['Users can create and assign tasks to different categories or projects.', 'Users can view tasks grouped by categories or projects.', 'Tasks can be moved between categories or projects.', 'Category or project information is accurately reflected in task views.', '']

Generated Test Cases:

## Test Cases for Task Management Tool - Task Categorization

Test Case 1

Test Case ID: TC_CAT_001

Test Case Title: Verify that users can create and assign tasks to different categories.

Test Case Description: This test case ensures that users are able to create new task categories and assign newly created tasks to these categories.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in.

- User has access to create tasks.

Test Data:

- Task Name: "Grocery Shopping"

- Category Name: "Personal Errands"

Test Steps:

1. Navigate to the "Tasks" section.

2. Click on "Create New Task".

3. Enter the Task Name: "Grocery Shopping".

4. Click on "Choose Category".

5. Select "Create New Category".

6. Enter the Category Name: "Personal Errands".

7. Save the category.

8. Verify the task is assigned to "Personal Errands" category.

Postconditions:

- The task "Grocery Shopping" is successfully created and assigned to the "Personal Errands" category.

Expected Result: The system should allow users to create new categories and assign tasks to them.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: TC_CAT_002

Test Case Title: Verify that users can view tasks grouped by categories.

Test Case Description: This test case validates that the system correctly displays tasks organized by their assigned categories, allowing users to easily locate tasks within specific categories.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in.

- At least two tasks are assigned to different categories.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Tasks" section.

2. Select the view mode as "Categorized".

3. Verify that tasks are grouped and displayed under their respective categories.

Postconditions: None

Expected Result: The system should accurately display tasks grouped by their assigned categories.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: TC_CAT_003

Test Case Title: Verify that tasks can be moved between categories.

Test Case Description: This test case confirms that users can seamlessly move existing tasks from one category to another, ensuring flexibility in task organization.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

- User is logged in.

- At least two categories exist.

- A task is assigned to a category.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task you want to move.

2. Open the task details.

3. Click on the "Edit Category" option.

4. Select a different category from the list.

5. Save the changes.

6. Navigate to the new category.

7. Verify that the moved task is now present in the new category.

Postconditions:

- The task is successfully moved to the new category.

Expected Result: The system should allow users to easily move tasks between different categories.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: TC_CAT_004

Test Case Title: Verify that category information is accurately reflected in task views.

Test Case Description: This test case ensures that when viewing individual tasks, the category information associated with the task is displayed accurately and consistently across all task views.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

- User is logged in.
- A task is assigned to a category.

Test Data: No test data needed.

Test Steps:

1. Navigate to a task.
2. Open the task details.
3. Locate the "Category" field.
4. Verify that the displayed category matches the category assigned to the task.

Postconditions: None

Expected Result: The system should accurately display the correct category information for each task across all task views.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: TC_CAT_005

Test Case Title: Verify that deleting a category does not delete the tasks assigned to it.

Test Case Description: This test case verifies that when a category is deleted, the tasks previously assigned to that category are not deleted and are moved to a default category or a designated Uncategorized category.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

- User is logged in.
- A category with assigned tasks exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the category settings.
2. Choose the option to delete the category.
3. Confirm the deletion.
4. Navigate to the "Tasks" section.
5. Verify that the tasks previously assigned to the deleted category are not deleted and are either moved to a default category or displayed under an "Uncategorized" category.

Postconditions:

- The category is successfully deleted.

Expected Result: The system should not delete tasks when their assigned category is deleted, ensuring data integrity.

Severity: Critical

Type of Testing: Functional Testing, Negative Testing

Test Case Approach: Destructive

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 Cases for Task Management Tool - Task Search Feature

Test Case 1:

Test Case ID: TMS-TS-001

Test Case Title: Verify that users can search for tasks by name.

Test Case Description: This test case verifies that the task search functionality correctly returns tasks matching the entered name, including partial matches and case-insensitive searches.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in.

- At least one task exists.

Test Data:

- Existing task name: "Write Project Report"

- Partial task name: "Report"

- Case-varied task name: "write project report"

Test Steps:

1. Navigate to the task search bar.

2. Enter the full existing task name ("Write Project Report").

3. Observe search results.

4. Enter the partial task name ("Report").

5. Observe search results.

6. Enter the case-varied task name ("write project report").

7. Observe search results.

Postconditions: None

Expected Result:

- The search returns the correct task when searching with the full task name.

- The search returns relevant tasks containing the partial name, regardless of case.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMS-TS-002

Test Case Title: Verify that users can search for tasks by description.

Test Case Description: This test case verifies that users can search for tasks using keywords or phrases present in the task description.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in.

- At least one task exists with a detailed description.

Test Data:

- Task with description: "Design website layout, including homepage, product pages, and contact form."

- Search keyword: "homepage"

Test Steps:

1. Navigate to the task search bar.

2. Enter the keyword ("homepage") from the task description.

3. Observe search results.

Postconditions: None

Expected Result: The search accurately returns tasks containing the keyword in their description.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMS-TS-003

Test Case Title: Verify that users can filter search results by category, priority, or due date.

Test Case Description: This test case checks if users can refine search results using filters like category, priority level, and due date range.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in.
- Tasks exist with different categories, priorities, and due dates.

Test Data: No test data needed

Test Steps:

1. Perform a general search that returns multiple results.
2. Apply a filter by category (e.g., "Marketing").
3. Observe the filtered results.
4. Apply a filter by priority (e.g., "High").
5. Observe the filtered results.
6. Apply a filter by due date range (e.g., "Next week").
7. Observe the filtered results.

Postconditions: None

Expected Result: Each filter should accurately narrow down the search results based on the selected criteria.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMS-TS-004

Test Case Title: Verify that search results are accurate and relevant.

Test Case Description: This test case ensures that the search functionality only displays tasks that match the entered search criteria and that irrelevant results are not displayed.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in.
- A variety of tasks exist.

Test Data:

- Unique keyword not present in any task.

Test Steps:

1. Enter a unique keyword that doesn't exist in any task name or description.
2. Observe search results.
3. Perform searches using a combination of name, description, and filters.
4. Verify that each returned task matches the applied search criteria.

Postconditions: None

Expected Result:

- The search should not display any results when an irrelevant keyword is used.
- All displayed tasks should accurately reflect the entered search terms and filters.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Negative, Positive

Test Case 5:

Test Case ID: TMS-TS-005

Test Case Title: Verify search functionality with empty search input.

Test Case Description: This test case checks the system's behavior when the search is performed without entering any search criteria.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: No test data needed

Test Steps:

1. Navigate to the task search bar.
2. Without entering any text, click the search button (or equivalent action).
3. Observe the system's response.

Postconditions: None

Expected Result: The system should either display all tasks or provide a message indicating that the search criteria are required.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Negative

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 Cases for Task Collaboration Feature:

Test Case 1:

Test Case ID: TC_Collab_001

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

Test Case Description: This test case verifies that users can successfully add comments to existing tasks. It ensures that the comment input field functions correctly, comments are saved and displayed properly, and different users can interact through comments.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* User is logged in.

* A task is created.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task view.

2. Locate the comment section.

3. Enter a comment in the input field.

4. Click the "Add Comment" button.

5. Verify the comment is displayed below the input field.

Postconditions:

* The comment is saved and associated with the task.

* The comment is visible to other users collaborating on the task.

Expected Result: The system successfully adds the comment to the task and displays it correctly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_Collab_002

Test Case Title: Verify that users can attach files to tasks.

Test Case Description: This test case ensures users can attach various file types to tasks. It covers successful uploads, file size limitations, and validation checks for supported formats.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* User is logged in.

* A task is created.

Test Data: Different file types (e.g., .doc, .pdf, .jpg).

Test Steps:

1. Navigate to the task view.

2. Locate the file attachment section.

3. Click "Choose File" and select a file to upload.

4. Verify the file name appears in the attachment list.

5. Repeat steps 3-4 with different file types and sizes.

Postconditions:

* Attached files are stored securely.

* Users can download or preview the files.

Expected Result: The system successfully uploads and displays various file types within the task's attachment section.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_Collab_003

Test Case Title: Verify that users can share tasks with other users.

Test Case Description: This test case validates the task sharing functionality, ensuring users can share tasks with specific individuals or groups. It checks for appropriate access control and notifications for shared tasks.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* User is logged in.

* A task is created.

* Other users exist within the platform.

Test Data: No test data needed.

Test Steps:

1. Navigate to the task view.

2. Locate the "Share Task" option.

3. Select the desired user(s) or group(s) to share with.

4. Click "Share" to confirm.

5. Verify the selected users/groups have access to the task.

Postconditions:

* Shared task appears in the recipient's task list.

* Notifications are sent to users about the shared task.

Expected Result: The system shares the task with selected users or groups, providing them appropriate access and sending relevant notifications.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_Collab_004

Test Case Title: Verify that real-time updates occur for task comments and attachments.

Test Case Description: This test case ensures that any changes made to comments or attachments are reflected in real-time for all users collaborating on the task.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* Two users are logged in and collaborating on the same task.

* The task has existing comments and attachments.

Test Data: No test data needed.

Test Steps:

1. User 1 adds a new comment to the task.

2. User 2 verifies that the new comment appears instantly on their screen without needing to refresh the page.

3. User 1 uploads a new file attachment to the task.

4. User 2 verifies that the new file attachment appears instantly on their screen without needing to refresh the page.

Postconditions: None

Expected Result: Both users should see the updated comments and attachments without any delay, ensuring seamless collaboration.

Severity: Major

Type of Testing: Functional Testing, Real-time Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_Collab_005

Test Case Title: Verify that task collaboration features enhance team communication.

Test Case Description: This test case focuses on the overall impact of collaboration features on team communication. It assesses the ease of use, clarity of information exchange, and efficiency of collaboration using the provided tools.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

* Multiple users are actively collaborating on various tasks.

Test Data: No test data needed.

Test Steps:

1. Observe users' interactions and communication flow within tasks.

2. Evaluate the clarity and organization of comments.

3. Assess the effectiveness of file sharing and its impact on collaboration.

4. Conduct user surveys or interviews to gather feedback on their experience with collaboration features.

Postconditions: None

Expected Result: Task collaboration features should demonstrably improve communication flow, reduce misunderstandings, and contribute to efficient task completion through shared information and timely updates.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

--

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 Cases for Online Learning Platform - Course Creation Tools

Test Case 1:

Test Case ID: TC_CC_001

Test Case Title: Verify that instructors can create courses using various multimedia elements.

Test Case Description: This test case verifies that instructors can successfully create courses using different multimedia components, including videos, presentations, quizzes, and assignments, ensuring a comprehensive and engaging learning experience for students.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- Instructor is logged in.
- Course creation page is accessible.

Test Data: Sample video file, presentation slides, quiz questions, and assignment details.

Test Steps:

1. Navigate to the course creation page.
2. Input course details (title, description, etc.).
3. Upload and embed a video lecture.
4. Add a presentation with multiple slides.
5. Create a quiz with various question types (multiple-choice, true/false).

6. Set up an assignment with submission instructions.

7. Save the course.

Postconditions:

- Course is saved with all added multimedia elements.

Expected Result: The system should successfully save the course with all the added multimedia elements (video, presentation, quiz, assignment).

Severity: Critical

Type of Testing: Functional, User Interface (UI) Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_CC_002

Test Case Title: Verify that the course creation tools are user-friendly and require minimal technical skills.

Test Case Description: This test case aims to evaluate the user-friendliness of the course creation tools by observing instructors with varying technical skills using the platform. The goal is to ensure the tools are intuitive and require minimal technical expertise.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- Instructor accounts with varying technical skills are created.

Test Data: No test data needed

Test Steps:

1. Provide instructors with minimal instructions on using the course creation tools.

2. Observe instructors as they navigate the interface and attempt to create a basic course.

3. Record any difficulties or points of confusion encountered by the instructors.

4. Gather feedback from instructors on the usability and intuitiveness of the tools.

Postconditions:

- Collect feedback from instructors for analysis and potential improvements.

Expected Result: Instructors with varying technical skills should be able to navigate the platform and create a basic course with minimal difficulty. The tools should be self-explanatory and intuitive.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_CC_003

Test Case Title: Verify that instructors can preview courses before publishing.

Test Case Description: This test case ensures that instructors can preview the entire course content and flow from a student's perspective before making it live, allowing for final adjustments and quality assurance.

Test Suite: Course Creation

Test Priority: High

Preconditions:

- Instructor is logged in.

- A course is created with at least one module.

Test Data: No test data needed

Test Steps:

1. Go to the "My Courses" section.

2. Select the course to preview.

3. Click on the "Preview" button.

4. Navigate through the course modules, including videos, presentations, quizzes, and assignments.

Postconditions: No specific postconditions.

Expected Result: The system should accurately display the course content as it would appear to students, allowing instructors to review and adjust the material before publishing.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_CC_004

Test Case Title: Verify that instructors can save courses as drafts.

Test Case Description: This test case ensures that the system provides instructors the flexibility to save their course work in progress as drafts and come back later to continue editing, preventing data loss and offering flexibility in course development.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

- Instructor is logged in.
- Course creation page is accessible.

Test Data: Partially filled course details.

Test Steps:

1. Navigate to the course creation page.
2. Fill in some course details (title, partial description, etc.).
3. Instead of publishing, click on the "Save as Draft" button.
4. Log out of the platform.
5. Log back in and navigate to the "Drafts" section.

Postconditions:

- The partially filled course should be saved and accessible in the "Drafts" section for future editing.

Expected Result: The system should save the incomplete course as a draft, allowing the instructor to return later and continue editing without losing previously entered data.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_CC_005

Test Case Title: Verify that instructors can edit courses post-publishing.

Test Case Description: This test case validates the functionality for instructors to make necessary updates or modifications to courses even after they have been published and made available to students, ensuring course content remains relevant and up-to-date.

Test Suite: Course Creation

Test Priority: Medium

Preconditions:

- Instructor is logged in.
- A course is created and published.

Test Data: Updated course information (e.g., new module, revised quiz question).

Test Steps:

1. Navigate to the "My Courses" section.
2. Select the published course to edit.
3. Click the "Edit Course" button.
4. Make the desired changes (e.g., add a new module, edit existing content).
5. Save the changes.

Postconditions:

- The published course should reflect the changes made.

Expected Result: The system should successfully implement the changes to the published course, making the updated content accessible to enrolled students.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 50

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Interactive Lessons

Feature description -

Lessons are thoughtfully designed to foster learner engagement through interactive components such as quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these interactive elements enhance comprehension and retention of course material.

Userstory title -

As a learner, I want interactive lessons so that I can engage with the content more effectively.

Acceptance criteria -

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided after each interactive component.', 'Interactive lessons can be accessed on various devices.']

Generated Test Cases:

## Test Cases for Interactive Lessons on Online Learning Platform:

Test Case 1:

Test Case ID: IL-001

Test Case Title: Verify that lessons include quizzes, polls, simulations, and discussions.

Test Case Description: This test case verifies the presence and functionality of all required interactive elements (quizzes, polls, simulations, discussions) within a lesson on the platform.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

  - Learner is logged in.
  - Learner has accessed a lesson.

Test Data: No test data needed

Test Steps:

  1. Navigate to a lesson.
  2. Check for the presence of a quiz section within the lesson.
  3. Check for the presence of a poll within the lesson.
  4. Check for the presence of a simulation activity within the lesson.
  5. Check for the presence of a discussion forum linked to the lesson.

Postconditions: None

Expected Result: Each of the interactive elements (quiz, poll, simulation, discussion) should be present and accessible within the lesson.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: IL-002

Test Case Title: Verify that interactive elements are integrated seamlessly into the lesson content.

Test Case Description: This test case verifies that interactive elements are not presented in isolation but are logically integrated within the lesson flow to enhance learning.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has accessed a lesson.

Test Data: No test data needed

Test Steps:

1. Go through the lesson content.
2. Observe the placement and relevance of each interactive element (quiz, poll, simulation, discussion).
3. Check if the elements appear at logical points to reinforce the presented material.

Postconditions: None

Expected Result: All interactive elements should be meaningfully incorporated within the lesson, contributing to the learning experience and not disrupting the flow.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: IL-003

Test Case Title: Verify that learners can track their progress through interactive activities.

Test Case Description: This test case ensures the platform provides learners with clear visibility and tracking of their performance and completion status on interactive components within a lesson.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has participated in at least one interactive activity within a lesson.

Test Data: No test data needed

Test Steps:

1. Complete a quiz, participate in a poll, finish a simulation, or engage in a discussion.
2. Navigate to the lesson progress section or dashboard.
3. Check for the display of completion status or scores related to the interactive activities.

Postconditions: None

Expected Result: The platform should accurately reflect the learner's progress and achievements on interactive elements, allowing them to monitor their learning journey effectively.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: IL-004

Test Case Title: Verify that feedback is provided after each interactive component.

Test Case Description: This test case validates that learners receive immediate and informative feedback after engaging with each type of interactive component within a lesson.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has accessed a lesson and interacted with a quiz, poll, simulation, or discussion.

Test Data: Various answers for quizzes and polls, different outcomes for simulations.

Test Steps:

1. Answer a quiz question.
2. Submit an answer to a poll.
3. Complete a simulation activity.
4. Post in a lesson discussion forum.
5. Observe the system response after each interaction.

Postconditions: None

Expected Result: The platform should provide immediate feedback for quizzes and polls (correct/incorrect), relevant feedback after a simulation (based on actions taken), and facilitate feedback within discussion forums (instructor/peer responses).

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: IL-005

Test Case Title: Verify that interactive lessons can be accessed on various devices.

Test Case Description: This test case checks the responsiveness and functionality of interactive lessons across multiple devices (desktop, mobile, tablet) ensuring learners can access and engage with content seamlessly regardless of their chosen device.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is logged in.
- Lesson with interactive components is available.

Test Data: No test data needed

Test Steps:

1. Access the online learning platform on a desktop computer.
2. Open a lesson and interact with the quizzes, polls, simulations, and discussions.
3. Repeat steps 1 and 2 using a mobile phone.
4. Repeat steps 1 and 2 using a tablet.

Postconditions: None

Expected Result: The interactive lessons should load correctly and function as expected across all devices, maintaining usability and a consistent user experience.

Severity: Major

Type of Testing: Compatibility Testing, Usability Testing

Test Case Approach: Positive

Data - 51

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Personalized Learning Paths

Feature description -

Learners have the flexibility to tailor their learning journey by selecting courses, modules, and activities aligned with their unique interests, preferences, and learning objectives. This personalized approach enables learners to pursue their educational goals at their own pace and according to their individual learning styles.

Userstory title -

As a learner, I want personalized learning paths so that I can focus on topics that interest me and progress at my own pace.

Acceptance criteria -

['Learners can choose courses and modules based on their interests.', 'Personalized learning paths adjust dynamically based on learner progress.', 'Learners receive recommendations for new courses and modules.', "Progress tracking is personalized to each learner's path.", 'Learners can adjust their learning paths at any time.']

Generated Test Cases:

## Test Cases for Personalized Learning Paths:

Test Case 1:

Test Case ID: PLP-001

Test Case Title: Verify that learners can choose courses and modules based on their interests.

Test Case Description: This test case verifies that the platform allows learners to browse and select courses and modules based on their chosen interests. It ensures that the selection process is intuitive and that chosen courses are added to the learner's personalized learning path.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

  - Learner is logged in.
  - Learner has completed the initial interest selection process.

Test Data: No test data needed

Test Steps:

  1. Navigate to the "Browse Courses" or "Explore" section.
  2. Observe the displayed courses/modules, ensuring they align with the learner's chosen interests.
  3. Select a course/module that aligns with the learner's interests.
  4. Click "Add to My Learning Path" or a similar action button.
  5. Navigate to "My Learning Path."

Postconditions:

  - Verify that the selected course/module appears in the learner's "My Learning Path" section.

Expected Result: The system should successfully display courses matching the learner's interests, and allow learners to add them to their personalized learning path.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: PLP-002

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress.

Test Case Description: This test case ensures that the learning path dynamically updates based on the learner's progress. This involves marking modules as complete, tracking scores, and adjusting future recommendations accordingly.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has started a course/module on their personalized learning path.

Test Data: No test data needed

Test Steps:

1. Access a course/module within "My Learning Path" and complete a learning activity.
2. Achieve a passing score or mark the activity as complete.
3. Observe any changes in the personalized learning path structure, recommended next steps, or overall progress indicators.

Postconditions:

- Verify that the system marks the completed activity accordingly.
- Check for updates in the personalized learning path reflecting the completed activity.

Expected Result: The system should accurately track progress and dynamically adjust the learning path, potentially suggesting more advanced modules or related topics based on the learner's performance.

Severity: Major

Type of Testing: Functional Testing, Regression Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: PLP-003

Test Case Title: Verify that learners receive recommendations for new courses and modules.

Test Case Description: This test case validates the recommendation engine for suggesting new learning content. It assesses the relevance of recommendations based on the learner's chosen interests, learning path progress, and potentially, popular courses within the platform.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- Learner is logged in.

- Learner has an active personalized learning path.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Recommendations" or "Suggested Courses" section (or a similar area).

2. Observe the recommended courses/modules.

3. Check if the recommendations align with the learner's initial interests and current progress within their chosen courses.

Postconditions: N/A

Expected Result: The platform should provide relevant course/module recommendations aligned with the learner's interests and learning progress.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: PLP-004

Test Case Title: Verify that progress tracking is personalized to each learner's path.

Test Case Description: This test case confirms that progress tracking accurately reflects the learner's individual path. This means the system should not just track overall course completion but also consider modules within the personalized path, providing a granular view of the learner's advancement.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in.

- Learner has made progress within their personalized learning path (completed modules, quizzes, etc.).

Test Data: No test data needed

Test Steps:

1. Go to the "My Progress" or "Dashboard" section.
2. Verify that the displayed progress reflects completed modules, scores, and overall advancement within the personalized learning path.
3. Compare this displayed progress with the actual progress made by the learner.

Postconditions: N/A

Expected Result: The system should accurately and individually track the learner's progress within their personalized learning path, presenting a clear overview of their achievements.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: PLP-005

Test Case Title: Verify that learners can adjust their learning paths at any time.

Test Case Description: This test case ensures the flexibility of the platform by allowing learners to modify their chosen paths. This includes adding or removing courses, rearranging the learning order, and ensuring these changes are reflected accurately in the platform.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- Learner is logged in.
- Learner has an active personalized learning path with at least one course/module.

Test Data: No test data needed

Test Steps:

1. Navigate to "My Learning Path."
2. Attempt to remove an existing course/module.
3. Add a new course/module to the path.
4. Rearrange the order of courses/modules if the platform allows it.

Postconditions:

- Verify that the changes made to the learning path are saved and reflected accurately.

Expected Result: The system should allow learners to adjust their learning path flexibly, saving any modifications made and providing a seamless user experience.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

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 Cases for Online Learning Platform - Progress Tracking Feature

Test Case Creation Template:

Test Case ID	Test Case Title	Test Case Description	Test Suite	Test Priority	Preconditions	Test Data	Test Steps	Postconditions	Expected Result	Severity	Type of Testing	Test Case Approach
OLPP-001	Verify that progress tracking shows completed activities and grades.	This test case verifies that the progress tracking feature accurately displays a list of completed activities within a course, along with the grades achieved for each activity.	Progress Tracking	High	- User is logged in. - User is enrolled in a course with at least one completed activity.	No test data needed	1. Navigate to the course page. 2. Access the "Progress" section. 3. Verify that completed activities are listed. 4. Verify that the correct grades are displayed for each completed activity.	- None	The system accurately displays a list of completed activities and their corresponding grades.	Major	Functional Testing	Positive
OLPP-002	Verify that learners can view detailed reports on their progress.	This test case verifies that learners can access and view detailed reports that provide comprehensive insights into their overall progress within a course, including graphical representations and breakdowns of performance.	Progress Tracking

High | - User is logged in. - User has progressed sufficiently in a course to generate data for a detailed report. | No test data needed | 1. Navigate to the course page. 2. Access the "Progress" section. 3. Locate and select the option to view a detailed progress report. 4. Verify that the report displays comprehensive information, including overall progress, grades, time spent, and potentially areas for improvement. | - None | The system generates a detailed progress report containing comprehensive information about the learner's performance. | Major | Functional Testing | Positive |

| OLPP-003 | Verify that achievements are highlighted in the progress tracker. | This test case checks if the platform effectively highlights achievements, such as badges or milestones, within the progress tracker, providing learners with visual cues of their accomplishments. | Progress Tracking | Medium | - User is logged in. - User has earned at least one achievement in a course. | No test data needed | 1. Navigate to the course page. 2. Access the "Progress" section. 3. Verify that earned achievements are prominently displayed. 4. Verify that achievement details are accessible (e.g., clicking on the achievement provides more information). | - None | Earned achievements are visually highlighted within the progress tracker, enhancing the user's sense of accomplishment. | Minor | Functional Testing | Positive |

| OLPP-004 | Verify that learners receive notifications about their progress. | This test case ensures that the system can successfully send notifications to learners regarding their progress, including updates on grades, completion of milestones, or areas requiring attention. | Progress Tracking | Medium | - User is logged in. - User has enabled notifications for progress updates. - The system has predefined triggers for progress notifications (e.g., achieving a certain grade, completing a module). | No test data needed | 1. Trigger a progress notification event (e.g., complete a quiz with a high score). 2. Verify that the user receives a notification. 3. Verify that the notification content accurately reflects the progress update. 4. Check the notification delivery method (e.g., in-app, email). | - None | Learners receive timely and accurate notifications about their progress, delivered through their preferred channels. | Major | Functional Testing | Positive |

| OLPP-005 | Verify that progress tracking can be exported as reports. | This test case verifies the functionality to export progress tracking data in a commonly used report format (e.g., PDF, CSV) for offline viewing or sharing purposes. | Progress Tracking | Medium | - User is logged in. - User has progress data available for export. | No test data needed | 1. Navigate to the progress tracking section. 2. Locate and select the option to

export progress data. 3. Choose the desired report format (e.g., PDF, CSV). 4. Download the exported report. 5. Verify that the downloaded report contains accurate and well-formatted progress information. | - None | Learners can easily export their progress tracking data in a user-friendly format for offline viewing or sharing. | Minor | Functional Testing | 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 Cases for Online Learning Platform - Assessment and Grading

Test Case 1:

Test Case ID: OLPAssess-001

Test Case Title: Verify that instructors can create different types of assessments.

Test Case Description: This test case ensures that instructors can successfully create quizzes, tests, and assignments within the platform.

Test Suite: Assessment Creation

Test Priority: High

Preconditions:

* Instructor is logged in.

* A course is created.

Test Data: No test data needed.

Test Steps:

1. Navigate to the course page.

2. Access the assessment creation section.

3. Select 'Create Quiz', provide a title, description, and save.

4. Repeat steps 2-3 for 'Create Test' and 'Create Assignment'.

Postconditions:

* The platform should display the newly created quiz, test, and assignment within the course.

Expected Result: The system successfully allows the creation of different assessment types (quiz, test, assignment).

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPAssess-002

Test Case Title: Verify that assessments can include various question types.

Test Case Description: This test case verifies that instructors can add different question types to their assessments, including MCQs, short answers, and essays.

Test Suite: Question Type Variety

Test Priority: High

Preconditions:

* Instructor is logged in.

* An assessment (quiz/test/assignment) is created.

Test Data: Sample questions for each question type: MCQ, short answer, essay.

Test Steps:

1. Open the created assessment for editing.

2. Add a new question and select 'Multiple Choice' as the type.

3. Input question, options, and correct answer. Save the question.

4. Repeat steps 2-3 for 'Short Answer' and 'Essay' question types.

Postconditions:

* The assessment should contain at least one question of each tested type.

Expected Result: The system allows the addition of different question types (MCQ, short answer, essay) to assessments.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPAssess-003

Test Case Title: Verify that the system provides automatic grading for objective questions.

Test Case Description: This test case ensures the platform accurately auto-grades objective questions (MCQs) and provides immediate feedback to learners upon submission.

Test Suite: Automatic Grading

Test Priority: High

Preconditions:

* Instructor is logged in.

* An assessment with at least one MCQ question is created and published.

* A learner is enrolled in the course.

Test Data: Correct and incorrect answers for the MCQ question.

Test Steps:

1. Log in as the enrolled learner.

2. Access the published assessment and answer the MCQ question.

3. Submit the assessment.

Postconditions:

* The learner should be able to view their score for the MCQ question immediately.

Expected Result: The system automatically grades the MCQ question, providing immediate feedback to the learner.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPAssess-004

Test Case Title: Verify that the system allows manual grading for subjective questions.

Test Case Description: This test case checks that instructors can manually grade subjective questions (short answers, essays) and provide feedback to learners.

Test Suite: Manual Grading

Test Priority: High

Preconditions:

* Instructor is logged in.

* An assessment with at least one short answer or essay question is created and published.

* A learner submits the assessment with answers to the subjective question(s).

Test Data: Learner's submitted answers for the subjective question(s).

Test Steps:

1. As the instructor, access the submitted assessment.

2. Navigate to the subjective question(s).

3. Input a grade and provide written feedback for the learner.

4. Save the grading and feedback.

Postconditions:

* The learner should be able to view the grade and feedback for the subjective question.

Expected Result: The system successfully allows manual grading and feedback provision for subjective questions.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPAssess-005

Test Case Title: Verify that feedback is provided to learners after grading.

Test Case Description: This test case confirms that learners receive feedback (both automated and manual) on their submitted assessments.

Test Suite: Feedback Delivery

Test Priority: High

Preconditions:

* An assessment with both objective and subjective questions is created, published, and graded.

* A learner has submitted the assessment.

Test Data: No test data needed.

Test Steps:

1. Log in as the learner who submitted the assessment.

2. Navigate to the 'Grades' or 'Submissions' section.

3. View the graded assessment.

Postconditions:

* The platform should clearly display the learner's grade and any provided feedback.

Expected Result: The learner can access their grade and detailed feedback (automatic for objective questions, manual for subjective questions) on the submitted assessment.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

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 Cases for Online Learning Platform - Discussion Forums:

Test Case 1:

* **Test Case ID:** OLPD-Forum-001

* **Test Case Title:** Verify that learners can post questions and responses in forums

* **Test Case Description:** This test case verifies that learners can successfully post new questions and responses within a specific course forum.

* **Test Suite:** Discussion Forums

* **Test Priority:** High
* **Preconditions:**
    * Learner is logged in.
    * Learner has joined the course with the active forum.
* **Test Data:**
    * Sample question: "Can someone explain the concept discussed in the last lecture?"
    * Sample response: "I found this resource helpful for understanding the concept: [link]"
* **Test Steps:**
    1. Navigate to the desired course page.
    2. Access the discussion forum for that course.
    3. Click on the "Create New Post" button (or similar).
    4. Input the sample question in the post body.
    5. Submit the question to the forum.
    6. Locate an existing post and click "Reply."
    7. Input the sample response in the reply field.
    8. Submit the response.
* **Postconditions:**
    * The question and response are visible to other forum participants.
* **Expected Result:** The learner can successfully post both a new question and a response within the forum.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** OLPD-Forum-002

* **Test Case Title:** Verify that forums support threaded discussions

* **Test Case Description:** This test case checks the functionality of threaded discussions, ensuring replies are nested correctly and visually associated with the original post.

* **Test Suite:** Discussion Forums

* **Test Priority:** High

* **Preconditions:**

    * Forum has at least one post with multiple replies.

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

* **Test Steps:**

    1. Navigate to a forum with existing posts and replies.

    2. Identify a post with multiple replies.

    3. Observe the visual representation of the replies.

    4. Verify that replies are indented or visually nested under the original post.

    5. Attempt to reply to both the original post and an existing reply.

* **Postconditions:** None

* **Expected Result:** Replies are clearly associated with the correct parent post, creating a clear and understandable threaded discussion flow.

* **Severity:** Major

* **Type of Testing:** Functional, Usability Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** OLPD-Forum-003

* **Test Case Title:** Verify that instructors can moderate discussions

* **Test Case Description:** This test case confirms that instructors have the necessary permissions to perform moderation actions within the forum, ensuring a respectful and constructive learning environment.
* **Test Suite:** Discussion Forums
* **Test Priority:** High
* **Preconditions:**
    * An instructor account is logged in.
    * A forum exists with at least one student post.
* **Test Data:**
    * Sample inappropriate post: "This course is terrible!"
* **Test Steps:**
    1. Locate a student's post within the forum.
    2. Attempt to delete the post as the instructor.
    3. Attempt to edit the content of the student's post.
    4. Locate the "Report" or "Flag" functionality (if available).
    5. Attempt to respond to the student's post as the instructor.
* **Postconditions:**
    * The inappropriate post is removed or hidden from the forum.
* **Expected Result:** The instructor can successfully delete, edit, and respond to posts, demonstrating moderation capabilities.
* **Severity:** Critical
* **Type of Testing:** Functional, Security Testing
* **Test Case Approach:** Positive, Negative (attempting actions beyond moderation permissions)

Test Case 4:

* **Test Case ID:** OLPD-Forum-004

* **Test Case Title:** Verify that learners receive notifications for new posts
* **Test Case Description:** This test case ensures learners are notified about new posts in the forums they participate in, keeping them engaged and informed.
* **Test Suite:** Discussion Forums
* **Test Priority:** Medium
* **Preconditions:**
    * Learner account is logged in and has notification settings enabled.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. As a different user, post a new question or reply to an existing thread in a forum the learner is participating in.
    2. Observe if the learner receives a notification (in-app or via email, depending on settings).
    3. Verify the notification content accurately reflects the new post in the forum.
    4. Check if the notification directs the learner to the correct forum thread upon clicking.
* **Postconditions:** None
* **Expected Result:** Learners receive timely and accurate notifications about new posts, promoting engagement and facilitating active participation in discussions.
* **Severity:** Minor
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** OLPD-Forum-005
* **Test Case Title:** Verify that forums are accessible on all devices
* **Test Case Description:** This test confirms the responsiveness of the discussion forum feature, ensuring

accessibility and usability across a range of devices and screen sizes.

* **Test Suite:** Discussion Forums

* **Test Priority:** High

* **Preconditions:**

    * A stable internet connection is available.

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

* **Test Steps:**

    1. Access the online learning platform and navigate to a discussion forum using a desktop computer.

    2. Repeat step 1 using a smartphone (iOS and Android).

    3. Repeat step 1 using a tablet device.

    4. On each device, attempt to navigate the forum, read posts, create new posts, and reply to existing threads.

    5. Observe the layout, formatting, and functionality of the forum on each device.

* **Postconditions:** None

* **Expected Result:** The discussion forum should be accessible and fully functional on all devices (desktop, mobile, tablet) without significant layout issues or loss of functionality.

* **Severity:** Major

* **Type of Testing:** Usability Testing, Compatibility Testing

* **Test Case Approach:** Positive

Data - 55

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Live Classes and Webinars

Feature description -

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

Userstory title -

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

Acceptance criteria -

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Live Classes and Webinars:

Test Case 1:

* **Test Case ID:** OLP-LCW-001
* **Test Case Title:** Verify that learners can join live classes and webinars through video conferencing tools.
* **Test Case Description:** This test case verifies that learners can successfully access and participate in live classes and webinars using the integrated video conferencing tools.
* **Test Suite:** Live Classes and Webinars

* **Test Priority:** High

* **Preconditions:**

    * Learner is logged in.

    * Learner has registered for the live class/webinar.

    * The live session is scheduled and active.

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

* **Test Steps:**

    1. Navigate to the upcoming live classes/webinars section.

    2. Select the desired live class/webinar to join.

    3. Click on the "Join Session" button.

    4. Verify successful connection to the video conferencing tool and access to the session.

* **Postconditions:**

    * None

* **Expected Result:** The learner should be able to join the live session seamlessly through the integrated video conferencing tool. Audio and video should function correctly.

* **Severity:** Critical

* **Type of Testing:** Functional, Integration Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** OLP-LCW-002

* **Test Case Title:** Verify that learners can ask questions in real-time during live sessions.

* **Test Case Description:** This test case ensures that learners can actively engage in live sessions by submitting questions to the instructor through a real-time chat or Q&A feature.

* **Test Suite:** Live Classes and Webinars

* **Test Priority:** High

* **Preconditions:**
    * Learner has joined a live class/webinar.
    * Real-time chat or Q&A feature is enabled for the session.
* **Test Data:** Sample questions to be asked.
* **Test Steps:**
    1. Locate the chat or Q&A section within the video conferencing interface.
    2. Type a question in the designated area.
    3. Submit the question.
    4. Observe if the question is visible to the instructor and other participants (if applicable).
* **Postconditions:**
    * None
* **Expected Result:** The learner's question should be successfully submitted and visible within the designated chat or Q&A area.
* **Severity:** Major
* **Type of Testing:** Functional, Usability Testing
* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** OLP-LCW-003
* **Test Case Title:** Verify that live sessions are recorded and available for later viewing.
* **Test Case Description:** This test case confirms that completed live classes and webinars are recorded and accessible to registered learners for on-demand viewing.
* **Test Suite:** Live Classes and Webinars
* **Test Priority:** Medium
* **Preconditions:**
    * A live class/webinar session has concluded.

   * The session was set to be recorded.

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

* **Test Steps:**

   1. Navigate to the completed live classes/webinars section.

   2. Locate the specific recorded session.

   3. Attempt to access and play the recording.

* **Postconditions:**

   * None

* **Expected Result:** The recorded session should be available for viewing, with functional playback controls.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** OLP-LCW-004

* **Test Case Title:** Verify that instructors can share their screen and resources during live sessions.

* **Test Case Description:** This test case validates the screen sharing and resource sharing functionality for instructors during live sessions.

* **Test Suite:** Live Classes and Webinars

* **Test Priority:** Medium

* **Preconditions:**

   * Instructor is logged in and has initiated a live class/webinar.

   * Test files or resources are prepared for sharing.

* **Test Data:** Sample presentation slides, documents, or images.

* **Test Steps:**

1. As the instructor, locate and access the screen sharing functionality within the video conferencing tool.
2. Select the desired screen or application window to share.
3. Verify if the shared content is visible to the learners.
4. Test uploading and sharing additional resources like files or documents through the available tools.

* **Postconditions:**
    * End the screen sharing and remove shared resources.
* **Expected Result:** The instructor should be able to share their screen or specific application windows effectively. Learners should be able to view the shared content without issues. Uploaded resources should be accessible.
* **Severity:** Major
* **Type of Testing:** Functional, Integration Testing
* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** OLP-LCW-005
* **Test Case Title:** Verify that learners receive notifications before live sessions start.
* **Test Case Description:** This test case ensures the system sends timely notifications to remind learners of upcoming live sessions they have registered for.
* **Test Suite:** Live Classes and Webinars
* **Test Priority:** Medium
* **Preconditions:**
    * Learner is logged in and registered for an upcoming live session.
    * Notifications are enabled in user settings.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Verify the notification settings for live classes/webinars are enabled.

2. Observe the delivery of notifications within a reasonable timeframe before the scheduled start of the session.

3. Check if notifications contain essential details such as session title, date, and time.

* **Postconditions:** None

* **Expected Result:** Learners should receive clear and accurate notifications about upcoming live sessions they are registered for, delivered via their preferred notification channels (e.g., email, platform notifications).

* **Severity:** Minor

* **Type of Testing:** Functional, Usability Testing

* **Test Case Approach:** Positive

--

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 Cases for Online Learning Platform - Mobile Accessibility:

Test Case 1:

Test Case ID: OLP-MA-001

Test Case Title: Verify that the platform is accessible on smartphones.

Test Case Description: This test case aims to confirm that learners can access the online learning platform on various smartphone devices and operating systems without encountering any compatibility issues.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Stable internet connection is available.

- A range of smartphones with different operating systems (iOS, Android) and screen sizes are available.

Test Data: No test data needed

Test Steps:

1. Open the learning platform website on a smartphone using a mobile browser.

2. Observe the platform's loading time and responsiveness.

3. Navigate through different sections like courses, lectures, quizzes.

4. Attempt to log in (if applicable) and access course materials.

5. Verify the display and functionality of interactive elements.

Postconditions:

- No significant performance issues encountered.

- Platform elements are displayed and function correctly.

Expected Result: The platform should be accessible, responsive, and functional on different smartphones, providing a seamless user experience.

Severity: Major

Type of Testing: Compatibility Testing, Usability Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLP-MA-002

Test Case Title: Verify that the platform is accessible on tablets.

Test Case Description: This test case ensures the online learning platform functions as intended on various tablet devices, considering different operating systems, screen sizes, and orientations.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Active internet connection is available.

- A variety of tablets with different operating systems (iOS, Android) and screen sizes are available.

Test Data: No test data needed

Test Steps:

1. Access the online learning platform website on a tablet using its web browser.

2. Evaluate the platform's loading speed and overall responsiveness.

3. Navigate through different sections, including courses, lectures, quizzes, and the forum.

4. Test the platform's behavior in both portrait and landscape orientations.

5. Attempt to play video lectures, access downloadable resources, and participate in the discussion forum.

Postconditions:

- No significant performance bottlenecks observed.

- The platform's elements display and function correctly.

Expected Result: The online learning platform should be accessible, responsive, and functional on different tablet devices, ensuring a smooth user experience across various screen sizes and orientations.

Severity: Major

Type of Testing: Compatibility Testing, Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLP-MA-003

Test Case Title: Verify that the mobile UI is optimized for touch navigation.

Test Case Description: This test case aims to confirm that all interactive elements on the platform respond accurately and efficiently to touch inputs, ensuring a user-friendly experience on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- A smartphone or tablet with touch screen capabilities.

- The platform is accessible on the chosen device.

Test Data: No test data needed

Test Steps:

1. Access different sections of the platform on a smartphone or tablet.

2. Test tapping, swiping, and scrolling gestures.

3. Interact with buttons, links, menus, and input fields.

4. Assess the responsiveness and accuracy of each interaction.

5. Verify if elements like dropdown menus and sliders are easily navigable using touch gestures.

Postconditions: No unexpected behavior or delays encountered.

Expected Result: All interactive elements should respond correctly and promptly to touch gestures without any lag or misinterpretation, providing a smooth and intuitive user experience.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLP-MA-004

Test Case Title: Verify that course materials are accessible and readable on mobile devices.

Test Case Description: This test confirms that various course materials, including text, images, videos, and downloadable documents, are appropriately formatted, displayed, and accessible on mobile devices without compromising readability or quality.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- The learning platform is accessible on a mobile device.

- At least one course with various learning materials is available.

Test Data: Different types of course materials - text, images, videos, downloadable documents (PDFs, PPTs).

Test Steps:

1. Access a course on the platform using a mobile device.

2. Open and interact with different types of learning materials.

3. Check the readability of the text, clarity of images, and playback functionality of videos.

4. Download a document and verify its accessibility and formatting.

5. Rotate the device to ensure responsive design and readability in both portrait and landscape modes.

Postconditions: All materials should remain accessible and functional.

Expected Result: All course materials should be displayed clearly and formatted correctly on the mobile device, ensuring easy readability, accessibility, and optimal user experience. Videos should play without issues, and downloaded documents should maintain their formatting.

Severity: Major

Type of Testing: Compatibility Testing, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP-MA-005

Test Case Title: Verify that the platform supports offline access to downloaded materials.

Test Case Description: This test case validates that learners can access downloaded course materials offline, ensuring continuous learning even without an active internet connection.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- The platform is accessible on a mobile device.

- The user is logged in and has downloaded course materials.

Test Data: Previously downloaded course materials.

Test Steps:

1. Download various learning materials (documents, videos) while online.

2. Turn off the internet connection on the mobile device.

3. Access the downloaded course materials within the platform.

4. Verify that documents are accessible and videos can be played offline.

5. Attempt to access materials that haven't been downloaded.

Postconditions: Internet connection can be restored.

Expected Result: All downloaded course materials should be accessible offline. The platform should clearly indicate unavailable materials that haven't been downloaded, ensuring a transparent and user-friendly experience.

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 Cases for Online Learning Platform - Resource Library

Test Case 1:

Test Case ID: RL-001

Test Case Title: Verify that the resource library includes e-books, articles, and reference guides.

Test Case Description: This test case aims to confirm that the resource library contains the specified types of materials: e-books, articles, and reference guides, ensuring a diverse range of supplementary content for learners.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has accessed a course with an active resource library.

Test Data: No test data needed.

Test Steps:

1. Navigate to the course page.
2. Access the "Resource Library" section.
3. Verify the presence of sections or filters for "E-books," "Articles," and "Reference Guides."
4. Randomly select and open items from each category to confirm content type.

Postconditions: None.

Expected Result: The resource library should clearly display and provide access to e-books, articles, and reference guides.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: RL-002

Test Case Title: Verify that learners can search for specific materials in the resource library.

Test Case Description: This test case ensures the functionality of the resource library's search feature, allowing learners to find specific materials using keywords, titles, or author names.

Test Suite: Resource Library

Test Priority: High

Preconditions:

  - Learner is logged in.
  - Learner has accessed the resource library.

Test Data:

  - Keywords related to existing resources (e.g., topic, author name, title).

Test Steps:

  1. Locate the search bar within the resource library.
  2. Enter a relevant keyword, title, or author name.
  3. Initiate the search.
  4. Verify that the displayed results match the search query.
  5. Open a result to confirm its relevance to the search term.

Postconditions: None.

Expected Result: The search function should accurately return relevant resources based on the provided search criteria.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: RL-003

Test Case Title: Verify that resources are categorized and tagged for easy navigation.

Test Case Description: This test case checks if the resources are well-organized within the library using categories and tags, enabling learners to browse and locate materials efficiently.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- Learner is logged in.
- Learner is within the resource library.

Test Data: No test data needed.

Test Steps:

1. Observe if resources are categorized (e.g., by topic, type, level).
2. Check for the presence of tags associated with resources.
3. Select a category and verify that the displayed resources belong to that category.
4. Click on a tag and confirm that the results include resources associated with that tag.

Postconditions: None.

Expected Result: Resources should be logically categorized and tagged, facilitating easy navigation and discovery for learners.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: RL-004

Test Case Title: Verify that learners can download resources for offline access.

Test Case Description: This test case confirms the functionality of downloading resources, enabling learners to access materials offline and enhancing learning flexibility.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has selected a downloadable resource.

Test Data: No test data needed.

Test Steps:

1. Open a resource that offers a download option (e.g., e-book, PDF article).
2. Click on the download button or link.
3. Choose a save location and confirm the download.
4. Verify that the downloaded file is accessible offline.

Postconditions:

- Delete the downloaded file to maintain a clean testing environment.

Expected Result: Learners should be able to download and access resources offline without any errors.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: RL-005

Test Case Title: Verify that instructors can recommend resources to learners.

Test Case Description: This test case ensures that instructors have the ability to recommend specific resources to learners, enhancing the learning experience by providing curated content suggestions.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- Instructor is logged in.
- Instructor has access to the resource library and course management features.

Test Data:

- A selected resource to recommend.

Test Steps:

1. Navigate to the course management or resource library section as an instructor.
2. Locate the option to recommend or suggest resources.
3. Select a specific resource from the library.
4. Add a brief message or reason for recommendation (optional).
5. Submit the recommendation.
6. Verify that learners can view the recommended resource within the course or designated area.

Postconditions:

- Remove the recommendation to maintain a clean testing environment.

Expected Result: Instructors should be able to recommend specific resources, and learners should be able to view these recommendations within the platform.

Severity: Minor

Type of Testing: Functional Testing

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 Cases for Online Learning Platform - Peer Review Feature

Test Case 1:

Test Case ID: PR-001

Test Case Title: Verify that learners can submit assignments for peer review.

Test Case Description: This test case verifies that learners can successfully submit their completed assignments for peer review within a specific course.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has joined a course with peer review enabled.
- Learner has completed an assignment.

Test Data: A completed assignment file (e.g., document, presentation, spreadsheet).

Test Steps:

1. Navigate to the desired course.
2. Locate the assignment designated for peer review.
3. Click on the "Submit for Peer Review" button.
4. Upload the completed assignment file.
5. Confirm submission.

Postconditions:

- The assignment is successfully submitted for peer review.
- The learner receives a confirmation message or notification.

Expected Result: The system should allow learners to submit their assignments for peer review without encountering errors.

Severity: Major

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case 2:

Test Case ID: PR-002

Test Case Title: Verify that peers can provide feedback on submitted assignments.

Test Case Description: This test case ensures that learners assigned as reviewers can access, review, and provide feedback on submitted assignments.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Reviewer is logged in.
- Reviewer is assigned an assignment for review.

Test Data: No test data needed.

Test Steps:

1. Navigate to the peer review section of the course.
2. Access the assigned assignment for review.
3. Review the submitted assignment content.
4. Provide feedback using the provided tools (e.g., comments, ratings, rubrics).
5. Submit the feedback.

Postconditions:

- Feedback is successfully submitted for the assigned assignment.

Expected Result: The platform should allow reviewers to provide comprehensive feedback using the available tools.

Severity: Major

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case 3:

Test Case ID: PR-003

Test Case Title: Verify that a structured review process guides the feedback.

Test Case Description: This test case validates that the peer review process provides a structured framework or guidelines to ensure consistent and effective feedback.

Test Suite: Peer Review

Test Priority: Medium

Preconditions:

- Learner is logged in and participating in peer review.

Test Data: No test data needed.

Test Steps:

1. Access an assignment open for peer review.
2. Review the provided guidelines, rubrics, or criteria for providing feedback.
3. Attempt to submit feedback without adhering to the structured format.

Postconditions: N/A

Expected Result: The system should enforce the structured review process, prompting users to adhere to the provided guidelines and preventing submission of incomplete or unstructured feedback.

Severity: Minor

Type of Testing: Usability, Functional

Test Case Approach: Negative

Test Case 4:

Test Case ID: PR-004

Test Case Title: Verify that learners can view feedback on their work.

Test Case Description: This test case ensures that learners can access and view the feedback provided by their peers on their submitted assignments.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in.

- Learner has submitted an assignment for peer review.

- Feedback has been provided on the submitted assignment.

Test Data: No test data needed.

Test Steps:

1. Navigate to the submitted assignment within the course.

2. Access the feedback section for the assignment.

3. View the provided feedback from peers.

Postconditions: N/A

Expected Result: The system should display the received feedback clearly and comprehensively, allowing learners to understand their strengths and areas for improvement.

Severity: Major

Type of Testing: Functional, System

Test Case Approach: Positive

Test Case 5:

Test Case ID: PR-005

Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback.

Test Case Description: This test case validates that the platform maintains the anonymity of reviewers throughout the peer review process, ensuring unbiased feedback.

Test Suite: Peer Review

Test Priority: Medium

Preconditions:

- Multiple learners are participating in peer review.

- Feedback has been provided on submitted assignments.

Test Data: No test data needed.

Test Steps:

1. As a learner, submit an assignment for peer review.

2. As another learner, provide feedback on the submitted assignment.

3. Log in as the original learner (submitter).

4. View the received feedback.

Postconditions: N/A

Expected Result: The system should not reveal the identities of the reviewers during any stage of the process, promoting impartial and constructive feedback.

Severity: Major

Type of Testing: Security, Functional

Test Case Approach: Positive

Data - 59

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Certificates and Badges

Feature description -

Upon successful completion of courses or specific milestones, learners receive certificates and badges to recognize their achievements and accomplishments. These credentials provide tangible recognition of learners' efforts, skills, and expertise, enhancing their credentials and demonstrating their commitment to continuous learning and professional development.

Userstory title -

As a learner, I want to earn certificates and badges so that I can showcase my achievements and skills.

Acceptance criteria -

['Learners earn certificates upon course completion.', 'Badges are awarded for specific achievements and milestones.', 'Certificates and badges are accessible through learner profiles.', 'Learners can share their certificates and badges on social media.', 'Instructors can issue custom badges for special achievements.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Certificates and Badges

Test Case 1

Test Case ID: OLP-CB-001

Test Case Title: Verify that learners earn certificates upon course completion.

Test Case Description: This test case verifies that the system correctly issues certificates to learners who successfully complete a course, meeting all requirements. This includes completing all modules, passing required assessments, and achieving a passing grade.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

* Learner is logged in.

* Learner has enrolled in a course.

* Learner has completed all course requirements.

Test Data: No test data needed.

Test Steps:

1. Navigate to the course the learner has completed.

2. Verify the course completion status is displayed as "Completed."

3. Navigate to the learner's profile.

4. Verify that a certificate for the completed course is available.

Postconditions: None

Expected Result: The learner's profile should display a downloadable certificate for the completed course, confirming their achievement.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: OLP-CB-002

Test Case Title: Verify that badges are awarded for specific achievements and milestones.

Test Case Description: This test case verifies that the system automatically awards badges to learners when they achieve specific milestones or accomplish predefined achievements within courses.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

* Learner is logged in.

* Learner has joined a course with defined badge criteria.

Test Data:

* Badge criteria (e.g., completing a specific module, achieving a high score on a quiz).

Test Steps:

1. As a learner, complete the action that fulfills the badge criteria.

2. Navigate to the learner's profile.

3. Verify that the corresponding badge is displayed on the learner's profile.

Postconditions: None

Expected Result: The system should automatically award the badge to the learner upon meeting the specific requirements.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: OLP-CB-003

Test Case Title: Verify that certificates and badges are accessible through learner profiles.

Test Case Description: This test case verifies that learners can easily access and view their earned certificates and badges through a dedicated section on their profiles.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

* Learner is logged in.

* Learner has earned at least one certificate or badge.

Test Data: No test data needed.

Test Steps:

1. Go to the learner's profile.

2. Locate the section dedicated to certificates and badges.

3. Verify that all earned certificates and badges are displayed correctly.

Postconditions: None

Expected Result: The learner's profile should have a dedicated section that clearly displays all earned certificates and badges. Each item should be clearly identifiable and accessible for viewing or downloading.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: OLP-CB-004

Test Case Title: Verify that learners can share their certificates and badges on social media.

Test Case Description: This test case checks the functionality for learners to share their earned certificates and badges on various social media platforms, promoting their achievements and the learning platform.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

* Learner is logged in.

* Learner has earned at least one certificate or badge.

* Social media sharing options are enabled.

Test Data: No test data needed

Test Steps:

1. Navigate to the learner's profile and access the certificates and badges section.

2. Select a certificate or badge to share.

3. Choose a social media platform from the available options.

4. Verify that the sharing functionality redirects to the chosen platform with the correct information.

Postconditions: None

Expected Result: The learner should be able to seamlessly share their selected certificate or badge on the chosen social media platform with a pre-populated message or image.

Severity: Minor

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: OLP-CB-005

Test Case Title: Verify that instructors can issue custom badges for special achievements.

Test Case Description: This test case confirms if instructors have the capability to create and award custom badges to learners for unique achievements or exceptional performance within their courses.

Test Suite: Certificates and Badges

Test Priority: Low

Preconditions:

* Instructor is logged in.

* Instructor has necessary permissions to manage badges.

Test Data:

* Badge title

* Badge description

* Badge criteria

Test Steps:

1. Navigate to the course management section as an instructor.

2. Locate the badge management area and select to create a new badge.

3. Input the badge details, including title, description, and criteria for earning.

4. Save the new custom badge.

5. Verify that the newly created badge is available for awarding to learners.

Postconditions: Remove the custom badge from the course.

Expected Result: The instructor should be able to successfully create and define a custom badge, which will then be available for awarding to learners in the course.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

--

Data - 60

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Gamification Elements

Feature description -

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

Userstory title -

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

Acceptance criteria -

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Gamification Elements

Test Case 1

Test Case ID: OLP-GF-001

Test Case Title: Verify that learners earn points for completing activities.

Test Case Description: This test case aims to confirm that the system accurately awards points to learners upon successful completion of various learning activities, such as watching video lectures, passing quizzes, and participating in forum discussions.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has access to learning activities.

Test Data: No test data needed.

Test Steps:

1. Access and complete a video lecture.
2. Successfully pass a quiz related to the course material.
3. Participate actively in a course-related forum discussion.
4. Navigate to the gamification dashboard.

Postconditions: None

Expected Result: The gamification dashboard reflects the awarded points for each completed activity:

watching video lectures, passing quizzes, and participating in forum discussions.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: OLP-GF-002

Test Case Title: Verify that levels are achieved based on accumulated points.

Test Case Description: This test case ensures that the system accurately tracks accumulated points and grants level advancements to learners when they reach pre-defined point thresholds.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has earned enough points to reach the next level.

Test Data: No test data needed.

Test Steps:

1. Complete various learning activities to accumulate points.
2. Reach the point threshold required for the next level.

Postconditions: None

Expected Result: The system automatically upgrades the learner's level upon reaching the required points and displays the new level on their profile and gamification dashboard.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: OLP-GF-003

Test Case Title: Verify that leaderboards display top-performing learners.

Test Case Description: This test case ensures the leaderboard functionality accurately ranks and displays learners based on their accumulated points, promoting healthy competition.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Multiple learners are participating in the platform.
- Learners have earned points through various activities.

Test Data: No test data needed

Test Steps:

1. Navigate to the leaderboard section of the platform.

Postconditions: None

Expected Result: The leaderboard displays learners ranked from highest to lowest based on their accumulated points. The leaderboard should be updated dynamically as learners earn more points.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: OLP-GF-004

Test Case Title: Verify that badges are awarded for reaching milestones.

Test Case Description: This test case ensures the system awards badges to learners who achieve specific milestones, such as completing a course, achieving a high score on a quiz, or maintaining a consistent learning streak.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Learner is logged in.
- Learner has met the pre-defined criteria for earning badges (e.g., completed a course).

Test Data: No test data needed

Test Steps:

1. Complete a course or achieve a specific milestone.
2. Navigate to the learner's profile or achievements section.

Postconditions: None

Expected Result: The system automatically awards the corresponding badge to the learner's profile. The badge should be visible on their profile and potentially on the leaderboard.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: OLP-GF-005

Test Case Title: Verify that learners can view their progress in a gamification dashboard.

Test Case Description: This test case ensures the gamification dashboard provides a comprehensive overview of the learner's progress, including their current level, accumulated points, earned badges, and leaderboard ranking.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is logged in.

Test Data: No test data needed

Test Steps:

1. Navigate to the gamification dashboard.

Postconditions: None

Expected Result: The dashboard displays the learner's current level, total accumulated points, earned badges, leaderboard position, and potentially a progress bar towards the next level. The information displayed should be accurate and up-to-date.

Severity: Major

Type of Testing: Functional Testing

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 Cases for Multilingual Support on Online Learning Platform:

Test Case 1:

Test Case ID: ML-001

Test Case Title: Verify that the platform interface is available in multiple languages.

Test Case Description: This test case aims to confirm that the user interface of the platform, including menus, buttons, and navigation elements, are correctly translated and displayed in the selected languages.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

  - Access to the online learning platform
  - A stable internet connection

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

Test Steps:

1. Navigate to the platform's homepage.
2. Locate the language selection option (e.g., dropdown menu, language icons).
3. Select a different language from the available options.
4. Observe the interface elements for correct translation and display in the chosen language.
5. Repeat steps 3-4 for other supported languages.

Postconditions: None

Expected Result: The platform's interface should accurately reflect the selected language without any display errors or missing translations.

Severity: Major

Type of Testing: Functional, Localization Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: ML-002

Test Case Title: Verify that course materials are available in different languages.

Test Case Description: This test case verifies that the platform supports course materials in multiple languages, including lecture notes, videos, quizzes, and downloadable resources.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- User is logged in.
- At least one course offers multilingual content.

Test Data: Course with multilingual materials available.

Test Steps:

1. Navigate to a course offering multilingual content.
2. Select the desired language for course materials.
3. Access different types of course materials, such as lectures, videos, quizzes, and downloads.
4. Verify that the selected content is displayed in the chosen language.

Postconditions: None

Expected Result: All course materials should be presented in the selected language, ensuring consistency and accessibility for learners.

Severity: Major

Type of Testing: Functional, Localization Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: ML-003

Test Case Title: Verify that learners can select their preferred language at any time.

Test Case Description: This test case ensures that users can seamlessly switch between preferred languages throughout their learning journey without encountering any disruptions or data loss.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- User is logged in.
- User has previously selected a preferred language.

Test Data: No test data needed.

Test Steps:

1. Navigate to different sections of the platform (e.g., dashboard, course catalog, profile settings).
2. Locate the language selection option.

3. Switch to a different language.
4. Verify that the language change is applied instantly across all platform sections.
5. Confirm that user progress and data are retained after the language switch.

Postconditions: None

Expected Result: The platform should dynamically adapt to the new language selection without affecting user data or requiring a restart.

Severity: Minor

Type of Testing: Functional, Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: ML-004

Test Case Title: Verify that multilingual support includes translated help and support resources.

Test Case Description: This test case ensures accessibility to translated help resources, guaranteeing that learners can find assistance in their preferred language.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- Access to the online learning platform.

Test Data: No test data needed.

Test Steps:

1. Navigate to the help and support section of the platform.
2. Check for language selection options within the help section.
3. Select a different language.
4. Verify that help articles, FAQs, and contact information are available in the chosen language.

Postconditions: None

Expected Result: All help and support resources should be translated into the selected languages, ensuring clarity and accessibility for learners seeking assistance.

Severity: Minor

Type of Testing: Functional, Localization Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: ML-005

Test Case Title: Verify that instructors can provide content in multiple languages.

Test Case Description: This test case aims to confirm that the platform allows instructors to upload and manage course materials in different languages, promoting inclusivity and accessibility for a wider audience.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- Instructor account is logged in.
- Access to course creation/editing features.

Test Data: Course materials in different languages.

Test Steps:

1. Navigate to the course creation/editing section.
2. Attempt to upload course materials in different languages (e.g., lecture notes in Spanish, subtitles for videos in French).
3. Verify that the platform accepts and processes these multilingual materials.
4. Publish the course and confirm that learners can access materials in their chosen language.

Postconditions: None

Expected Result: The platform should successfully store and display multilingual course materials as intended by the instructor, providing a seamless experience for learners across different languages.

Severity: Major

Type of Testing: Functional, Usability 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 Cases for Online Learning Platform - Accessibility Features

Test Case 1

Test Case ID: OLPAF-001

Test Case Title: Verify that the platform is compatible with screen readers.

Test Case Description: This test case aims to ensure that the online learning platform is fully navigable and usable with popular screen reader software.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- Screen reader software (e.g., NVDA, JAWS) is installed and running.
- The online learning platform is accessed using a compatible browser.

Test Data: No test data needed.

Test Steps:

1. Navigate through the platform's core features (e.g., course catalog, lectures, quizzes) using only keyboard shortcuts and screen reader commands.
2. Verify that all interactive elements (buttons, links, forms) are correctly identified and interpreted by the screen reader.
3. Check that the screen reader accurately describes images and other visual content using alt text.

Postconditions: None.

Expected Result: The screen reader should accurately interpret and vocalize all content and functionalities, allowing users to navigate and interact with the platform effectively.

Severity: Blocker

Type of Testing: Accessibility Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: OLPAF-002

Test Case Title: Verify that learners can adjust font sizes for better readability.

Test Case Description: This test case validates the functionality of adjusting font sizes within the platform to enhance readability for users with visual impairments.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- A user account on the online learning platform.

Test Data: No test data needed.

Test Steps:

1. Log in to the platform and navigate to the user settings or accessibility options.
2. Locate the font size adjustment controls.
3. Increase the font size to the maximum and minimum available options.
4. Observe if the font size changes are reflected accurately across all platform sections (e.g., course content, menus, forum posts).

Postconditions: None.

Expected Result: The platform should allow users to adjust font sizes smoothly and consistently, improving readability according to their needs.

Severity: Major

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: OLPAF-003

Test Case Title: Verify that color contrast options are available for improved visibility.

Test Case Description: This test ensures that the platform provides adjustable color contrast options to accommodate users with low vision or color blindness.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- A user account on the platform.

Test Data: No test data needed.

Test Steps:

1. Access the platform's user settings or accessibility options.
2. Locate the color contrast or theme settings.
3. Switch between different contrast themes (e.g., light mode, dark mode, high contrast).
4. Verify that the selected theme is applied consistently throughout the platform, including text, backgrounds, and interactive elements.

Postconditions: None.

Expected Result: The platform should offer a range of color contrast themes to enhance visual clarity and accessibility for users with varying visual needs.

Severity: Major

Type of Testing: Usability Testing, Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: OLPAF-004

Test Case Title: Verify that accessibility features are easy to activate and customize.

Test Case Description: This test assesses the ease of access and customization of the platform's accessibility features for users with disabilities.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- A user account on the platform.

Test Data: No test data needed.

Test Steps:

1. Locate and access the platform's accessibility settings.
2. Evaluate the clarity and conciseness of instructions for activating features (e.g., keyboard shortcuts, screen reader mode).
3. Test the customization options for each accessibility feature (e.g., adjusting font size increments, choosing color palettes).
4. Verify if the changes are saved and applied consistently across user sessions.

Postconditions: None.

Expected Result: Users should be able to locate, activate, and customize accessibility features without difficulty, ensuring a personalized and accessible learning experience.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: OLPAF-005

Test Case Title: Verify that support resources are available to assist learners with disabilities.

Test Case Description: This test confirms the availability and accessibility of support resources specifically designed for users with disabilities.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- Access to the online learning platform.

Test Data: No test data needed.

Test Steps:

1. Locate the platform's help center, FAQ section, or dedicated accessibility resources page.
2. Verify that the resources provide clear and concise information on using the platform's accessibility features.
3. Check for contact information for technical support or disability services related to the platform.
4. Evaluate if the resources are available in accessible formats (e.g., plain language, alternative text for images).

Postconditions: None.

Expected Result: The platform should offer comprehensive and easily accessible support resources that address the specific needs of learners with disabilities, ensuring a smooth and inclusive learning experience.

Severity: Minor

Type of Testing: Usability 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 Cases for Online Learning Platform - Discussion Moderation

Test Case 1:

Test Case ID: OLPD-DM-001

Test Case Title: Verify that instructors can edit posts in discussions.

Test Case Description: This test case verifies the functionality for instructors to edit student posts within the discussion forum, ensuring they can correct misinformation, improve clarity, or moderate language appropriately.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in.
- A discussion forum with at least one student post exists.

Test Data: Modified post content.

Test Steps:

1. Navigate to the discussion forum.
2. Locate a student post.
3. Click on the "Edit" option for the post.
4. Modify the post content.
5. Save the changes.

Postconditions:

- The edited post is displayed correctly in the forum.

Expected Result: The system allows the instructor to edit the post, and the changes are reflected accurately in the discussion thread.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPD-DM-002

Test Case Title: Verify that instructors can mute disruptive users.

Test Case Description: This test case ensures that instructors can temporarily mute users exhibiting disruptive behavior in the forum, restricting their posting privileges for a defined period.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in.
- A discussion forum with at least one disruptive user exists.

Test Data: Duration of the mute (e.g., 24 hours, 1 week).

Test Steps:

1. Identify a user displaying disruptive behavior.
2. Access the user's profile or options within the forum.
3. Select the "Mute User" function.
4. Choose the duration for the mute.
5. Confirm the muting action.

Postconditions:

- The muted user is prevented from posting new messages.
- An indicator of the user's muted status is visible to the instructor.

Expected Result: The system successfully mutes the selected user, preventing further posting for the specified duration.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPD-DM-003

Test Case Title: Verify that moderation actions are logged and reported.

Test Case Description: This test case checks the system's ability to record all moderation actions taken by instructors, including details like the action, timestamp, user involved, and reason for moderation.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- Instructor performs at least one moderation action (edit, delete, mute, ban).

Test Data: No test data needed.

Test Steps:

1. Navigate to the moderation log or report section.
2. Verify that the performed moderation action is logged.
3. Check the accuracy of the log details, including timestamp, user, and action description.

Postconditions: None

Expected Result: The system maintains a comprehensive log of all moderation activities, providing accurate details for review and accountability.

Severity: Minor

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPD-DM-004

Test Case Title: Verify that community guidelines are visible to all participants.

Test Case Description: This test case ensures that community guidelines for discussions are easily accessible to all users, promoting understanding of expected behavior and platform rules.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- User is logged in (can be a student or an instructor).

Test Data: No test data needed.

Test Steps:

1. Navigate to the discussion forum.

2. Locate a link or section for "Community Guidelines" or a similar label.

3. Verify that clicking the link displays the complete guidelines.

4. Ensure the guidelines are clear, concise, and easily understandable.

Postconditions: None

Expected Result: The system provides clear and readily accessible community guidelines within the discussion forum for all users to reference.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPD-DM-005

Test Case Title: Verify that notifications are sent for moderated actions.

Test Case Description: This test case checks the functionality of the system to send appropriate notifications to users involved in moderation actions, keeping them informed about post edits, deletions, or account restrictions.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- An instructor performs a moderation action that triggers a notification (e.g., mutes a user, edits a post).

- Notification settings are enabled for the affected user.

Test Data: No test data needed.

Test Steps:

1. Trigger a moderation action that should generate a notification.
2. Log in as the user who should receive the notification.
3. Check for the notification within the platform (e.g., notification bell, inbox).
4. Verify that the notification message is clear, informative, and reflects the moderation action taken.

Postconditions: None

Expected Result: The system successfully delivers clear and accurate notifications to users regarding moderation actions that directly affect them.

Severity: Minor

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 Cases for Online Learning Platform - Learning Analytics

Test Case 1:

Test Case ID: LA-001

Test Case Title: Verify that learning analytics dashboards display engagement metrics.

Test Case Description: This test case verifies that instructors can view various engagement metrics on the learning analytics dashboards, including but not limited to, course views, lecture completion rates, forum activity, resource downloads, and time spent on each module.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in.
- A course with enrolled students and learning activities exists.

Test Data: No test data needed.

Test Steps:

1. Navigate to the course dashboard.

2. Access the learning analytics section.

3. Verify that the dashboard displays metrics related to:

    - Course views

    - Lecture completion rates

    - Forum posts and replies

    - Resource downloads

    - Time spent on each module

Postconditions: None

Expected Result: The learning analytics dashboard should clearly present all specified engagement metrics with accurate data reflecting student activity.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: LA-002

Test Case Title: Verify that performance trends are visualized over time.

Test Case Description: This test case aims to confirm that instructors can view visualizations of student performance trends over different time periods. This includes trends related to quiz scores, assignment submissions, and overall course progress.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in.

- Course data spanning across different time periods is available (e.g., past week, month, entire course duration).

Test Data: No test data needed.

Test Steps:

1. Navigate to the learning analytics dashboard for a specific course.
2. Locate the performance trends visualization section.
3. Select different timeframes for analysis (e.g., week, month, all-time).
4. Verify that the visualizations accurately represent performance trends over the chosen timeframe, including:
    - Average quiz scores
    - Assignment submission rates
    - Overall course progress

Postconditions: None

Expected Result: The system should present clear and accurate visualizations of performance trends, allowing instructors to understand learner progress over time.

Severity: Major

Type of Testing: Functional Testing, Data Visualization Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: LA-003

Test Case Title: Verify that instructors can access detailed reports on learner progress.

Test Case Description: This test case ensures that instructors can access detailed reports on individual learner progress, including their activity, quiz scores, assignment submissions, and areas where they might be struggling.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in.
- Course has student data including quiz scores, assignment submissions, and module completion.

Test Data: No test data needed.

Test Steps:

1. From the learning analytics dashboard, navigate to the learner progress section.
2. Select a specific student from the list of enrolled learners.
3. Verify that the system displays a detailed report for the selected student, including information on:
    - Overall course progress
    - Quiz scores and attempts
    - Assignment submissions and grades
    - Modules completed and time spent
    - Forum activity
4. Verify that the instructor can filter the report by specific modules or activities.

Postconditions: None

Expected Result: The system should generate comprehensive learner progress reports, offering instructors detailed insights into each student's performance.

Severity: Major

Type of Testing: Functional Testing, Report Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: LA-004

Test Case Title: Verify that analytics data is updated in real-time.

Test Case Description: This test case validates that the learning analytics dashboard and reports are updated in real-time to reflect the most current student activity and performance data.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in and viewing the learning analytics dashboard.
- A student is actively participating in the course (e.g., completing a quiz, submitting an assignment).

Test Data:

- Student submission for a quiz or assignment.

Test Steps:

1. While viewing the learning analytics dashboard, have a student submit a quiz or assignment.
2. Observe the relevant analytics metrics on the dashboard (e.g., quiz attempts, assignment submissions).
3. Verify that the dashboard updates in real-time to reflect the student's recent activity.
4. Navigate to the detailed learner progress report for the student.
5. Confirm that the report also reflects the real-time updates.

Postconditions: None

Expected Result: The learning analytics dashboard and reports should dynamically update in real-time to accurately represent the most up-to-date student data.

Severity: Critical

Type of Testing: Functional Testing, Real-Time Data Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: LA-005

Test Case Title: Verify that instructors can export analytics reports for further analysis.

Test Case Description: This test case confirms that instructors can export learning analytics reports in various formats (e.g., CSV, Excel) for further analysis or sharing with stakeholders.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

- Instructor is logged in and viewing a learning analytics report (either general dashboard or individual learner report).

Test Data: No test data needed.

Test Steps:

1. Locate the export functionality within the report view.
2. Select the desired export format (e.g., CSV, Excel).
3. Initiate the report download.
4. Verify that the report downloads successfully in the chosen format.
5. Open the downloaded report and confirm that it contains all the relevant data displayed within the platform's interface.

Postconditions: None

Expected Result: Instructors should be able to export comprehensive learning analytics reports in commonly used formats for further analysis outside the platform.

Severity: Minor

Type of Testing: Functional Testing, Export/Import 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 Cases for Online Learning Platform - Social Learning Features

Test Case 1:

Test Case ID: OLPSLF-001

Test Case Title: Verify that learners can create and join groups for projects.

Test Case Description: This test case verifies the functionality of learners to create new groups for specific projects and join existing groups. It ensures that the process is intuitive and functions correctly, including error handling for invalid group names or duplicate entries.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in.

Test Data:

- Project name: "Software Testing Fundamentals"
- Group name: "Test Masters"

Test Steps:

1. Navigate to the "Projects" section.
2. Select the project "Software Testing Fundamentals."
3. Click on the "Groups" tab.
4. Click on the "Create Group" button.
5. Enter group name "Test Masters" and submit.
6. Verify the group is created and displayed.
7. Search for the group "Test Masters" and join the group.
8. Verify successful joining of the group.

Postconditions:

- Leave the created group.

Expected Result: The system should allow the learner to create a new group, successfully join the created group, and display the group correctly within the project.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPSLF-002

Test Case Title: Verify that collaboration tools are available for group communication.

Test Case Description: This test case ensures that within a group, learners have access to and can effectively utilize various communication tools, such as a chat feature, file sharing, and forum discussions. It also checks for notification functionalities for new messages or activities.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner is part of a group.

Test Data:

- Test message: "Hello Team! Let's collaborate."
- Test file: A small document (e.g., .txt, .docx)

Test Steps:

1. Navigate to the group's page.
2. Locate and utilize the available communication tools (e.g., chat, forum).
3. Send a test message in the group chat.
4. Upload a test file in the file-sharing section.
5. Verify that the message and file are successfully sent and visible to other group members.
6. Check for notifications related to new messages or uploaded files.

Postconditions: None

Expected Result: The collaboration tools should function correctly, allowing learners to communicate effectively within the group. Notifications should be timely and informative.

Severity: Major

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPSLF-003

Test Case Title: Verify that learners can share resources and feedback within groups.

Test Case Description: This test case examines the process of sharing learning resources and providing feedback within the group. It ensures that learners can easily upload, download, and share relevant documents, links, or feedback messages within their groups.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner is part of a group.

Test Data:

- Resource to share: Relevant document or website link
- Feedback: "This resource is very helpful! Thanks for sharing."

Test Steps:

1. Navigate to the group's page.
2. Locate the section for sharing resources.
3. Upload a resource or paste a relevant link.
4. Provide feedback on an existing resource shared by another group member.
5. Verify that the uploaded resource and feedback are visible to other group members.

Postconditions: None

Expected Result: Learners should be able to seamlessly share resources and provide constructive feedback within their groups, fostering a collaborative learning environment.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPSLF-004

Test Case Title: Verify that instructors can assign group projects and monitor progress.

Test Case Description: This test case focuses on the instructor's role in managing group projects. It ensures that instructors can create and assign group projects, track the progress of each group, and have access to individual contributions within the group.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Instructor is logged in.
- A group exists with learners.

Test Data:

- Project title: "Collaborative Research Paper"
- Project description: "Research and write a paper on?"

Test Steps:

1. Navigate to the "Courses" section as an instructor.
2. Select a course and navigate to the "Projects" tab.
3. Click on "Create Group Project."
4. Enter the project title, description, and assign it to a specific group.
5. Monitor the group's progress on the assigned project.
6. Access individual contributions and participation levels within the group.

Postconditions: None

Expected Result: The platform should allow instructors to effectively manage group projects, track progress, and access individual contributions to assess participation and learning outcomes.

Severity: Critical

Type of Testing: Functional Testing, User Acceptance Testing (UAT)

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPSLF-005

Test Case Title: Verify that social sharing features allow learners to share achievements.

Test Case Description: This test case checks the functionality of social sharing features integrated with the platform. It ensures that learners can share their achievements, such as completing a course, finishing a project, or earning a badge, on various social media platforms.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner is logged in.
- Learner has achieved something shareable (completed a course, finished a project, etc.)

Test Data: No test data needed

Test Steps:

1. Complete a course or achieve a milestone (e.g., finish a project, earn a badge).
2. Locate the option to share the achievement on social media.
3. Choose a social media platform to share on (e.g., Facebook, Twitter, LinkedIn).
4. Verify that the achievement is successfully shared on the chosen platform with relevant details.

Postconditions: None

Expected Result: Learners should be able to easily share their learning achievements on different social media platforms, promoting engagement and visibility for the learning platform.

Severity: Minor

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Data - 66

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Authoring Tools

Feature description -

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

Userstory title -

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

Acceptance criteria -

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Content Authoring Tools:

Test Case 1

Test Case ID: OLP-CAT-001

Test Case Title: Verify that instructors can create multimedia presentations.

Test Case Description: This test case verifies that the content authoring tools allow instructors to create multimedia presentations incorporating various elements like slides, text, images, audio, and video.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor is logged in.

- Course is created.

Test Data: Sample text, images, audio, and video files.

Test Steps:

1. Navigate to the course creation section.

2. Select the option to add new content.

3. Choose the "Multimedia Presentation" content type.

4. Add slides, text, images, audio, and video elements to the presentation.

5. Save the presentation.

Postconditions:

- Multimedia presentation is saved successfully.

Expected Result: The system should allow the instructor to seamlessly create a multimedia presentation with the desired elements.

Severity: Major

Type of Testing: Functional, Usability Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: OLP-CAT-002

Test Case Title: Verify that interactive modules can be developed and integrated.

Test Case Description: This test case ensures that instructors can create and integrate interactive modules, such as quizzes, simulations, and drag-and-drop activities, within their course content.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor is logged in.

- Course is created.

Test Data: Sample quiz questions, simulation parameters, drag-and-drop elements.

Test Steps:

1. Go to the course content creation area.

2. Choose the option to add interactive content.

3. Select the desired interactive module type (quiz, simulation, etc.).

4. Configure the module with appropriate parameters and content.

5. Integrate the module within the desired section of the course.

Postconditions:

- Interactive module is successfully integrated and functions as expected.

Expected Result: The system should allow for easy creation, configuration, and integration of various interactive modules within the course.

Severity: Major

Type of Testing: Functional, Integration Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: OLP-CAT-003

Test Case Title: Verify that rich media resources are supported.

Test Case Description: This test case checks the platform's compatibility with various rich media formats, ensuring instructors can upload and embed diverse resources like videos and animations.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

- Instructor is logged in.
- Course is created.

Test Data: Video files in various formats (e.g., MP4, MOV, AVI), animation files (e.g., GIF, SWF).

Test Steps:

1. Go to the content editor for a specific course section.
2. Select the option to add rich media content.
3. Upload or embed video and animation files in different formats.
4. Preview the uploaded/embedded media within the course content.

Postconditions:

- Rich media content is displayed correctly and plays smoothly.

Expected Result: The system should successfully upload, embed, and display various rich media formats without any compatibility issues.

Severity: Major

Type of Testing: Functional, Compatibility Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: OLP-CAT-004

Test Case Title: Verify that content authoring tools are user-friendly and intuitive.

Test Case Description: This test case evaluates the user-friendliness and intuitiveness of the content authoring tools from an instructor's perspective, ensuring ease of use and navigation.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

- Instructor is logged in.

Test Data: No test data needed.

Test Steps:

1. Access the content authoring tools section.
2. Explore different features and functionalities within the tools.
3. Attempt creating various content types using the provided tools.
4. Evaluate the ease of navigation, clarity of instructions, and overall user experience.

Postconditions: No specific postconditions.

Expected Result: The content authoring tools should be user-friendly, easy to navigate, and intuitive for instructors with varying technical skills.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: OLP-CAT-005

Test Case Title: Verify that instructors can preview and edit content before publishing.

Test Case Description: This test case ensures that instructors have the functionality to preview and edit their created content before making it live for students.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

- Instructor has created course content using the authoring tools.

Test Data: No test data needed.

Test Steps:

1. Navigate to the created course content.

2. Select the option to preview the content.

3. Review the content for any errors or required modifications.

4. Edit the content as needed using the available tools.

5. Preview the updated content again to ensure changes are reflected.

Postconditions:

- Content is saved after editing.

Expected Result: The system should allow instructors to preview their created content in a student-facing view and enable them to make edits and save changes before publishing.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 67

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Paths and Curriculum

Feature description -

The platform offers structured learning paths and curricula to guide learners through sequential courses, modules, and topics aligned with specific learning objectives. By providing clear pathways and learning trajectories, the platform streamlines the learning process, facilitates skill development, and ensures coherence and continuity in learners' educational journeys.

Userstory title -

As a learner, I want structured learning paths and curricula so that I can achieve my learning objectives in a systematic manner.

Acceptance criteria -

['Learning paths include sequential courses and modules.', 'Curricula are aligned with specific learning objectives.', 'Learners can track their progress within a learning path.', 'Instructors can customize learning paths for different needs.', 'Notifications are sent for upcoming courses or milestones.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Learning Paths and Curricula:

Test Case 1:

* **Test Case ID:** LP-001

* **Test Case Title:** Verify that learning paths include sequential courses and modules.

* **Test Case Description:** This test case verifies that learning paths are structured with courses and modules arranged in a logical and sequential order. It ensures learners can progress through the content in a structured manner.

* **Test Suite:** Learning Paths and Curricula
* **Test Priority:** High
* **Preconditions:**
    * A learning path with at least two courses is created.
    * Each course has at least two modules.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Navigate to the learning path.
    2. Verify that courses within the path are displayed in the intended sequence.
    3. Enter each course and verify that modules within the course are also displayed sequentially.
* **Postconditions:** None
* **Expected Result:** Courses and modules within the learning path are presented in a logical and sequential order as defined by the instructor.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** LP-002
* **Test Case Title:** Verify that curricula are aligned with specific learning objectives.
* **Test Case Description:** This test case ensures that each curriculum is designed to meet specific learning objectives, aligning the learning path with the desired skills and knowledge learners should gain.
* **Test Suite:** Learning Paths and Curricula
* **Test Priority:** High
* **Preconditions:**
    * A curriculum with defined learning objectives is available.

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

* **Test Steps:**

   1. Review the curriculum description and stated learning objectives.

   2. Analyze the courses and modules included in the curriculum.

   3. Verify that the content and structure of the curriculum effectively address each learning objective.

* **Postconditions:** None

* **Expected Result:** The curriculum content and structure are closely aligned with the stated learning objectives, ensuring learners achieve the desired skills and knowledge upon completion.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** LP-003

* **Test Case Title:** Verify that learners can track their progress within a learning path.

* **Test Case Description:** This test case ensures learners can monitor their advancement through the learning path, including completed courses, modules, and overall progress.

* **Test Suite:** Learning Paths and Curricula

* **Test Priority:** High

* **Preconditions:**

   * Learner is enrolled in a learning path.

   * Learner has completed at least one module.

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

* **Test Steps:**

   1. Navigate to the learning path as a learner.

   2. Verify a progress bar or visual indicator displaying the completion status.

3. Check that completed courses and modules are clearly marked.

4. Ensure the overall progress percentage reflects the completed content accurately.

* **Postconditions:** None

* **Expected Result:** The platform accurately displays the learner's progress, including completed courses/modules and overall percentage completion.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** LP-004

* **Test Case Title:** Verify that instructors can customize learning paths for different needs.

* **Test Case Description:** This test case ensures instructors have the functionality to tailor learning paths to suit various learner needs, such as adding/removing courses, adjusting the sequence, or incorporating additional resources.

* **Test Suite:** Learning Paths and Curricula

* **Test Priority:** High

* **Preconditions:**

  * Instructor account with necessary permissions.

  * Existing learning path available.

* **Test Data:**

  * Course to be added.

  * Module to be removed.

* **Test Steps:**

  1. Login as an instructor and access the learning path customization interface.

  2. Add a new course to the learning path.

   3. Remove an existing module from a course within the path.

   4. Adjust the sequence of courses or modules.

   5. Save the changes made to the learning path.

* **Postconditions:** Customized learning path is saved and accessible to learners.

* **Expected Result:** The platform allows instructors to successfully modify the learning path structure and content according to specific needs.

* **Severity:** Major

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** LP-005

* **Test Case Title:** Verify that notifications are sent for upcoming courses or milestones.

* **Test Case Description:** This test case verifies that learners receive timely notifications about upcoming courses, deadlines for modules, or other important milestones within their enrolled learning path.

* **Test Suite:** Learning Paths and Curricula

* **Test Priority:** Medium

* **Preconditions:**

   * Learner is enrolled in a learning path with future course start dates or milestones.

   * Notification settings are enabled for the learner.

* **Test Data:**

   * Scheduled start date of a future course.

   * Upcoming milestone within a module.

* **Test Steps:**

   1. Configure the system to trigger notifications for upcoming courses or milestones.

   2. As a learner, verify the receipt of notifications via the chosen communication channel (e.g., email,

platform notification).

3. Check if the notification content accurately reflects the upcoming event (course start date, milestone details).

* **Postconditions:** None

* **Expected Result:** The system successfully sends timely and informative notifications to learners regarding upcoming courses, deadlines, or milestones within their enrolled learning paths.

* **Severity:** Minor

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Data - 68

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Attendance Tracking

Feature description -

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and accountability. By recording attendance data and participation metrics, instructors can assess learner engagement, identify areas for improvement, and provide targeted support and intervention as needed.

Userstory title -

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Acceptance criteria -

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation metrics.', 'Notifications are sent for missed sessions.']

Generated Test Cases:

## Test Case 1

Test Case ID: TC_ATT_001

Test Case Title: Verify that attendance is automatically recorded during live sessions.

Test Case Description: This test case verifies that the system automatically records attendance for learners who join a live session. It checks if the system accurately captures the start and end time of a learner's presence in the session.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Instructor has scheduled a live session.
- Learner is enrolled in the course with the live session.

Test Data: No test data needed

Test Steps:

1. Log in as a learner.
2. Join the scheduled live session.
3. Stay in the session for a certain duration.
4. Leave the session.

Postconditions:

- Verify learner's attendance record in the system.

Expected Result: The system should record the learner's attendance for the session, capturing the correct start and end times.

Severity: Major

Type of Testing: Functional, System Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TC_ATT_002

Test Case Title: Verify that instructors can view attendance reports for each session.

Test Case Description: This test case ensures that instructors can access and view detailed attendance reports for each live session they conduct.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Instructor has conducted at least one live session with recorded learner attendance.

Test Data: No test data needed

Test Steps:

1. Log in as an instructor.

2. Navigate to the course with the completed live session.

3. Access the attendance records for the specific session.

Postconditions: None

Expected Result: The system should display a comprehensive attendance report, including the list of learners, their attendance status (present, absent), and session duration.

Severity: Major

Type of Testing: Functional, System Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TC_ATT_003

Test Case Title: Verify that learners can see their attendance records.

Test Case Description: This test case validates that learners can view their attendance records for the courses they are enrolled in.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- Learner has attended at least one live session.

Test Data: No test data needed

Test Steps:

1. Log in as a learner.

2. Navigate to the "My Courses" or similar section.

3. Access the attendance tab or section within the course.

Postconditions: None

Expected Result: The system should display the learner's attendance records, showing their presence or absence for each session.

Severity: Major

Type of Testing: Functional, System Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TC_ATT_004

Test Case Title: Verify that attendance data is integrated with participation metrics.

Test Case Description: This test case ensures that the system integrates attendance data with other participation metrics, like engagement in chat, polls, or quizzes during the live session.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Learner has attended a live session and participated in activities like chat or polls.

Test Data: No test data needed

Test Steps:

1. Log in as an instructor.

2. Access the attendance report for a session where learners participated in activities.

Postconditions: None

Expected Result: The attendance report should display not only attendance status but also reflect the learner's participation level through available metrics.

Severity: Minor

Type of Testing: Integration Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: TC_ATT_005

Test Case Title: Verify that notifications are sent for missed sessions.

Test Case Description: This test case verifies that the system sends appropriate notifications to learners who miss a scheduled live session.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Learner is enrolled in a course with upcoming live sessions.

Test Data: No test data needed

Test Steps:

1. As a learner, do not join a scheduled live session.

2. Check for notifications within the platform and configured notification channels (e.g., email).

Postconditions: None

Expected Result: The learner should receive a notification about the missed session, prompting them to catch up or access recorded content.

Severity: Minor

Type of Testing: Functional, System Testing

Test Case Approach: Negative

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 Cases for Online Learning Platform - Content Licensing and Rights Management:

Test Case 1:

Test Case ID: CLRM-001

Test Case Title: Verify that content licenses are assigned to each resource.

Test Case Description: This test case ensures that every uploaded resource has a content license associated with it. This includes verifying that the system prompts content creators to select a license during upload and displays the selected license on the resource page.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* Content creator is logged in.

* Content creator has initiated the resource upload process.

Test Data:

* Sample course resource (e.g., PDF document, video file).

* Different content license types (e.g., Creative Commons, All Rights Reserved).

Test Steps:

1. Navigate to the resource upload page.

2. Upload the sample course resource.

3. Observe the system prompt for selecting a content license.

4. Select different content license types and proceed.

5. After successful upload, view the resource page and verify the assigned license is displayed.

Postconditions:

* The uploaded resource is associated with the selected license.

* The selected license is visible on the resource page.

Expected Result: The system should successfully assign the selected content license to each uploaded resource and display it clearly.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: CLRM-002

Test Case Title: Verify that usage rights and permissions are clearly defined for each content license.

Test Case Description: This test verifies that the platform provides clear and concise descriptions of usage rights and permissions associated with each available content license.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* User is browsing available courses/resources.

Test Data: No test data needed.

Test Steps:

1. Navigate to a resource page.

2. Locate and click on the displayed content license information.

3. Observe the system display of usage rights and permissions for that license.

Postconditions: N/A

Expected Result: The system should display a clear and comprehensive description of usage rights and permissions for each content license.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: CLRM-003

Test Case Title: Verify that content creators can set restrictions on usage and distribution.

Test Case Description: This test case ensures that content creators have the ability to define specific restrictions regarding the usage and distribution of their resources, even under pre-defined licenses.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* Content creator is logged in.

* Content creator is in the process of uploading or editing a resource.

Test Data:

* Sample course resource.

Test Steps:

1. Navigate to the resource upload or edit page.

2. Locate the section for setting usage and distribution restrictions.

3. Attempt to set specific restrictions (e.g., non-commercial use only, no derivatives).

4. Save the resource and verify the applied restrictions.

Postconditions:

* The uploaded resource should reflect the set usage and distribution restrictions.

Expected Result: The system should allow content creators to set additional restrictions on the usage and distribution of their resources, beyond the standard license terms.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: CLRM-004

Test Case Title: Verify that instructors and learners are informed of content licensing terms.

Test Case Description: This test confirms that both instructors and learners are clearly informed of the content licensing terms before accessing any course materials.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* User is logged in (either as an instructor or learner).

* User is attempting to access a course or resource.

Test Data: No test data needed.

Test Steps:

1. As an instructor, attempt to access a course page.

2. As a learner, attempt to access a course resource.

3. Observe the system behavior regarding the display of licensing information.

Postconditions: N/A

Expected Result: Before accessing any course material, the system should clearly display the relevant content license information and require acknowledgement from both instructors and learners.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: CLRM-005

Test Case Title: Verify that notifications are sent for license expiration or renewal.

Test Case Description: This test ensures the system sends timely notifications to content creators regarding upcoming license expirations or required renewals, allowing them to take appropriate actions.

Test Suite: Content Licensing and Rights Management

Test Priority: Medium

Preconditions:

* Content creator is logged in.

* System has a record of content licenses and their expiry dates.

Test Data:

* Simulated license expiry dates (approaching and expired).

Test Steps:

1. Simulate an approaching license expiry date for a resource.

2. Simulate a passed license expiry date for another resource.

3. Monitor the system for generated notifications.

Postconditions: N/A

Expected Result: The system should send timely email notifications to the content creator regarding approaching license expirations and actions required. It should also notify the content creator of any expired licenses.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 70

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Versioning

Feature description -

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

Userstory title -

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

Acceptance criteria -

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.', 'Notifications are sent for significant content updates.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Content Versioning:

Test Case 1:

* **Test Case ID:** OLPCV-001

* **Test Case Title:** Verify that instructors can create and save multiple versions of content.

* **Test Case Description:** This test case ensures instructors can create new versions of course content, save changes within a version, and manage multiple versions effectively.

* **Test Suite:** Content Versioning

* **Test Priority:** High

* **Preconditions:**

    * Instructor is logged in.

    * Instructor has an active course.

* **Test Data:** Sample lecture notes, presentation slides, and quiz questions.

* **Test Steps:**

    1. Navigate to the course content management section.

    2. Create new course content (e.g., lecture notes).

    3. Save the content as Version 1.

    4. Make modifications to the content.

    5. Save the modified content as Version 2.

    6. Repeat steps 4-5 to create multiple versions.

* **Postconditions:**

    * Multiple versions of the content are saved and accessible.

* **Expected Result:** The system should allow creating, saving, and managing multiple versions of content without data loss or corruption.

* **Severity:** Critical

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 2:

* **Test Case ID:** OLPCV-002
* **Test Case Title:** Verify that version history is maintained and accessible.
* **Test Case Description:** This test case validates that the system accurately records and displays the version history of course content, including details like version number, date modified, and author.
* **Test Suite:** Content Versioning
* **Test Priority:** High
* **Preconditions:**
    * Instructor is logged in.
    * Multiple versions of content exist within a course.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Navigate to a piece of content with multiple versions.
    2. Access the version history feature.
    3. Verify that the version history displays all saved versions.
    4. Check that each version displays its creation date, author, and version number.
* **Postconditions:** None
* **Expected Result:** The system should accurately record and present a comprehensive version history for each content item.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 3:

* **Test Case ID:** OLPCV-003

* **Test Case Title:** Verify that instructors can compare different versions of content.
* **Test Case Description:** This test case ensures instructors can easily compare any two versions of their content to view changes, additions, or deletions.
* **Test Suite:** Content Versioning
* **Test Priority:** High
* **Preconditions:**
    * Instructor is logged in.
    * At least two versions of content exist.
* **Test Data:** No test data needed.
* **Test Steps:**
    1. Access the version history of a content item.
    2. Select two different versions for comparison.
    3. Trigger the comparison function (e.g., "Compare Versions").
    4. Verify that the system highlights the differences between the selected versions.
* **Postconditions:** None
* **Expected Result:** The system should provide a clear and concise visual comparison of the selected content versions, highlighting any discrepancies.
* **Severity:** Major
* **Type of Testing:** Functional Testing
* **Test Case Approach:** Positive

Test Case 4:

* **Test Case ID:** OLPCV-004
* **Test Case Title:** Verify that previous versions can be restored if needed.
* **Test Case Description:** This test case confirms that instructors have the functionality to revert to any previously saved version of their content.

* **Test Suite:** Content Versioning

* **Test Priority:** High

* **Preconditions:**

    * Instructor is logged in.

    * Multiple versions of content exist.

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

* **Test Steps:**

    1. Access the version history of a content item.

    2. Select a previous version to restore.

    3. Confirm the restoration action.

    4. Verify that the selected version becomes the current active version.

* **Postconditions:**

    * The restored version is marked as the current version.

* **Expected Result:** The system should allow instructors to easily revert to a previous version of their content, making it the active version.

* **Severity:** Critical

* **Type of Testing:** Functional Testing

* **Test Case Approach:** Positive

Test Case 5:

* **Test Case ID:** OLPCV-005

* **Test Case Title:** Verify that notifications are sent for significant content updates.

* **Test Case Description:** This test case ensures that the system generates and sends notifications to instructors upon saving a new version of content.

* **Test Suite:** Content Versioning

* **Test Priority:** Medium

* **Preconditions:**

    * Instructor is logged in.

    * Notification settings are enabled for content updates.

* **Test Data:** Sample changes to existing course content.

* **Test Steps:**

    1. Make significant changes to an existing piece of content.

    2. Save the changes as a new version.

    3. Check for a notification indicating the creation of a new version.

    4. Verify that the notification includes relevant details (e.g., content title, version number, timestamp).

* **Postconditions:** None

* **Expected Result:** The system should trigger a notification to the instructor upon saving a new version, providing information about the update.

* **Severity:** Minor

* **Type of Testing:** Functional Testing

* **Test Case Approach:** 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 Cases for Online Learning Platform API Integration:

Test Case 1:

Test Case ID: API-001

Test Case Title: Verify that APIs are available for key platform functionalities.

Test Case Description: This test verifies the availability and functionality of APIs for crucial platform features like user authentication, course retrieval, content access, and user progress tracking.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API documentation is available.
- API endpoints are accessible.

Test Data: API request payloads for different functionalities (e.g., user login credentials, course ID for retrieval).

Test Steps:

1. Send API requests to access various key platform functionalities (user authentication, course listing, content access, progress update).

2. Verify the HTTP status codes of the API responses (expecting 200 OK for successful requests).

3. Validate the response data format and content to ensure correctness and completeness.

Postconditions: None

Expected Result: The APIs for key functionalities should respond with valid data and expected status codes, indicating proper functionality and data accessibility.

Severity: Blocker

Type of Testing: API Testing, Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: API-002

Test Case Title: Verify that comprehensive API documentation is provided.

Test Case Description: This test case ensures the API documentation is thorough, clear, and user-friendly for developers to understand and integrate the APIs effectively.

Test Suite: API Integration

Test Priority: High

Preconditions: API documentation is published and accessible.

Test Data: No test data needed

Test Steps:

1. Review the API documentation for completeness, covering all endpoints, request/response formats, authentication methods, and error handling.

2. Evaluate the clarity and organization of the documentation, ensuring easy navigation and understanding.

3. Verify the inclusion of code samples and examples for different programming languages to assist developers.

Postconditions: Documentation improvements to be suggested based on the review.

Expected Result: The API documentation should be comprehensive, well-structured, and easy to understand, enabling developers to integrate with the platform seamlessly.

Severity: Major

Type of Testing: Documentation Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: API-003

Test Case Title: Verify that secure access protocols are implemented.

Test Case Description: This test case aims to validate the implementation of secure access protocols like OAuth 2.0 or API keys to protect sensitive user data and platform functionalities.

Test Suite: API Integration

Test Priority: High

Preconditions:

- API authentication mechanisms are set up.
- Test accounts with different access levels are available.

Test Data: Invalid or expired API keys, unauthorized access attempts.

Test Steps:

1. Attempt to access API endpoints with invalid or expired API keys.
2. Test unauthorized access attempts to restricted resources or functionalities.
3. Verify the system rejects unauthorized requests with appropriate HTTP error codes (e.g., 401 Unauthorized, 403 Forbidden).

Postconditions: None

Expected Result: The platform should enforce secure access protocols and reject unauthorized requests, ensuring the protection of sensitive data and functionalities.

Severity: Critical

Type of Testing: Security Testing, API Testing

Test Case Approach: Negative

Test Case 4:

Test Case ID: API-004

Test Case Title: Verify that third-party applications can authenticate and access APIs.

Test Case Description: This test case ensures that third-party applications can successfully authenticate and interact with the platform's APIs using provided credentials and access tokens.

Test Suite: API Integration

Test Priority: High

Preconditions:

- A third-party application is set up for testing with valid API credentials.
- The application has necessary permissions to access specific platform resources.

Test Data: Valid API credentials (API key, OAuth 2.0 tokens) for the third-party application.

Test Steps:

1. Initiate an authentication request from the third-party application using provided credentials.
2. Obtain an access token upon successful authentication.
3. Send API requests to access platform resources using the acquired access token.
4. Verify the platform allows access to authorized resources and functionalities.

Postconditions: None

Expected Result: The third-party application should successfully authenticate, obtain an access token, and access permitted platform resources, demonstrating seamless integration.

Severity: Major

Type of Testing: Integration Testing, API Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: API-005

Test Case Title: Verify that rate limiting and monitoring features are in place.

Test Case Description: This test case validates the implementation of rate limiting and monitoring mechanisms to prevent abuse, ensure platform stability, and track API usage.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

- Rate limiting rules are configured for the APIs.
- Monitoring tools are set up to track API usage.

Test Data: Scripts or tools to generate a high volume of API requests exceeding defined rate limits.

Test Steps:

1. Send a large number of requests to specific API endpoints exceeding the defined rate limits.
2. Observe the platform's response and verify if rate limiting is enforced effectively (e.g., HTTP 429 Too Many Requests).
3. Analyze the monitoring data to ensure API usage is tracked, and alerts are generated for abnormal activities.

Postconditions: None

Expected Result: The platform should effectively enforce rate limits, prevent abuse, and provide monitoring data for analysis and performance optimization.

Severity: Major

Type of Testing: Performance Testing, Security Testing, API Testing

Test Case Approach: Negative, Destructive

Data - 72

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

## Test Cases for Online Learning Platform - Data Analytics and Insights

Test Case 1:

Test Case ID: OLP-DAI-001

Test Case Title: Verify that data visualization tools are available for analysis

Test Case Description: This test case aims to verify that the platform provides various data visualization tools, such as charts, graphs, and dashboards, to present learner performance data in an easy-to-understand and insightful manner.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in.
- Course with learner data is available.

Test Data: No test data needed.

Test Steps:

1. Navigate to the Data Analytics section.
2. Select the desired course or learner group.
3. Choose different data points for analysis (e.g., completion rates, quiz scores, forum participation).
4. Verify that the platform generates appropriate visualizations (e.g., bar charts, line graphs, pie charts).

Postconditions: N/A

Expected Result: The platform should provide a variety of data visualization tools, allowing administrators to analyze data effectively.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLP-DAI-002

Test Case Title: Verify that reports can be generated for learner performance metrics

Test Case Description: This test case ensures that administrators can generate comprehensive reports on various learner performance metrics.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in.
- Course with learner data is available.

Test Data: Select specific date range, learner group, and performance metrics.

Test Steps:

1. Navigate to the Reports section.
2. Choose the desired report type (e.g., course progress, learner engagement, quiz performance).
3. Select the data range and learner group for the report.
4. Specify the desired performance metrics to include in the report.
5. Generate the report.

Postconditions:

- Verify that the generated report is downloadable.

Expected Result: The platform should generate accurate and customizable reports reflecting the selected learner performance metrics.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLP-DAI-003

Test Case Title: Verify that real-time analytics provide insights into engagement and outcomes

Test Case Description: This test case aims to validate the platform's ability to provide real-time insights into learner engagement and learning outcomes.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in.
- Course with active learners is available.

Test Data: No test data needed.

Test Steps:

1. Navigate to the Real-time Analytics dashboard.
2. Monitor the live data streams related to learner engagement (e.g., active users, forum posts, resource downloads).
3. Observe the real-time updates on learning outcomes (e.g., quiz attempts, average scores).

Postconditions: N/A

Expected Result: The platform should display real-time data updates on learner engagement and outcomes, providing immediate insights.

Severity: Major

Type of Testing: Functional Testing, Performance Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLP-DAI-004

Test Case Title: Verify that data can be exported for further analysis

Test Case Description: This test case ensures that administrators can export raw data in various formats for further analysis using external tools.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator is logged in.

- Data is available for export.

Test Data: No test data needed.

Test Steps:

1. Navigate to the Data Export section.

2. Select the desired data set for export (e.g., learner performance, course completion).

3. Choose the desired export format (e.g., CSV, Excel, JSON).

4. Initiate the data export process.

Postconditions:

- Verify that the exported file is in the correct format and downloadable.

Expected Result: The platform should allow exporting the selected data in the chosen format, ensuring compatibility with external analysis tools.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP-DAI-005

Test Case Title: Verify that notifications are sent for significant data trends or anomalies

Test Case Description: This test case checks if the platform triggers notifications to alert administrators about significant data trends or potential anomalies.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator is logged in.
- Notification settings are enabled.

Test Data: Simulate significant data changes (e.g., sudden drop in course completion rates, unusual spike in quiz failures).

Test Steps:

1. Configure notification preferences for specific data trends and anomalies.
2. Trigger a significant data change within the platform.
3. Monitor the administrator's notification center for alerts.

Postconditions: N/A

Expected Result: The platform should send timely and informative notifications about the configured data trends or anomalies, allowing administrators to take prompt action.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive, Negative (testing scenarios where notifications should not be sent)

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 Cases for Online Learning Platform - User Roles and Permissions

Test Case 1:

Test Case ID: URP-001

Test Case Title: Verify that administrators can create and assign user roles.

Test Case Description: This test case ensures that administrators can successfully create new user roles with custom names and assign those roles to existing platform users.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

* Administrator is logged in.

* No existing user role with the chosen name exists.

Test Data: New role name: "Teaching Assistant"

Test Steps:

1. Navigate to the User Management section.

2. Select the option to create a new user role.

3. Enter the new role name ("Teaching Assistant").

4. Save the new user role.

5. Select an existing user and navigate to their role assignment settings.

6. Assign the newly created "Teaching Assistant" role to the user.

Postconditions:

* New role "Teaching Assistant" is created and visible in the list of roles.

* The selected user has the "Teaching Assistant" role assigned.

Expected Result: The system should allow the administrator to create a new user role and assign it to a user successfully.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: URP-002

Test Case Title: Verify that permissions are customizable for each role.

Test Case Description: This test case checks that administrators can customize permissions for different user roles, granting or restricting access to specific platform functionalities.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

* Administrator is logged in.

* At least one user role (e.g., "Instructor") exists.

Test Data: Permission to grant: "Create and edit courses"

Test Steps:

1. Navigate to the User Roles section.

2. Select the existing user role ("Instructor").

3. Locate the permission settings for this role.

4. Grant the "Create and edit courses" permission to the Instructor role.

5. Save the changes made to the role permissions.

Postconditions:

* The Instructor role now has the "Create and edit courses" permission enabled.

Expected Result: The system should allow the administrator to successfully grant specific permissions to a chosen user role.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: URP-003

Test Case Title: Verify that users are restricted to functionalities based on their roles.

Test Case Description: This test case validates that users can only access and perform actions permitted by their assigned role.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

* Two users are logged in: one with the "Student" role and another with the "Instructor" role.

* The "Instructor" role has "Create and edit courses" permission, while the "Student" role does not.

Test Data: No test data needed

Test Steps:

1. Log in as the "Student" user.

2. Attempt to access the course creation functionality.

3. Log out and log in as the "Instructor" user.

4. Attempt to access the same course creation functionality.

Postconditions: No changes to user roles or permissions.

Expected Result:

* The "Student" user should be denied access to course creation.

* The "Instructor" user should be able to access and utilize the course creation functionality.

Severity: Critical

Type of Testing: Security Testing, Functional Testing

Test Case Approach: Negative, Positive

Test Case 4:

Test Case ID: URP-004

Test Case Title: Verify that changes to roles and permissions are logged.

Test Case Description: This test case ensures that any modifications made to user roles and their permissions are logged by the system for auditing purposes.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

* Administrator is logged in.

Test Data: New permission to grant: "Moderate discussion forums"

Test Steps:

1. Navigate to the User Roles section.

2. Select an existing user role.

3. Grant the "Moderate discussion forums" permission to the selected role.

4. Save the changes made to the role permissions.

5. Access the system logs.

Postconditions:

* The selected role now has the "Moderate discussion forums" permission.

Expected Result: The system logs should accurately record the administrator's action, including the timestamp, the user (administrator) who made the change, the specific change made (permission granted), and the role affected.

Severity: Minor

Type of Testing: Security Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: URP-005

Test Case Title: Verify that users are notified of their assigned roles and permissions.

Test Case Description: This test case verifies that users receive appropriate notifications informing them about their assigned roles and the permissions associated with those roles upon first login after role assignment or modification.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

* A user account exists without any roles assigned.

Test Data: Role to assign: "Teaching Assistant"

Test Steps:

1. As an administrator, assign the "Teaching Assistant" role to the user.

2. Log out of the administrator account.

3. Log in as the user who was assigned the new role.

Postconditions:

* User is now logged in with the "Teaching Assistant" role.

Expected Result: Upon login, the user should receive a clear notification informing them about their assigned role ("Teaching Assistant") and a summary of the permissions associated with this role. This notification could be displayed as a pop-up, banner, or within the user's dashboard.

Severity: Minor

Type of Testing: Usability Testing

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 1:

Test Case ID: TC_FRS_001

Test Case Title: Verify that learners can rate courses and instructors on a scale.

Test Case Description: This test case verifies the functionality of the rating system, ensuring learners can rate courses and instructors on the defined scale. It covers various rating inputs to confirm accurate recording and display.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- Learner is logged in.

- Learner has accessed a completed course.

Test Data:

- Valid rating inputs within the defined scale (e.g., 1-5 stars).

Test Steps:

1. Navigate to a completed course page.

2. Locate the rating section for the course and the instructor.

3. Select different rating values for both the course and the instructor.

4. Submit the ratings.

5. Verify the selected ratings are displayed correctly on the course page.

Postconditions:

- Ratings are saved and associated with the specific course and instructor.

Expected Result: The system should accurately record and display the selected ratings for the course and the instructor.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 2:

Test Case ID: TC_FRS_002

Test Case Title: Verify that feedback forms are available for detailed comments.

Test Case Description: This test case ensures the availability and functionality of feedback forms for providing detailed comments on courses and instructors.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- Learner is logged in.

- Learner has accessed a completed course.

Test Data:

- Sample feedback text (positive and negative).

Test Steps:

1. Navigate to the feedback section of a completed course.

2. Locate the text box for providing feedback.

3. Enter sample feedback text.

4. Submit the feedback.

Postconditions:

- Feedback is submitted and stored in the system.

Expected Result: The system should successfully submit and store the feedback provided by the learner.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 3:

Test Case ID: TC_FRS_003

Test Case Title: Verify that instructors can view and respond to feedback.

Test Case Description: This test case verifies if instructors can access, view, and respond to learner feedback submitted for their courses.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- Instructor is logged in.

- Learner feedback is available for the instructor's course.

Test Data: No test data needed.

Test Steps:

1. Navigate to the course management section as an instructor.

2. Select the course with available learner feedback.

3. Access the feedback section for the selected course.

4. View submitted feedback.

5. Locate the option to respond to feedback.

Postconditions:

- Instructor can view all feedback and respond if desired.

Expected Result: The system should allow instructors to view all learner feedback for their courses and provide a means to respond to individual feedback entries.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 4:

Test Case ID: TC_FRS_004

Test Case Title: Verify that aggregated ratings are displayed publicly.

Test Case Description: This test case ensures that the system correctly calculates and displays aggregated ratings for courses and instructors publicly, allowing potential learners to make informed decisions.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

- The platform has courses with existing ratings.

Test Data: No test data needed.

Test Steps:

1. Navigate to a course page as a guest user (not logged in).
2. Locate the aggregated rating display for the course and the instructor.
3. Verify that the displayed rating reflects the average of all individual ratings.

Postconditions: No specific postconditions.

Expected Result: The system should accurately calculate and display the average rating for both the course and the instructor, visible to all users, including those not logged in.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 5:

Test Case ID: TC_FRS_005

Test Case Title: Verify that anonymous feedback options are available.

Test Case Description: This test case ensures the functionality of anonymous feedback submission, allowing learners to provide feedback without revealing their identity, encouraging honest reviews.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

- Learner is logged in.
- Learner has accessed a course's feedback form.

Test Data:

- Sample feedback text.

Test Steps:

1. Navigate to the feedback form for a specific course.
2. Locate the option to submit feedback anonymously.
3. Select the anonymous feedback option.
4. Provide feedback in the text box.
5. Submit the feedback.

Postconditions: Feedback is stored without being linked to the learner's account.

Expected Result: The system should successfully submit the feedback without associating it with the learner's account information, ensuring anonymity.

Severity: Minor

Type of Testing: Functional Testing & Security 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 Cases for Social Media Integration Feature:

Test Case 1:

Test Case ID: SM_Integration_001

Test Case Title: Verify that learners can connect their social media accounts to the platform

Test Case Description: This test case verifies that learners can successfully connect their social media accounts (Facebook, Twitter, LinkedIn) to the online learning platform.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has active social media accounts.

Test Data: No test data needed.

Test Steps:

1. Navigate to the learner's profile settings.
2. Select the "Social Media Integration" option.
3. Choose a social media platform (Facebook, Twitter, LinkedIn).
4. Follow the authentication process to grant access to the platform.

Postconditions:

- The selected social media account is linked to the platform.

Expected Result: The system should successfully link the learner's social media account to the platform, displaying a confirmation message upon successful connection.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SM_Integration_002

Test Case Title: Verify that achievements and course completions can be shared on social media

Test Case Description: This test case verifies that learners can share their achievements and course completions on connected social media platforms.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- Learner is logged in.

- Learner has connected at least one social media account.

Test Data:

- Achievement: Completing a course module.

Test Steps:

1. Trigger an achievement event (e.g., complete a course module).

2. Check for the option to share the achievement on social media.

3. Choose a connected social media platform for sharing.

4. Customize the sharing message (optional).

5. Confirm sharing.

Postconditions:

- The achievement/completion is shared on the selected platform.

Expected Result: The achievement/completion, along with a pre-defined message and a link to the platform/course, should be successfully posted on the selected social media account.

Severity: Major

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SM_Integration_003

Test Case Title: Verify that privacy settings control the visibility of shared content

Test Case Description: This test case verifies that learners can control the visibility of their shared content on social media platforms using the platform's privacy settings.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- Learner is logged in.
- Learner has connected at least one social media account.

Test Data: No test data needed.

Test Steps:

1. Navigate to the platform's privacy settings.
2. Locate the section for social media sharing controls.
3. Adjust the visibility settings for different types of shared content (e.g., public, connections only, private).
4. Attempt to share an achievement or course completion on a connected social media platform.

Postconditions:

- The visibility of the shared content on social media aligns with the selected privacy settings.

Expected Result: The shared content's visibility on the social media platform should accurately reflect the learner's chosen privacy settings on the learning platform.

Severity: Major

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SM_Integration_004

Test Case Title: Verify that instructors can promote courses and updates through social media

Test Case Description: This test case verifies that instructors can utilize the social media integration to promote their courses and share updates with potential learners.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- Instructor is logged in.
- Instructor has connected at least one social media account.

Test Data:

- Sample course information and updates.

Test Steps:

1. Navigate to the instructor's dashboard or course management section.
2. Locate the options for promoting courses and sharing updates.
3. Choose the social media platform for promotion.
4. Input the promotional message or update.
5. Confirm sharing.

Postconditions:

- The course promotion or update is shared on the selected social media platform.

Expected Result: The course information and updates should be successfully posted on the instructor's selected social media account, including relevant links and details.

Severity: Minor

Type of Testing: Functional Testing, Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SM_Integration_005

Test Case Title: Verify that analytics track the reach and engagement of shared content

Test Case Description: This test case verifies that the platform's analytics system accurately tracks and displays data related to the reach and engagement (likes, shares, comments) of content shared on social media.

Test Suite: Social Media Integration

Test Priority: Low

Preconditions:

- Admin user is logged in.
- Content has been shared on social media via the platform.

Test Data: No test data needed.

Test Steps:

1. Access the platform's analytics dashboard.
2. Locate the section for social media insights.
3. Verify the display of metrics like reach (number of users who saw the content), engagement (likes, shares, comments), and click-through rates.

Postconditions:

- Analytics data accurately reflects the social media engagement with shared content.

Expected Result: The analytics dashboard should provide accurate and comprehensive data on the reach and engagement metrics of the content shared through the platform's social media integration.

Severity: Minor

Type of Testing: Functional Testing, Data-Driven 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 1

Test Case ID: FTA-001

Test Case Title: Verify that the app records data on distance, duration, pace, and calories burned during an activity.

Test Case Description: This test case verifies the app's ability to accurately capture and display essential activity data, including distance, duration, pace, and calories burned, for various activity types like running, walking, and cycling.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

- User is logged in.

- User has granted necessary permissions (location, health data).

Test Data: No test data needed.

Test Steps:

1. Start a new activity tracking session.

2. Select an activity type (e.g., running, walking, cycling).

3. Perform the activity for a predetermined time and distance.

4. Stop the activity tracking session.

5. Verify that the app displays the recorded distance, duration, pace, and estimated calories burned.

Postconditions:

- Recorded activity data is saved.

Expected Result: The app accurately records and displays the distance, duration, pace, and estimated calories burned for the selected activity type.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: FTA-002

Test Case Title: Verify that users can view summaries of their recorded activities.

Test Case Description: This test case ensures users can easily access and view summarized data for their past activities, including daily, weekly, or monthly overviews, allowing them to track progress and identify trends.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

- User is logged in.

- User has recorded at least one activity.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Activity History" or "Activity Summary" section.

2. Select a desired timeframe for viewing the activity summary (daily, weekly, monthly).

3. Verify that the app displays a summary of activities for the selected timeframe.

4. Check for clear and concise presentation of key activity metrics.

Postconditions: No specific postconditions.

Expected Result: The app displays a clear and informative summary of the user's activities for the chosen timeframe, including relevant metrics.

Severity: Major

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: FTA-003

Test Case Title: Verify that activity data is stored for historical analysis and can be accessed by the user.

Test Case Description: This test case checks the app's ability to store activity data over time and allow users to retrieve and analyze their historical fitness progress.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

- User has recorded activities over a period of time.

Test Data: No test data needed.

Test Steps:

1. Navigate to the activity history or data export section.

2. Attempt to access and view activity data from previous days, weeks, or months.

3. Verify that the app retains and displays historical activity data accurately.

4. Check if users can filter or sort historical data by date, activity type, or other criteria.

Postconditions: No specific postconditions.

Expected Result: The app allows users to access, view, and potentially analyze their historical activity data without any data loss.

Severity: Major

Type of Testing: Functional Testing, Data Validation

Test Case Approach: Positive

## Test Case 4

Test Case ID: FTA-004

Test Case Title: Verify that activity tracking can be paused and resumed without data loss.

Test Case Description: This test case ensures the app accurately handles interruptions during activity tracking, allowing users to pause and resume sessions without compromising the integrity of recorded data.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: No test data needed.

Test Steps:

1. Start a new activity tracking session.

2. Pause the activity tracking after a short period.

3. Wait for a specific duration (e.g., 5 minutes).

4. Resume the activity tracking.

5. Continue the activity for a short period and then stop the session.

6. Verify that the app accurately records the total duration, distance, and other metrics considering both active periods.

Postconditions: Recorded activity data is saved.

Expected Result: The app allows pausing and resuming an activity tracking session without losing or corrupting recorded data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: FTA-005

Test Case Title: Verify that users receive notifications for milestones achieved during activity tracking.

Test Case Description: This test case checks if the app provides timely and motivational notifications to users upon reaching predefined milestones during their activities, enhancing user engagement.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

- User has enabled notifications for activity milestones.

Test Data: Pre-set milestones for distance, calories burned, or duration (e.g., 5km, 500 calories, 30 minutes).

Test Steps:

1. Start an activity tracking session.

2. Perform the activity until a predefined milestone is reached.

3. Verify that the app delivers a notification to the user upon reaching the milestone.

4. Check the content and format of the notification for clarity and relevance.

Postconditions: No specific postconditions.

Expected Result: The app sends a clear and timely notification to the user when they achieve a predefined milestone during an activity.

Severity: Minor

Type of Testing: Functional Testing, Usability 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 Cases for Fitness Tracker App - GPS Tracking Feature

Test Case 1:

Test Case ID: FT-GPS-001

Test Case Title: Verify that GPS tracking accurately records route data.

Test Case Description: This test case verifies the accuracy of GPS tracking by comparing the recorded route data with a known distance and elevation profile.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

* User is logged in.

* GPS is enabled on the device.

* User has granted location permissions to the app.

Test Data: A pre-defined outdoor route with known distance and elevation changes (e.g., a local park trail).

Test Steps:

1. Start a new outdoor activity tracking session.

2. Begin moving along the pre-defined route.

3. Complete the route and end the tracking session.

4. Compare the recorded distance and elevation profile with the known values of the pre-defined route.

Postconditions:

* The recorded route data is saved to the user's activity history.

Expected Result: The recorded route data should closely match the known distance and elevation profile of the pre-defined route, with an acceptable margin of error (e.g., within 5%).

Severity: Major

Type of Testing: Functional, System Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-GPS-002

Test Case Title: Verify that users can view detailed maps and elevation profiles.

Test Case Description: This test verifies if users can access and interact with detailed maps and elevation profiles for their tracked activities.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

* User has completed an outdoor activity with GPS tracking enabled.

Test Data: No test data needed.

Test Steps:

1. Navigate to the activity history section of the app.

2. Select the completed activity with GPS data.

3. Verify that a map displaying the recorded route is visible.

4. Check for options to zoom in/out and pan the map.

5. Verify that an elevation profile corresponding to the route is displayed.

Postconditions: None

Expected Result: The app should display a detailed map of the recorded route with zoom and pan functionalities. An elevation profile accurately representing the route's altitude changes should also be accessible.

Severity: Major

Type of Testing: Functional, Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-GPS-003

Test Case Title: Verify that performance data includes location-based metrics.

Test Case Description: This test case checks whether the app provides location-based metrics such as average pace/speed, split times, and elevation gain/loss for tracked activities.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

* User has completed an outdoor activity with GPS tracking enabled.

Test Data: No test data needed.

Test Steps:

1. Access the performance data summary for the completed activity.

2. Verify that the summary includes average pace/speed.

3. Check for the presence of split times at regular intervals (e.g., every kilometer/mile).

4. Confirm that total elevation gain and loss are displayed.

Postconditions: None

Expected Result: The activity summary should accurately display location-based metrics, including average pace/speed, split times, and elevation gain/loss, based on the recorded GPS data.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-GPS-004

Test Case Title: Verify that routes can be saved and revisited.

Test Case Description: This test case ensures users can save completed routes and access them later for revisiting or sharing purposes.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

* User has completed an outdoor activity with GPS tracking enabled.

Test Data: No test data needed.

Test Steps:

1. After completing the activity, locate the option to save the route.

2. Save the route with a recognizable name.

3. Navigate to the "Saved Routes" section within the app.

4. Verify that the saved route appears in the list.

5. Select the saved route to view its details and map.

Postconditions: The saved route is stored within the user's profile and accessible for future use.

Expected Result: The app should allow users to save completed routes. Saved routes should be accessible for viewing, sharing, or initiating a new activity following the same route.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-GPS-005

Test Case Title: Verify that notifications are provided for deviations from planned routes.

Test Case Description: This test case confirms that the app alerts users if they deviate significantly from a pre-planned route during an activity.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

* User has planned a route using the app's route planning feature.

* User has started an activity following the planned route with GPS tracking enabled.

Test Data: No test data needed.

Test Steps:

1. During the activity, deviate significantly from the planned route.

2. Observe if the app provides a notification indicating the deviation.

3. Check if the notification offers options to return to the planned route or continue off-route.

Postconditions: The app may continue tracking the user's actual route even after deviation.

Expected Result: The app should notify the user, visually or audibly, when a significant deviation from the planned route is detected. The notification should offer options to get back on track or continue with the current path.

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 Cases for Fitness Tracker App - Heart Rate Monitoring

Test Case 1:

Test Case ID: FTHRM-001

Test Case Title: Verify that heart rate data is recorded in real-time.

Test Case Description: This test case verifies that the app accurately captures and displays heart rate data in real-time from both built-in sensors and compatible wearable devices.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in.
- Heart rate sensor is activated (built-in or wearable).

Test Data: No test data needed

Test Steps:

1. Navigate to the "Heart Rate" section of the app.
2. Initiate heart rate monitoring.
3. Observe the heart rate display for real-time updates.
4. Compare the displayed heart rate to a secondary heart rate source (e.g., pulse oximeter).

Postconditions: None

Expected Result: The app should display real-time heart rate readings that are consistent with the secondary source, with minimal lag.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTHRM-002

Test Case Title: Verify that users can set and monitor target heart rate zones.

Test Case Description: This test case validates the functionality of setting personalized target heart rate zones and receiving feedback during workouts.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in.
- User has input their age and resting heart rate in settings.

Test Data: User-defined target heart rate zones (e.g., Fat Burn, Cardio, Peak).

Test Steps:

1. Navigate to the "Heart Rate Zones" section in settings.
2. Input desired target heart rate ranges for each zone.
3. Start a workout and monitor heart rate feedback.

Postconditions: None

Expected Result: The app should visually indicate which zone the user is currently in during their workout, providing real-time feedback.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTHRM-003

Test Case Title: Verify that historical heart rate data is stored and viewable.

Test Case Description: This test case ensures that the app accurately stores historical heart rate data and provides users with meaningful ways to access and visualize it.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

- User is logged in.
- Heart rate data from at least one workout session is available.

Test Data: No test data needed

Test Steps:

1. Navigate to the "History" or "Progress" section.
2. Select a past workout session.
3. View the detailed heart rate data for that session.
4. Verify that the data is displayed clearly in a graph or table format, including average heart rate, time spent in each zone, and maximum heart rate.

Postconditions: None

Expected Result: The app should display accurate and detailed historical heart rate data for selected workout sessions.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTHRM-004

Test Case Title: Verify that alerts are provided for abnormal heart rate readings.

Test Case Description: This test case validates that the app correctly identifies and alerts the user about potentially dangerous heart rate readings during workouts.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in.
- User has enabled heart rate alerts in settings.

Test Data: Simulated abnormally high or low heart rate readings.

Test Steps:

1. Set up a simulated workout with controlled heart rate input (e.g., manually inputting data).
2. Input heart rate readings that exceed the user-defined safe range.
3. Observe the app for visual or audio alerts.

Postconditions: None

Expected Result: The app should display a clear warning message and/or sound an alarm when abnormal heart rate readings are detected, prompting the user to check their health.

Severity: Critical

Type of Testing: Functional Testing, Security Testing

Test Case Approach: Negative

Test Case 5:

Test Case ID: FTHRM-005

Test Case Title: Verify that heart rate data is compatible with various wearable devices.

Test Case Description: This test ensures that the app can successfully connect and receive accurate heart rate data from a range of popular wearable fitness trackers.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in.
- User has a compatible wearable device (e.g., Fitbit, Garmin, Apple Watch).

Test Data: A selection of different compatible wearable devices.

Test Steps:

1. Navigate to the app's settings and access the "Devices" or "Connections" section.
2. Follow the on-screen instructions to pair the wearable device with the app.
3. Initiate heart rate monitoring on both the app and the wearable device.
4. Compare the displayed heart rate readings between the app and the device.

Postconditions: Unpair the device after testing if necessary.

Expected Result: The app should successfully connect and display heart rate data from all tested compatible wearable devices with minimal discrepancies.

Severity: Major

Type of Testing: Integration 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 1

Test Case ID: TC_CalorieTracking_001

Test Case Title: Verify that users can log daily food intake and calorie consumption.

Test Case Description: This test case aims to ensure that users can successfully log their food intake for different meals and the app accurately records the calorie consumption for each entry.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has navigated to the calorie tracking section.

Test Data:

- Meal type (Breakfast, Lunch, Dinner, Snack)
- Food item (e.g., Apple, Salad, Chicken Breast)

- Quantity (e.g., 1 medium, 100 grams, 1 serving)

Test Steps:

1. Select a meal type from the options.

2. Search for the desired food item in the database.

3. Select the food item and specify the consumed quantity.

4. Save the food log entry.

5. Verify that the food item, quantity, and calculated calories are displayed correctly in the food log.

Postconditions:

- The food log should reflect the accurate entry.

Expected Result: The app should allow users to easily log various food items with their respective quantities and display the calculated calorie consumption for each meal entry.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 2

Test Case ID: TC_CalorieTracking_002

Test Case Title: Verify that the app calculates and displays total calories consumed and burned.

Test Case Description: This test case aims to verify that the app accurately calculates and displays the total calories consumed from food logs and calories burned from recorded physical activities.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged in.

- User has logged at least one food item.

- User has recorded at least one physical activity.

Test Data: No test data needed

Test Steps:

1. Navigate to the calorie tracking dashboard.
2. Verify that the total calories consumed for the day are displayed.
3. Verify that the total calories burned for the day are displayed.
4. Verify that the difference between consumed and burned calories is calculated and displayed clearly.

Postconditions: None

Expected Result: The app should accurately calculate and prominently display the total calories consumed, burned, and the remaining calorie balance.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 3

Test Case ID: TC_CalorieTracking_003

Test Case Title: Verify that users can set daily calorie goals and track progress.

Test Case Description: This test case ensures users can set personalized daily calorie goals and the app effectively tracks their progress toward their goals.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged in.

Test Data:

- Desired daily calorie intake goal (e.g., 1800 calories)

Test Steps:

1. Navigate to the calorie goal setting section.

2. Enter the desired daily calorie intake goal.
3. Save the calorie goal.
4. Log food intake throughout the day.
5. Verify that the app visually represents the progress toward the set calorie goal (e.g., using a progress bar, percentage, or remaining calories).

Postconditions: None

Expected Result: The app should allow setting a daily calorie goal and provide clear visual feedback on the user's progress, updating dynamically as they log food consumption.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 4

Test Case ID: TC_CalorieTracking_004

Test Case Title: Verify that food items and calorie information are searchable in the database.

Test Case Description: This test case ensures that the food database is comprehensive and users can easily search for specific food items and their corresponding calorie information.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User is in the food logging section.

Test Data:

- List of common and unique food items (e.g., Apple, Broccoli, Quinoa, specific brand-name cereal)

Test Steps:

1. Use the search bar to look up each food item from the test data.

2. Verify that the search results accurately display the intended food items.

3. Select a few food items from the search results.

4. Verify that the correct calorie information for different serving sizes is displayed for each selected item.

Postconditions: None

Expected Result: The app should have a robust search functionality that allows users to find desired food items, even with partial words or brand names, and display accurate calorie information.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

## Test Case 5

Test Case ID: TC_CalorieTracking_005

Test Case Title: Verify that users receive alerts for exceeding calorie goals.

Test Case Description: This test case validates that the app provides timely alerts to users when they approach or exceed their pre-set daily calorie goals.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

- User has set a daily calorie intake goal.

Test Data:

- Food log entries that would lead to reaching and exceeding the set calorie goal.

Test Steps:

1. Log food items until reaching near the daily calorie goal.

2. Continue logging food items that exceed the set calorie goal.

3. Verify that the app displays a clear alert notification when the user approaches their calorie goal (e.g., a

pop-up message, visual cue on the progress bar).

4. Verify that a different alert notification is displayed when the user exceeds their daily calorie goal.

Postconditions: None

Expected Result: The app should issue timely and distinct alerts to users when they are nearing and when they have exceeded their set daily calorie goal, promoting awareness and control over food consumption.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

--

Data - 80

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Sleep Tracking Feature

Test Case 1:

Test Case ID: FT-ST-001

Test Case Title: Verify that sleep data is recorded including duration and quality.

Test Case Description: This test case ensures that the app accurately records and displays sleep duration and quality metrics after a sleep session.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has granted necessary permissions (e.g., motion activity, heart rate).
- User has initiated sleep tracking before going to bed.

Test Data: No test data needed.

Test Steps:

1. Start sleep tracking before going to sleep.
2. Sleep for a period.
3. Stop sleep tracking upon waking up.
4. Navigate to the sleep tracking section of the app.

Postconditions:

- Sleep tracking data is stored.

Expected Result: The app should display the sleep duration and a quality metric (e.g., score, rating) based on the recorded data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-ST-002

Test Case Title: Verify that users can view detailed sleep stages and patterns.

Test Case Description: This test case validates that the app provides a detailed breakdown of sleep stages (e.g., light, deep, REM) and their duration.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- Sleep data from at least one sleep session is available.

Test Data: No test data needed.

Test Steps:

1. Navigate to the sleep tracking section of the app.
2. Select a recorded sleep session to view details.

Postconditions:

- No specific postconditions.

Expected Result: The app should display a visual representation (e.g., graph, chart) of sleep stages and their duration for the selected session.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-ST-003

Test Case Title: Verify that the app provides insights into sleep disturbances.

Test Case Description: This test case assesses if the app identifies and provides insights into factors that may have disrupted sleep.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- Sleep data containing disturbances (e.g., movements, heart rate spikes) is available.

Test Data: No test data needed.

Test Steps:

1. Navigate to the sleep details for a session with recorded disturbances.
2. Check if the app highlights or provides insights into potential causes of the disturbances.

Postconditions:

- No specific postconditions.

Expected Result: The app should identify periods of wakefulness or restlessness and potentially correlate them with factors like movement or heart rate variations.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-ST-004

Test Case Title: Verify that users receive recommendations for improving sleep quality.

Test Case Description: This test case verifies that the app provides personalized recommendations to users based on their sleep patterns and identified issues.

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- The app has analyzed sufficient sleep data to generate personalized recommendations.

Test Data: No test data needed.

Test Steps:

1. Navigate to the sleep tracking section.
2. Look for a dedicated section or feature offering sleep improvement recommendations.

Postconditions:

- No specific postconditions.

Expected Result: Based on the user's sleep data, the app should provide relevant recommendations such as establishing a consistent sleep schedule, creating a relaxing bedtime routine, or suggesting improvements to sleep environment.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-ST-005

Test Case Title: Verify that sleep data can be viewed in daily, weekly, and monthly summaries.

Test Case Description: This test case ensures that users can access and view their sleep data in different summarized views.

Test Suite: Sleep Tracking

Test Priority: Low

Preconditions:

- Sleep data for multiple days, weeks, or months is available.

Test Data: No test data needed.

Test Steps:

1. Navigate to the sleep tracking section.
2. Check for options to switch between daily, weekly, and monthly data views.
3. Select each view option and ensure data is displayed accordingly.

Postconditions:

- No specific postconditions.

Expected Result: The app should accurately present sleep data (e.g., duration, quality) in daily, weekly, and monthly summaries, allowing users to analyze trends over time.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 81

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Logging

Feature description -

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

Userstory title -

As a user, I want to log my workouts so that I can track my progress and improve my performance.

Acceptance criteria -

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Workout Logging Feature

Test Case 1:

Test Case ID: FT-WL-001

Test Case Title: Verify that users can log various workout types with details.

Test Case Description: This test case verifies that users can successfully log different types of workouts with details such as exercise name, duration, and intensity level. It ensures that the app accurately records and displays the entered information.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

  - User is logged in.

- User has navigated to the workout logging section.

Test Data:

- Workout Type: Running

- Exercise Name: Outdoor Run

- Duration: 45 minutes

- Intensity: Moderate

- Notes: Beautiful day for a run!

Test Steps:

1. Select "Log Workout."

2. Choose "Running" as the workout type.

3. Enter "Outdoor Run" as the exercise name.

4. Set the duration to "45 minutes."

5. Select "Moderate" for intensity.

6. Add the note "Beautiful day for a run!"

7. Save the workout log.

Postconditions:

- The workout log is saved successfully.

- The log is accessible in the workout history.

Expected Result: The app should successfully record and display the workout log with all entered details.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-WL-002

Test Case Title: Verify that workout logs include duration, intensity, and notes.

Test Case Description: This test case validates that the app correctly captures and displays the duration, intensity level, and any additional notes entered by the user for each logged workout.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User is logged in.
- A workout has been previously logged.

Test Data: No test data needed

Test Steps:

1. Navigate to the workout history.
2. Select a previously logged workout.

Postconditions: None

Expected Result: The app should display the duration, intensity level, and notes associated with the selected workout.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-WL-003

Test Case Title: Verify that users can edit and delete workout entries.

Test Case Description: This test case ensures users can modify or remove incorrect or unwanted workout entries from their log, maintaining data accuracy.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged in.

- A workout has been previously logged.

Test Data: No test data needed

Test Steps:

1. Navigate to the workout history.

2. Select a previously logged workout.

3. Choose the "Edit" option.

4. Modify the workout details (e.g., change duration, intensity, notes).

5. Save the changes.

6. Select the same workout entry again.

7. Choose the "Delete" option.

8. Confirm the deletion.

Postconditions:

- The edited workout log reflects the changes made.

- The deleted workout log is removed from the history.

Expected Result: The app should allow editing and deleting workout entries successfully, updating or removing them from the log accordingly.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-WL-004

Test Case Title: Verify that data is stored and viewable for tracking progress.

Test Case Description: This test case confirms that logged workout data is persistently stored by the application and can be accessed by the user for reviewing their exercise history and progress over time.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User is logged in.
- Multiple workouts have been logged over different dates.

Test Data: No test data needed

Test Steps:

1. Navigate to the workout history.
2. Observe the list of logged workouts.

Postconditions: None

Expected Result: The app should display a chronological list of logged workouts, allowing the user to view their complete workout history.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-WL-005

Test Case Title: Verify that users can filter logs by date and workout type.

Test Case Description: This test case validates the filtering functionality, ensuring users can easily find specific workout entries in their history by filtering based on the date of the workout and/or the type of workout performed.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged in.
- Multiple workouts of different types have been logged on various dates.

Test Data: No test data needed

Test Steps:

1. Navigate to the workout history.
2. Select the filter option.
3. Apply a filter by date range (e.g., last week, last month).
4. Observe the filtered workout logs.
5. Clear the date filter.
6. Apply a filter by workout type (e.g., Running, Strength Training).
7. Observe the filtered workout logs.

Postconditions: None

Expected Result: The app should accurately filter and display workout logs that match the selected date range and/or workout type criteria.

Severity: Minor

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:

## Fitness Tracker App - Goal Setting Test Cases

Test Case 1:

Test Case ID: FT-GS-001

Test Case Title: Verify that users can set goals for various fitness metrics.

Test Case Description: This test case ensures users can set goals for different fitness metrics like steps taken, calories burned, distance traveled, and active minutes.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User is logged in.

- User is on the "Set Goals" screen.

Test Data: No test data needed

Test Steps:

1. Select a fitness metric from the available options (e.g., steps taken, calories burned).
2. Input a target value for the chosen metric.
3. Select a start date for the goal (optional).
4. Save the goal.

Postconditions:

- The goal is displayed on the user's goal dashboard.

Expected Result: The system should successfully save and display the set goal for the chosen fitness metric.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-GS-002

Test Case Title: Verify that progress tracking is available for each goal.

Test Case Description: This test case confirms that users can track their progress toward each set goal, including visual representations like progress bars and percentage completion.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User is logged in.
- User has set at least one fitness goal.

Test Data: No test data needed

Test Steps:

1. Navigate to the "Goals" section of the app.

2. Select an active goal.

3. Observe the progress details displayed, including current progress, target value, and remaining amount.

Postconditions: N/A

Expected Result: The system should accurately display the user's current progress towards their set goal, including visual representations.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-GS-003

Test Case Title: Verify that users receive motivational feedback and reminders.

Test Case Description: This test case validates that the app provides users with encouraging messages and reminders to stay motivated and on track with their goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- User is logged in.

- User has set at least one fitness goal.

Test Data: No test data needed

Test Steps:

1. Reach a milestone in a set goal (e.g., 50% completion).

2. Observe if the app displays a congratulatory message.

3. Set a reminder for a specific goal.

4. Verify the reminder notification is received at the scheduled time.

Postconditions: N/A

Expected Result: The system should display motivational messages upon reaching milestones and send timely reminders as set by the user.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-GS-004

Test Case Title: Verify that goals can be adjusted or updated as needed.

Test Case Description: This test case ensures users can modify existing goals, including changing the target value, start date, or deleting the goal entirely.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- User is logged in.
- User has set at least one fitness goal.

Test Data: No test data needed

Test Steps:

1. Go to the "Goals" section.
2. Select a goal to edit.
3. Modify the target value, start date, or choose to delete the goal.
4. Save the changes.

Postconditions:

- Updated goal is reflected in the goal dashboard.

- Deleted goal is removed from the dashboard.

Expected Result: The system should successfully update or delete the chosen goal based on user modifications.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-GS-005

Test Case Title: Verify that the app provides a summary of goal attainment.

Test Case Description: This test case checks that the app offers users a comprehensive overview of their goal achievement, potentially including completed goals, average performance, and areas for improvement.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

- User is logged in.
- User has completed at least one goal.

Test Data: No test data needed

Test Steps:

1. Navigate to a "Goal Summary" or "Progress Overview" section.
2. Verify the display of completed goals with achieved dates.
3. Observe if the summary provides insights or statistics related to goal performance.

Postconditions: N/A

Expected Result: The app should present a clear and informative summary of the user's goal attainment history, highlighting successes and areas for improvement.

Severity: Minor

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 Cases for Fitness Tracker App - Workout Plans Feature

Test Case 1

Test Case ID: FTap-WP-001

Test Case Title: Verify that users can select workout plans based on fitness levels

Test Case Description: This test case verifies that the app allows users to choose from a range of workout plans categorized by different fitness levels (e.g., Beginner, Intermediate, Advanced).

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in.
- User has navigated to the "Workout Plans" section.

Test Data: No test data needed

Test Steps:

1. Observe the available workout plans.
2. Verify that plans are categorized by fitness levels (e.g., Beginner, Intermediate, Advanced).
3. Select a fitness level category.
4. Confirm that the displayed plans align with the selected fitness level.

Postconditions: None

Expected Result: The app should display a variety of workout plans categorized by fitness levels, allowing users to easily find plans suitable for their experience.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: FTap-WP-002

Test Case Title: Verify that workout plans include detailed exercise routines and schedules

Test Case Description: This test case ensures that each workout plan provides a comprehensive breakdown of exercises, sets, reps, rest periods, and a clear schedule for the week/month.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in.
- User is viewing a specific workout plan.

Test Data: No test data needed

Test Steps:

1. Select a workout plan from any fitness level.
2. Verify that the plan includes a detailed breakdown of exercises for each workout session.
3. Confirm that each exercise includes information on sets, reps, and recommended rest periods.
4. Ensure the plan presents a clear schedule, outlining workout frequency and days.

Postconditions: None

Expected Result: The app should provide detailed exercise routines and schedules within each workout plan, offering users clear guidance on their training program.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: FTap-WP-003

Test Case Title: Verify that users can track adherence and progress within a workout plan

Test Case Description: This test case confirms that users can log completed workouts, track their progress against the plan's schedule, and view relevant metrics related to their adherence.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in.
- User has started following a workout plan.

Test Data: Completed workout data

Test Steps:

1. Complete a workout session within the chosen plan.
2. Log the workout within the app, recording relevant metrics.
3. Navigate to the workout plan section.
4. Verify that the app visually indicates completed workouts within the plan's schedule.
5. Confirm that users can access a summary of their adherence (e.g., workouts completed vs. planned).

Postconditions: None

Expected Result: The app should allow users to log workouts, visually track their adherence to the plan's schedule, and provide insights into their overall progress.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: FTap-WP-004

Test Case Title: Verify that the app provides notifications and reminders for workouts

Test Case Description: This test case ensures that the app sends timely notifications to remind users of upcoming workouts according to their chosen plan's schedule.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is logged in.
- User has granted notification permissions to the app.
- User has started following a workout plan.

Test Data: No test data needed

Test Steps:

1. Observe the time remaining until the next scheduled workout.
2. Verify that the app delivers a notification reminding the user of the upcoming workout.
3. Confirm that the notification includes relevant details such as the workout name and scheduled time.

Postconditions: None

Expected Result: The app should send timely and informative notifications to remind users of upcoming workout sessions, helping them stay on track with their plan.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: FTap-WP-005

Test Case Title: Verify that users can adjust workout plans based on progress and preferences.

Test Case Description: This test case ensures users can customize workout plans by swapping exercises, adjusting workout frequency, or switching to a different plan altogether.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is logged in.
- User is viewing a specific workout plan.

Test Data: No test data needed

Test Steps:

1. Explore the available options to modify the current workout plan.
2. Attempt to swap exercises within a workout session.
3. Try adjusting the workout frequency (e.g., from 3 days/week to 2 days/week).
4. Explore the option to switch to a different workout plan entirely.

Postconditions: None

Expected Result: The app should provide flexibility, allowing users to personalize their workout plans by swapping exercises, adjusting workout frequency, or switching to different plans based on their progress, preferences, and needs.

Severity: Minor

Type of Testing: Functional Testing, Usability 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 Cases for Fitness Tracker App - Community Support Feature

Test Case 1:

Test Case ID: FT-CS-001

Test Case Title: Verify that users can join and participate in community groups.

Test Case Description: This test case verifies that users can successfully browse available community groups, send join requests, and participate in group activities once accepted.

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is logged in.
- Community groups are created.

Test Data: No test data needed

Test Steps:

1. Navigate to the Community section.
2. Browse available groups.
3. Select a group and send a join request.
4. (Assuming approval) Access the group and participate in discussions or challenges.

Postconditions:

- User is added to the selected group.
- User activity is reflected in the group.

Expected Result: The user should be able to join community groups and engage in group activities.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-CS-002

Test Case Title: Verify that users can share progress updates and achievements.

Test Case Description: This test case ensures that users can share their fitness progress, achievements, and milestones within their joined community groups.

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is logged in.
- User is a member of at least one community group.

Test Data: Sample progress update text, achievement image (optional)

Test Steps:

1. Navigate to the desired community group.

2. Locate the "Share Update" or similar option.

3. Enter a progress update message (optional: include an image).

4. Submit the update.

Postconditions:

- The update is visible to other group members.

Expected Result: The user's progress update is successfully shared within the group.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-CS-003

Test Case Title: Verify that the app facilitates participation in community challenges.

Test Case Description: This test case checks that users can view, join, participate in, and track their progress within community challenges.

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is logged in.

- Active community challenges are available.

Test Data: No test data needed

Test Steps:

1. Navigate to the Community Challenges section.

2. Browse available challenges.

3. Join a challenge.

4. Track progress within the challenge.

Postconditions:

- User's challenge participation and progress are recorded.

Expected Result: The user should be able to join and participate in community challenges and their progress should be tracked accurately.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-CS-004

Test Case Title: Verify that users can offer and receive support and encouragement.

Test Case Description: This test case ensures users can interact with each other by providing and receiving support, encouragement, or feedback on shared updates within the community.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- User is logged in.
- User is a member of a community group with active posts.

Test Data: Sample encouraging message or comment

Test Steps:

1. View a community group post or update.
2. Leave a comment offering support or encouragement.
3. Observe other users' comments on their own posts.

Postconditions:

- Comments are visible to other users in the group.

Expected Result: Users should be able to successfully post and view comments, fostering a supportive community environment.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-CS-005

Test Case Title: Verify that community activity and engagement are tracked.

Test Case Description: This test case checks if the app tracks and potentially displays relevant metrics related to community engagement, such as the number of active users, posts, likes, or challenge participation.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- App has been used for a period with community interaction.

Test Data: No test data needed

Test Steps:

1. Navigate to relevant sections where community engagement might be displayed (e.g., community overview, user profiles).
2. Observe if metrics like active users, posts, or challenge participation are tracked and displayed.

Postconditions: N/A

Expected Result: The app should track and potentially display relevant metrics to provide insights into community activity and engagement.

Severity: Minor

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:

## Fitness Tracker App - Wearable Device Integration Test Cases

Here are 5 test cases designed to cover the acceptance criteria for the "Integration with Wearable Devices" feature:

Test Case 1:

Test Case ID: FIT-WID-001

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: This test case aims to confirm that the Fitness Tracker App can successfully connect and integrate with a variety of wearable devices from different manufacturers, ensuring broad compatibility.

Test Suite: Wearable Device Integration

Test Priority: High

Preconditions:

- The Fitness Tracker App is installed on the test device.
- A range of wearable devices from different manufacturers are available (e.g., Fitbit, Garmin, Apple Watch).

Test Data: List of supported wearable devices

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the "Settings" or "Devices" section.
3. Select "Add New Device."
4. Choose a wearable device from the list of supported devices.
5. Follow the on-screen instructions to complete the pairing process for each device.

Postconditions:

- The app should successfully connect to each wearable device.

Expected Result: The app should display a confirmation message upon successful connection with each wearable device, indicating proper integration.

Severity: Blocker

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FIT-WID-002

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: This test case verifies the app's ability to accurately receive and synchronize activity data (steps, heart rate, calories burned) from a connected wearable device.

Test Suite: Wearable Device Integration

Test Priority: High

Preconditions:

- A wearable device is successfully paired with the app.
- The wearable device has recorded activity data.

Test Data: No test data needed

Test Steps:

1. Initiate a data synchronization process from the app.
2. Monitor the app for data updates.

Postconditions: N/A

Expected Result: The app should display the synced activity data from the wearable device, matching the data recorded by the wearable device itself.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FIT-WID-003

Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: This test validates the app's ability to receive and display real-time data updates from a connected wearable during an ongoing activity.

Test Suite: Wearable Device Integration

Test Priority: High

Preconditions:

- A wearable device is actively connected to the app.
- The user is currently engaged in a physical activity while wearing the device.

Test Data: No test data needed

Test Steps:

1. Start an activity tracking session on both the app and the wearable device.
2. Observe the app's display for real-time updates of steps, heart rate, and calories burned.

Postconditions: N/A

Expected Result: The app should dynamically update with the real-time data transmitted from the wearable device with minimal delay.

Severity: Major

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FIT-WID-004

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: This test ensures the app collects and displays all the critical data points ? steps taken, heart rate, and calories burned ? transmitted by the wearable device.

Test Suite: Wearable Device Integration

Test Priority: Medium

Preconditions:

- A wearable device is synced with the app.

- The wearable has recorded data for all three metrics: steps, heart rate, and calories burned.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the data display section.

2. Verify that the app displays data for steps taken.

3. Verify that the app displays data for heart rate.

4. Verify that the app displays data for calories burned.

Postconditions: N/A

Expected Result: The app should accurately present the data for all three metrics received from the wearable device.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FIT-WID-005

Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: This test evaluates the app's ability to guide users through common connectivity problems with their wearable devices.

Test Suite: Wearable Device Integration

Test Priority: Medium

Preconditions:

- The app has a dedicated section for troubleshooting wearable connectivity issues.

Test Data: List of potential connectivity issues (e.g., Bluetooth disabled, device out of range, pairing errors)

Test Steps:

1. Simulate a connectivity issue (e.g., disable Bluetooth).
2. Attempt to connect or sync data with the wearable device.
3. Access the app's troubleshooting section.
4. Follow the provided instructions to resolve the simulated issue.

Postconditions:

- Connectivity between the app and wearable device is restored.

Expected Result: The app should provide clear and effective solutions within the troubleshooting section to address common connectivity problems.

Severity: Minor

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:

## Fitness Tracker App - Water Intake Tracking Test Cases

Here are 5 test cases designed to cover the acceptance criteria for the Water Intake Tracking feature:

Test Case 1

Test Case ID: FIT-Water-001

Test Case Title: Verify that users can log their daily water intake.

Test Case Description: This test case verifies the functionality for users to manually input and record their water consumption throughout the day. This includes adding individual entries for water intake with specific amounts and timestamps.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

* User is logged into the app.

* Water Intake Tracking feature is enabled.

Test Data:

* Water intake amount: 250 ml, Time: 10:00 AM

* Water intake amount: 500 ml, Time: 12:30 PM

* Water intake amount: 300 ml, Time: 2:00 PM

Test Steps:

1. Navigate to the Water Intake Tracking feature.

2. Select the option to add a new water intake record.

3. Input the amount of water consumed (e.g., 250 ml).

4. Input the time of consumption (e.g., 10:00 AM).

5. Save the water intake record.

6. Repeat steps 2-5 for different amounts and times.

Postconditions:

* Water intake records are saved accurately in the user's log.

Expected Result: The app successfully records and displays all entered water intake entries with correct amounts and timestamps.

Severity: Major

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case 2

Test Case ID: FIT-Water-002

Test Case Title: Verify that the app calculates total water consumption for the day.

Test Case Description: This test case checks the app's ability to accurately calculate and display the user's total water intake throughout the day, based on their logged entries.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

* User is logged in.

* At least one water intake record is logged for the current day.

Test Data: No test data needed

Test Steps:

1. Navigate to the Water Intake Tracking feature.

2. Observe the displayed total water consumption for the day.

Postconditions: N/A

Expected Result: The app displays the correct total water consumption for the day, reflecting the sum of all logged water intake entries.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: FIT-Water-003

Test Case Title: Verify that users can set daily hydration goals.

Test Case Description: This test case ensures users can easily set personalized daily hydration goals within the app.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

* User is logged into the app.

Test Data:

* Daily hydration goal: 2000 ml

Test Steps:

1. Navigate to the Water Intake Tracking feature settings.

2. Locate the "Daily Goal" or similar option.

3. Input the desired daily water intake goal (e.g., 2000 ml).

4. Save the changes to the hydration goal.

Postconditions: The user's daily hydration goal is updated within the app.

Expected Result: The app successfully saves and displays the user's chosen daily hydration goal.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: FIT-Water-004

Test Case Title: Verify that reminders are provided to encourage regular water intake.

Test Case Description: This test case verifies the app's reminder functionality for prompting users to drink water at regular intervals, based on their set preferences.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

* User is logged in.

* Daily hydration goal is set.

* Reminder frequency is customized (e.g., every 2 hours).

Test Data: No test data needed

Test Steps:

1. Wait for the set reminder interval to pass (e.g., 2 hours).

2. Observe if the app delivers a reminder notification to drink water.

Postconditions: N/A

Expected Result: The app sends timely and customizable reminder notifications, prompting users to stay hydrated throughout the day.

Severity: Minor

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case 5

Test Case ID: FIT-Water-005

Test Case Title: Verify that users can view hydration progress and trends.

Test Case Description: This test case verifies that the app provides visual representations of the user's water intake progress, potentially through graphs, charts, or other data visualization methods.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

* User is logged in.

* Water intake data for multiple days is available.

Test Data: No test data needed

Test Steps:

1. Navigate to the Water Intake Tracking feature.

2. Locate the section for viewing hydration history or trends.

Postconditions: N/A

Expected Result: The app displays clear and informative visuals (graphs, charts, etc.) depicting the user's hydration progress over time, allowing them to analyze their hydration habits.

Severity: Minor

Type of Testing: Functional, Usability

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 Cases for Fitness Tracker App - Nutrition Tracking:

Test Case 1:

Test Case ID: FT-NT-001

Test Case Title: Verify that users can log their daily meals and snacks.

Test Case Description: This test case ensures that users can successfully add, edit, and delete meal entries including food items and portion sizes, for different meals throughout the day.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has navigated to the Nutrition Tracking section.

Test Data:

- Meal type (Breakfast, Lunch, Dinner, Snack)
- Food items (e.g., Oatmeal, Chicken Salad, Apple)
- Portion sizes (e.g., 1 cup, 100 grams, 1 medium)

Test Steps:

1. Navigate to the "Add Meal" section.
2. Select a meal type from the available options.
3. Search for and select food items from the food database.
4. Input the portion size for each food item.
5. Save the meal entry.
6. Verify that the meal entry is displayed correctly on the daily food log.
7. Edit the meal entry by modifying food items or portion sizes.
8. Save the changes and verify the updated information on the log.
9. Delete the meal entry and confirm its removal from the log.

Postconditions:

- The daily food log reflects accurate entries.

Expected Result: The app should allow adding, editing, and deleting meal entries accurately reflecting the user's daily food intake.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-NT-002

Test Case Title: Verify that the app calculates macronutrient intake (carbs, fats, proteins).

Test Case Description: This test case verifies that the app accurately calculates and displays the total macronutrient intake based on the logged food items and their respective portion sizes.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has logged at least one meal with food items and portion sizes.

Test Data:

- Pre-verified food items with known macronutrient values (e.g., 100g chicken breast: 0g carbs, 3g fat, 30g protein)

Test Steps:

1. Navigate to the daily food log or nutrition summary page.
2. Observe the displayed values for total calories, carbs, fats, and protein consumed.
3. Manually calculate the expected macronutrient intake based on the logged food data.
4. Compare the app's calculated values with the manually calculated values.

Postconditions: N/A

Expected Result: The app should accurately calculate and display the total calorie and macronutrient intake, matching the expected values based on logged food data.

Severity: Critical

Type of Testing: Functional Testing, Calculation Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-NT-003

Test Case Title: Verify that users can set nutritional goals and track progress.

Test Case Description: This test case ensures that users can set personalized daily or weekly goals for calorie and macronutrient intake, and the app tracks their progress against these goals.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has access to the nutrition goal setting section.

Test Data:

- Calorie goal (e.g., 2000 calories/day)
- Macronutrient goals (e.g., 50% carbs, 30% protein, 20% fats)

Test Steps:

1. Navigate to the goal setting section within Nutrition Tracking.
2. Input desired calorie intake goal.
3. Set target percentages for carbohydrate, protein, and fat intake.
4. Save the set goals.
5. Log meals throughout the day.
6. View the progress bar or visual representation of goal progress in relation to actual intake.

Postconditions:

- The app should store the user's set goals.

Expected Result: The app should allow users to input and save their nutritional goals and provide clear visual feedback on their progress towards those goals as they log meals.

Severity: Major

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-NT-004

Test Case Title: Verify that the app provides nutritional information and suggestions.

Test Case Description: This test case assesses whether the app provides detailed nutritional information for individual food items and offers personalized suggestions or recommendations based on logged meals and set goals.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has logged at least one meal.
- User may or may not have set nutritional goals (test both scenarios).

Test Data:

- Sample logged meals with varying nutritional content.

Test Steps:

1. Select a logged food item from the food diary.
2. Verify that the app displays detailed nutritional information, including calories, macronutrient breakdown, vitamins, minerals, etc.

3. Observe if the app provides any suggestions or recommendations based on the logged meal, such as alternative food choices or portion adjustments to better align with potential goals.

Postconditions: N/A

Expected Result: The app should display comprehensive nutritional information for individual food items and optionally provide relevant suggestions based on user data and goals.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-NT-005

Test Case Title: Verify that users can view their nutritional history and trends.

Test Case Description: This test case ensures that users can access historical data of their logged meals and nutritional intake over different time periods and identify trends in their eating habits.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has logged meal data for at least one week.

Test Data:

- No specific test data needed.

Test Steps:

1. Navigate to the "History" or "Trends" section within Nutrition Tracking.
2. Select different timeframes to view past data (e.g., last week, last month).
3. Verify that the app displays a graphical representation or a summary of calorie and macronutrient intake over the chosen periods.

4. Check for features that highlight potential trends, patterns, or inconsistencies in the user's nutritional habits.

Postconditions: N/A

Expected Result: The app should provide users with a clear and easy-to-understand overview of their nutritional history through graphs, charts, or summaries, allowing them to identify trends and patterns in their eating habits.

Severity: Minor

Type of Testing: Functional Testing, UI Testing

Test Case Approach: Positive

Data - 88

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Progress Reports

Feature description -

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

Userstory title -

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

Acceptance criteria -

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Progress Reports:

Test Case 1:

Test Case ID: FT-PR-001

Test Case Title: Verify that users receive weekly and monthly progress reports.

Test Case Description: This test case verifies that the app generates and delivers both weekly and monthly progress reports to the user, covering the specified reporting periods.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

- User is logged in.
- User has tracking data for at least one week.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Progress' or 'Reports' section of the app.
2. Verify that a weekly progress report is available for the current week.
3. Wait until the next week begins (or manipulate the device date/time).
4. Verify a new weekly report is generated, and a monthly report is available.

Postconditions: None

Expected Result: The app should successfully generate and display both weekly and monthly progress reports to the user, reflecting accurate data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-PR-002

Test Case Title: Verify that reports include activity levels, workout performance, and goal attainment.

Test Case Description: This test case ensures that progress reports comprehensively present data on user activity levels, details of workout performance (if applicable), and progress made towards set fitness goals.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

  - User is logged in.
  - User has at least one week of tracking data, including workouts and goals.

Test Data: Sample data for various activities, workouts, and a set goal (e.g., steps, calories, distance, workout type, goal target).

Test Steps:

  1. Generate a weekly or monthly progress report.
  2. Verify the report displays accurate data on activity levels (steps, calories, distance).
  3. Verify the report includes details of workout performance (type, duration, calories burned).
  4. Verify the report shows progress made towards the user's set goal, including percentage completion.

Postconditions: None

Expected Result: The progress report should clearly present all relevant data points: activity levels, workout performance, and goal attainment progress.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-PR-003

Test Case Title: Verify that users can view detailed breakdowns of achievements and trends.

Test Case Description: This test case ensures users can access in-depth breakdowns of their achievements and identify trends within their activity and performance data over time.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in.
- User has several weeks of tracking data available.

Test Data: No specific test data needed.

Test Steps:

1. Generate a weekly or monthly progress report.
2. Locate and select an achievement or metric within the report to view details.
3. Verify the app provides a detailed breakdown of the selected item (e.g., daily steps for the week, calorie trends over a month).
4. Verify users can easily navigate between different data breakdowns within the report.

Postconditions: None

Expected Result: The app should allow users to drill down into specific achievements and view clear, informative breakdowns and trend analysis of their data.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-PR-004

Test Case Title: Verify that the app provides recommendations based on report findings.

Test Case Description: This test case confirms that the app analyzes report data and provides personalized recommendations to users, suggesting improvements to their fitness routines or goal adjustments.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in.
- User has sufficient tracking data for the app to generate meaningful recommendations (e.g., several weeks of activity).

Test Data: No specific test data needed.

Test Steps:

1. Generate a weekly or monthly progress report.
2. Verify the report includes a section for 'Recommendations' or 'Insights.'
3. Confirm that recommendations are relevant to the user's report findings, such as suggesting an increase in activity if goals are consistently exceeded or offering workout variations based on performance.

Postconditions: None

Expected Result: The app should provide tailored recommendations based on individual user data, aiming to optimize their fitness journey.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-PR-005

Test Case Title: Verify that users can customize the frequency and content of reports.

Test Case Description: This test case verifies that users have the option to customize the frequency of receiving reports (e.g., weekly, bi-weekly, monthly) and personalize the content displayed.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: No specific test data needed.

Test Steps:

1. Navigate to the app's settings or preferences related to progress reports.
2. Verify users can adjust the report frequency (weekly, bi-weekly, monthly).
3. Confirm users can select specific data points to include or exclude from reports (e.g., steps, calories, workout details).
4. Save the changes and generate a new report to confirm the customizations are applied.

Postconditions: None

Expected Result: The app should allow for flexible report customization, enabling users to personalize frequency and content based on their preferences.

Severity: Minor

Type of Testing: Usability 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 Cases for Fitness Tracker App - In-App Challenges:

Test Case 1:

Test Case ID: FTA-IC-001

Test Case Title: Verify that users can join various in-app challenges.

Test Case Description: This test case aims to verify that users can successfully browse and join different types of in-app challenges available within the fitness tracker app.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app.
- The app has active in-app challenges.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Challenges" section of the app.
2. Verify that multiple challenges are displayed with details (title, description, duration, etc.).
3. Select a challenge and view its detailed information.
4. Tap on the "Join Challenge" button.
5. Confirm successful enrollment in the selected challenge.

Postconditions:

- The user is registered for the selected challenge.

Expected Result: The app allows users to explore different in-app challenges and successfully join them.

Severity: Major

Type of Testing: Functional, User Interface (UI) Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTA-IC-002

Test Case Title: Verify that challenges have clear objectives and rules.

Test Case Description: This test ensures that each in-app challenge within the fitness tracker app clearly defines its objectives, rules, and any specific criteria for participation or completion.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app.
- The app has active in-app challenges.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Challenges" section.
2. Select any challenge and view its detailed information.
3. Verify the presence of a clear and concise description of the challenge's objective.
4. Check for clearly stated rules, guidelines, or requirements for participation.
5. Ensure that the criteria for successful completion of the challenge are well-defined.

Postconditions: None.

Expected Result: Each challenge should have easily understandable objectives and rules displayed to the user.

Severity: Major

Type of Testing: Functional, Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTA-IC-003

Test Case Title: Verify that users can track their progress within the challenge.

Test Case Description: This test case confirms that users can easily monitor their real-time progress within an active in-app challenge, including relevant metrics and comparisons.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app.

- User has joined an active in-app challenge.

Test Data: No test data needed.

Test Steps:

1. Navigate to the active challenge section.

2. Select the joined challenge.

3. Verify that the current progress towards the challenge's objective is clearly displayed.

4. Check for visual representations of progress, such as progress bars, charts, or graphs.

5. Ensure that relevant metrics (steps, calories, distance, etc.) are updated and accurately reflected in the progress tracking.

Postconditions: None.

Expected Result: The app effectively tracks and displays the user's progress within the challenge.

Severity: Major

Type of Testing: Functional, UI Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTA-IC-004

Test Case Title: Verify that rewards are provided for achieving challenge milestones.

Test Case Description: This test case validates that the app correctly awards users with appropriate rewards upon successfully reaching specific milestones within an in-app challenge.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app.

- User is participating in an active in-app challenge.

Test Data: Predefined challenge milestone criteria.

Test Steps:

1. Ensure the user achieves a predefined milestone within the challenge.
2. Trigger the app to recognize the achievement of the milestone.
3. Verify that the app displays a notification or confirmation of the reward unlocked.
4. Check that the reward (badges, points, virtual items, etc.) is successfully credited to the user's account.
5. Confirm that the reward system aligns with the challenge's rules and predefined milestone criteria.

Postconditions:

- The user's profile reflects the awarded reward.

Expected Result: The app successfully recognizes and rewards users for achieving challenge milestones.

Severity: Major

Type of Testing: Functional, Positive

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTA-IC-005

Test Case Title: Verify that users can compete with friends and view leaderboards.

Test Case Description: This test case checks the functionality of the leaderboard system, allowing users to compare their progress with friends and other participants within an active challenge.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged into the app.
- User has joined a challenge that allows friend comparisons or has a leaderboard.

Test Data: No test data needed.

Test Steps:

1. Navigate to the active challenge with the leaderboard feature.
2. Verify that the leaderboard is accessible and displays the ranking of participants.
3. Check that the leaderboard accurately reflects the progress of the user and other participants.
4. If applicable, confirm that the user can search for and add friends to compare progress within the challenge.
5. Ensure that the displayed rankings are updated in real-time or at reasonable intervals.

Postconditions: None.

Expected Result: Users can effectively compare their progress with others and the leaderboard accurately reflects the standings.

Severity: Major

Type of Testing: Functional, UI Testing

Test Case Approach: Positive

Data - 90

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Customizable Dashboards

Feature description -

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

Userstory title -

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Acceptance criteria -

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Customizable Dashboards

Test Case 1:

Test Case ID: FT-CD-001

Test Case Title: Verify that users can select which metrics to display on their dashboard.

Test Case Description: This test case aims to verify that users can successfully choose and display their preferred fitness metrics on the dashboard. This includes adding and removing metrics from the display.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged into the app.
- User has navigated to the dashboard customization screen.

Test Data: No test data needed.

Test Steps:

1. Access the list of available fitness metrics.

2. Select a metric not currently displayed on the dashboard.

3. Verify that the selected metric is added to the dashboard display.

4. Select a metric currently displayed on the dashboard.

5. Verify that the selected metric is removed from the dashboard display.

Postconditions:

- The user's selected metrics are saved and persist across sessions.

Expected Result: The system should allow users to freely add or remove metrics from their dashboard view.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT-CD-002

Test Case Title: Verify that the app allows customization of dashboard layout and theme.

Test Case Description: This test case verifies the customization options for the dashboard layout and theme, including rearranging widgets and applying different themes.

Test Suite: Customizable Dashboards

Test Priority: Medium

Preconditions:

- User is logged into the app.

- User is on the dashboard customization screen.

Test Data: No test data needed

Test Steps:

1. Attempt to rearrange the position of displayed metrics on the dashboard (e.g., drag and drop).

2. Verify that the layout changes are applied and saved successfully.

3. Access the theme selection option for the dashboard.

4. Choose a different theme from the available options.

5. Verify that the selected theme is applied to the dashboard.

Postconditions:

- Customized layout and theme should persist across app sessions.

Expected Result: The system should provide options to customize the arrangement of elements and the visual theme of the dashboard.

Severity: Minor

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT-CD-003

Test Case Title: Verify that users can save and update their dashboard settings.

Test Case Description: This test case ensures that any changes made to the dashboard configuration are saved correctly by the app and persist across different sessions.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged into the app.

- User has made changes to their dashboard settings (metrics, layout, or theme).

Test Data: No test data needed

Test Steps:

1. Navigate away from the dashboard customization screen (e.g., to another section of the app).

2. Return to the dashboard screen.

3. Verify that all previously applied customization settings are still in effect.

Postconditions: None

Expected Result: All customized dashboard settings should be retained and loaded correctly each time the user accesses the dashboard.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT-CD-004

Test Case Title: Verify that dashboard displays are dynamic and update in real-time.

Test Case Description: This test simulates user activity and verifies if the displayed metrics on the dashboard update accordingly and reflect the latest data.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in and actively using the app.

- The app is connected to a data source (wearable device or manual input).

Test Data: Simulated user activity data (e.g., steps taken, heart rate).

Test Steps:

1. Initiate a simulated activity that would affect one of the displayed metrics.

2. Observe the corresponding metric displayed on the dashboard.

3. Verify that the metric updates in real-time to reflect the simulated activity.

Postconditions: None

Expected Result: The dashboard should dynamically update displayed metrics based on real-time user activity or data input.

Severity: Major

Type of Testing: Performance Testing, Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT-CD-005

Test Case Title: Verify that users can reset the dashboard to default settings.

Test Case Description: This test case confirms the functionality of resetting the dashboard to its original default configuration.

Test Suite: Customizable Dashboards

Test Priority: Medium

Preconditions:

- User is logged into the app.
- User has customized their dashboard settings.

Test Data: No test data needed.

Test Steps:

1. Locate the option to reset the dashboard settings (usually found in settings or customization menus).
2. Trigger the reset functionality.
3. Confirm the reset action (if a confirmation prompt is part of the app's design).
4. Verify that the dashboard display reverts to the original default settings for metrics, layout, and theme.

Postconditions:

- User preferences should be removed, and the dashboard should reflect the app's standard configuration.

Expected Result: The system should successfully revert all dashboard settings to their default state upon user request.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 91

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Social Sharing

Feature description -

Users can share workout summaries, achievements, and progress updates on social media platforms, inspiring others and fostering accountability. Social sharing enables users to celebrate their successes, seek support from their social networks, and stay motivated on their fitness journey.

Userstory title -

As a user, I want to share my fitness achievements on social media so that I can inspire others and stay motivated.

Acceptance criteria -

['Users can share workout summaries and achievements.', 'The app supports sharing on multiple social media platforms.', 'Users can customize the content and format of shared posts.', 'Sharing features are integrated seamlessly with social media.', 'Users can track the engagement of their shared posts.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Social Sharing Feature

Test Case 1:

Test Case ID: FTS-SS-001

Test Case Title: Verify that users can share workout summaries and achievements.

Test Case Description: This test case aims to confirm that users can successfully share summaries of their completed workouts, including details such as workout type, duration, distance, calories burned, and any achieved milestones, to their chosen social media platforms.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

  - User is logged in.
  - User has completed a workout and saved the data.
  - User has granted the app necessary social media permissions.

Test Data: Sample workout summary with relevant details.

Test Steps:

  1. Navigate to the 'Workout History' section.
  2. Select a completed workout.
  3. Tap on the 'Share' button.
  4. Choose a social media platform from the available options.
  5. Edit the pre-populated post content (optional).
  6. Confirm sharing.

Postconditions:

  - The workout summary is successfully posted on the selected social media platform.

Expected Result: The app should seamlessly share the workout summary to the chosen social media platform with the user's customized content.

Severity: Major

Type of Testing: Functional, Integration

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTS-SS-002

Test Case Title: Verify that the app supports sharing on multiple social media platforms.

Test Case Description: This test will ensure the application allows users to share their fitness achievements on various social media platforms by checking the integration and functionality of sharing options.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- User is logged in.
- User is on the screen with sharing options.

Test Data: No test data needed.

Test Steps:

1. Initiate the sharing process for a workout summary or achievement.
2. Observe the list of available social media platforms for sharing.
3. Verify the presence of popular platforms (e.g., Facebook, Instagram, Twitter).
4. Attempt sharing on different platforms.

Postconditions: None

Expected Result: The app should display a variety of popular social media platforms as sharing options, and each option should function correctly.

Severity: Major

Type of Testing: Integration, Usability

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTS-SS-003

Test Case Title: Verify that users can customize the content and format of shared posts.

Test Case Description: This test case validates that users have the flexibility to personalize their shared posts, including adding captions, emojis, and modifying the pre-populated workout details before sharing on social media.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- User is logged in.
- User has initiated the sharing process for a workout summary.

Test Data: Sample customized captions, emojis.

Test Steps:

1. Initiate sharing a workout summary and proceed to the editing section.
2. Attempt to add a custom caption to the post.
3. Check the functionality of adding emojis or stickers.
4. Modify the pre-filled workout details (e.g., distance, duration).
5. Share the customized post.

Postconditions:

- The shared post on the social media platform should reflect the user's customizations.

Expected Result: The app should allow users to edit the pre-populated post content, including adding captions, emojis, and modifying workout details, before sharing.

Severity: Minor

Type of Testing: Functional, Usability

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTS-SS-004

Test Case Title: Verify that sharing features are integrated seamlessly with social media.

Test Case Description: This test aims to ensure the app smoothly integrates with social media platforms during the sharing process. It verifies that users are appropriately redirected to their social media accounts for authentication and posting.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- User is logged in.
- User has not granted the app social media permissions.

Test Data: No test data needed.

Test Steps:

1. Attempt to share a workout summary on a social media platform.
2. Observe if the app redirects to the chosen social media platform for authentication.
3. Authenticate with social media credentials (if prompted).
4. Confirm if the app returns to the correct screen after successful sharing.

Postconditions:

- App permissions are granted.

Expected Result: The app should redirect the user to the chosen social media platform for authentication (if necessary) and return to the app after successful sharing.

Severity: Critical

Type of Testing: Integration

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTS-SS-005

Test Case Title: Verify that users can track the engagement of their shared posts.

Test Case Description: This test case checks if users can view and track the engagement metrics of their shared posts, such as likes, comments, and shares, directly within the app.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- User is logged in.
- User has previously shared posts on social media.

Test Data: No test data needed.

Test Steps:

1. Navigate to a section within the app that displays shared posts (e.g., profile, activity feed).
2. Verify if the shared posts display engagement metrics like likes, comments, and shares.
3. Interact with a shared post by liking or commenting.
4. Observe if the engagement metrics update accordingly within the app.

Postconditions: None

Expected Result: The app should display accurate engagement metrics (likes, comments, shares) for each shared post and reflect any changes in real-time.

Severity: Minor

Type of Testing: Functional, Integration

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 Cases for Fitness Tracker App - Rest and Recovery Tracking:

Test Case 1:

Test Case ID: FTRA-001

Test Case Title: Verify that users can log rest days and recovery activities.

Test Case Description: This test case verifies that users can successfully log both full rest days and specific recovery activities, including the activity type and duration.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

   - User is logged in.
   - User has enabled activity tracking.

Test Data:

   - Rest Day entry
   - Recovery activity: Yoga, Duration: 30 minutes

Test Steps:

   1. Navigate to the "Log Activity" or "Add Entry" section.
   2. Select the option to log a "Rest Day".
   3. Save the entry.
   4. Navigate back to the "Log Activity" section.
   5. Select "Add Recovery Activity".
   6. Choose "Yoga" as the activity type and set the duration to 30 minutes.
   7. Save the entry.

Postconditions:

   - Rest Day and Recovery activity are visible in the user's activity log.

Expected Result: The app successfully records both the full rest day and the specific recovery activity with accurate details.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTRA-002

Test Case Title: Verify that the app tracks metrics such as resting heart rate and sleep quality.

Test Case Description: This test case ensures the app accurately collects, displays, and records resting heart rate and sleep quality data, either through manual input or integration with wearable devices.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has enabled heart rate and sleep tracking (if applicable, through integrations).

Test Data:

- Resting Heart Rate: 65 bpm
- Sleep Quality: 8 hours, "Good" rating

Test Steps:

1. If manual input: Navigate to the "Heart Rate" or "Sleep" section and input the data.
2. If integrated: Ensure the app syncs data from the wearable device.
3. View the recorded resting heart rate and sleep quality data within the app.

Postconditions:

- Data is stored and displayed accurately in the user profile.

Expected Result: The app accurately displays and stores the resting heart rate and sleep quality data, reflecting the input or synced data.

Severity: Major

Type of Testing: Integration Testing (if using wearables), Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTRA-003

Test Case Title: Verify that users receive recommendations for recovery based on their activities.

Test Case Description: This test case checks if the app provides personalized recovery recommendations (like suggesting a rest day, light activity, or specific recovery techniques) based on the user's logged activity intensity and frequency.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has logged various workouts and activities over a period of time (e.g., at least one week).

Test Data: No test data needed

Test Steps:

1. Navigate to the "Recommendations" or "Recovery" section.
2. Observe the recommendations provided by the app.
3. Compare the recommendations to the user's recent activity history.

Postconditions: N/A

Expected Result: The app generates relevant and personalized recommendations for recovery activities or rest, taking into account the user's logged activity data and aiming to prevent overtraining.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTRA-004

Test Case Title: Verify that the app provides alerts for insufficient recovery time.

Test Case Description: This test case ensures the app generates timely alerts or notifications if a user attempts to log a high-intensity workout without sufficient rest or after several consecutive high-intensity training days.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has logged a series of high-intensity workouts on consecutive days.

Test Data: No test data needed

Test Steps:

1. Attempt to log another high-intensity workout.
2. Observe if the app generates an alert or notification.

Postconditions: N/A

Expected Result: The app successfully displays an alert or notification, informing the user of insufficient recovery time and potential risks of overtraining. The app may also suggest alternative activities or a rest day.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Negative

Test Case 5:

Test Case ID: FTRA-005

Test Case Title: Verify that users can view trends and history of their recovery data.

Test Case Description: This test case confirms that users can access and view historical data related to their rest days, recovery activities, resting heart rate, and sleep quality over different time periods (e.g., weekly, monthly).

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has logged recovery data for a period of time.

Test Data: No test data needed

Test Steps:

1. Navigate to the "History" or "Progress" section of the app.
2. Select the "Recovery" data view (if applicable).
3. View the historical data, ensuring different time period selections are available.

Postconditions: N/A

Expected Result: The app displays the user's historical recovery data accurately and allows for visualization of trends over different time frames.

Severity: Minor

Type of Testing: Functional Testing

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:

## Fitness Tracker App - Audio Coaching Test Cases

Test Case 1:

Test Case ID: FTA-AC-001

Test Case Title: Verify that users can access a variety of audio-guided workouts.

Test Case Description: This test case ensures users can browse and select from a diverse library of audio-guided workouts within the app.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in.
- User has enabled audio coaching in settings.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Workouts' section.

2. Select 'Audio Coaching'.

3. Verify that a list of various workout types is displayed (e.g., Yoga, HIIT, Strength Training).

4. Select different workout types and confirm the availability of multiple audio-guided options within each category.

Postconditions: None

Expected Result: The app should display a diverse list of audio-guided workouts categorized by workout type.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTA-AC-002

Test Case Title: Verify that audio coaching includes motivational messages and coaching cues.

Test Case Description: This test case verifies the presence and effectiveness of motivational messages and coaching cues during an audio-guided workout.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

* User is logged in.

* User has started an audio-guided workout.

Test Data: No test data needed.

Test Steps:

1. Begin an audio-guided workout.

2. Listen for and document the presence of motivational messages (e.g., "Keep going!", "You got this!").

3. Identify and document the presence of specific coaching cues related to the workout (e.g., form correction, breathing instructions).

Postconditions: None

Expected Result: The audio coaching should provide clear and motivating messages throughout the workout, along with relevant coaching cues to guide the user.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTA-AC-003

Test Case Title: Verify that users can choose coaching based on workout type and intensity.

Test Case Description: This test case confirms users can filter and select audio coaching options tailored to their chosen workout type and preferred intensity level.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

* User is logged in.

* User is browsing the audio-guided workout library.

Test Data: No test data needed.

Test Steps:

1. Select a specific workout type (e.g., Running).

2. Verify if filtering options for intensity level are available (e.g., Beginner, Intermediate, Advanced).

3. Apply different intensity filters and confirm that the displayed audio coaching options change accordingly.

Postconditions: None

Expected Result: The app should offer filtering options for intensity levels within each workout type, displaying relevant coaching choices.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTA-AC-004

Test Case Title: Verify that audio coaching integrates seamlessly with other app features.

Test Case Description: This test case ensures smooth integration of audio coaching with other app functionalities, such as workout tracking and music playback.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

* User is logged in.

* User has started an audio-guided workout.

Test Data: No test data needed.

Test Steps:

1. Start an audio-guided workout.

2. Minimize the app or switch to another app while the workout is in progress.

3. Verify that audio coaching continues seamlessly in the background.

4. Open the music player within the app and start playing music.

5. Confirm that audio coaching volume adjusts automatically for clear music playback without interrupting the workout guidance.

Postconditions: None

Expected Result: Audio coaching should function seamlessly alongside other app features, including background operation and audio adjustments for concurrent music playback.

Severity: Critical

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTA-AC-005

Test Case Title: Verify that users can control audio settings and preferences.

Test Case Description: This test case ensures users have control over audio settings related to coaching volume, voice selection, and other preferences.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

* User is logged in.

Test Data: No test data needed.

Test Steps:

1. Navigate to the app settings.
2. Locate and access audio settings for coaching.
3. Verify options to adjust coaching volume independently from the device volume.
4. Confirm the availability of different coach voice options (if applicable).
5. Modify various audio settings and confirm that changes are applied correctly.

Postconditions: None

Expected Result: The app should provide users with control over audio settings, allowing them to customize their audio coaching experience.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 94

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Health Apps

Feature description -

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

Userstory title -

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

Acceptance criteria -

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

Generated Test Cases:

## Test Cases for Fitness Tracker App - Integration with Health Apps:

Test Case 1:

Test Case ID: FTA-HA-01

Test Case Title: Verify that the app integrates with popular health and wellness apps.

Test Case Description: This test case verifies the successful integration of the Fitness Tracker app with a selection of popular health and wellness apps (e.g., MyFitnessPal, Headspace, Calm). It ensures that users can connect their accounts and authorize data sharing between the apps.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

   - The Fitness Tracker app is installed and launched.
   - The user has an active account with the chosen health and wellness apps.
   - Necessary permissions for data access are granted.

Test Data: List of popular health and wellness apps for integration.

Test Steps:

   1. Navigate to the "Settings" or "Integrations" section of the Fitness Tracker app.
   2. Select the desired health app from the list of available integrations.
   3. Follow the on-screen instructions to log in to the health app account and authorize data sharing.
   4. Verify that the connection is successful and that the Fitness Tracker app recognizes the integrated app.

Postconditions:

   - The Fitness Tracker app is connected to the selected health app.
   - Data sharing between the apps is enabled.

Expected Result: The Fitness Tracker app successfully integrates with the chosen health and wellness app, allowing users to connect their accounts and enable data synchronization.

Severity: Blocker

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTA-HA-02

Test Case Title: Verify that users can sync data from connected health apps.

Test Case Description: This test case verifies the app's ability to synchronize data from connected health apps. It ensures that relevant health metrics (e.g., steps, calories, sleep data) are fetched from the integrated apps and displayed correctly within the Fitness Tracker app.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- The Fitness Tracker app is integrated with at least one health app.
- Data synchronization is enabled for the connected app.

Test Data: No test data needed.

Test Steps:

1. Initiate a manual data synchronization with the connected health app.
2. Observe the data synchronization process within the Fitness Tracker app.
3. Verify that relevant health metrics from the connected app are fetched and displayed correctly.
4. Check for data consistency between the connected app and the Fitness Tracker app.

Postconditions:

- Data from the connected health app is synced with the Fitness Tracker app.

Expected Result: The Fitness Tracker app successfully retrieves and displays data from the connected health app, ensuring data accuracy and consistency.

Severity: Critical

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTA-HA-03

Test Case Title: Verify that the app provides a consolidated view of health metrics.

Test Case Description: This test case ensures that the Fitness Tracker app presents a clear and organized consolidated view of the user's health metrics from both the Fitness Tracker app itself and any connected health apps.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- The user has synced data from at least one health app.

Test Data: No test data needed.

Test Steps:

1. Navigate to the main dashboard or health summary section of the app.
2. Verify that the displayed data includes metrics from both the Fitness Tracker app and connected apps.
3. Check that the data is presented in a clear, understandable, and well-organized manner.
4. Ensure that users can easily differentiate between data sources (e.g., Fitness Tracker app vs. connected app).

Postconditions: None

Expected Result: The Fitness Tracker app displays a comprehensive view of the user's health, integrating data from connected apps in a user-friendly and informative way.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTA-HA-04

Test Case Title: Verify that users receive insights based on combined data from multiple apps.

Test Case Description: This test case verifies that the Fitness Tracker app leverages the combined data from connected apps to generate personalized insights and recommendations. This could include insights related to activity levels, sleep patterns, nutrition, and mental well-being.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- The Fitness Tracker app has collected sufficient data from both the app itself and connected apps.
- The app's algorithms are configured to generate insights based on combined data.

Test Data: No test data needed.

Test Steps:

1. Navigate to the insights or recommendations section of the app.
2. Observe if the app generates personalized insights that consider data from multiple sources.
3. Verify that the insights are relevant, accurate, and potentially actionable for the user.
4. Check if the app provides explanations or context for the generated insights.

Postconditions: None

Expected Result: The app provides valuable and actionable insights based on the analysis of combined data from the Fitness Tracker app and connected apps, contributing to a more holistic understanding of the user's well-being.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTA-HA-05

Test Case Title: Verify that integration settings are easy to manage and configure.

Test Case Description: This test case focuses on the usability of the integration settings. It ensures that users can easily manage connected apps, control data sharing preferences, and disconnect apps as needed.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- The user is logged into the Fitness Tracker app.

Test Data: No test data needed.

Test Steps:

1. Navigate to the settings or integrations section where users can manage connected apps.
2. Attempt to connect a new health app, modify data sharing settings for an existing connection, and disconnect an app.
3. Observe the user interface and navigation for managing integrations.
4. Verify that the process is intuitive, user-friendly, and provides clear instructions.
5. Confirm that changes to integration settings are saved and applied correctly.

Postconditions:

- Integration settings are updated according to user preferences.

Expected Result: Users should be able to easily connect, manage, and disconnect health apps within the Fitness Tracker app, controlling their data sharing preferences with ease and clarity.

Severity: Minor

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 Cases for Fitness Tracker App - Body Composition Tracking:

Test Case 1:

Test Case ID: FCT-BCT-001

Test Case Title: Verify that users can log body composition metrics.

Test Case Description: This test case aims to ensure users can successfully input and save their body composition metrics, including weight, body fat percentage, and muscle mass.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has navigated to the Body Composition section.

Test Data:

- Weight: 75 kg
- Body Fat Percentage: 20%
- Muscle Mass: 60 kg

Test Steps:

1. Click on the "Add Entry" button.
2. Enter the desired values for weight, body fat percentage, and muscle mass.
3. Click on the "Save" button.

Postconditions:

- Data is saved successfully.
- The user is returned to the main Body Composition view.

Expected Result: The system should successfully save the entered body composition data and display it in the user's history.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FCT-BCT-002

Test Case Title: Verify that the app calculates and displays BMI based on logged data.

Test Case Description: This test case aims to confirm the app accurately calculates and displays the user's BMI based on their inputted weight and height information.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has logged their weight and height in the app.

Test Data: No test data needed (uses pre-existing user data)

Test Steps:

1. Navigate to the Body Composition section.
2. Locate the BMI display area.

Postconditions: N/A

Expected Result: The app should display the user's BMI calculated based on their logged weight and height.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: FCT-BCT-003

Test Case Title: Verify that users can view trends and history of body composition changes.

Test Case Description: This test case verifies that users can access and view historical data of their logged body composition metrics, including the ability to visualize trends over time.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has logged body composition data for at least two different dates.

Test Data: No test data needed (uses pre-existing user data)

Test Steps:

1. Navigate to the Body Composition section.
2. Select the "History" or "Trends" view.

Postconditions: N/A

Expected Result: The app should display a chronological log of the user's body composition data, potentially with graphical representations like charts or graphs.

Severity: Minor

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: FCT-BCT-004

Test Case Title: Verify that the app provides insights and recommendations based on body composition data.

Test Case Description: This test case aims to confirm the app generates personalized insights and recommendations based on the user's logged body composition data and potentially other relevant information (like activity levels or goals).

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has logged sufficient body composition data to trigger insights.

Test Data: No test data needed (uses pre-existing user data)

Test Steps:

1. Navigate to the Body Composition section.
2. Locate the "Insights" or "Recommendations" section.

Postconditions: N/A

Expected Result: The app should provide relevant and personalized insights or recommendations based on the user's data, such as suggesting adjustments to training or calorie intake.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FCT-BCT-005

Test Case Title: Verify that users can set goals for body composition metrics.

Test Case Description: This test case ensures users can set and manage personalized goals for various body composition metrics within the app.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has navigated to the Body Composition section.

Test Data:

- Target Weight: 70 kg
- Target Body Fat Percentage: 18%

Test Steps:

1. Locate and select the "Set Goals" option.
2. Choose the desired body composition metric (e.g., weight, body fat percentage).
3. Input the desired target value for the chosen metric.
4. Save the goal.

Postconditions:

- The set goal is saved within the app.
- The user can view and track progress toward their set goal.

Expected Result: The app successfully saves the user's goal and potentially provides visualizations or tools for tracking progress toward that goal.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 96

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Analysis

Feature description -

The app offers detailed workout analysis tools, including performance metrics, trends, and insights into exercise intensity, duration, and calorie burn. Workout analysis helps users assess the effectiveness of their workouts, identify areas for improvement, and optimize training strategies for better results.

Userstory title -

As a user, I want to analyze my workouts so that I can assess my performance and optimize my training strategies.

Acceptance criteria -

['Users can view detailed performance metrics for each workout.', 'The app provides trends and insights into workout intensity and duration.', 'Users can track calorie burn and other health metrics.', 'The app offers recommendations based on workout analysis.', 'Users can compare current workouts with past performance.']

Generated Test Cases:

## Fitness Tracker App - Workout Analysis Test Cases

Test Case 1

Test Case ID: FTA-WA-001

Test Case Title: Verify that users can view detailed performance metrics for each workout.

Test Case Description: This test case aims to confirm that the app accurately displays relevant performance metrics for completed workouts, including but not limited to duration, distance, calories burned, average heart rate, and steps taken.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

*   User is logged in.

*   User has completed at least one workout.

*   Workout data is synced with the app.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Workouts' section.

2. Select a completed workout from the list.

3. Verify that the workout summary screen displays the following metrics:

    *   Workout date and time

    *   Workout type

    *   Duration

    *   Distance

    *   Calories burned

    *   Average heart rate

    *   Steps taken (if applicable)

Postconditions: None.

Expected Result: The app displays all relevant performance metrics for the selected workout accurately.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2

Test Case ID: FTA-WA-002

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration.

Test Case Description: This test case verifies that the app generates visualizations or summaries showing trends in workout intensity and duration over a defined period, allowing users to identify patterns and assess their activity levels.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

*   User is logged in.
*   User has logged multiple workouts over a period of time.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Workout Analysis' section.
2. Select the desired timeframe for analysis (e.g., past week, month, or custom range).
3. Verify that the app displays trends related to workout intensity and duration. This could be in the form of:
    *   Graphs showcasing workout duration over time
    *   Charts displaying average heart rate zones during workouts
    *   Insights highlighting changes in workout intensity or frequency

Postconditions: None.

Expected Result: The app provides clear and insightful visualizations or summaries of workout intensity and duration trends over the selected timeframe.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3

Test Case ID: FTA-WA-003

Test Case Title: Verify that users can track calorie burn and other health metrics.

Test Case Description: This test case ensures that the app accurately tracks and displays calorie expenditure during workouts, along with other relevant health metrics, allowing users to monitor their progress and make informed decisions about their fitness goals.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

*   User is logged in.
*   User has completed a workout with calorie tracking enabled.

Test Data: No test data needed.

Test Steps:

1. Complete a workout while tracking calories through the app or a connected device.
2. Navigate to the 'Workout Summary' for the completed session.
3. Verify that the app displays the following:
    *   Total calories burned during the workout
    *   Active calories vs. total calories burned (if applicable)
    *   Breakdown of calorie burn by workout phase or interval (if applicable)
4. If applicable, verify that the app displays other relevant health metrics tracked during the workout, such as average heart rate, steps taken, or distance covered.

Postconditions: None.

Expected Result: The app accurately tracks and displays calorie burn and other relevant health metrics for completed workouts.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4

Test Case ID: FTA-WA-004

Test Case Title: Verify that the app offers recommendations based on workout analysis.

Test Case Description: This test case assesses the app's ability to provide personalized recommendations based on analyzed workout data. The recommendations should aim to help users improve their training routines, achieve fitness goals, or address potential issues identified through data analysis.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

*   User is logged in.
*   User has logged a sufficient amount of workout data for the app to generate personalized recommendations.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Workout Analysis' section or a dedicated 'Recommendations' area within the app.
2. Verify that the app provides personalized recommendations based on workout history, trends, and user-defined goals (if applicable).
3. Recommendations should relate to aspects like:
    *   Adjustments to workout frequency, intensity, or duration
    *   Suggestions for trying new workout types or routines
    *   Reminders for rest and recovery based on activity levels
    *   Tips for improving form or technique based on tracked data (if available)

Postconditions: None.

Expected Result: The app provides personalized recommendations tailored to the user's workout history and potentially their fitness goals.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5

Test Case ID: FTA-WA-005

Test Case Title: Verify that users can compare current workouts with past performance.

Test Case Description: This test case confirms that users can easily compare their performance on current workouts with past workout data, allowing them to track progress and identify areas of improvement or decline.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

*   User is logged in.
*   User has completed multiple workouts of the same type.

Test Data: No test data needed.

Test Steps:

1. Navigate to the 'Workouts' section.
2. Select a completed workout to view its summary.
3. Verify that the app provides an option to compare the selected workout with:
    *   The user's previous workout of the same type.
    *   The user's average performance for that workout type over a defined period.
4. Verify that the comparison view clearly displays relevant metrics for both the current workout and the compared data, highlighting areas of improvement or decline.

Postconditions: None.

Expected Result: The app enables users to easily compare their performance on current workouts with past data for the same workout type, visually highlighting progress and areas for potential improvement.

Severity: Minor

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 1:

Test Case ID: FT-WI-001

Test Case Title: Verify that the app provides current weather conditions and forecasts.

Test Case Description: This test case verifies that the app displays accurate and up-to-date weather information, including current conditions and forecasts for the user's location. This ensures users have the necessary information to plan their outdoor activities effectively.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- The app is installed and launched.
- The user has granted location permissions to the app.
- An active internet connection is available.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the weather section.
2. Verify that the current weather conditions are displayed, including temperature, humidity, wind speed and direction, and an appropriate weather icon.
3. Verify that the hourly and daily weather forecasts are displayed, including temperature, precipitation probability, and relevant weather icons.

Postconditions: None

Expected Result: The app should accurately display the current weather conditions and forecasts for the user's location, matching data from reliable weather sources.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 2:

Test Case ID: FT-WI-002

Test Case Title: Verify that users can view weather details for different locations.

Test Case Description: This test verifies that users can search for and view weather information for locations other than their current location, allowing them to plan for trips or check on conditions in other areas.

Test Suite: Weather Integration

Test Priority: Medium

Preconditions:

  - The app is installed and launched.
  - An active internet connection is available.

Test Data: A valid city name (e.g., "London", "Tokyo")

Test Steps:

  1. Open the app and navigate to the weather section.
  2. Locate the search bar or location selection option.
  3. Enter the desired location and initiate the search.
  4. Verify that the app displays the current weather conditions and forecasts for the searched location.

Postconditions: None

Expected Result: The app should accurately display the weather information for the searched location, including current conditions and forecasts.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 3:

Test Case ID: FT-WI-003

Test Case Title: Verify that weather data is updated in real-time.

Test Case Description: This test ensures that the app provides users with the most up-to-date weather information by verifying that the data is updated in real-time or at regular intervals.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- The app is installed and launched.
- An active internet connection is available.

Test Data: No test data needed

Test Steps:

1. Open the app and navigate to the weather section.
2. Note the current time and the last updated timestamp for the weather data.
3. Wait for a predetermined interval (e.g., 5 minutes).
4. Refresh the weather data within the app.
5. Verify that the last updated timestamp has been updated, reflecting the data refresh.

Postconditions: None

Expected Result: The app should display updated weather information, and the last updated timestamp should reflect the data refresh, ensuring users have access to the most recent weather data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 4:

Test Case ID: FT-WI-004

Test Case Title: Verify that the app offers recommendations based on weather conditions.

Test Case Description: This test case checks if the app provides relevant workout recommendations based on current and forecasted weather conditions, promoting user safety and enhancing the workout experience.

Test Suite: Weather Integration

Test Priority: Medium

Preconditions:

- The app is installed and launched.
- An active internet connection is available.
- Location permissions are granted.

Test Data: No test data needed

Test Steps:

1. Navigate to the weather section or the workout recommendation section.
2. Observe the provided workout recommendations.
3. Verify that the recommendations are appropriate for the current or forecasted weather conditions.
4. For example, the app should recommend indoor workouts during extreme heat or thunderstorms and outdoor activities during pleasant weather.

Postconditions: None

Expected Result: The app should provide workout recommendations that align with the current and forecasted weather conditions, ensuring user safety and comfort.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

## Test Case 5:

Test Case ID: FT-WI-005

Test Case Title: Verify that users can receive weather alerts for extreme conditions.

Test Case Description: This test case verifies that the app can issue timely and accurate weather alerts to users in case of extreme weather conditions, such as heavy rain, thunderstorms, or heat waves. This ensures user safety and allows for appropriate adjustments to workout plans.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- The app is installed and launched.
- An active internet connection is available.
- Users have enabled notifications for weather alerts within the app.

Test Data: Simulated extreme weather conditions (e.g., thunderstorm warning for user's location)

Test Steps:

1. Simulate an extreme weather alert for the user's location.
2. Observe the app's behavior and notification system.
3. Verify that the app issues a timely and clear notification to the user, alerting them to the extreme weather condition.
4. Verify that the notification includes relevant information, such as the type of alert, the affected area, and the expected duration.

Postconditions: Dismiss the weather alert notification.

Expected Result: The app should successfully deliver a clear and timely weather alert to the user, providing them with sufficient information to take necessary precautions and adjust their outdoor workout plans accordingly.

Severity: Critical

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Data - 98

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Mood Tracking

Feature description -

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

Userstory title -

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

Acceptance criteria -

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and energy levels.']

Generated Test Cases:

## Fitness Tracker App - Mood Tracking Test Cases

Test Case 1:

Test Case ID: FTM-001

Test Case Title: Verify that users can log their mood, energy levels, and motivation.

Test Case Description: This test case ensures that users can successfully input their mood, energy levels, and motivation before and after workouts. It verifies the functionality of the input fields and the saving mechanism for this data.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

- User is logged in.

- User has completed a workout.

Test Data:

- Mood: "Happy"

- Energy Level: "High"

- Motivation: "Motivated"

Test Steps:

1. Navigate to the "Workout Summary" screen.

2. Locate the "Mood & Energy" section.

3. Select the appropriate mood from the list (e.g., happy, stressed, calm).

4. Choose the corresponding energy level (e.g., high, moderate, low).

5. Indicate motivation level (e.g., motivated, neutral, unmotivated).

6. Save the mood and energy log.

Postconditions:

- Mood, energy levels, and motivation are saved and associated with the workout.

Expected Result: The app should successfully record and display the entered mood, energy level, and motivation data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTM-002

Test Case Title: Verify that the app provides insights into mood and energy trends over time.

Test Case Description: This test case checks if the app effectively analyzes and presents trends in the user's logged mood and energy data. It ensures that users can view historical data and identify patterns in their mental well-being related to their workouts.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

- User is logged in.
- User has logged mood and energy data for multiple workouts over a period of time (e.g., one week).

Test Data: No test data needed.

Test Steps:

1. Go to the "Mood Insights" or "Trends" section of the app.
2. Select a timeframe for analysis (e.g., past week, past month).
3. View the displayed charts, graphs, or summaries of mood and energy trends.

Postconditions: None

Expected Result: The app should display clear and accurate visualizations of mood and energy trends, allowing users to see patterns and changes over time.

Severity: Major

Type of Testing: Functional Testing, Data Analysis

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTM-003

Test Case Title: Verify that users receive recommendations based on their mood and energy data.

Test Case Description: This test case validates that the app provides personalized recommendations to the user based on their logged mood and energy levels. Recommendations could include workout suggestions, mindfulness exercises, or motivational tips.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User is logged in.
- User has logged consistent mood and energy data.

Test Data: No test data needed.

Test Steps:

1. Navigate to the "Recommendations" or "Insights" section.
2. Observe the provided recommendations.
3. Verify that the recommendations are relevant to the user?s logged mood and energy patterns.

Postconditions: None

Expected Result: The app should generate personalized recommendations based on the user's mood and energy data, promoting well-being and informed exercise choices.

Severity: Minor

Type of Testing: Functional Testing, Recommendation Engine

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTM-004

Test Case Title: Verify that the app supports journaling for personal reflections on mood and workouts.

Test Case Description: This test case confirms the functionality of the journaling feature, ensuring users can add personal notes related to their mood, energy levels, and workout experiences.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User is logged in.

- User has completed a workout.

Test Data: Free-text journaling entry related to mood and workout.

Test Steps:

1. Navigate to the "Workout Summary" or a dedicated "Journal" section.

2. Locate the journaling input field.

3. Enter a personal reflection on mood, energy levels, or the workout itself.

4. Save the journal entry.

Postconditions:

- The journal entry is saved and linked to the workout.

Expected Result: The app should allow users to easily add and save journal entries, providing a space for personal reflections and enhancing self-awareness.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTM-005

Test Case Title: Verify that users can set goals for improving mood and energy levels.

Test Case Description: This test case ensures users can set specific, measurable, achievable, relevant, and time-bound (SMART) goals related to improving their mood and energy levels through exercise.

Test Suite: Mood Tracking, Goal Setting

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: Goal: "Increase average energy level by 1 point within two weeks."

Test Steps:

1. Go to the "Goals" section of the app.
2. Select the option to "Create a New Goal."
3. Choose "Mood" or "Energy Level" as the goal category.
4. Define the specific goal (e.g., increase energy level, improve mood consistency).
5. Set a target timeframe for achieving the goal.
6. Save the new goal.

Postconditions:

- The mood/energy goal is saved and displayed in the user's active goals.

Expected Result: The app should enable users to set clear and trackable goals for improving mood and energy, promoting motivation and a sense of progress.

Severity: Minor

Type of Testing: Functional Testing, Goal Setting

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 Cases for Recipe Sharing Platform - Recipe Creation Feature

Test Case 1:

Test Case ID: RSP-RC-001

Test Case Title: Verify that users can input and save ingredients and cooking instructions.

Test Case Description: This test case ensures that the recipe creation form allows users to successfully enter and save a list of ingredients and detailed cooking instructions.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in.

- User has navigated to the recipe creation page.

Test Data:

- Recipe title: "Classic Chocolate Chip Cookies"

- Ingredients: Flour, sugar, butter, eggs, chocolate chips (with specific quantities)

- Cooking instructions: Detailed steps for mixing, baking, and cooling the cookies.

Test Steps:

1. Enter the recipe title in the designated field.

2. Input the list of ingredients, including quantities, in the ingredients section.

3. Enter the detailed cooking instructions in the instructions section.

4. Click the "Save Recipe" button.

Postconditions:

- The recipe is saved to the user's profile.

- The user is redirected to the recipe view page.

Expected Result: The system successfully saves the recipe with the provided ingredients and instructions.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: RSP-RC-002

Test Case Title: Verify that users can upload photos for their recipes.

Test Case Description: This test case checks the functionality of uploading photos during recipe creation, ensuring users can enhance their recipes with visuals.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in.

- User is creating a new recipe.

Test Data:

- Valid image file (JPEG or PNG format)

Test Steps:

1. On the recipe creation form, locate the image upload section.

2. Click the "Upload Image" or "Choose File" button.

3. Select the desired image file from the local device.

4. Click "Open" or "Upload" to initiate the upload.

5. Verify the image preview is displayed correctly.

Postconditions:

- The image is uploaded and associated with the recipe.

Expected Result: The chosen image is successfully uploaded and displayed within the recipe preview without errors.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: RSP-RC-003

Test Case Title: Verify that recipes are saved in the user's profile.

Test Case Description: This test verifies that after creating a recipe, it is correctly associated with the user's profile and accessible within their "My Recipes" section.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in.

- User has created at least one recipe.

Test Data: No test data needed.

Test Steps:

1. Navigate to the user profile page.

2. Locate the "My Recipes" section.

3. Verify that the created recipe is listed within this section.

Postconditions: None

Expected Result: The created recipe is displayed within the user's profile under "My Recipes."

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: RSP-RC-004

Test Case Title: Verify that recipes are accessible to other users on the platform.

Test Case Description: This test case confirms that once a recipe is created, it becomes visible and accessible to other users browsing the platform's recipe database.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- A recipe has been created and saved by a user.

- The recipe's visibility is set to "Public" (if applicable).

Test Data: No test data needed.

Test Steps:

1. Log in as a different user account or browse the platform as a guest.

2. Utilize the recipe search or browse functionality to locate the specific recipe.

Postconditions: None

Expected Result: The created recipe is successfully found and displayed in the search results or browsing page.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: RSP-RC-005

Test Case Title: Verify that users receive a confirmation upon successful recipe submission.

Test Case Description: This test case checks that a confirmation message is displayed to the user after they successfully create and save a new recipe.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in.
- User has completed the recipe creation process and clicked "Save Recipe."

Test Data: No test data needed.

Test Steps:

1. Observe the system response after clicking "Save Recipe."

Postconditions: None

Expected Result: A clear confirmation message, such as "Recipe successfully saved!" or similar, is displayed to the user.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 100

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Discovery

Feature description -

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

Userstory title -

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance criteria -

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

Generated Test Cases:

## Test Cases for Recipe Sharing Platform - Recipe Discovery Feature

Test Case 1:

Test Case ID: RS-RD-001

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

Test Case Description: This test case ensures that the search functionality correctly returns recipes matching the entered keywords. It verifies various keyword combinations and their impact on search results.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User is logged in.
- The recipe database contains recipes with various keywords.

Test Data:

- Keywords: "Pasta", "Chicken", "Vegetarian", "Chocolate Cake"

Test Steps:

1. Navigate to the recipe search bar.
2. Enter a single keyword (e.g., "Pasta").
3. Observe the search results.
4. Repeat steps 2-3 for different keyword combinations (e.g., "Chicken Pasta", "Vegetarian Chocolate Cake").

Postconditions: None

Expected Result: The system should display a list of recipes containing the entered keywords in their title, description, or ingredients. The results should update dynamically with each keyword entered.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: RS-RD-002

Test Case Title: Verify that users can search for recipes using ingredients.

Test Case Description: This test case validates the ingredient-based search functionality, ensuring users can find recipes based on specific ingredients they have or prefer.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User is logged in.
- Recipes in the database have a detailed list of ingredients.

Test Data:

- Ingredients: "Chicken", "Broccoli", "Garlic", "Parmesan Cheese"

Test Steps:

1. Access the recipe search bar.
2. Enter a single ingredient (e.g., "Chicken").
3. Observe the search results.
4. Combine multiple ingredients (e.g., "Chicken, Broccoli, Garlic") and repeat step 3.

Postconditions: None

Expected Result: The system displays a list of recipes using the entered ingredients. Results should dynamically update as ingredients are added or removed.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: RS-RD-003

Test Case Title: Verify that users can filter search results by cuisine type, dietary preferences, and other criteria.

Test Case Description: This test case confirms the functionality of the search filter options, allowing users to refine their search based on various criteria.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- A search query (keywords or ingredients) has been made.
- Search filters for cuisine, dietary preferences, and other criteria are available.

Test Data:

- Cuisine type: Italian, Mexican, Indian
- Dietary preference: Vegetarian, Vegan, Gluten-free

Test Steps:

1. Perform a recipe search.
2. Apply a filter for cuisine type (e.g., Italian).
3. Observe the change in search results.
4. Apply additional filters (e.g., Vegetarian) while the cuisine filter is active.
5. Clear all filters and observe the return to the original search results.

Postconditions: None

Expected Result: Search results accurately reflect the applied filters. The system should allow combining multiple filters and revert to the unfiltered results when filters are cleared.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: RS-RD-004

Test Case Title: Verify that recipes in the search results are accessible and detailed.

Test Case Description: This test case ensures that users can access the full recipe details from the search results page, including ingredients, instructions, and additional information.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- Search results are displayed.

Test Data: No test data needed.

Test Steps:

1. Perform a recipe search.
2. Click on a recipe from the search results.

Postconditions: None

Expected Result: The system should navigate to the detailed recipe page, displaying the recipe title, author, ingredients list, cooking instructions, nutritional information (if available), user ratings/reviews, and social sharing options.

Severity: Major

Type of Testing: Functional Testing, Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: RS-RD-005

Test Case Title: Verify that users can save search results for future reference.

Test Case Description: This test case confirms the functionality of saving search results, allowing users to bookmark recipes for later use.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

- User is logged in.

- Search results are displayed.

Test Data: No test data needed.

Test Steps:

1. Execute a recipe search.

2. Select the option to save the search results (e.g., click a "Save Search" button).

3. Navigate away from the search results page.

4. Access the user's saved items or search history section.

Postconditions: None

Expected Result: The saved search results should be accessible from the user's profile or a dedicated section for saved searches. The saved search should include the original keywords or filters used.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive
