

Generated Test Cases via gpt-4 LLM Model and Few-shot Prompting Technique

Data - 1

Project name -

Social Media Platform

Project description -

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

Feature name -

User Profiles

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TS01

Test Case Title: Verify that users can add and edit their name, bio, profile picture, location, and interests

Test Case Description: This test case validates the functionality of adding and editing user's name, bio, profile picture, location, and interests on profile

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in
- User is on their profile page

Test Data: User's personal details including name, bio, profile picture, location, and interests

Test Steps:

1. Navigate to the profile edit page
2. Input or edit the name, bio, location, and interests field
3. Upload or change the profile picture
4. Click on the save changes button

Postconditions:

- Changes are saved on user profile

Expected Result: User's details are successfully updated on the profile

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TS02

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

Test Case Description: This test case verifies the ability to upload a profile picture in standard image formats (JPEG, PNG)

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

- User is logged in
- User is on their profile edit page

Test Data: Profile pictures in JPEG and PNG formats

Test Steps:

1. Navigate to profile edit page
2. Click on 'Upload Picture'
3. Select an image on JPEG format
4. Click on 'Upload Picture' again
5. Select an image on PNG format

Postconditions:

- Profile picture is uploaded
- Profile picture displays correctly

Expected Result: Users can successfully upload profile picture in JPEG and PNG formats

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TS03

Test Case Title: Verify that users can set privacy settings for profile fields

Test Case Description: This test verifies that users can change the privacy settings of each profile field

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in
- User is on their profile page

Test Data: Privacy settings

Test Steps:

1. Navigate to profile settings
2. Change privacy settings for each field
3. Save changes

Postconditions:

- Privacy settings are updated

Expected Result: The user-profile field becomes private according to the selected setting

Severity: Major

Type of Testing: Security

Test Case Approach: Positive

Test Case 4:

Test Case ID: TS04

Test Case Title: Verify that the profile is saved successfully

Test Case Description: This test case validates the save function of the profile details

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in
- User is on their profile edit page

Test Data: User's personal details

Test Steps:

1. Edit user profile details
2. Click save

Postconditions:

- User profile details are saved

Expected Result: The newly updated details should be visible on the profile page

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TS05

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

Test Case Description: Ensuring that the user gets a confirmation upon saving their profile to validate that the information has been successfully saved

Test Suite: User Profiles

Test Priority: Medium

Preconditions:

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

Test Data: User's personal details

Test Steps:

1. Edit the user profile details
2. Click Save

Postconditions:

- Confirmation message is received

Expected Result: Users receive a notification or message of successful profile saving

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 2

Project name -

Social Media Platform

Project description -

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

Feature name -

News Feed

Feature description -

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

Userstory title -

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

Acceptance criteria -

['The news feed displays posts from followed accounts.', 'Content is shown in chronological or algorithmic

order.', 'Users can interact with posts (like, comment, share).', 'The feed updates in real-time.', 'Users can filter content based on type or topic.']

Generated Test Cases:

Test Case 1:

Test Case ID: A01

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

Test Case Description: To ensure that the posts from the followed accounts of the user are displayed on the news feed.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- User has followed some accounts

Test Data: Post information from followed accounts.

Test Steps:

1. Login to account
2. Go to news feed
3. Scroll and view the posts

Postconditions:

- Posts from followed accounts are viewed

Expected Result: The system should display the posts from the followed accounts in the news feed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: A02

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

Test Case Description: To ensure that the posts are displayed on the news feed in chronological or as per audience interest order.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- User has some previous posts in the news feed

Test Data: Date and time of posts, interests of the user, related account interaction data.

Test Steps:

1. Login to account
2. Go to news feed
3. Check the order of the displayed posts

Postconditions:

- Order of the posts is determined

Expected Result: The system should display the posts in chronological or as per audience interest order.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: A03

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

Test Case Description: To ensure that users on the social media platform can interact with posts by liking, commenting, and sharing them.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- There are some posts in the news feed on which the user can interact

Test Data: Existing posts data

Test Steps:

1. Go to news feed
2. Select a post
3. Click on like, comment, and share
4. Enter comment if applicable and submit

Postconditions:

- User interaction with the post is achieved

Expected Result: The system should enable the user to interact with the posts.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: A04

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

Test Case Description: To ensure that the social media platform displays real-time updates in news feed.

Test Suite: News Feed

Test Priority: High

Preconditions:

- User is logged in
- User's followed accounts are actively posting content

Test Data: Real-time posts data from followed accounts

Test Steps:

1. Login to account
2. Go to news feed
3. Note down the latest post
4. Wait for new posts to populate
5. Refresh the news feed

Postconditions:

- News feed is updated in real-time

Expected Result: The system should update the news feed with real-time posts.

Severity: Major

Type of Testing: System

Test Case Approach: Positive

Test Case 5:

Test Case ID: A05

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

Test Case Description: To ensure that the users can filter the news feed posts based on their interests or type of content.

Test Suite: News Feed

Test Priority: Medium

Preconditions:

- User is logged in

- Various types of content (photo, video, text, etc.) are in the news feed

Test Data: Content type (photo, video, text, etc.), user preferences for topics

Test Steps:

1. Go to news feed
2. Select filter option
3. Choose a type or topic
4. Apply the filter

Postconditions:

- News feed is filtered according to the type or topic selected

Expected Result: The system should filter the news feed according to the user-selected type or topic.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 3

Project name -

Social Media Platform

Project description -

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

Feature name -

Messaging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: MSG001

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

Test Case Description: This test case is designed to confirm that the messaging system allows users to compose and send text messages to other users.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged into the account
- The receiver's profile exists

Test Data: Text message.

Test Steps:

1. Navigate to the messaging feature.
2. Select the friend to whom you want to send the message.
3. Compose a text message.

4. Click on the send button.

Postconditions:

- Message is sent to the receiver.

Expected Result: The system sends the text message to the chosen receiver and the sent message appears in the chat window.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: MSG002

Test Case Title: Verify real-time messaging feature.

Test Case Description: This test case is designed to validate that messages can be sent and received in real-time.

Test Suite: Messaging

Test Priority: High

Preconditions:

- Both users are logged into their accounts
- The users are connected

Test Data: Text message

Test Steps:

1. Navigate to the messaging feature.
2. Select a friend.
3. Compose a text message.
4. Click on the send button.

Postconditions:

- Receiver receives the message in real-time.

Expected Result: Upon sending the message, the receiver gets the text message instantly.

Severity: Major

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: MSG003

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

Test Case Description: Ensure that the platform allows users to start not only private conversations but also create group chats.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged into the account
- Chosen friends' profiles exist on the platform

Test Data: Friend's names

Test Steps:

1. Navigate to the messaging feature.
2. Click on "Create a new chat".
3. Select multiple friends for a group chat.
4. Click on "Start Conversation".

Postconditions:

- New group chat is created.

Expected Result: The system creates a new group chat with the selected friends appearing in the chat window.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: MSG004

Test Case Title: Verify that messages show read receipts.

Test Case Description: Confirm that the system establishes a visual indicator(std. is to display 'Seen' or 'Delivered') when messages are read by the receiver.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged into the account.
- Receiver is online and has opened the message.

Test Data: Text message

Test Steps:

1. Navigate to the messaging feature.
2. Select a friend.
3. Send a message.
4. Wait for the friend to open the message.

Postconditions:

- Read receipt is displayed.

Expected Result: The system shows 'Seen' under the sent message indicating that the message has been read by the receiver.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: MSG005

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

Test Case Description: Ensure that the platform alerts users when they receive new messages.

Test Suite: Messaging

Test Priority: High

Preconditions:

- User is logged into the account.
- User has received a new message.

Test Data: No test data needed

Test Steps:

1. Receive a new message from a friend.
2. Check for notifications.

Postconditions:

- New message notification is displayed.

Expected Result: The system triggers a notification, displaying the number of new messages unread by the user.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 4

Project name -

Social Media Platform

Project description -

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

Feature name -

Content Sharing

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TC_CSP_01

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

Test Case Description: Check if the social media platform allows users to create, post, and share text, photos, and videos successfully.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in
- User is on the create post page

Test Data: Text post, photo, and video files

Test Steps:

1. Click on 'Create Post' button
2. Add post content (text, photo, or video)
3. Click on 'Post' button

Postconditions:

- Post is uploaded and visible on the user's timeline and followers' feed

Expected Result: The post (text/photo/video) is successfully shared.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_CSP_02

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

Test Case Description: Ensure that the platform allows users to set the privacy of their posts to followers only or public.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the create post page

Test Data: Text post, photo, and video files

Test Steps:

1. Click on 'Create Post' button
2. Add post content (text/photo/video)
3. Choose privacy settings - followers only/public
4. Click on 'Post' button

Postconditions:

- Post is uploaded and visible based on chosen privacy level

Expected Result: User's post is visible only to followers or to everyone based on chosen setting.

Severity: Minor

Type of Testing: Security

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_CSP_03

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

Test Case Description: Ensure that users are able to modify or remove posts after they have been published.

Test Suite: Content Sharing

Test Priority: High

Preconditions:

- User is logged in
- Post is already created

Test Data: Modified text for the post

Test Steps:

1. Go to user's own post
2. Click on 'Edit Post' or 'Delete Post'

3. If editing, make changes and click 'Save Changes'

Postconditions:

- Post is updated or deleted

Expected Result: The user's post is successfully edited or removed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_CSP_04

Test Case Title: Verify that posts support hashtags and tagging

Test Case Description: Validate if the system successfully creates posts that include hashtags and tagging of other users.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the create post page

Test Data: Text post with hashtag and tagged user

Test Steps:

1. Click on 'Create Post' button
2. Add post content with hashtag and tagged user
3. Click on 'Post' button

Postconditions:

- Post is uploaded and visible on the user's timeline and followers' feed
- Tagged user gets notification

Expected Result: The post with hashtags and tagged users is successfully shared. Tagged users receive a notification.

Severity: Medium

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_CSP_05

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

Test Case Description: This test ensures that users can interact with posts via likes, comments, and shares and the original poster gets notifications of the interactions.

Test Suite: Content Sharing

Test Priority: Medium

Preconditions:

- Two users are logged in
- User 1 has published a post

Test Data: No test data needed

Test Steps:

1. User 2 views User 1's post
2. User 2 likes/comments/shares the post
3. User 1 checks notifications

Postconditions:

- Interaction is registered and visible on post
- User 1 received notification

Expected Result: Users can interact with posts through likes, comments, and shares, and the original poster can see these interactions.

Severity: Medium

Type of Testing: Functionality

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 Case 1:

Test Case ID: SMT_Ps001

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

Test Case Description: Ensure the system provides options to set user profile privacy level to public, private, or restricted.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User Account is created and logged in
- Access to the privacy settings page

Test Data: User profile privacy settings

Test Steps:

1. Go to privacy settings
2. Select public, private, or restricted
3. Save changes

Postconditions:

- Profile privacy is updated

Expected Result: The system updates the profile privacy setting as selected by the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMT_Ps002

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

Test Case Description: Ensuring that users can change their privacy settings whenever they want.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User Account is created and logged in
- Access to the privacy settings page

Test Data: Changes in privacy settings

Test Steps:

1. Go to privacy settings
2. Change privacy level
3. Save changes

Postconditions:

- Profile privacy settings are updated

Expected Result: The system should reflect the changes immediately on the user's privacy settings.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMT_Ps003

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

Test Case Description: Ensure that user can set the visibility of each individual post to public, private, or restricted.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in
- User has posts on the profile

Test Data: Post visibility setting

Test Steps:

1. Select a post
2. Go to post settings
3. Change post visibility
4. Save changes

Postconditions:

- Post visibility is updated

Expected Result: The system updates the post visibility as per the user's selection.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMT_Ps004

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

Test Case Description: Ensure that the system applies the user's privacy settings immediately after they are saved.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in
- User has changed privacy settings

Test Data: No test data needed

Test Steps:

1. Change privacy settings
2. Save changes
3. Check profile from another account

Postconditions:

- Updated privacy settings are implemented

Expected Result: The system should implement the updated privacy settings immediately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMT_Ps005

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

Test Case Description: Confirm that the system sends a notification to the user when their privacy settings are changed.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in
- Access to the privacy settings page

Test Data: Privacy settings changes

Test Steps:

1. Go to privacy settings
2. Change the privacy settings
3. Save changes

Postconditions:

- User receives notification regarding privacy settings changes

Expected Result: The system sends a notification of changes to the user right after the privacy settings are updated.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 6

Project name -

Social Media Platform

Project description -

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

Feature name -

Notifications

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: SMP0N1

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

Test Case Description: The test case is designed to ascertain that users are correctly notified whenever their posts receive likes, comments or shares.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has posts with likes, comments, or shares

Test Data: User's post data with likes, shares, comments

Test Steps:

1. Log in to the platform
2. Navigate to the notifications area
3. Look for new notifications about likes, comments, and shares

Postconditions:

- User has viewed the notifications

Expected Result: The system correctly displays new notifications for likes, comments, and shares on user posts.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMP0N2

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

Test Case Description: The test focuses on evaluating whether the system promptly notifies users of new friend requests and mentions in posts.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has new friend requests or mentions

Test Data: Friend request details, User post mentions

Test Steps:

1. Log in to the platform
2. Navigate to the notifications area
3. Look for new notifications about friend requests and mentions

Postconditions:

- User viewed the notifications

Expected Result: System should display new notifications for friend requests and mentions correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMP0N3

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

Test Case Description: This test aims to check whether the system displays notifications in real-time as soon as relevant activity occurs.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- Relevant activity triggering a notification occurs

Test Data: Real time activity data

Test Steps:

1. Log in to the platform
2. Engage in activity that triggers notifications (like, comment, share, send a friend request)
3. Verify that notifications appear instantly

Postconditions:

- User views the notification

Expected Result: The system should display notifications in real-time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMP0N4

Test Case Title: Verify that users can customize notification preferences

Test Case Description: The test centers on verifying that users can modify their notification preferences, including alert types and frequency.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in

Test Data: User's notification preference settings

Test Steps:

1. Log in to the platform
2. Navigate to the user settings section
3. Change the notification settings as desired
4. Save changes

Postconditions:

- User has updated preferred notification settings

Expected Result: The system allows a user to customize their notification preferences and saves changes correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMP0N5

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

Test Case Description: The test is designed to ensure that every notification carries a link directing users to the corresponding content or activity.

Test Suite: Notifications

Test Priority: High

Preconditions:

- User is logged in
- Notifications are present

Test Data: User's notifications details

Test Steps:

1. Log in to the platform
2. Navigate to the notifications area
3. Click on a single notification

Postconditions:

- User is redirected to the relevant activity from the notification

Expected Result: The system should include a link in every notification that redirects the user to the related activity on the platform.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: SM1-Search-01

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

Test Case Description: This test case checks if the system allows users to search for content using specific keywords or hashtags.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in
- There is content available matching the keywords or hashtags used for searching

Test Data: Keywords or Hashtags

Test Steps:

1. Log in to the platform
2. Go to the search bar

3. Enter a keyword or a hashtag

4. Click on the search button

Postconditions:

- The system initiates a search using the entered keyword or hashtag

Expected Result: The system retrieves and displays relevant content matching the keyword or hashtag.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: SM1-Search-02

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

Test Case Description: This test case validates if the system includes profiles, posts, and topics in its search results.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in
- Keywords used for search have matching profiles, posts, and topics

Test Data: Keywords

Test Steps:

1. Log in to the platform
2. Go to the search bar
3. Type a keyword
4. Execute the search

Postconditions:

- The search result is displayed

Expected Result: The system displays the search results including profiles, posts, and topics matching the keyword.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: SM1-Search-03

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

Test Case Description: This test case verifies if the system displays search results based on their relevance to the search query.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in
- There are various search results for the entered keyword

Test Data: Keywords

Test Steps:

1. Log in
2. Go to the search bar
3. Type a keyword
4. Execute the search

Postconditions:

- The search result is displayed in an order

Expected Result: The system arranges the search results in the order of their relevance to the keyword.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case 4:

Test Case ID: SM1-Search-04

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

Test Case Description: This test case validates if the system allows users to filter their search results by content type.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in
- There are various types of content available for the entered keyword

Test Data: Keywords

Test Steps:

1. Log in
2. Go to the search bar
3. Enter a keyword
4. Execute the search
5. Use the content filter option

Postconditions:

- The search results are filtered

Expected Result: The system filters the search results based on the selected content type.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: SM1-Search-05

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

Test Case Description: This test case checks if the system provides suggestions while users type their search query.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in
- There is content available that matches the partial keyword entry during typing

Test Data: Partial Keywords

Test Steps:

1. Log in
2. Go to the search bar
3. Start typing a keyword

Postconditions:

- Suggestions are shown

Expected Result: The system provides suggestions as the user types, matching the partial keyword entry.

Severity: Minor

Type of Testing: Usability

Test Case Approach: Positive

Project name -

Social Media Platform

Project description -

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

Feature name -

Hashtags and Tagging

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: HTT001

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

Test Case Description: Ensure that the system allows users to add relevant hashtags to their posts to categorize and make their content discoverable.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in
- User has created a post

Test Data: Text-based Post content with hashtags

Test Steps:

1. Navigate to the create post section
2. Write the post content
3. Add relevant hashtags in the post content
4. Click on Post

Postconditions:

- Post is successfully created with hashtags

Expected Result: The system successfully adds hashtags to the posts, making them clickable and leading to related content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: HTT002

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

Test Case Description: This test case will verify whether the system allows users to tag other entities which includes other users, brands and locations.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in
- User is posting a post

Test Data: Tagged entity (user profile, brand, location)

Test Steps:

1. Go to create post section
2. Enter post content
3. Tag relevant profiles, brands, or locations
4. Post the content

Postconditions:

- Post is created with tagged entities

Expected Result: The system allows users to tag other profiles, brands, and locations in their post content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: HTT003

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

Test Case Description: This test case will confirm whether hashtags and tags in a user's post are clickable and direct to related content.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

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

Test Data: No test data needed

Test Steps:

1. Navigate to the post with hashtags or tags
2. Click on the hashtag or tag

Postconditions:

- User is directed to a page with related content

Expected Result: The system enables users to click on hashtags or tags and be redirected to related content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: HTT004

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

Test Case Description: This test case validates the system's ability for users to search and find content using hashtags.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

- User is logged in

Test Data: Hashtag for searching

Test Steps:

1. Go to the search bar
2. Input a hashtag
3. Press enter

Postconditions:

- Search results showing posts with the entered hashtag are shown

Expected Result: The system allows users to search for posts or content using hashtags.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: HTT005

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

Test Case Description: This test case ensures that the system suggests popular hashtags as users begin to type them in a post text.

Test Suite: Hashtags and Tagging

Test Priority: Medium

Preconditions:

- User is logged in
- User is creating a post

Test Data: No test data needed

Test Steps:

1. Navigate to the create post section
2. Begin to type a hashtag in the post content

Postconditions:

- System suggests popular hashtags before user finishes typing

Expected Result: The system provides autosuggestions for popular hashtags as users begin to type a hashtag in their post content.

Severity: Minor

Type of Testing: Functional Testing

Data - 9

Project name -

Social Media Platform

Project description -

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

Feature name -

Emojis and Reactions

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: EmojiReact01

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

Test Case Description: Testing the system functionality to allow users to react to posts using various emojis.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in
- There are available posts to react to

Test Data: User posts, emojis

Test Steps:

1. Navigate to a post
2. Click on the reaction option
3. Select an emoji to react with
4. Confirm reaction

Postconditions:

- User's reaction is displayed on the post

Expected Result: The system allows the user to react to a post with the chosen emoji.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: EmojiReact02

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

Test Case Description: Evaluate whether the system correctly supports adding emojis in comments and

messages.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in
- User is in a chat or post comment section

Test Data: Emojis, messaging and commenting field

Test Steps:

1. Open a chat or a post
2. Click on the message/comment field
3. Click on the emoji button
4. Select an emoji and send the message/comment

Postconditions:

- Emoji appears in the message or comment

Expected Result: The system allows the user to include an emoji in a message or comment.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: EmojiReact03

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

Test Case Description: Check if the system displays the correct number of reactions on the posts

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in
- Post has reactions

Test Data: Post with reactions

Test Steps:

1. Navigate to a post with reactions
2. Look at the reaction count

Postconditions:

- The count displayed corresponds to the post reactions

Expected Result: The system correctly displays the count of reactions on a post.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: EmojiReact04

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

Test Case Description: To confirm that the system offers a varied selection of emojis for users.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

- User is logged in

Test Data: Range of emojis

Test Steps:

1. Navigate to the emoji selection
2. Scroll through the different emojis available

Postconditions:

- A wide range of emojis are available for selection

Expected Result: The system provides a wide selection of emojis for user usage in posts, reactions, messages, and comments.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case 5:

Test Case ID: EmojiReact05

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

Test Case Description: Check whether the system allows users to change or remove their reactions from a post.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

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

Test Data: Post with user reaction

Test Steps:

1. Navigate to a post where the user has reacted
2. Click on the reaction
3. Choose to change or remove the reaction

Postconditions:

- User's initial reaction is changed or removed

Expected Result: The system allows users to successfully change or remove their reactions from a post.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 10

Project name -

Social Media Platform

Project description -

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

Feature name -

Live Streaming

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: LS001

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

Test Case Description: This test will ensure that the system allows users to start and stop live streams easily.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User is on the live streaming page

Test Data: No test data needed

Test Steps:

1. Go to the live streaming page
2. Click on the start live stream button
3. Broadcast for a few minutes
4. Click on the stop live stream button

Postconditions:

- Live stream is stopped

Expected Result: The live stream starts and stops without any issues.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: LS002

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

Test Case Description: This test will ensure that users can set the visibility of their live streams to public or only to their followers.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User is on the live streaming page

Test Data: No test data needed

Test Steps:

1. Go to the live streaming page
2. Before starting the stream, select the visibility option
3. Choose either public or followers
4. Save the settings and start the live stream

Postconditions:

- Live stream starts with the selected visibility settings

Expected Result: The live stream starts with the selected visibility settings, allowing only the selected audience to view it.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case 3:

Test Case ID: LS003

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

Test Case Description: This test will ensure that the system allows viewers to interact with the streamer by

posting comments and reactions.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User is watching a live stream

Test Data: User comments and reactions

Test Steps:

1. Go to a live streaming page
2. During the live stream, post a comment
3. React to the live stream through emojis

Postconditions:

- Comment and reaction appear on the live stream

Expected Result: The comments and reactions from the viewer are successfully posted and appear on the live stream.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: LS004

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

Test Case Description: This test case checks if the system correctly displays viewer count and live comments during the live stream.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in
- User is hosting a live stream

Test Data: No test data needed

Test Steps:

1. Go to the live streaming page
2. Start a live stream
3. Check the displayed viewer count and live comments during the stream

Postconditions:

- Live stream displays viewer count and live comments correctly

Expected Result: The system correctly displays the number of viewers and the live comments during the live stream.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: LS005

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

Test Case Description: This test case ensures that users can save their live stream for later viewing.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

- User is logged in

- User is on the live streaming page

Test Data: No test data needed

Test Steps:

1. Go to the live streaming page
2. Start a live stream
3. Stop the live stream
4. Choose the option to save the live stream for later viewing

Postconditions:

- Live stream is saved and can be viewed later

Expected Result: The system correctly saves the live stream and it successfully appears in the saved streams list.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: SMP01

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

Test Case Description: This test case focuses on testing if the social media platform allows users to create stories consisting of photos, videos and text.

Test Suite: Stories

Test Priority: High

Preconditions:

- User account is created and active.
- User is logged in.
- User has access to photos, videos, and text overlays.

Test Data: User credentials, Photos, Videos, Text overlays

Test Steps:

1. Go to Stories section.

2. Click on 'Create Story' button.
3. Upload a photo or video.
4. Add text overlay.
5. Click on 'Post Story'.

Postconditions: The story is created with photos, videos, and text overlays.

Expected Result: The system allows users to create a story with photos, videos, and text overlays and displays it on the user's profile and in follower's feed.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: SMP02

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

Test Case Description: This case ensures that the stories created by users automatically disappear from their profile and followers' feed after a 24-hour duration.

Test Suite: Stories

Test Priority: High

Preconditions:

- User is logged in.
- User has posted a story.

Test Data: User credentials, Time

Test Steps:

1. Log in to the system after 24 hours of story creation.
2. Navigate to the 'Stories' section in the user profile.
3. Check the availability of the story.

Postconditions: The story is automatically removed from the profile and followers' feed after 24 hours.

Expected Result: The system should automatically remove the story from the platform after 24 hours.

Severity: Major

Type of Testing: Time-based Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: SMP03

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

Test Case Description: This case checks whether the social media platform allows users to see the list of people who have viewed their stories.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- User has posted a story.
- Other users have viewed the posted story.

Test Data: User credentials, Story Data

Test Steps:

1. Go to Stories section.
2. Open a posted story.
3. Check the list of viewers.

Postconditions: The posted story reveals the list of viewers.

Expected Result: The system should show the list of other users who have viewed the posted story.

Severity: Medium

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: SMP04

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

Test Case Description: This case ensures that the stories created by users can be shared with all followers or with a selected group of followers.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- User is logged in
- User has followers
- User has created a story.

Test Data: User credentials, Followers data

Test Steps:

1. Go to Stories section.
2. Choose to create a story.
3. Select the option to share with all followers or only with selected followers.
4. Post the story.

Postconditions: The story is posted and shared with the intended audience.

Expected Result: The system should allow users to share stories with all followers or only with a chosen group of followers.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: SMP05

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

Test Case Description: This case verifies that the system allows users to add stickers and apply filters to their photos or videos while creating a story.

Test Suite: Stories

Test Priority: Medium

Preconditions:

- User is logged in.
- User has access to photos, videos, stickers, and filters.

Test Data: User credentials, Photos, Videos, Stickers, Filters

Test Steps:

1. Go to Stories section.
2. Click on 'Create Story' button.
3. Upload a photo or video.
4. Apply filters.
5. Add stickers.
6. Post the story.

Postconditions: A story is created with stickers and filters.

Expected Result: The system should allow users to add stickers and apply filters to their photos or videos in the stories.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Project name -

Social Media Platform

Project description -

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

Feature name -

Analytics and Insights

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: AC1

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

Test Case Description: User should be able to access and view data related to follower demographics such as age, gender, location.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in
- User has followers

Test Data: Follower demographic data

Test Steps:

1. Navigate to the Analytics and Insights section
2. Click on the link for follower demographics
3. Review the displayed information

Postconditions:

- Analytics and set of insights are displayed

Expected Result: The system will display follower demographic data in a clear and understandable format.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: AC2

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

Test Case Description: The user should be able to see post reach and engagement metrics such as likes, shares, and comments.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in
- User has posted content

Test Data: Post reach and engagement data

Test Steps:

1. Navigate to the Analytics and Insights section
2. Click on the 'Post Reach and Engagement' section
3. Review the displayed information

Postconditions:

- Post reach and engagement metrics are displayed

Expected Result: The system displays the engagement metrics of the user's posts.

Severity: Major

Type of Testing: Usability testing

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: AC3

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

Test Case Description: User should be able to track audience growth trends over a specified period of time.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

- User is logged in
- User has generated sufficient data for trend analysis

Test Data: Audience growth trend data

Test Steps:

1. Navigate to the Analytics and Insights section
2. Click on 'Audience Growth Trends'
3. Review the displayed trends graphically

Postconditions:

- Audience growth trends are shown on insights

Expected Result: The system shows the graphical representation of audience growth trends.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: AC4

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

Test Case Description: Analytics should be displayed in a user-friendly format, including graphs, charts, and interactive elements.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

- User is logged in
- User has generated some Analytics and Insights data

Test Data: No test data needed

Test Steps:

1. Navigate to the Analytics and Insights section
2. Review the format in which data is displayed

Postconditions:

- Analytics are displayed in a user-friendly format

Expected Result: The system displays all analytics data in a user-friendly format.

Severity: Minor

Type of Testing: Usability Testing

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: AC5

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

Test Case Description: Users should be able to filter the displayed analytics data by time range and type of content.

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. Set the filter parameters for time range and content type
3. Verify that the displayed data aligns with the set filter parameters

Postconditions:

- Filtered Analytics and Insights are displayed

Expected Result: The system applies the user set filter parameters to the displayed data.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 13

Project name -

Social Media Platform

Project description -

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

Feature name -

Events and Groups

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: SMP1

Test Case Title: Verify that users can create events

Test Case Description: Ensure that the system allows users to create new events with all necessary details such as date, time, location, and description.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- User is logged in
- User has the permissions to create events

Test Data: Event details (Date, Time, Location, Description)

Test Steps:

1. Navigate to events page
2. Click on the "Create Event" button
3. Enter all required event details
4. Click on "Save Event"

Postconditions:

- Event is created successfully

Expected Result: The system successfully creates a new event and displays it on the events page.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMP2

Test Case Title: Verify that users can join events

Test Case Description: Check if the system allows users to join already created events.

Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

- User is logged in
- Events are available to join

Test Data: No test data needed

Test Steps:

1. Navigate to events page
2. Browse available events
3. Click on "Join Event"

Postconditions:

- User joined the event successfully.

Expected Result: The system changes the status of the user to 'Joined' for the respective event.

Severity: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMP3

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

Test Case Description: This test case validates if users can join a group based on their interests.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- User is logged in
- Groups are available to join

Test Data: No test data needed

Test Steps:

1. Navigate to groups page
2. Search groups by interest
3. Select a group
4. Click on "Join Group" button

Postconditions:

- User successfully joined a group.

Expected Result: The system adds the user to the selected group.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMP4

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

Test Case Description: Check if users can organize their group discussions based on different topics.

Test Suite: Events and Groups

Test Priority: Medium

Preconditions:

- User is logged in
- User is joined in a group

Test Data: Group discussion details

Test Steps:

1. Navigate to group page
2. Click on 'Start a Discussion'
3. Fill all required fields including topic

4. Click on 'Post'

Postconditions:

- Discussion is posted successfully.

Expected Result: A new discussion is started in the group and is organized under the provided topic.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMP5

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

Test Case Description: This test case ensures users can confirm their presence for an event and view attendees list.

Test Suite: Events and Groups

Test Priority: High

Preconditions:

- User is logged in
- User has joined an event

Test Data: No test data needed

Test Steps:

1. Navigate to the event page which the user has joined
2. Click on 'RSVP' for the event
3. Select 'I am going'
4. Click on 'Attendees' to view the list

Postconditions:

- User RSVP status is updated

- Attendees list is displayed

Expected Result: The system updates the user's RSVP status and provides a list of all attendees for the event.

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 Case 1:

Test Case ID: SMT01

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

Test Case Description: This test case verifies if the users can apply for the monetization options in the system.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in
- User's profile is completely filled

Test Data: User profile details, Monetization options

Test Steps:

1. Log into the account.
2. Navigate to Monetization Options.
3. Apply for a monetization option.

Postconditions:

- Monetization application is submitted

Expected Result: The system should let user apply for monetization options and gives a message of successful application.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: SMT02

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

Test Case Description: This test case is to ensure that the users have monetization options like advertising revenue and sponsored content.

Test Suite: Monetization Options

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Monetization options

Test Steps:

1. Log into the account.
2. Navigate to Monetization Options.
3. Check for options like advertising revenue and sponsored content.

Postconditions:

- User is aware of the monetization options

Expected Result: Options like advertising revenue and sponsored content should be available for users in monetization options.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: SMT03

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

Test Case Description: This test case is to verify if the users can receive payments through the platform.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in
- Payment details are set up

Test Data: Earnings details

Test Steps:

1. Log into the account.
2. Navigate to earnings section.
3. Initiate a withdrawal.

Postconditions:

- User has withdrawn earnings

Expected Result: Users should be able to initiate a withdrawal and receive payments through the platform.

Severity: Blocker

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: SMT04

Test Case Title: Verify that detailed earnings reports are available to the users

Test Case Description: This test case checks if the users can access detailed earnings reports.

Test Suite: Monetization Options

Test Priority: High

Preconditions:

- User is logged in
- User has earned revenue

Test Data: Earnings report

Test Steps:

1. Log into the account.
2. Navigate to earnings section.
3. Check for detailed earnings report.

Postconditions:

- User has viewed earnings report

Expected Result: Users should be able to view a detailed report of their earnings.

Severity: Critical

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: SMT05

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

Test Case Description: This test case checks whether eligibility requirements for monetization are clearly stated.

Test Suite: Monetization Options

Test Priority: Low

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Log into the account.

2. Navigate to Monetization Options.

3. Check for eligibility requirements.

Postconditions:

- User has viewed eligibility requirements

Expected Result: Eligibility criteria for monetization options should be clearly stated.

Severity: Minor

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Data - 15

Project name -

Social Media Platform

Project description -

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

Feature name -

Explore or Discover Tab

Feature description -

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

Userstory title -

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

topics.

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: EXP001

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

Test Case Description: The test case aims to validate that the Explore tab in the social media platform curates and displays trending and popular content from across the platform.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- User is logged into the social media platform
- There are trending and popular posts on the platform

Test Data: User login credentials, Trending posts data

Test Steps:

1. Log into the platform
2. Click on the explore tab
3. Observe the posts displayed

Postconditions:

- User is able to access Explore tab

Expected Result: The Explore tab showcases trending and popular posts from all over the platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: EXP002

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

Test Case Description: The test aims to ensure that the social media platform personalizes the content in the Explore tab based on users' interests.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

- User is logged into the social media platform
- User has indicated interests or liked similar content on the platform

Test Data: User login credentials, User interest data

Test Steps:

1. Log into the platform
2. Click on the explore tab
3. Verify if the displayed content aligns with the user's interests or previous activity

Postconditions:

- Personalized content is displayed

Expected Result: The content displayed on the Explore tab is personalized based on the user's interests.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: EXP003

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

Test Case Description: This test case is designed to ensure that users are able to filter the content displayed in the Explore tab by category or topic.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged into the social media platform
- There are multiple categories or topics of content available

Test Data: User login credentials, Content categories and topics

Test Steps:

1. Log into the platform
2. Click on the explore tab
3. Apply filter by category or topic
4. Observe the displayed content

Postconditions:

- Content is filtered and displayed as per chosen category or topic

Expected Result: The Explore tab allows the user to filter content by category or topic, displaying the filtered results accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: EXP004

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

Test Case Description: This test case will ensure that the Explore tab is updated with fresh new content on a regular basis.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged into the social media platform
- New content is being added to the platform regularly

Test Data: User login credentials, Time-stamped content data

Test Steps:

1. Log into the platform
2. Click on the explore tab
3. Observe the content displayed
4. Refresh the explore tab after some time
5. Observe if new content is displayed

Postconditions:

- Explore tab is refreshed with new content

Expected Result: The Explore tab updates regularly to display new trending content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: EXP005

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

Test Case Description: This test case will ensure that users are able to save or follow content that interests

them directly from the Explore tab.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

- User is logged into the social media platform
- Content displayed can be saved or followed

Test Data: User login credentials

Test Steps:

1. Log into the platform
2. Click on the explore tab
3. Choose a content to save or follow
4. Click on the save or follow button
5. Check that the action is reflected accurately in user's account

Postconditions:

- User has successfully saved or followed content from the explore tab

Expected Result: The Explore tab allows users to save or follow the content directly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 16

Project name -

Social Media Platform

Project description -

Develop a comprehensive social media platform where users can create profiles, connect with friends, share

posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending topics and communities. The platform should include features like news feed, messaging, user profiles, privacy settings, notifications, and analytics.

Feature name -

Accessibility Features

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: A01

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

Test Case Description: Check if the platform provides alternative text representations for all images that can be read by screen readers and understood by visually impaired users.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in

- Image posts are available

Test Data: Image with alt text caption

Test Steps:

1. Log in to the platform
2. Go to a post having an image
3. Hover over the image to check for alt text or use a screen reader to identify the alt text

Postconditions:

- Alt text displayed for images

Expected Result: The alt text is provided for all images in a readable manner, providing an accurate description of the image content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: A02

Test Case Title: Verify that screen reader compatibility is available

Test Case Description: Ensure that the platform can be navigated and interacted with successfully using a screen reader device/software to aid visually impaired users.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in
- Screen reader software is installed

Test Data: No test data needed

Test Steps:

1. Log in to the platform using the screen reader
2. Navigate through the main screen and features
3. Check if all the buttons, texts, and other elements are properly mentioned and understandable through the screen reader

Postconditions:

- Screen reader software can effectively interpret information on the platform

Expected Result: The screen reader should be able to navigate the platform and interpret the information accurately.

Severity: Critical

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: A03

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

Test Case Description: Test the platform's feature that allows users to navigate through pages, menus, options, etc., using defined keyboard shortcuts.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: List of pre-defined keyboard shortcuts

Test Steps:

1. Log in to the platform

2. Use various shortcuts to navigate through pages, open menus, like, comment, share, etc.
3. Check if every shortcut is functioning as expected

Postconditions:

- Users can navigate and interact with the platform using keyboard shortcuts

Expected Result: The keyboard shortcuts work effectively, providing quicker navigation and interactions.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: A04

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

Test Case Description: Check if the platform provides options to modify the color contrast and font sizes to aid users with visual impairments.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data required

Test Steps:

1. Log in to the platform
2. Go to user preferences/settings
3. Find and adjust the color contrast and font size settings

Postconditions:

- Settings are saved and changes reflect throughout the platform

Expected Result: Users can easily adjust the color contrast and font size according to their preferences, enhancing their user experience.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: A05

Test Case Title: Verify that accessibility features are clearly documented

Test Case Description: Ensure documentation provides clear instructions on how to enable and use each accessibility feature.

Test Suite: Accessibility Features

Test Priority: Low

Preconditions:

- User has access to the platform documentation

Test Data: Accessibility feature documentation

Test Steps:

1. Find and open the platform's user guide or related documentation
2. Navigate and find the Accessibility Features section
3. Evaluate for clarity and completeness of instructions

Postconditions:

- Documentation provides clear instructions on the use of Accessibility Features

Expected Result: The platform's documentation should provide users with disabilities instructions that are clear and comprehensive.

Severity: Minor

Type of Testing: Documentation Review

Test Case Approach: Positive

Data - 17

Project name -

Social Media Platform

Project description -

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

Feature name -

Safety and Moderation Tools

Feature description -

Safety and moderation tools are designed to protect users from harmful or inappropriate content and interactions on the platform. This may include features such as content moderation algorithms, reporting tools, blocking capabilities, comment filters, and community guidelines enforcement, fostering a safer and more positive online environment.

Userstory title -

As a user, I want safety and moderation tools so that I can have a safe and positive experience on the platform.

Acceptance criteria -

['Content moderation algorithms filter inappropriate content.', 'Users can report harmful or inappropriate content.', 'Blocking capabilities are available to users.', 'Comment filters prevent offensive language.', 'Community guidelines are enforced consistently.']

Generated Test Cases:

Test Case 1:

Test Case ID: SMPT1

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

Test Case Description: Ensure that the posted content is screened and filtered by the moderation algorithms to stop inappropriate content from being posted.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- User attempts to post inappropriate content

Test Data: User profile information, content text

Test Steps:

1. Go to the 'Create a Post' section
2. Draft inappropriate content
3. Attempt to post the content

Postconditions:

- Content is not posted
- Reason for content rejection is provided

Expected Result: The system stops the content from being posted and explains why.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

Test Case 2:

Test Case ID: SMPT2

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

Test Case Description: Ensure that users have the ability to report any harmful or inappropriate content they come across on the platform.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- Offensive content is present

Test Data: User profile information, offensive content details

Test Steps:

1. Navigate to offending post
2. Click on report button
3. Fill out report form and submit

Postconditions:

- Report is submitted
- Notification of successful submission

Expected Result: The system successfully submits the report and notifies the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMPT3

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

Test Case Description: Ensure that users can block other users to prevent any unwanted interaction or communication.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- Offensive user is on platform

Test Data: User profile information, Offensive user profile information

Test Steps:

1. Navigate to offending user's profile
2. Click on 'Block User' button
3. Confirm the action

Postconditions:

- Offending user is blocked
- Notification of successful block operation

Expected Result: The system blocks the user and sends a confirmation notification.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMPT4

Test Case Title: Verify that comment filters prevent offensive language

Test Case Description: Ensure that comments containing offensive language get filtered out by the platform's content filter.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- Offensive comment is attempted

Test Data: User profile information, Offensive comment text

Test Steps:

1. Navigate to a post
2. Type an offensive comment
3. Attempt to post the comment

Postconditions:

- Comment does not get posted
- Notification of why the comment was not posted

Expected Result: The system prevents the offensive comment from being posted and informs the user why.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

Test Case 5:

Test Case ID: SMPT5

Test Case Title: Verify that community guidelines are enforced consistently

Test Case Description: Ensure that any posts violating the community guidelines are not allowed in the platform.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- Offensive post is drafted in violation of community guidelines

Test Data: User profile information, Offensive post content

Test Steps:

1. Navigate to 'Create a Post' section
2. Draft a post in violation of community guidelines
3. Attempt to post the content

Postconditions:

- Post is not displayed on the platform
- Notification of violation of guidelines

Expected Result: The system denies the post due to the violation of community guidelines and explains the reason to the user.

Severity: Major

Type of Testing: Functional

Test Case Approach: 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 Case 1:

Test Case ID: ML1

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

Test Case Description: This test will validate the functionality of the user's ability to set a preferred language for the interface.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User has an active account on the platform
- User is logged in

Test Data: Available language options

Test Steps:

1. Navigate to the user's profile settings.
2. Locate the language preference section.
3. Choose a language from the given options.

4. Set the chosen language as the preferred language.

5. Save changes.

Postconditions:

- Interface language is updated

Expected Result: The platform changes to the user's preferred language once the setting is saved.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: ML2

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

Test Case Description: This test will validate the content translation functionalities of the platform.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

- User has an active account on the platform
- User is logged in

Test Data: Various posts and messages in different languages

Test Steps:

1. Navigate to a post or message written in a language different from the preferred language.
2. Click on translate button.

Postconditions:

- Post or message is translated to the preferred language

Expected Result: The platform translates the selected post/message in user's preferred language.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: ML3

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

Test Case Description: This test will confirm if posts and messages can be translated into multiple languages.

Test Suite: Multi-Language Support

Test Priority: Medium

Preconditions:

- User has an active account on the platform
- User is logged in

Test Data: Posts and messages in various languages

Test Steps:

1. Navigate to a post or message.
2. Click on the translate option.
3. Select the desired language from the options.

Postconditions:

- Selected post or message is translated into the chosen language

Expected Result: The platform translates the post/message into the chosen language.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: ML4

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

Test Case Description: This test is to check if the user's language preferences are effectively saved and remain consistent across sessions.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User has an active account on the platform
- User is logged in
- User has set a preferred language.

Test Data: No test data needed

Test Steps:

1. Log out of the platform.
2. Log back into the platform.

Postconditions:

- Language choice persists across sessions

Expected Result: The platform maintains the user's preferred language across different sessions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: ML5

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

Test Case Description: Check whether the platform supports a wide range of languages for interface localization and content translation.

Test Suite: Multi-Language Support

Test Priority: Low

Preconditions:

- User is logged in to the platform

Test Data: Collection of different languages to be offered by the platform

Test Steps:

1. Navigate to the user's profile settings.
2. Locate the language selection option.
3. Check the available languages in the dropdown.

Postconditions:

- User is aware of the available languages

Expected Result: The platform offers a wide range of languages for interface localization and content translation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 19

Project name -

Social Media Platform

Project description -

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

Feature name -

Cross-Platform Integration

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TCP1

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

Test Case Description: Ensure the system provides an option for users to link their accounts with different social media platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is logged in
- User has an existing account on other social media platforms

Test Data: Social media account credentials

Test Steps:

1. Go to account settings
2. Choose the option to link other social media accounts
3. Enter the credentials of the other social media accounts
4. Confirm linking

Postconditions:

- User's account is linked with other social media platforms

Expected Result: The system links the user's account to the selected social media platforms and confirms the linking.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TCP2

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

Test Case Description: Ensure the system provides an option for users to share their posts across multiple social media platforms.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is logged in
- User has linked their accounts with other social media platforms

Test Data: User post content

Test Steps:

1. Create a new post

2. Choose the option to share post on other platforms
3. Select the platforms to share the post
4. Confirm sharing

Postconditions:

- Post is shared on other platforms

Expected Result: The system shares the user's post to the selected platforms and confirms the cross-posting.

Severity: Major

Type of Testing: Integration

Test Case Approach: Positive

Test Case 3:

Test Case ID: TCP3

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

Test Case Description: Ensure the system allows users to sign in using SSO, enhancing user convenience.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User has an existing account on the SSO provider platform

Test Data: SSO account credentials

Test Steps:

1. Go to the login page
2. Select the option to log in using SSO
3. Enter the SSO account credentials
4. Confirm login

Postconditions:

- User is logged in using SSO

Expected Result: The system allows the user to log in using SSO and lands them on the home page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TCP4

Test Case Title: Verify the integration with external media sources

Test Case Description: Make sure the system allows users to integrate and share content from external media sources.

Test Suite: Cross-Platform Integration

Test Priority: Medium

Preconditions:

- User is logged in
- External media sources are available

Test Data: External media source URL

Test Steps:

1. Create a new post
2. Choose the option to add content from external media source
3. Paste the URL of the media source
4. Confirm post creation

Postconditions:

- Post with integrated media is created

Expected Result: The system allows the user to integrate external media into their post and confirms the post creation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TCP5

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

Test Case Description: Validate the system's capability to share content seamlessly across different platforms, without hindering the user experience.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

- User is logged in
- User has linked their accounts with other social media platforms

Test Data: User post content

Test Steps:

1. Create a new post
2. Choose the option to share post on other platforms
3. Select the platforms to share the post
4. Confirm sharing

Postconditions:

- Post is shared on other platforms

Expected Result: The system promptly shares the post on the selected platforms without any glitches, ensuring a seamless sharing experience.

Severity: Major

Type of Testing: End-to-End

Test Case Approach: Positive

Data - 20

Project name -

Social Media Platform

Project description -

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

Feature name -

Gamification and Rewards

Feature description -

Gamification and rewards enhance user engagement by incorporating game-like elements such as points, badges, leaderboards, and challenges. Users can earn rewards for their activities, achievements, and contributions on the platform, fostering a sense of accomplishment and encouraging continued interaction and participation.

Userstory title -

As a user, I want gamification and rewards so that I can earn recognition and incentives for my activities.

Acceptance criteria -

['Users can earn points and badges for their activities.', 'Leaderboards display top users based on points or achievements.', 'Challenges are available for users to complete and earn rewards.', 'Rewards can include virtual goods, discounts, or exclusive content.', 'Users can view their progress and achievements in a dedicated section.']

Generated Test Cases:

Test Case 1:

Test Case ID: GR01

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

Test Case Description: Ensure the gamification system of the social media platform allows users to earn points and badges for various activities like making posts, commenting, and sharing.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- A user is logged into the platform.
- The user is aware of the gamification system and how to earn points or badges.

Test Data: User account and activities

Test Steps:

1. Login to user account.
2. Perform an activity such as making a post.
3. Check if points or badges are earned for the performed activity.

Postconditions:

- Points or badges are awarded for the activity performed.

Expected Result: A user earns points or badges after completion of an activity on the platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: GR02

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

Test Case Description: The test case verifies if the platform properly displays a leaderboard that ranks top users based on earned points or badges achieved.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- Multiple users have performed activities that earn points or badges.

Test Data: User accounts, points or badges earned

Test Steps:

1. View leaderboards.
2. Verify if the leaderboard ranks users based on points or achievements.

Postconditions:

- Leaderboard updates with top-ranking users.

Expected Result: The leaderboard correctly displays the top users based on their points or achievements.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: GR03

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

Test Case Description: The test ensures that the platform offers challenges to users that can be completed to earn rewards.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- User is logged in.

Test Data: User account

Test Steps:

1. Login to User account.
2. Browse for available challenges.
3. Verify if challenges are completable and rewards can be earned.

Postconditions:

- Rewards are given upon completion of challenges.

Expected Result: Users are able to complete challenges and earn rewards.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: GR04

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

Test Case Description: The test ensures that the platform offers rewards such as virtual goods, discounts, or exclusive content.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- User is logged in.
- User has completed a challenge or achieved certain points or badges.

Test Data: User account, reward details

Test Steps:

1. Login to User account.
2. Verify if rewards such as virtual goods, discounts, or exclusive content are provided.

Postconditions:

- Rewards are given to the user.

Expected Result: Users are able to receive rewards such as virtual goods, discounts, or exclusive content.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: GR05

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

Test Case Description: The test ensures that the platform allows users to view their gamification progress and achievements in a dedicated section.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User is logged in.

Test Data: User account, points, badges, or achievements detail

Test Steps:

1. Login to User account.
2. Navigate to the dedicated section for viewing gamification progress and achievements.
3. Verify if the section correctly displays users' progress and achievements.

Postconditions:

- Progress and achievements are updated in the relevant section.

Expected Result: Users are able to view their progress and achievements in a dedicated section.

Severity: Major

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 Case 1:

Test Case ID: TC01

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

Test Case Description: This test case is designed to validate that the Task Creation feature allows the user to input the 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, description, due date, and priority level

Test Steps:

1. Go to the "Create Task" page
2. Enter a task title
3. Enter a task description
4. Select a due date
5. Select a priority level

Postconditions:

- Task information is entered

Expected Result: The system successfully saves the entered task information.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC02

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

Test Case Description: This test case is designed to validate that the Task Creation feature allows the user to assign tasks to specific team members.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- User is logged in
- User is on the "Create Task" page
- Task contains the necessary information (title, description, due date, priority level)

Test Data: Team member information

Test Steps:

1. Go to the "Create Task" page
2. Enter a task with the necessary information
3. Assign the task to a specific team member

Postconditions:

- Task is assigned to a team member

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

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC03

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

Test Case Description: This test case is designed to validate that the Task Creation feature saves the created tasks and allows the user to view them later.

Test Suite: Task Creation

Test Priority: High

Preconditions:

- User is logged in
- User has created a task

Test Data: No test data needed

Test Steps:

1. Go to the "View Tasks" page
2. Locate the created task

Postconditions:

- Task is displayed

Expected Result: The system successfully displays the created task.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC04

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

Test Case Description: This test case is designed to validate that the Task Creation feature allows users to edit the details of a task after it has been created.

Test Suite: Task Creation

Test Priority: Medium

Preconditions:

- User is logged in
- User has created a task

Test Data: New task details

Test Steps:

1. Go to the "View Tasks" page
2. Select the task to be edited
3. Edit the task details
4. Save changes

Postconditions:

- Task details are updated

Expected Result: The system successfully updates the task, and the changes are saved and displayed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC05

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

Test Case Description: This test case is designed to validate that the Task Creation feature requires certain fields, such as the task title, to be input before a task can be created.

Test Suite: Task Creation

Test Priority: High

Preconditions:

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

Test Data: Partial task details

Test Steps:

1. Go to the "Create Task" page
2. Enter task details, leaving the title blank
3. Attempt to save the task

Postconditions:

- Task is not saved

Expected Result: The system prevents task saving and prompts the user to complete the required fields.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Negative

Data - 22

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Assignment

Feature description -

Tasks can be assigned to specific team members, delegating responsibility and ensuring clarity about who is

accountable for each task. Assignees receive notifications about their assigned tasks, helping them stay informed and focused on their responsibilities.

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TMT001

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

Test Case Description: Ensure that the system allows users to assign tasks to specific team members.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- User is logged in
- Task is created
- Team members are added

Test Data: Task details, Team member names

Test Steps:

1. Open the created task
2. Click on the assign button
3. Choose a team member from the list

4. Confirm task assignment

Postconditions:

- Task is assigned to selected team member

Expected Result: The system assigns the task to the selected team member and reflects in the task details.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT002

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

Test Case Description: The system must notify team members when a task is assigned to them.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- Task is assigned to a team member

Test Data: Notification settings

Test Steps:

1. Assign a task to a team member
2. Login as the team member
3. Check notifications

Postconditions:

- Notification received

Expected Result: The system sends a notification of task assignment to the selected team member.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT003

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

Test Case Description: Ensure that assigned tasks are displayed in the assignee's task list.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

- Task is assigned to a team member

Test Data: No test data needed

Test Steps:

1. Assign the task to a team member
2. Login as the team member
3. Check the task list

Postconditions:

- Task appears in the task list

Expected Result: The system shows the assigned tasks in the assignee's task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT004

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

Test Case Description: Ensure that the system allows reassigning tasks to different team members.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

- Task is assigned to a team member
- More than one team member is present

Test Data: Task Details and Team member names

Test Steps:

1. Open the existing assigned task
2. Click on the reassign button
3. Choose a different team member from the list
4. Confirm task reassignment

Postconditions:

- Task is reassigned and notifications are sent

Expected Result: The system allows reassignment of tasks to different team members and send out notifications.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT005

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

Test Case Description: Ensure that the system keeps track of task assignment history.

Test Suite: Task Assignment

Test Priority: Low

Preconditions:

- Task has been reassigned at least once

Test Data: No test data needed

Test Steps:

1. Open the existing reassigned task
2. Navigate to task history
3. Check assignment history

Postconditions:

- Task assignment history is updated

Expected Result: The system keeps track of task assignment history and reflects every change in assignment accurately.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 23

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Deadline Reminders

Feature description -

Users receive reminders and notifications for upcoming task deadlines, ensuring that tasks are completed on time and preventing delays. Reminders can be customized based on users' preferences and urgency levels.

Userstory title -

As a user, I want to receive reminders for upcoming task deadlines so that I can complete tasks on time.

Acceptance criteria -

['Users can set reminders for tasks based on deadlines.', 'Reminders can be customized by urgency level.', 'The system sends notifications for upcoming deadlines.', 'Users can choose the delivery method for reminders (e.g., email, in-app).', 'Reminder settings can be adjusted or disabled.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT1

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

Test Case Description: This test case will ensure that the system enables users to set reminders for tasks according to their respective deadlines.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged in
- Task is created with a specified deadline

Test Data: Task details with deadlines

Test Steps:

1. Select a task
2. Click on set reminder
3. Choose a reminder time based on the deadline
4. Save the setting

Postconditions:

- Reminder is set for the task

Expected Result: The system sets the reminder successfully for the selected task based on the deadline.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: TMT2

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

Test Case Description: This test case checks the user's ability to customize task reminders based on the level of urgency.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged in
- Reminder for a task is set

Test Data: Task details and urgency level

Test Steps:

1. Select a task with a reminder
2. Click on customize reminder
3. Set the urgency level for the reminder
4. Save changes

Postconditions:

- Reminder is customized with urgency level

Expected Result: The system customizes the reminder based on the chosen urgency level.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: TMT3

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

Test Case Description: This test ensures that the system sends out notifications for upcoming task deadlines to a user.

Test Suite: Task Deadline Reminders

Test Priority: High

Preconditions:

- User is logged in
- Task with a set deadline and reminder is created

Test Data: No test data needed

Test Steps:

1. Wait for the reminder time
2. Check if a notification is received for the upcoming deadline

Postconditions:

- User receives notification for upcoming deadline

Expected Result: The system successfully sends a notification for the upcoming task deadline.

Severity: Critical

Type of Testing: Integration

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: TMT4

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

Test Case Description: This test ensures that users can select their desired delivery method (e.g., email, in-app) for task deadline reminders.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- User is logged in
- Reminder for a task is set

Test Data: Election of choice for reminder delivery method

Test Steps:

1. Go to reminder settings
2. Choose the delivery method for reminders
3. Save changes

Postconditions:

- Delivery method for reminders is updated

Expected Result: The system allows the user to choose a delivery method for reminders and applies the changes successfully.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: TMT5

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

Test Case Description: This test case checks if users can adjust or disable reminder settings for task deadlines.

Test Suite: Task Deadline Reminders

Test Priority: Low

Preconditions:

- User is logged in
- Reminder for a task is set

Test Data: No test data needed

Test Steps:

1. Go to reminder settings of a task
2. Adjust or disable the reminder
3. Save changes

Postconditions:

- Reminder setting for the task is updated

Expected Result: The system allows the user to adjust or disable reminder settings successfully.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 24

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

Users can prioritize tasks based on their urgency and importance, allowing them to focus on high-priority tasks and manage their workload effectively. Tasks can be categorized into different priority levels, such as high, medium, or low, based on their importance to the project or team.

Userstory title -

As a user, I want to prioritize tasks so that I can manage my workload effectively.

Acceptance criteria -

['Users can set priority levels for tasks (High, Medium, Low).', 'Tasks can be sorted based on priority levels.', 'Users can change the priority level of a task.', 'The system visually distinguishes tasks based on priority.', 'Tasks without a specified priority default to Medium.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT-TP-01

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

Test Case Description: This test case will verify that the system allows users to assign priority levels (High, Medium, Low) to tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in and has access to create tasks
- Task to be assigned a priority level is available

Test Data: Task details

Test Steps:

1. Log into the Task Management system
2. Select a task
3. Assign a priority level to the task
4. Save the task

Postconditions:

- Priority level of the task is updated

Expected Result: The selected task should be assigned the chosen priority level.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: TMT-TP-02

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

Test Case Description: This test case checks whether the system is able to sort tasks based on their assigned priority levels.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in and has tasks with different priority levels assigned

Test Data: No test data needed

Test Steps:

1. Log into the Task Management system
2. Navigate to the tasks dashboard
3. Initiate task sorting by priority

Postconditions:

- Sorting of tasks by priority is displayed

Expected Result: The tasks should be sorted based on their priority levels.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: TMT-TP-03

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

Test Case Description: This test case ensures that users can change the previously set priority level of a task to a different level.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in and has access to change tasks
- Task previously assigned a priority level is available

Test Data: Task details

Test Steps:

1. Log into the Task Management system
2. Select a task
3. Change the task's priority
4. Save the task

Postconditions:

- Priority level of the task is changed

Expected Result: The selected task should reflect the newly chosen priority level.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: TMT-TP-04

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

Test Case Description: This test case checks if the system is able to distinguish tasks visually based on their assigned priority levels.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in and has tasks assigned with different priority levels

Test Data: No test data needed

Test Steps:

1. Log into the Task Management system
2. Navigate to the tasks dashboard
3. Verify the visual distinction of tasks based on priority

Postconditions:

- Visual distinction of tasks based on priority is confirmed

Expected Result: The tasks should be visually distinguished based on their priority levels.

Severity: Minor

Type of Testing: Usability Testing

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: TMT-TP-05

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

Test Case Description: This test case verifies that any task created without a specified priority level defaults to Medium.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in and has access to create tasks

Test Data: Task details

Test Steps:

1. Log into the Task Management system
2. Create a new task without assigning a priority level
3. Save the task

Postconditions:

- The task with no assigned priority level is saved

Expected Result: The task without a specified priority level should default to Medium.

Severity: Minor

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Data - 25

Project name -

Task Management Tool

Project description -

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

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

Feature name -

Subtasks

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1:

Test Case ID: TMT01

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

Test Case Description: The purpose of this case is to verify that the system allows a user to successfully create a subtask under a main task.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Main task exists

Test Data: Subtask details

Test Steps:

1. Select a main task
2. Click on the "Add Subtask" button
3. Provide subtask details in the form
4. Click on the "Save" button

Postconditions:

- Subtask is created under the main task

Expected Result: The system creates the subtask under the chosen main task successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT02

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

Test Case Description: The objective of this case is to validate whether the system allows users to set independent due dates and assignees for a subtask.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Subtask exists

Test Data: Due date and assignee details

Test Steps:

1. Select a subtask

2. Click on "Edit subtask"
3. Set due date and assignee
4. Save the changes

Postconditions:

- Due date and assignee assigned to the subtask are updated

Expected Result: The system allows users to set and save due dates and assignees for a subtask successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT03

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

Test Case Description: This test verifies whether the system accurately tracks the progress status of each subtask.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Subtask is created and assigned

Test Data: No test data needed

Test Steps:

1. Go to the subtask progress section
2. Update the status of the subtask
3. Verify if the subtask progress reflects the update

Postconditions:

- Subtask progress is tracked and updated

Expected Result: The system records and displays the updated progress status of the subtask accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT04

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

Test Case Description: The objective of this case is to ascertain if the system allows users to mark a subtask as complete, without the need to complete the main task

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Subtask is created

Test Data: No test data needed

Test Steps:

1. Go to the subtask section
2. Select a subtask
3. Mark this subtask as complete

Postconditions:

- Subtask is marked as complete

Expected Result: The system successfully allows users to independently mark a subtask as complete.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT05

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

Test Case Description: The purpose of this case is to ascertain that the system enables users to set dependencies between subtasks.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Multiple subtasks exist

Test Data: No test data needed

Test Steps:

1. Select a subtask
2. Click on "Set dependencies"
3. Choose a second subtask as the dependent task
4. Save the dependency

Postconditions:

- Dependency between subtasks is created

Expected Result: The system allows users to set up dependencies between subtasks successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 26

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Dependencies

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Test Case 1

Test Case ID: TC001

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

Test Case Description: This test checks whether users are allowed to set dependencies between different tasks

within a project.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- Project and tasks are created

Test Data: Task names

Test Steps:

1. Select a task
2. Click on the dependencies option
3. Select another task as a dependent task
4. Confirm the settings

Postconditions:

- Dependencies are established between tasks

Expected Result: The system should accept and display the dependencies between the chosen tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: TC002

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

Test Case Description: This test ensures that the system accurately displays the defined dependencies in task details.

Test Suite: Task Dependencies

Test Priority: Medium

Preconditions:

- User has set dependencies between tasks

Test Data: No test data needed

Test Steps:

1. Open a task
2. Navigate to task details
3. Check dependency details

Postconditions:

- Dependency details are displayed

Expected Result: The system successfully displays the dependencies in the task details.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: TC003

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

Test Case Description: This test examines whether the system prevents marking a task as complete when its dependent tasks are not completed.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User has set dependencies between tasks

Test Data: Task names

Test Steps:

1. Pick a task which has a dependent task

2. Attempt to mark the task as complete

Postconditions:

- Task remains incomplete

Expected Result: The system should not allow marking the task as complete as long as its dependent task is incomplete.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Negative

Test Case 4

Test Case ID: TC004

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

Test Case Description: This test ensures the usability of the feature by allowing users to update or remove existing dependencies between tasks.

Test Suite: Task Dependencies

Test Priority: Low

Preconditions:

- User has set dependencies between tasks

Test Data: Task names

Test Steps:

1. Open a task
2. Navigate to dependencies
3. Remove or change a dependency
4. Save changes

Postconditions:

- Dependencies are updated

Expected Result: The system allows users to effectively update or remove dependencies between tasks.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: TC005

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

Test Case Description: To ensure that task dependencies are managed properly, the system needs to provide alerts or warnings when incomplete tasks have dependent tasks linked to them.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User has set dependencies between tasks

Test Data: Task names

Test Steps:

1. Open a task with incomplete dependent tasks
2. Receive a warning or alert

Postconditions:

- User is alerted

Expected Result: The system provides an alert or warning indicating that the dependent tasks of the opened task are not completed.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Data - 27

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Comments and Collaboration

Feature description -

Users can collaborate on tasks by leaving comments, attaching files, and engaging in discussions directly within the task interface. This feature facilitates communication and teamwork, allowing team members to provide updates, share feedback, and ask questions related to the task.

Userstory title -

As a user, I want to comment on tasks so that I can collaborate with my team.

Acceptance criteria -

['Users can add comments to tasks.', 'Comments can include text, attachments, and links.', 'Users can reply to and edit their comments.', 'The system notifies users of new comments.', 'Comments are displayed in chronological order.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC1_ADD_CMT

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

Test Case Description: The test case ensures that users are able to successfully add comments to tasks in the

Task Management Tool.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in
- A task exists in the project

Test Data: Task information, Comment text

Test Steps:

1. Navigate to the task
2. Click on the comment section
3. Enter the comment text
4. Click on the add comment button

Postconditions:

- Comment is added to task

Expected Result: The system successfully adds the comment to the task, and the new comment is displayed in the task comment section.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC2_CMT_ATTACHMENT

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

Test Case Description: This test verifies the ability of users to add text, attachments, and links to their comments.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in
- A task exists in the project

Test Data: Task information, Comment text, File attachment, URLs

Test Steps:

1. Navigate to the task
2. Click on the comment section
3. Enter the comment text, attachment, and links
4. Click on the add comment button

Postconditions:

- Comment with text, attachment and link is added.

Expected Result: The system successfully adds the comment with text, file attachment, and links to the selected task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC3_REPLY_EDIT_CMT

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

Test Case Description: The test validates the system's ability to allow users to reply to comments and edit their own comments.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in
- A task with comments exists

Test Data: Task information, Text for comment reply and editing

Test Steps:

1. Navigate to the task
2. Locate the comment to reply to or edit
3. Click on the reply button or the edit button
4. Enter the reply or editing text
5. Click on the confirm button

Postconditions:

- Comment is replied or edited

Expected Result: The system successfully replies to the selected comment or edits the user's own comment.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC4_NOTIFY_CMT

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

Test Case Description: The test case checks if the system sends a notification to users whenever a new comment is added.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

- User is logged in
- A task with comments exists

Test Data: No test data needed

Test Steps:

1. Navigate to the task

2. Add a new comment

3. Check if the system sends a notification

Postconditions:

- Notification for new comment is received

Expected Result: The system sends a notification to the user when a new comment is added to a task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC5_CHRONOLOGICAL_CMT

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

Test Case Description: This test verifies that all comments for a task are displayed in chronological order.

Test Suite: Task Comments and Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- A task with multiple comments exists

Test Data: No test data needed

Test Steps:

1. Navigate to the task
2. View the list of comments
3. Verify that comments are displayed in chronological order

Postconditions:

- Comments are viewed in chronological order

Expected Result: The system successfully displays all comments in chronological order, from the oldest to the newest.

Severity: Minor

Type of Testing: Functional

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 Case 1:

Test Case ID: TMT1

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

Test Case Description: This test case verifies that the software can handle multiple file attachments to a single task.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Multiple files of different types (images, documents, etc.)

Test Steps:

1. Navigate to the desired task
2. Click on the 'Add Attachments' button
3. Select multiple files to attach
4. Click on the 'Upload' button

Postconditions:

- Files are attached to the task

Expected Result: The system attaches and displays the selected files for the respective task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT2

Test Case Title: Verify that the system can handle different file types

Test Case Description: This case verifies the system's capacity to attach and retrieve different file types like documents, images, etc.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Different file types (doc, pdf, jpg, png, etc.)

Test Steps:

1. Navigate to the desired task
2. Click on the 'Add Attachments' button
3. Select a file of a specific type to attach
4. Click on the 'Upload' button

Postconditions:

- File of specific type is attached to the task

Expected Result: The system allows users to attach files of various types to tasks.

Severity: Major

Type of Testing: Compatibility

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT3

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

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

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

- User is logged in
- File is attached to a specific task

Test Data: No test data needed

Test Steps:

1. Navigate to the desired task
2. Click on the attached file
3. Click on the 'Download' button

Postconditions:

- File is downloaded

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

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT4

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

Test Case Description: This test is to check whether the system restricts users from uploading an overly large file.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: A large file (size greater than the maximum allowable size)

Test Steps:

1. Navigate to the desired task
2. Click on the 'Add Attachments' button
3. Select a large file to attach
4. Click on the 'Upload' button

Postconditions:

- System shows error message for large file

Expected Result: The system displays an error message indicating the file is too large to upload.

Severity: Major

Type of Testing: Functional

Test Case Approach: Negative

Test Case 5:

Test Case ID: TMT5

Test Case Title: Verify that users can remove attachments

Test Case Description: This test case validates the functionality that allows users to remove attachments from tasks.

Test Suite: File Attachments

Test Priority: Low

Preconditions:

- User is logged in
- Task is created and files are attached

Test Data: No test data needed

Test Steps:

1. Navigate to the desired task

2. Click on the attached file
3. Click on the 'Remove' button

Postconditions:

- File is removed from the task attachments

Expected Result: The system allows the user to remove the desired attachment.

Severity: Minor

Type of Testing: Functional

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 Case 1:

Test Case ID: tc-rt001

Test Case Title: Verify that users can set tasks to recur at specified intervals (daily, weekly, monthly).

Test Case Description: This test case is designed to ensure that users can set tasks to repeat at specific intervals such as daily, weekly, or monthly.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Task details, Recurrence interval (daily, weekly, monthly)

Test Steps:

1. Go to the selected task
2. Click on the recurrence setting
3. Choose an interval (daily, weekly, monthly)
4. Save changes

Postconditions:

- Task is set to recurring

Expected Result: The system sets the task to recur at the specified interval.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: tc-rt002

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

Test Case Description: This test case checks if the system automatically creates new instances of tasks that are set to recur.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- User is logged in
- Task is set to recurring

Test Data: Recurring task details

Test Steps:

1. Wait for the recurrence interval to pass
2. Check the task list for a new instance of the recurring task

Postconditions:

- New instance of recurring task is generated

Expected Result: The system automatically generates a new instance of the recurring task.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: tc-rt003

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

Test Case Description: This test case ensures users can change the recurrence interval or stop a task from

recurring altogether.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

- User is logged in
- Task is set to recurring

Test Data: Recurring task details and new recurrence interval

Test Steps:

1. Go to the selected recurring task
2. Click on the recurrence setting
3. Modify the recurrence interval or cancel the recurrence
4. Save changes

Postconditions:

- Recurrence pattern is updated or task is no longer recurring

Expected Result: The system updates the recurrence pattern according to the user's modifications or cancels the recurrence.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: tc-rt004

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

Test Case Description: This test case is designed to ensure that the system sends a notification to users about upcoming instances of their recurring tasks.

Test Suite: Recurring Tasks

Test Priority: High

Preconditions:

- User is logged in
- Task is set to recurring

Test Data: No test data needed

Test Steps:

1. Wait for the recurrence interval to nearly pass
2. Check for a notification of the upcoming task

Postconditions:

- User receives notification about upcoming instance of the task

Expected Result: The system sends a notification about the upcoming instance of the recurring task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: tc-rt005

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 view a complete list of all occurrences of a recurring task.

Test Suite: Recurring Tasks

Test Priority: Medium

Preconditions:

- User is logged in
- Task is set to recurring

Test Data: Recurring task details

Test Steps:

1. Go to task list

2. Select the recurring task

3. Click on 'View all instances'

Postconditions:

- User views list of all instances of the recurring task

Expected Result: The system displays a list of all instances of the selected recurring task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 30

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tags and Labels

Feature description -

Tasks can be categorized and labeled using tags, making it easier to organize and filter tasks based on different criteria such as project, priority, or status. Tags provide a flexible way to categorize tasks and customize task views based on users' preferences.

Userstory title -

As a user, I want to use tags and labels to categorize tasks so that I can easily organize and filter them.

Acceptance criteria -

['Users can add tags and labels to tasks.', 'Tags can be customized and edited.', 'Tasks can be filtered and sorted based on tags and labels.', 'The system displays tags and labels prominently in the task list.', 'Users can remove tags and labels if needed.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT001

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

Test Case Description: Test to ensure that users are able to add tags and labels to tasks for better categorization and organisation.

Test Suite: Task Tags and Labels

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Test data consists of task details and the names of tags and labels.

Test Steps:

1. Go to the task
2. Click on 'Add tag/label'
3. Enter the name of the tag/label
4. Save the changes

Postconditions:

- Tag/label is added to the task

Expected Result: The system adds the tag/label to the task successfully.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: TMT002

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

Test Case Description: Test to ensure that users can customize and edit the tags that have been added to tasks.

Test Suite: Task Tags and Labels

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created with a tag

Test Data: Details of the modifications to the tag.

Test Steps:

1. Go to the task
2. Click on the tag
3. Modify the tag as needed
4. Save the changes

Postconditions:

- Tag is updated

Expected Result: The system updates the tag successfully.

Severity: Minor

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: TMT003

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

Test Case Description: Test to ensure that tasks can be filtered and sorted successfully based on their tags and labels.

Test Suite: Task Tags and Labels

Test Priority: High

Preconditions:

- User is logged in
- Tasks are created with tags and labels

Test Data: No test data needed

Test Steps:

1. Open the task list
2. Click on 'Filter'
3. Select to filter or sort by a specific tag or label

Postconditions:

- Task list is filtered or sorted

Expected Result: The system filters or sorts the tasks based on the specified tag or label.

Severity: Critical

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: TMT004

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

Test Case Description: The test focuses on the visibility of tags and labels in the task list.

Test Suite: Task Tags and Labels

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks are created with tags and labels

Test Data: No test data needed

Test Steps:

1. Open the task list
2. Verify that tags and labels of each task are visibly displayed

Postconditions:

- No changes made

Expected Result: All the tags and labels of each task are prominently displayed in the task list.

Severity: Minor

Type of Testing: Usability Testing

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: TMT005

Test Case Title: Verify that users can remove tags and labels from tasks

Test Case Description: Test to ensure that tags and labels can be removed from a task by a user if needed.

Test Suite: Task Tags and Labels

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks are created with tags and labels

Test Data: No test data needed

Test Steps:

1. Go to the task
2. Click on the tag or label
3. Click 'Remove'

Postconditions:

- Tag/label is removed

Expected Result: The system successfully removes the tag/label from the task.

Severity: Minor

Type of Testing: Functional Testing

Test Case Behaviour: 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 Case 1:

Test Case ID: TMT01

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

Test Case Description: To verify that the system allows users to change the status of tasks to different stages such as To Do, In Progress, or Completed.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- Tasks have been created

Test Data: Task status details

Test Steps:

1. Select a task
2. Click on the status button
3. Update the status

Postconditions:

- Task status is updated

Expected Result: The system should successfully update and display the status of the selected tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT02

Test Case Title: Verify that the task status is prominently displayed in the task list

Test Case Description: Testing the system's ability to display the status of tasks well-visibly in the task list.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- Tasks have been created and their status updated

Test Data: No test data needed

Test Steps:

1. Open the task list
2. Check if task status is visible and clear

Postconditions:

- Task status is visible in the task list

Expected Result: The system displays each task's status well-visibly within the task list.

Severity: Major

Type of Testing: UI

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT03

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

Test Case Description: To ensure that the system can calculate and display the task's progress percentage.

Test Suite: Task Status Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks have been created

Test Data: Task progress details

Test Steps:

1. Select a task
2. Check the progress bar for progress percentage of the task

Postconditions:

- Task progress percentage is viewable

Expected Result: The system successfully calculates and displays task progress percentage.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT04

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

Test Case Description: Validate that the system sends out notifications whenever there is a change in the task status.

Test Suite: Task Status Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks have been created

Test Data: Task status details

Test Steps:

1. Select a task
2. Change the task status
3. Check if a notification is sent

Postconditions:

- Notification for task status change is received

Expected Result: The system effectively sends out a notification every time there is a change in the task status.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT05

Test Case Title: Verify that tasks can be filtered and sorted based on their status

Test Case Description: To ensure that the system allows users to filter and sort tasks based on their status.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

- User is logged in
- Tasks have been created and their status updated

Test Data: Task status details

Test Steps:

1. Open the task list
2. Use the filter/sort options to arrange tasks based on status

Postconditions:

- Tasks are filtered/sorted based on status

Expected Result: The system enables filtration and sorting of tasks based on their status.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 32

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Filtering and Sorting

Feature description -

Users can filter and sort tasks based on various criteria such as due date, priority, assignee, or project. Filtering and sorting options help users focus on specific tasks or view tasks from different perspectives, enhancing task management and organization.

Userstory title -

As a user, I want to filter and sort tasks so that I can focus on specific tasks or view tasks from different perspectives.

Acceptance criteria -

['Users can filter tasks based on criteria such as due date, priority, assignee, or project.', 'Tasks can be sorted by various attributes.', 'The system provides multiple filter and sort options in the task list.', 'Users can save and apply custom filter and sort settings.', 'Filters and sorts do not affect the actual data, only the view.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT1

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

Test Case Description: Check that users can filter tasks by due date, priority, assignee or project.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in
- Tasks are created

Test Data: Task details (due date, priority, assignee, project)

Test Steps:

1. Go to the task list
2. Click on the filter option
3. Select a filter criteria (due date, priority, assignee, project)

Postconditions:

- User sees the filtered tasks

Expected Result: The system provides an updated task list based on the selected filter criteria.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT2

Test Case Title: Verify that tasks can be sorted based on various attributes

Test Case Description: Ensure that users can sort their tasks by attributes like due date, priority etc.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in
- Tasks are created

Test Data: Task details

Test Steps:

1. Go to the task list
2. Click on the sort option
3. Choose an attribute to sort by

Postconditions:

- Tasks are sorted

Expected Result: Tasks are sorted according the chosen attribute.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT3

Test Case Title: Verify that the system provides multiple filter and sort options

Test Case Description: Check that multiple filter and sort options are available for task management.

Test Suite: Task Filtering and Sorting

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Go to the task list
2. Click on the filter or sort option

Postconditions:

- Multiple criteria for filter or sort appear

Expected Result: The system provides a list of multiple criteria for filter or sort when selected.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT4

Test Case Title: Verify that users can save and apply custom filter and sort settings

Test Case Description: Ensure that users can save their customized filter and sort settings and apply them later.

Test Suite: Task Filtering and Sorting

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Custom filter and sort settings

Test Steps:

1. Go to the task list
2. Apply filter or sort settings

3. Save the settings

4. Open a new task list and apply saved settings

Postconditions:

- Task list gets updated based on saved settings

Expected Result: The system applies the saved custom filter and sort settings on the new task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT5

Test Case Title: Verify that filters and sorts do not affect the actual data, only the view

Test Case Description: Ensure that implementing filters and sorts do not alter the original task data, but only change their view.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Note the initial task data
2. Apply filter or sort
3. Check the task data

Postconditions:

- Task data remains unchanged

Expected Result: The task data remains the same as the initial one after applying filter or sort.

Severity: Critical

Type of Testing: Functional

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 Case 1:

Test Case ID: TM1

Test Case Title: Verify that users can start and stop a timer for each task.

Test Case Description: To confirm the system allows users to accurately measure time spent on individual tasks by starting and stopping a timer.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- User is logged in
- At least one task has been created

Test Data: Task details

Test Steps:

1. Select a task
2. Start the task's timer
3. Stop the task's timer after some time

Postconditions:

- The task's timer is updated
- The duration between the start and stop is recorded

Expected Result: The timer's start and stop functionality accurately records the time spent on the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TM2

Test Case Title: Verify that the system records and displays total time spent on tasks.

Test Case Description: To confirm the system tracks and shows the total duration a user has spent on their tasks.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- User is logged in
- At least one task with tracked time exists

Test Data: No test data needed

Test Steps:

1. Navigate to the time tracking section
2. Confirm the display of total time spent

Postconditions:

- Total tracked time is shown

Expected Result: The system correctly calculates and displays the total time spent on tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TM3

Test Case Title: Verify that users can manually enter or adjust time entries.

Test Case Description: To ensure the system provides an option for users to manually input or modify time entries for tasks.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- At least one task exists

Test Data: Task details

Test Steps:

1. Select a task
2. Input or adjust the time entry
3. Save changes

Postconditions:

- The task's time entry is updated

Expected Result: The system allows users to manually enter or adjust time entries correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TM4

Test Case Title: Verify that time tracking data is stored and can be exported.

Test Case Description: To confirm the system properly stores time tracking data and allows for data export.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Task with tracked time exists

Test Data: No test data needed

Test Steps:

1. Navigate to the time tracking section
2. Confirm storage of time tracking data
3. Attempt to export the data

Postconditions:

- Time tracking data is exported

Expected Result: The system correctly stores and exports time tracking data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TM5

Test Case Title: Verify that the system provides reports on time spent per task or project.

Test Case Description: To ensure the system generates comprehensive reports reflecting time spent on tasks or projects.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- User is logged in
- At least one task or project with tracked time exists

Test Data: No test data needed

Test Steps:

1. Navigate to the reports section
2. Generate a report for a task or project

Postconditions:

- A report detailing time spent is generated

Expected Result: The system provides detailed reports tracking time spent on tasks or projects.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 34

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Templates

Feature description -

Users can create and use templates for common task types or project workflows, streamlining task creation and standardizing processes across projects. Task templates can include predefined task structures, descriptions, assignees, and due dates, saving time and ensuring consistency in task execution.

Userstory title -

As a user, I want to use task templates so that I can streamline task creation and ensure consistency.

Acceptance criteria -

['Users can create and save task templates.', 'Templates can include predefined task structures, descriptions, assignees, and due dates.', 'Users can apply templates to new tasks or projects.', 'Templates can be edited or deleted as needed.', 'The system displays a list of available templates for selection.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT001

Test Case Title: Verify that users can create and save task templates

Test Case Description: Ensure that the system allows users to create and save task templates for future use.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in
- User is on the task template creation page

Test Data: Task template details

Test Steps:

1. Click on the create task template button
2. Enter template details
3. Click on the save button

Postconditions:

- Task template is saved in the system

Expected Result: The system saves the created task template and can display it in the list of available templates.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT002

Test Case Title: Verify that templates can include predefined task structures, descriptions, assignees, and due

dates

Test Case Description: Ensure that the system allows users to include detailed information in task templates.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in
- User is on the task template creation page

Test Data: Task template details including task structure, description, assignees, and due date

Test Steps:

1. Click on the create task template button
2. Enter all required template details
3. Click on the save button

Postconditions:

- Detailed task template is saved in the system

Expected Result: The system permits the inclusion of all specified details and displays them correctly when viewing the saved template.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT003

Test Case Title: Verify that users can apply templates to new tasks or projects

Test Case Description: Ensure that the system allows users to use saved templates when creating new tasks or projects.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the new task creation page
- Task templates are saved in the system

Test Data: Saved task template

Test Steps:

1. Click on the create new task button
2. Choose a template from the list of saved templates
3. Apply the template

Postconditions:

- New task is created using the applied template

Expected Result: The system applies the chosen template to the new task, auto-filling relevant fields.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT004

Test Case Title: Verify that templates can be edited or deleted as needed

Test Case Description: Ensure that the system allows users to edit or delete task templates.

Test Suite: Task Templates

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the task templates page

Test Data: Saved task template

Test Steps:

1. Select a template from the list of saved templates
2. Click on the edit/delete button
3. Save changes or confirm deletion

Postconditions:

- Selected template is edited or deleted

Expected Result: The system updates or removes the selected template as per user's action.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT005

Test Case Title: Verify that the system displays a list of available templates for selection

Test Case Description: Ensure that the system displays a list of all saved task templates when creating a new task.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- User is logged in
- User is on the new task creation page

Test Data: No test data needed

Test Steps:

1. Click on the create new task button
2. Click on the select template button

Postconditions:

- List of saved templates displayed

Expected Result: The system displays a list of all saved templates for the user to choose from.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 35

Project name -

Task Management Tool

Project description -

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

Feature name -

Calendar Integration

Feature description -

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments alongside other scheduled events and appointments. Calendar integration helps users manage their time effectively and avoid scheduling conflicts.

Userstory title -

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

Acceptance criteria -

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update

automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

Generated Test Cases:

Test Case 1:

Test Case ID: CALI01

Test Case Title: Verify that users can sync tasks with external calendars

Test Case Description: Ensure that the system allows users to sync their tasks with external calendars such as Google Calendar and Outlook.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged into the task management tool
- Tasks have been created
- User has an external calendar account (Google Calendar or Outlook)

Test Data: User credentials for external calendar

Test Steps:

1. Click on "Calendar Integration" in the settings
2. Select the external calendar to sync with
3. Enter external calendar credentials and authorize access

Postconditions:

- Tasks are synced with the selected calendar

Expected Result: The system syncs the tasks with the selected external calendar and updates it when tasks change

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: CALI02

Test Case Title: Verify that the system displays task deadlines in the calendar view

Test Case Description: Ensure that the system displays task deadlines within the user's calendar once the tasks are synced.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged into the task management tool
- Tasks with deadlines have been created and synced with an external calendar

Test Data: No test data needed

Test Steps:

1. Open a synced task in the user's calendar
2. View task details

Postconditions:

- Task deadline is displayed in the calendar event

Expected Result: The system displays the task deadline within the calendar event details

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: CALI03

Test Case Title: Verify that users can set reminders for tasks via the calendar

Test Case Description: Ensure that the system allows users to set reminders for tasks that are synced with a calendar.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged into the task management tool
- Tasks have been created and synced with an external calendar

Test Data: No test data needed

Test Steps:

1. Open a synced task in the user's calendar
2. Set a reminder for the task

Postconditions:

- Reminder is set for the task in the calendar

Expected Result: The system allows setting of reminders for the task via the synced calendar.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: CALI04

Test Case Title: Verify that calendar events update automatically when tasks are changed

Test Case Description: To ensure that any changes to a task's details (due dates, descriptions, etc.) on the app are automatically updated in the sync calendar.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User is logged into the task management tool
- Tasks have been created and synced with an external calendar

Test Data: Updates to task details

Test Steps:

1. Make edits to a task in the task management tool
2. Open the edited task in the synced calendar

Postconditions:

- Changes to the task are reflected in the calendar

Expected Result: The system automatically updates the changes made to a task in the synced calendar.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: CALI05

Test Case Title: Verify that users can manage calendar sync settings within the app

Test Case Description: To check that the system has an option for users to manage their calendar sync settings from within the task management tool.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- User is logged into the task management tool
- Tasks have been created and synced with an external calendar

Test Data: No test data needed

Test Steps:

1. Navigate to "Calendar Integration" in the settings
2. Change sync settings
3. Save changes

Postconditions:

- Changes to sync settings are saved and applied

Expected Result: The system allows users to manage their calendar sync settings.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 36

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Duplication

Feature description -

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

Userstory title -

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

Acceptance criteria -

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT001

Test Case Title: Verify that users can duplicate tasks

Test Case Description: Ensure that users can duplicate an existing task in the task management system.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in
- There exists at least one task

Test Data: Existing task details

Test Steps:

1. Navigate to a task.
2. Click on the duplicate task button.
3. Confirm the task duplication process.

Postconditions:

- A new duplicate task is created

Expected Result: The system creates a new task with the same details as the original task and adds it to the task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT002

Test Case Title: Verify that the duplicated task retains all original details

Test Case Description: Ensure that the duplicated task retains all original details including descriptions, assignees, and deadlines.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

- User is logged in
- A task was duplicated

Test Data: Original task details

Test Steps:

1. Navigate to the duplicated task.
2. Check the details of the duplicated task.

Postconditions:

- No change

Expected Result: The system displays the duplicated task with the same details as the original task.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT003

Test Case Title: Verify that users can modify the duplicated task

Test Case Description: Ensure that users can make changes to the details of the duplicated task.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in
- A task was duplicated

Test Data: Updates to task details

Test Steps:

1. Navigate to the duplicated task.
2. Modify the task details.
3. Save the changes.

Postconditions:

- The duplicated task details are updated

Expected Result: The system saves the changes and displays the updated details of the duplicated task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT004

Test Case Title: Verify that the system clearly indicates that a task has been duplicated

Test Case Description: Ensure that the system provides a clear indication that a task is a duplicate.

Test Suite: Task Duplication

Test Priority: Low

Preconditions:

- User is logged in
- A task was duplicated

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.
2. Find the duplicated task.

Postconditions:

- No change

Expected Result: The system displays an indication (e.g., "duplicate" label) that the task is a duplicate of another task.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT005

Test Case Title: Verify that duplicated tasks are saved and displayed in the task list

Test Case Description: Ensure that duplicated tasks are properly saved and displayed in the overall task list.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in
- A task was duplicated

Test Data: No test data needed

Test Steps:

1. Navigate to the task list.

2. Check for the presence of the duplicated task.

Postconditions:

- No change

Expected Result: The system displays the duplicated task in the task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 37

Project name -

Task Management Tool

Project description -

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

Feature name -

Task History

Feature description -

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating accountability, and transparency in task management.

Userstory title -

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

Acceptance criteria -

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and

other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides timestamps for all changes.', 'Task history is accessible from the task details page.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT01

Test Case Title: Verify that users can view a detailed history of changes to tasks

Test Case Description: Ensure the system allows users to view a detailed history of changes, modifications, comments, and other activity related to a task.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in
- The task has at least one recorded history

Test Data: TaskID

Test Steps:

1. Navigate to Task Details page
2. Enter valid TaskID
3. Open "History" tab

Postconditions:

- History display shows all changes, modifications, comments, and activity related to the task

Expected Result: The system displays the history of changes for the specified task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT02

Test Case Title: Verify that the history includes modifications, comments, and other relevant activity

Test Case Description: Ensure the task history feature records and displays all types of changes like modifications, comments, and any task-related activities.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in
- Task has history of changes

Test Data: TaskID

Test Steps:

1. Navigate to Task Details page
2. Enter valid TaskID
3. Open "History" tab
4. Ascertain all types of changes have been recorded and displayed in the history

Postconditions:

- Display showing all recorded changes of the task

Expected Result: The system accurately displays all the changes, including modifications and comments, made in the task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT03

Test Case Title: Verify that users can filter the history by type of change or date

Test Case Description: Ensure the system provides an option to filter the task history by type of change or date.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in
- Task history is available

Test Data: Filter criteria: type of change, date

Test Steps:

1. Navigate to Task Details page
2. Open "History" tab
3. Apply filters by type of change or date

Postconditions:

- Task history is filtered based on the selected criteria

Expected Result: The system applies the filter and displays the filtered results in the task history.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT04

Test Case Title: Verify that the system provides timestamps for all changes in the task history

Test Case Description: Ensure that every task update or change recorded in the history is accompanied by a timestamp indicating when the change was made.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- User is logged in
- Task history is available

Test Data: No test data needed

Test Steps:

1. Navigate to Task Details page
2. Open "History" tab
3. Review the changes and confirm the presence of timestamps

Postconditions:

- Every change recorded in the task history has a timestamp

Expected Result: The system displays timestamps for all changes in the task history.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT05

Test Case Title: Verify that Task History is accessible from the Task Details page

Test Case Description: Ensure that the Task History can be accessed from the Task Details page.

Test Suite: Task History

Test Priority: High

Preconditions:

- User is logged in
- Task history for a certain task is available

Test Data: TaskID

Test Steps:

1. Navigate to Task Details page of a certain task
2. Look for "History" tab or link
3. Click to open Task History

Postconditions:

- Task History page is opened and visible

Expected Result: The system opens the Task History on the Task Details page.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Data - 38

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

Userstory title -

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

Acceptance criteria -

['Users receive notifications for task updates, comments, mentions, and other relevant activities.',

'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize notification settings based on preferences.', 'Notifications provide detailed information about the activity.',

'The system logs notification history for future reference.']

Generated Test Cases:

Test Case 1:

Test Case ID: TMT1

Test Case Title: Verify that the system sends notifications for task updates, comments, and mentions

Test Case Description: Ensure that the system sends notifications when there are task updates, comments from team members, and user mentions in the task activities.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- User is associated with a task

Test Data: Task updates/comments, User credentials

Test Steps:

1. Update a task associated with the User
2. Comment on a task associated with the User
3. Mention the User in a task activity

Postconditions:

- User receives notifications for updates, comments, and mentions

Expected Result: The notification is sent successfully to the user

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT2

Test Case Title: Verify that notifications can be delivered via email, push notifications, and in-app alerts

Test Case Description: Check the system's capability to deliver notifications through multiple channels - email, mobile push notifications, and in-app alerts.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- User is associated with a task
- Notification settings are turned on for the User

Test Data: User credentials, Email data

Test Steps:

1. Update a task associated with the User
2. Check if the User receives notification via an email, mobile, and in-app alert

Postconditions:

- Notifications are sent through multiple channels

Expected Result: Notifications have been sent to the user via all specified channels - Email, mobile, and in-app alerts

Severity: Major

Type of Testing: Integration

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT3

Test Case Title: Verify that users can customize notification settings based on preferences

Test Case Description: Ensure that the system allows users to customize notification settings according to their needs.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: User preferences for notifications

Test Steps:

1. Go to the user settings page
2. Customize notification settings
3. Save changes

Postconditions:

- User notification preferences saved

Expected Result: The system allows the User to customize notification settings successfully.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT4

Test Case Title: Verify that notifications provide detailed information about the activity

Test Case Description: The notification system should be able to send detailed information about task updates, comments from team members and user mentions

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- User is associated with a task

Test Data: Task updates/comments, User credentials

Test Steps:

1. Update a task associated with the User
2. Comment on a task associated with the User
3. Check the detail provided in the notifications

Postconditions:

- Notifications sent with detailed information

Expected Result: Notifications are sent with detailed information about the activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT5

Test Case Title: Verify that system logs notification history for future reference

Test Case Description: The system is expected to log all notifications that it sends to the user for future reference.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in
- User is associated with a task

Test Data: Task updates/comments, User credentials

Test Steps:

1. Update a task associated with the User
2. Check if the system logs the notification that was sent

Postconditions:

- Notification is logged

Expected Result: The system logs all the notifications for future reference.

Severity: Major

Type of Testing: System

Test Case Approach: Positive

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 Case 1:

Test Case ID: TMT1

Test Case Title: Verify that users can archive completed or inactive tasks

Test Case Description: Users should have the ability to archive tasks that have been completed or are inactive.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- User is logged in
- Task has been created

Test Data: Task details

Test Steps:

1. Navigate to the task list
2. Select a completed or inactive task
3. Click on the "Archive" button

Postconditions:

- Task is moved to the archive

Expected Result: Task is successfully moved from the active task list to the archived task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT2

Test Case Title: Verify that archived tasks are moved to a separate section for easy access

Test Case Description: After being archived, tasks should be moved to a separate, easily accessible section.

Test Suite: Task Archiving

Test Priority: High

Preconditions:

- Tasks have been archived

Test Data: Archived task details

Test Steps:

1. Navigate to the archive section
2. Verify that the recently archived task is present

Postconditions:

- Archived task is found in the designated section

Expected Result: Task appears successfully in the archive section after it has been archived.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT3

Test Case Title: Verify that users can search and view archived tasks

Test Case Description: Users should be able to easily search for and view any tasks that have been archived.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:

- Tasks have been archived

Test Data: Archived task details

Test Steps:

1. Navigate to the archive section
2. Enter task details in search bar
3. Verify that the task is shown in the search results

Postconditions:

- Search results show archived tasks

Expected Result: The system effectively locates and displays the desired archived task based on the details entered in the search bar.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT4

Test Case Title: Verify that the system provides an option to restore archived tasks to the active list

Test Case Description: Users should be able to restore an archived task back to the active task list.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:

- Tasks have been archived

Test Data: Archived task details

Test Steps:

1. Navigate to the archive section
2. Select an archived task
3. Click on the "Restore Task" button

Postconditions:

- Archived task is moved back to the active task list

Expected Result: The task is successfully restored to the active task list from the archived task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT5

Test Case Title: Verify that archived tasks retain all original details and history

Test Case Description: To ensure all necessary data is preserved, archived tasks should retain all of the original details and history.

Test Suite: Task Archiving

Test Priority: Low

Preconditions:

- Tasks have been archived

Test Data: No test data needed

Test Steps:

1. Navigate to the archive section
2. Select an archived task
3. View the tasks details and history

Postconditions:

- Archived task maintains its original details and history

Expected Result: All task details and history are still intact after the task has been archived.

Severity: Minor

Type of Testing: Functional

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 Case 1:

Test Case ID: TMTGA001

Test Case Title: Verify that users can invite external collaborators as guests with limited access

Test Case Description: This test case ensures that the user can invite external collaborators, granting them guest access with limited features within the platform.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in
- Task/projects are created

Test Data: External collaborator's email, Access level parameters

Test Steps:

1. Go to desired task or project
2. Click on the "Add Guest" button
3. Enter the email address of the external collaborator
4. Select the access level
5. Click on "Invite" button

Postconditions:

- Guest invitation is sent
- Access level is set for the external collaborator

Expected Result: System successfully invites the external collaborator as a guest with limited access and shows the status as pending until the collaborator accepts the invitation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMTGA002

Test Case Title: Verify that guests can view and collaborate on specific tasks or projects

Test Case Description: This test case is to verify that the guest users can view and collaborate on the specific tasks or projects assigned to them.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- Guest user has accepted the invitation

Test Data: No test data needed

Test Steps:

1. Login as a Guest User
2. Check the received tasks/projects
3. View and comment on the assigned tasks/projects

Postconditions:

- Changes/comments made by the guest user are recorded

Expected Result: The system provides access to assigned tasks/projects and allows guest users to view and collaborate on these tasks/projects.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMTGA003

Test Case Title: Verify distinctions between guest and full access roles

Test Case Description: This test case is designed to test whether the system provides clear distinctions between guest and full access roles.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- Guest user and full access user are logged in separately

Test Data: No test data needed

Test Steps:

1. Login as a Guest User
2. Browse through different sections of the platform
3. Repeat steps with full access user
4. Compare the level of access between the two types of users

Postconditions:

- Validation of distinct roles

Expected Result: The system must provide different levels of access to the guest users and full access users.

Severity: Major

Type of Testing: Security

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMTGA004

Test Case Title: Verify that users can manage and revoke guest access at any time

Test Case Description: This test case is to ensure that full access users have the ability to manage and revoke guest access at any time.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- User is logged in
- Guest access is already assigned

Test Data: Guest user's details

Test Steps:

1. Go to the project where the guest was added
2. Click on "Manage Access" in the settings
3. To manage permissions, adjust the permission settings
4. To revoke access, click on "Revoke Access"

Postconditions:

- Guest access is managed or revoked

Expected Result: User is able to manage or revoke the guest's access successfully.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMTGA005

Test Case Title: Verify that guests cannot access sensitive or restricted areas of the platform

Test Case Description: This test case checks that guests cannot access sensitive or restricted areas of the platform they were not granted access to.

Test Suite: Guest Access

Test Priority: High

Preconditions:

- Guest user has accepted the invitation

Test Data: No test data needed

Test Steps:

1. Login as a Guest User
2. Attempt to access restricted areas

Postconditions:

- Validation of access restrictions

Expected Result: The system restricts the guest from accessing sensitive or restricted areas of the platform.

Severity: Critical

Type of Testing: Security

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 Case 1:

Test Case ID: TMT001

Test Case Title: Verify that users can import tasks from external sources in various formats

Test Case Description: To ensure that users are able to import tasks into the tool from a variety of external sources in different formats such as CSV and Excel.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

- User is logged in
- There's an existing file in CSV or Excel formats with task data

Test Data: CSV and Excel task data files

Test Steps:

1. Navigate to the import tasks option
2. Choose the appropriate file

3. Start the import process
4. Check if the tasks from the file are successfully imported

Postconditions:

- Imported tasks are visible in the user's task list

Expected Result: All tasks from the imported file are successfully added to the tasks list without any data loss.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TMT002

Test Case Title: Verify that users can export tasks to other task management tools or file formats

Test Case Description: To ensure that users are able to export their tasks to different task management tools or in different file formats such as CSV and Excel.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

- User is logged in
- There are existing tasks in the user's list

Test Data: No test data needed

Test Steps:

1. Navigate to the export tasks option
2. Choose the preferred format for export
3. Start the export process
4. Check the exported file for correctness of data

Postconditions:

- Exported file is available for download

Expected Result: All the tasks are exported successfully in the selected file format without losing any data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TMT003

Test Case Title: Verify that the system provides a clear mapping of fields during import/export

Test Case Description: To ensure that the system provides a clear mapping of fields while importing or exporting tasks, facilitating correct and efficient data migration.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

- User is logged in
- There's an existing file in CSV or Excel formats with task data

Test Data: CSV and Excel task data files

Test Steps:

1. Navigate to the import/export tasks option
2. Choose the appropriate file or format
3. Check the mapped fields before starting the import/export process

Postconditions:

- Tasks are imported/exported based on field mappings

Expected Result: All fields are mapped correctly, allowing efficient data migration during import/export.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TMT004

Test Case Title: Verify that users can preview imported data before finalizing the import process

Test Case Description: To ensure users can view and validate the imported data before final confirmation, preventing any incorrect data import.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

- User is logged in
- There's an existing file in CSV or Excel formats with task data

Test Data: CSV and Excel task data files

Test Steps:

1. Navigate to the import tasks option
2. Choose the appropriate file
3. Preview the data being imported
4. Confirm or cancel the import based on the previewed data

Postconditions:

- Imported tasks are visible in the user's task list if the import process was confirmed

Expected Result: User can preview and validate the imported data before committing to the import process.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TMT005

Test Case Title: Verify that the system logs import/export activities for future reference

Test Case Description: To ensure the system effectively logs all import/export activities for user reference, aiding in troubleshooting and maintaining data transparency.

Test Suite: Task Import/Export

Test Priority: Low

Preconditions:

- User is logged in
- There's an existing file in CSV or Excel formats with task data

Test Data: CSV and Excel task data files

Test Steps:

1. Navigate to the import/export tasks option
2. Choose the appropriate file or format
3. Start the import/export process
4. Check the system logs for the respective import/export activity

Postconditions:

- The log of the respective import/export activity is available in the system logs

Expected Result: The system logs denote details of the import/export activity.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.

Userstory title -

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

Acceptance criteria -

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.', 'Priority changes are logged and tracked.']

Generated Test Cases:

Test Case 1:

Test Case ID: TPM01

Test Case Title: Verify that users can set and adjust task priorities

Test Case Description: This test case ensures that users can set and adjust priorities for the tasks they create.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Task has been created

Test Data: Task information and prioritization details

Test Steps:

1. Log into the task management tool
2. Open an existing task
3. Set a priority level
4. Save changes

Postconditions:

- The priority level for the task is set

Expected Result: The task's priority level is updated successfully, and the changes are visible.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TPM02

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

Test Case Description: This test case checks if tasks can be sorted and filtered based on priority levels.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

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

Test Data: Tasks with different priority levels

Test Steps:

1. Log into the task management tool
2. Go to tasks page
3. Implement a sort/filter based on priority levels.

Postconditions:

- Tasks are filtered/sorted based on priorities

Expected Result: Tasks should be sorted/filtered correctly based on priority level.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TPM03

Test Case Title: Verify that the system provides visual indicators of task priority

Test Case Description: This test case will validate the visual indicators that represent different priority levels for tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

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

Test Data: No test data needed

Test Steps:

1. Log into the task management tool
2. View tasks

Postconditions:

- Visual indicator of task priority is visible

Expected Result: Each task's priority is displayed with a distinct visual indicator.

Severity: Major

Type of Testing: GUI Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: TPM04

Test Case Title: Verify that users receive notifications for high-priority tasks

Test Case Description: This test case checks that the system sends notifications/alarms when high-priority tasks are due soon.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- High-priority task set with upcoming deadline

Test Data: High priority task information

Test Steps:

1. Log into the task management tool
2. Wait for the notification time

Postconditions:

- User receives a notification for high-priority task

Expected Result: Notification for high-priority task is received.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TPM05

Test Case Title: Verify that priority changes are logged and tracked

Test Case Description: This test ensures that changes made to the priority levels of tasks are tracked and logged.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Task with adjusted priority level

Test Data: Task information and priority change log

Test Steps:

1. Log into the task management tool
2. Set a new priority level for a task
3. View the change log for the task

Postconditions:

- Priority changes are logged

Expected Result: Priority level changes are logged and visible.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and communicating through comments or messages. Collaboration features aim to enhance teamwork and coordination on tasks and projects.

Userstory title -

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay updated on task progress.

Acceptance criteria -

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications for task updates and comments.', 'The system provides a clear view of task collaboration history.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC01

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

Test Case Description: Ensure that the system allows users to successfully assign tasks to team members, enhancing task collaboration and distribution.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Task is created
- Team members are added

Test Data: Task details, Team member names

Test Steps:

1. Navigate to the created task
2. Click on the 'Assign' button
3. Select a team member from the drop-down
4. Click on 'Confirm Assignment'

Postconditions:

- Task is assigned to the selected team member

Expected Result: The system assigns the task to the selected team member and updates the task details accordingly.

Severity: Major

Test Type: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: TC02

Test Case Title: Verify that users can share updates and communicate through comments or messages

Test Case Description: Ensure that the system supports communication and update sharing through comments or messages on tasks.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Task is assigned to a team member

Test Data: Comment or Message details

Test Steps:

1. Navigate to the relevant task
2. Enter the comment or message in the 'Add comment or message' field
3. Click on the 'Add' button

Postconditions:

- Comment or message is added to the task

Expected Result: The system successfully adds the entered comment or message to the task and reflects it for all team members assigned to that task.

Severity: Major

Test Type: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: TC03

Test Case Title: Verify that task updates and comments are visible to assigned team members

Test Case Description: Ensure that any updates or comments made on a task are visible to all team members assigned to the task.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- User is logged in

- Task is assigned to a team member
- Update or comment is added to the task

Test Data: No test data needed

Test Steps:

1. Navigate to the relevant task
2. View the updates and comments section

Postconditions:

- Updates and comments on the task are visible

Expected Result: The system shows all updates and comments related to the task to all assigned team members.

Severity: Major

Test Type: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: TC04

Test Case Title: Verify that users receive notifications for task updates and comments

Test Case Description: Ensure that the system sends out notifications to team members when there are updates or comments on tasks they are assigned to.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Task is assigned to a team member
- Update or comment is added on the task

Test Data: No test data needed

Test Steps:

1. Wait for the system to send out notifications
2. Click on the 'Notifications' tab
3. Verify received notifications related to task updates or comments

Postconditions:

- Notification received for the task update or comment

Expected Result: The system sends a notification whenever there is a comment or update on tasks a user is assigned to, and when viewed, these notifications correctly refer to the latest task updates or comments.

Severity: Major

Test Type: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: TC05

Test Case Title: Verify the system provides a clear view of task collaboration history

Test Case Description: Ensure that the system keeps track of and displays a clear view of all historical actions related to task collaboration, such as task assignments, updates, comments, and so on.

Test Suite: Task Collaboration

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks are created and assigned to team members
- Updates or comments have been added to tasks

Test Data: No test data needed

Test Steps:

1. Navigate to the relevant task

2. Click on 'View Collaboration History'

Postconditions:

- Collaboration history for the task is visible

Expected Result: The system provides a detailed view of the task collaboration history with all related actions and timeline.

Severity: Major

Test Type: Functional

Test Case Behaviour: Positive

Data - 44

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Tracking

Feature description -

Users can track the progress of tasks, view task statuses, and monitor deadlines. The task tracking feature aims to provide visibility into task progress and ensure timely completion of tasks.

Userstory title -

As a user, I want to track the progress of tasks so that I can monitor their status and ensure timely completion.

Acceptance criteria -

['Users can view task statuses.', 'Users can track task progress and deadlines.', 'Tasks are updated in real-time.',

'System provides visual indicators for task progress.', "]

Generated Test Cases:

Test Case 1:

Test Case ID: TCT1

Test Case Title: Verify that users can view task statuses

Test Case Description: This test case validates that the system allows the user to view the status of tasks in the task management system.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in
- Tasks are created in the system

Test Data: Task details

Test Steps:

1. Go to the task tracking page
2. Click on a specific task
3. Check the status of the task

Postconditions:

- Task status is visible

Expected Result: The system displays the current status of the selected task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TCT2

Test Case Title: Verify that users can track task progress and deadlines

Test Case Description: This case ensures that the system allows users to track the progress and deadlines of tasks effectively.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in
- Task with a deadline is created

Test Data: Task details including deadline

Test Steps:

1. Navigate to the task tracking page
2. Select a task with a deadline
3. Review task progress and deadline information

Postconditions:

- Task progress and deadline information is displayed

Expected Result: The system displays the progress and the remaining time until the deadline of the selected task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TCT3

Test Case Title: Verify that tasks are updated in real time

Test Case Description: Checks that the system updates the status and progress of tasks in real time as changes are made.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in
- Existing task

Test Data: Updated task details

Test Steps:

1. Select an existing task
2. Make updates to the task
3. Check for real-time updates on task tracking page

Postconditions:

- Task updates are visible in real-time

Expected Result: The system displays the changes made to the task in real time on the task tracking page.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TCT4

Test Case Title: Verify that the system provides visual indicators for task progress

Test Case Description: Ensures that the system provides visual indicators to represent task progress.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is logged in
- Task with a progress level assigned

Test Data: Task details

Test Steps:

1. Navigate to task tracking page
2. Select a task
3. Review visual indicators of task progress

Postconditions:

- Task progress visual indicators are visible

Expected Result: The system presents visual indicators reflecting the current progress of the selected task.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TCT5

Test Case Title: Verify that the task tracking page does not show tasks for unauthenticated users

Test Case Description: Ensures that only authenticated users can view and track tasks in the task management system.

Test Suite: Task Tracking

Test Priority: High

Preconditions:

- User is not logged in

Test Data: No test data needed

Test Steps:

1. Navigate to task tracking page

2. Attempt to view tasks

Postconditions:

- Unauthenticated user is prompted to log in

Expected Result: The system prevents unauthenticated users from accessing the task tracking page and prompts them to log in.

Severity: Major

Type of Testing: Security

Test Case Approach: Negative

Data - 45

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming deadlines. The notifications aim to keep users informed and engaged with their tasks.

Userstory title -

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my tasks effectively.

Acceptance criteria -

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.',
'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', "']

Generated Test Cases:

Test Case 1:

Test Case ID: NT1

Test Case Title: Verify that users receive notifications for new tasks.

Test Case Description: Ensure that users receive notifications when a new task is assigned to them, keeping them informed and engaged with work.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has been assigned a task

Test Data: Task details (Task ID, Task Name, Task Description)

Test Steps:

1. Log into the system
2. Assign a task to a user
3. Check if the designated user receives a notification for the new task

Postconditions:

- Notification for the new task is sent

Expected Result: The system sends a notification to the user when a new task is assigned to them.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: NT2

Test Case Title: Verify that users receive notifications for task updates and comments

Test Case Description: Ensure that the system immediately notifies the users about updates and comments on tasks assigned to them.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has been assigned a task

Test Data: Task updates and comments

Test Steps:

1. Login to system
2. Perform an update or leave a comment on a user assigned task
3. Check if the designated user receives a notification for the task update or comment

Postconditions:

- Notification for the task update or comment is sent

Expected Result: The system notifies the assigned user immediately about task updates or comments.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: NT3

Test Case Title: Verify that users receive reminders for upcoming deadlines.

Test Case Description: Ensure that the system sends reminders to users for upcoming task deadlines to help manage task completion effectively.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in
- User has been assigned a task with a deadline

Test Data: Task deadlines

Test Steps:

1. Login to system
2. Assign a task to a user with a near deadline
3. Validate if reminder notification for task deadline is sent to the user

Postconditions:

- User receives a reminder notification for task deadline

Expected Result: The system sends a reminder notification to the user for the upcoming task deadline.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: NT4

Test Case Title: Verify that notification settings can be customized by users.

Test Case Description: Ensure that a user can customize their notification settings, allowing them to control the number and types of notifications they receive.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Log in to system
2. Navigate to notification settings
3. Customize the notification settings as per user preference
4. Validate if the changes have been saved and applied

Postconditions:

- User's changes to notification settings are saved and applied

Expected Result: The system allows the user to customize their notification settings successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: NT5

Test Case Title: Verify that users do not receive notifications for tasks when notifications are turned off.

Test Case Description: Verify that no task notifications are received by the user when the notification settings are turned off as per user preference.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in
- User has switched off notifications in the notification settings

Test Data: Task details

Test Steps:

1. Log in to system
2. Navigate to notification settings
3. Switch off all or specific task notifications
4. Assign a task to the user
5. Validate that no notification is received for the assigned task

Postconditions:

- No notification is received for the task assigned

Expected Result: The system adheres to the user's preference and does not send notifications when they are switched off in the notification settings.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: TC001

Test Case Title: Verify that users can create and assign tasks to different categories or projects.

Test Case Description: Validate the ability of users to create tasks and assign them to specific categories or projects for efficient organization and management.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in
- Categories/projects are created

Test Data: Task information

Test Steps:

1. Click on "Create New Task"
2. Enter necessary task details
3. Select a category or project to assign the task

4. Save the task

Postconditions:

- New task is created
- New task is assigned to a specific category or project

Expected Result: The new task should be created and displayed under the assigned category or project.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC002

Test Case Title: Verify that users can view tasks grouped by categories or projects.

Test Case Description: This test case checks the ability of users to view tasks grouped under specific categories or projects.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in
- Tasks exist in one or more categories or projects

Test Data: No test data needed

Test Steps:

1. Navigate to the tasks section
2. Select a specific category or project
3. View the tasks listed under the selected category or project

Postconditions:

- Tasks grouped by selected category or project are displayed

Expected Result: The system should list all tasks under the selected category or project.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: TC003

Test Case Title: Verify that tasks can be moved between categories or projects.

Test Case Description: Check if users can move tasks from one category or project to another.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in
- Multiple tasks exist in various categories or projects

Test Data: Task and category/project details

Test Steps:

1. Select a task from a category/project
2. Choose a different category/project
3. Move the task to the selected category/project
4. Save changes

Postconditions:

- Task is moved to another category/project

Expected Result: The task should be successfully moved to the newly selected category or project and should no longer appear under the original category/project.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: TC004

Test Case Title: Verify that category or project information is accurately reflected in task views.

Test Case Description: Confirm that the task view accurately displays the assigned category or project information.

Test Suite: Task Categorization

Test Priority: High

Preconditions:

- User is logged in
- Tasks are assigned to categories or projects

Test Data: Task details

Test Steps:

1. Click on a task to view details
2. Check the category/project information

Postconditions:

- Category or project information is displayed accurately

Expected Result: The task view should correctly show the information of the category or project it is assigned to.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: TC005

Test Case Title: Verify that users cannot assign a task to non-existing categories or projects.

Test Case Description: Test whether the application prevents users from assigning tasks to categories or projects that don't exist.

Test Suite: Task Categorization

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created

Test Data: Task information

Test Steps:

1. Click on "Assign to category/project"
2. Enter a non-existing category or project name
3. Click OK

Postconditions:

- Error message is displayed

Expected Result: The application should display an error message indicating that the category or project does not exist.

Severity: Minor

Type of Testing: Negative

Test Case Approach: Negative

Data - 47

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Search

Feature description -

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due date. The task search feature aims to help users quickly find specific tasks within their project or task list.

Userstory title -

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within my project or task list.

Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search results by category, priority, or due date.', 'Search results are accurate and relevant.', '']

Generated Test Cases:

Test Case 1:

Test Case ID: T01

Test Case Title: Verify that users can search for tasks by name

Test Case Description: This test case is designed to ensure that the task search feature allows users to find specific tasks based on their names accurately.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in

- At least one task has been created

Test Data: Task names

Test Steps:

1. Navigate to the task search feature
2. Enter the name of a task
3. Submit the search

Postconditions:

- Search results are displayed

Expected Result: The system returns the correct task based on the provided task name.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: T02

Test Case Title: Verify that users can search for tasks by description

Test Case Description: This test is meant to validate whether the user can search tasks using the task's description effectively.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- User is logged in
- At least one task with a distinct description is created

Test Data: Task descriptions

Test Steps:

1. Go to the task search feature
2. Input a known task description
3. Submit the search

Postconditions:

- Search results are displayed

Expected Result: The system accurately displays the task(s) that match(es) the searched description.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: T03

Test Case Title: Verify that users can filter search results by category, priority or due date

Test Case Description: This test case is used to ensure that the task management tool allows users to filter task search results by category, priority, or due date.

Test Suite: Task Search

Test Priority: Medium

Preconditions:

- User is logged in
- Tasks have been created and assigned a category, priority, and due date

Test Data: Category, priority, due date

Test Steps:

1. Perform a task search
2. Apply any or all filters (category, priority, due date)
3. Review the search results

Postconditions:

- Search results are displayed

Expected Result: The system filters the task search results accurately based on the selected category, priority, or due date.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: T04

Test Case Title: Verify that search results are accurate and relevant

Test Case Description: This test case ensures that the search results for tasks are accurate, relevant, and return the appropriate tasks for the provided search criteria.

Test Suite: Task Search

Test Priority: High

Preconditions:

- User is logged in
- At least one task has been created

Test Data: Task names, descriptions, category, priority, due date

Test Steps:

1. Conduct a task search using different search criteria
2. Review the search results

Postconditions:

- Search results are displayed

Expected Result: The system displays search results that are accurate and relevant to the search criteria.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: T05

Test Case Title: Verify that user receives an appropriate message when no tasks match the search criteria

Test Case Description: This test case is to check if the system provides an appropriate message when there are no tasks matching the user's search criteria.

Test Suite: Task Search

Test Priority: Low

Preconditions:

- User is logged in

Test Data: Nonexistent task details

Test Steps:

1. Go to the task search feature
2. Enter the details of a non-existing task
3. Submit the search

Postconditions:

- Search results are displayed

Expected Result: The system informs the user that there are no tasks matching the search criteria.

Severity: Minor

Type of Testing: Functional

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 Case 1:

Test Case ID: TC01_Task_Collaboration

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

Test Case Description: This test case checks if users can successfully add comments onto tasks enhancing

communication and productivity.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Comments Input

Test Steps:

1. Navigate to the specific task
2. Use the comment option to add a comment
3. Save the changes

Postconditions:

- Comment is added to the task

Expected Result: Users can successfully add comments to tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC02_Task_Collaboration

Test Case Title: Verify that users can attach files to tasks

Test Case Description: This test case determines if users can successfully attach files to tasks.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in

- Task is created

Test Data: File to be attached

Test Steps:

1. Navigate to a specific task
2. Use the attachment option to attach a file
3. Save the changes

Postconditions:

- File is attached to the task

Expected Result: Users can successfully attach files to tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC03_Task_Collaboration

Test Case Title: Verify that users can share tasks with other users

Test Case Description: This test case checks if users can successfully share tasks with other users.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: No test data needed

Test Steps:

1. Select a specific task
2. Use the sharing option to share the task with another user

3. Save the changes

Postconditions:

- Task is shared with selected user

Expected Result: Users can successfully share tasks with other users.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC04_Task_Collaboration

Test Case Title: Verify that the task collaboration features work seamlessly

Test Case Description: This test case checks if all the task collaboration features including adding comments, attaching files and sharing tasks work seamlessly together.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Comments Input, File to be attached

Test Steps:

1. Navigate to a specific task
2. Use commenting, attaching, and sharing features
3. Save all changes

Postconditions:

- All task collaboration features are functioning as expected

Expected Result: All task collaboration features work seamlessly together enhancing the overall user

experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC05_Task_Collaboration

Test Case Title: Verify that the task collaboration features improve team communication

Test Case Description: This test case checks if the task collaboration features including adding comments, attaching files and sharing tasks effectively improve team communication.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: Comments Input, File to be attached

Test Steps:

1. Navigate to a specific task
2. Use commenting, attaching, and sharing features
3. Verify if these features have improved the communication within the team

Postconditions:

- All task collaboration features are functioning as expected

Expected Result: Task collaboration features effectively enhance team communication.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: CCT-001

Test Case Title: Verify that instructors can create courses using videos, presentations, quizzes, and

assignments

Test Case Description: This test case focuses on ensuring the system allows instructors to use a variety of multimedia elements when creating a course.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- Instructor account is created
- Instructor is logged in

Test Data: Sample educational resources (videos, presentations, quizzes, and assignments)

Test Steps:

1. Choose the option to create a course
2. Input course details
3. Upload videos, presentations, quizzes, and assignments
4. Save the course

Postconditions:

- Course is created with all the necessary educational resources

Expected Result: The system successfully creates a course that includes videos, presentations, quizzes, and assignments.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: CCT-002

Test Case Title: Verify that the tools are user-friendly and require minimal technical skills

Test Case Description: This test case validates the usability and simplicity of the course creation tools.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- Instructor account is created
- Instructor is logged in

Test Data: No test data needed

Test Steps:

1. Navigate to the course creation page
2. Interact with course creation tools
3. Evaluate ease of use and user-friendliness

Postconditions:

- Instructor successfully interacts with the course creation tools

Expected Result: The tools required for course creation are user-friendly and do not demand sophisticated technical skills.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case 3:

Test Case ID: CCT-003

Test Case Title: Verify that courses can be previewed before publishing

Test Case Description: This test case ensures that a 'preview' feature is available and functional where instructors can check the course before it is published.

Test Suite: Course Creation Tools

Test Priority: Medium

Preconditions:

- Instructor account is created
- Instructor is logged in
- Draft course is ready

Test Data: Pre-drafted course details

Test Steps:

1. Navigate to the draft course
2. Click on the preview button
3. Assess the course preview for its correctness

Postconditions:

- Course is previewed

Expected Result: The system successfully allows the instructor to preview the course before it is published.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: CCT-004

Test Case Title: Verify that instructors can save courses as drafts

Test Case Description: This test case verifies the ability of instructors to save the course they are building as a draft.

Test Suite: Course Creation Tools

Test Priority: Medium

Preconditions:

- Instructor account is created
- Instructor is logged in

Test Data: Sample course details

Test Steps:

1. Go to course creation page
2. Input course details
3. Select 'save as draft' option

Postconditions:

- Course is saved as draft

Expected Result: The system successfully enables the instructor to save the course as a draft.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: CCT-005

Test Case Title: Verify that courses can be edited post-publishing

Test Case Description: This test case checks if an instructor can make changes to a course after it has been published.

Test Suite: Course Creation Tools

Test Priority: Medium

Preconditions:

- Instructor account is created
- Instructor is logged in
- Course has been published

Test Data: Updated course details

Test Steps:

1. Navigate to the published course
2. Select course edit option

3. Make necessary changes and save

Postconditions:

- Course is updated

Expected Result: The system successfully allows the instructor to edit a course that has been published.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: IL001

Test Case Title: Verify that lessons include quizzes, polls, simulations, and discussions

Test Case Description: This test case is to ensure that the interactive lessons include the components such as quizzes, polls, simulations, and discussions that enhance the learning experience.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is registered
- Course or tutorial selected

Test Data: Interactive lesson content including quizzes, polls, simulations, and discussions

Test Steps:

1. Log in as a learner
2. Select a course or tutorial
3. Access a lesson
4. Check for quizzes, polls, simulations, and discussions in the lesson

Postconditions:

- Lesson opened

Expected Result: The interactive lesson includes quizzes, polls, simulations, and discussions

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: IL002

Test Case Title: Verify that interactive elements are integrated seamlessly into the lesson content

Test Case Description: This test case checks whether the interactive elements are well-integrated into the lesson content without interrupting the learning flow.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is registered
- Course or tutorial selected

Test Data: Interactive lesson content

Test Steps:

1. Log in as a learner
2. Select a course or tutorial
3. Access a lesson
4. Validate the integration of interactive elements

Postconditions:

- Lesson opened

Expected Result: Interactive elements are well-integrated into the lesson content

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: IL003

Test Case Title: Verify that learners can track their progress through activities

Test Case Description: This test case ensures that learners can track their progress through interactive elements in the lessons.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is registered
- Course or tutorial selected

Test Data: Learner's progress data

Test Steps:

1. Log in as a learner
2. Select a course or tutorial
3. Access a lesson
4. Participate in interactive activities
5. Check progress tracking functionality

Postconditions:

- Progress tracked

Expected Result: The system tracks and updates progress as learner interacts with lesson components

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: IL004

Test Case Title: Verify that feedback is provided after each interactive component

Test Case Description: This test case is designed to confirm that performance feedback is provided after each interactive element in the lesson.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is registered
- Course or tutorial selected

Test Data: Learner interaction and feedback data

Test Steps:

1. Log in as a learner
2. Select a course or tutorial
3. Access a lesson
4. Participate in interactive activities
5. Check if the feedback is provided

Postconditions:

- Feedback given

Expected Result: The learner receives feedback after interacting with each component

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: IL005

Test Case Title: Verify that interactive lessons can be accessed on various devices

Test Case Description: This test case is intended to confirm that the interactive lessons are accessible on different kinds of devices like smartphones, tablets, laptops, etc.

Test Suite: Interactive Lessons

Test Priority: High

Preconditions:

- Learner is registered
- Course or tutorial selected
- Different devices available for testing

Test Data: No test data needed

Test Steps:

1. Log in as a learner on different devices
2. Select a course or tutorial
3. Access a lesson
4. Check if the lesson interacts properly and elements are accessible

Postconditions:

- Lesson accessed on multiple devices

Expected Result: Interactive lessons should be accessible and functional across multiple devices.

Severity: Major

Type of Testing: Compatibility

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 Case 1:

Test Case ID: LP001

Test Case Title: Verify that learners can choose courses and modules based on their interests

Test Case Description: The test ensures that learners can select their desired courses and modules for personalized study.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in

- Courses and modules are available

Test Data: Course and Module Names

Test Steps:

1. Log into the learner account
2. Browse available courses and modules
3. Select courses and modules according to interest
4. Add selections to personal learning path

Postconditions:

- Course and module selections are saved

Expected Result: The system should enable learners to add their choice of courses and modules to their personalized learning path.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: LP002

Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress

Test Case Description: The test checks if the learning paths are dynamically changing based on the learner's progress.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner has started a course or module

Test Data: Learner progress data

Test Steps:

1. Log into the learner account
2. Check current learning path structure
3. Complete a section/module of a course
4. Review changes in the learning path

Postconditions:

- Learning path has updated based on progress

Expected Result: The system dynamically adjusts the learner's personalized learning path according to their progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: LP003

Test Case Title: Verify that learners receive recommendations for new courses and modules

Test Case Description: The focus of this test is to confirm that the learners receive course and module recommendations based on their learning path and interests.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- Learner has built a learning path

Test Data: No test data needed

Test Steps:

1. Log into the learner account
2. Review the personalized learning path
3. Check for any new recommendations

Postconditions:

- Recommendations are listed

Expected Result: The system offers recommendations to the learner for new courses and modules related to their personalized learning path.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: LP004

Test Case Title: Verify that progress tracking is personalized to each learner's path

Test Case Description: This test aims to ensure that the learner's progress is monitored and displayed in relation to their personalized learning path.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is engaged in a course

Test Data: Learner's progress data

Test Steps:

1. Log into the learner account
2. Navigate to the progress tracking feature
3. Review the progress statistics in relation to the courses and modules on the learning path

Postconditions:

- Progress tracking is displayed

Expected Result: The system displays the learner's progress that is personalized and linked to their individual learning path.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: LP005

Test Case Title: Verify that learners can adjust their learning paths at any time

Test Case Description: This test verifies that learners can make changes to their existing learning paths as and when desired.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- Learner has developed a learning path

Test Data: Course and module names

Test Steps:

1. Log into the learner account
2. View the current learning path
3. Make changes to the learning path by adding or removing courses/modules
4. Save the updated learning path

Postconditions:

- Changes in the learning path are saved

Expected Result: The system allows learners to make adjustments to their personalized learning paths as they wish.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 52

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Progress Tracking

Feature description -

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

Userstory title -

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Acceptance criteria -

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

Generated Test Cases:

Test Case 1:

Test Case ID: PT_01

Test Case Title: Verify that Progress Tracking shows completed activities and grades

Test Case Description: Ensure that the Progress Tracking feature displays both completed activities and the corresponding grades of the learners.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- The learner is logged in
- The learner has completed some activities

Test Data: Completed activities and grades

Test Steps:

1. Login as a learner
2. Go to the progress tracking section
3. Review the list of completed activities and grades

Postconditions:

- Progress details are updated

Expected Result: The system successfully displays the list of completed activities and their corresponding grades.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: PT_02

Test Case Title: Verify that learners can view detailed reports on their progress

Test Case Description: Ensure that the Progress Tracking feature provides a detailed report of the learners' progress.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

- The learner is logged in
- The learner has some progress

Test Data: Learning progress details

Test Steps:

1. Login as a learner
2. Go to the progress tracking section
3. Click on 'View Report'

Postconditions:

- Progress report is generated

Expected Result: The system successfully generates and displays a detailed progress report for the learner.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: PT_03

Test Case Title: Verify that achievements are highlighted in the progress tracker

Test Case Description: Ensure that the Progress Tracking feature highlights the achievements of the learners.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

- The learner is logged in
- The learner has some achievements

Test Data: Learning achievements details

Test Steps:

1. Login as a learner
2. Go to the progress tracking section
3. Review the achievements

Postconditions:

- Achievements are highlighted

Expected Result: The system successfully highlights the achievements in the progress tracker.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: PT_04

Test Case Title: Verify that learners receive notifications about their progress

Test Case Description: Ensure that the learners receive notifications regarding their progress updates.

Test Suite: Progress Tracking

Test Priority: Medium

Preconditions:

- The learner is logged in
- The learner has opted for progress notifications

Test Data: Progress Notifications

Test Steps:

1. Login as a learner
2. Complete an activity or quiz
3. Check for progress notifications

Postconditions:

- Progress notification is received

Expected Result: The system successfully sends a notification to the learner regarding their progress update.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: PT_05

Test Case Title: Verify that progress tracking can be exported as reports

Test Case Description: Ensure that the learners can export their progress tracking details as a report.

Test Suite: Progress Tracking

Test Priority: Low

Preconditions:

- The learner is logged in
- The learner has some progress

Test Data: No test data needed

Test Steps:

1. Login as a learner
2. Go to the progress tracking section
3. Click on ?Export Report?
4. Save the report

Postconditions:

- The report is saved

Expected Result: The system successfully generates and exports the learner's progress report.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 53

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Assessment and Grading

Feature description -

Instructors have the ability to create diverse assessments such as quizzes, tests, and assignments to evaluate learner comprehension and mastery of course content. Through timely feedback and accurate grading, instructors support learners in their academic development and provide valuable insights for continuous improvement.

Userstory title -

As an instructor, I want to create diverse assessments so that I can evaluate learner comprehension and provide feedback.

Acceptance criteria -

['Instructors can create quizzes, tests, and assignments.', 'Assessments can include various question types (MCQs, short answers, essays, etc.).', 'Automatic grading is available for objective questions.', 'Manual grading is available for subjective questions.', 'Feedback is provided to learners after grading.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC_001

Test Case Title: Verify that instructors can create diverse assessments

Test Case Description: Ensure that the system empowers instructors to create quizzes, tests, and assignments for learner assessments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- The Instructor is logged in
- Course outlined

Test Data: Assessment details such as syllabus, number of questions, etc

Test Steps:

1. Navigate to the assessment creation page
2. Select the assessment type (quiz/test/assignment)
3. Enter the relevant assessment details
4. Save and publish the assessment

Postconditions:

- Assessment should be created

Expected Result: The system allows instructors to create and publish a quiz, test, or assignments as per their input.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC_002

Test Case Title: Verify that various question types can be included in the assessments

Test Case Description: Confirm whether the system allows various question types (MCQs, short answers, essays, etc.) in assessments.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- The Instructor is logged in
- An assessment type is selected

Test Data: Question details like question type, answers, etc

Test Steps:

1. Navigate to the question creation/edits in the assessment
2. Choose the question type
3. Enter the question and its corresponding answers
4. Save these settings.

Postconditions:

- Questions of various types should be added

Expected Result: The system enables instructors to include MCQs, short answers, essays, and other types in the assessments.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC_003

Test Case Title: Verify that auto-grading is available for objective questions

Test Case Description: The system should be able to grade objective questions automatically, helping teachers to evaluate learners effectively and efficiently.

Test Suite: Assessment and Grading

Test Priority: Medium

Preconditions:

- Assessment with objective type questions is created and published
- Students have attempted the assessment

Test Data: No test data needed

Test Steps:

1. Navigate to the grading section
2. Check if the objective type questions are auto-graded.
3. Verify that grades are properly reflected in the system

Postconditions:

- Grades for objective type questions should be updated

Expected Result: The system accurately auto-grades objective questions and displays the right grades.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC_004

Test Case Title: Verify that manual grading is available for subjective questions

Test Case Description: The system should allow manual grading for subjective questions, providing the opportunity for teachers to evaluate learners' subjective answers.

Test Suite: Assessment and Grading

Test Priority: Medium

Preconditions:

- Assessment with subjective type questions is created and published
- Students have attempted the assessment

Test Data: No test data needed

Test Steps:

1. Navigate to the grading section
2. Check for ungraded subjective questions
3. Validate the functionality of grading subjective questions manually.

Postconditions:

- Grades for subjective questions should be updated

Expected Result: The system permits manual grading of subjective questions and updates student grades accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC_005

Test Case Title: Verify that feedback is provided to learners after grading

Test Case Description: Confirm if the system ensures timely feedback to learners after the grading, offering valuable insights for their learning path.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

- Assessment has been graded

Test Data: No test data needed

Test Steps:

1. Navigate to the feedback section
2. Check the functionality to provide feedback to learners
3. Validate if the feedback is visible to the learner.

Postconditions:

- Feedback should be visible to learners

Expected Result: The system enables instructors to provide feedback after grading, which is subsequently accessible to learners.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: DFTC01

Test Case Title: Verify that learners can post questions and responses in forums

Test Case Description: This test case validates whether the learners are able to ask questions and respond to posts in the discussion forums.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- Learner is logged in
- Discussion forum for a course exists.

Test Data: Sample text for question and response

Test Steps:

1. Navigate to the chosen course's discussion forum
2. Click 'New Post'
3. Enter the question in text box and submit
4. Select the post just created and respond to it

Postconditions:

- New post and response are visible in discussion forum.

Expected Result: The system allows learners to post questions and responses successfully in the forum.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: DFTC02

Test Case Title: Verify that forums support threaded discussions

Test Case Description: This case checks if the system supports threaded discussions, where multiple learners can reply to a single post.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- Multiple learners are logged in
- Existing post in discussion forum

Test Data: Responses to existing post

Test Steps:

1. Go to existing post
2. Multiple learners respond to the post
3. Review the organisation of responses under the post

Postconditions:

- Responses are nested under the parent post in a threaded manner: responses show hierarchical relationship

Expected Result: The forum supports threaded discussions, visible as discussions nested under the parent post.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: DFTC03

Test Case Title: Verify that instructors can moderate discussions

Test Case Description: This case will test if instructors have necessary permissions to moderate discussions in the forum.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- Instructor is logged in
- Existing post in discussion forum
- Moderator abilities (delete, edit, pin, etc.)

Test Data: Existing post to be moderated

Test Steps:

1. Instructor goes to a discussion forum post
2. Instructor uses moderation abilities (delete, edit, pin, etc.)

Postconditions:

- Changes made by instructor are reflected in the discussion forum

Expected Result: The instructor can moderate discussion forums successfully.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: DFTC04

Test Case Title: Verify that learners receive notifications for new posts

Test Case Description: The aim is to check that learners receive notifications for new posts in discussion forums they participate in.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- Learner is logged in
- Discussion forum for a course exists
- Settings set to receive notifications

Test Data: New post in the thread learner participates in

Test Steps:

1. Post new discussion in the forum
2. Check learner's notifications

Postconditions:

- New post is visible in learner's notifications

Expected Result: Learners receive notifications for new posts in discussion forums.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: DFTC05

Test Case Title: Verify that discussion forums are accessible on all devices

Test Case Description: This case aims to test if learners can access and interact with discussion forums accurately on various devices (desktop, tablet, mobile).

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

- Learner is logged in
- Available devices for testing: desktop, tablet, and mobile
- Discussion forum for a course exists

Test Data: No test data needed

Test Steps:

1. Access discussion forum on a desktop
2. Access discussion forum on a tablet
3. Access discussion forum on a mobile

Postconditions:

- The discussion forum is accessible, functional, and optimally viewable on all tested devices.

Expected Result: The system offers multi-device accessibility for discussion forums.

Severity: Major

Type of Testing: Compatibility

Test Case Behaviour: Positive

Data - 55

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Live Classes and Webinars

Feature description -

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A opportunities, and direct interaction between instructors and learners, replicating the immersive experience of traditional classroom settings.

Userstory title -

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

Acceptance criteria -

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources during sessions.', 'Notifications are sent before sessions start.']

Generated Test Cases:

Test Case 1:

Test Case ID: OLP001

Test Case Title: Verify that the platform facilitates live classes and webinars

Test Case Description: Ensure that the platform provides online video conferencing tools for live interactive classes and webinars.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Learner is logged into the platform
- Instructor scheduled a live class or webinar

Test Data: Live class or webinar schedule details

Test Steps:

1. Check the live classes or webinars section in the platform
2. Open the scheduled online class or webinar
3. Connect to the live class or webinar

Postconditions:

- Learner attended the live class or webinar

Expected Result: The platform facilitates live interactive sessions without any interruptions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLP002

Test Case Title: Verify that learners can interact in real-time

Test Case Description: Ensure that the platform provides features for learners to ask questions and interact with instructors during live sessions.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Learner is logged into the platform
- Live session is ongoing

Test Data: None

Test Steps:

1. Access an ongoing live session
2. Try to raise a question or a query.

Postconditions:

- Learner was able to interact in the live session

Expected Result: The platform allows learners to ask questions and interact with instructors in real-time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLP003

Test Case Title: Verify that live classes and webinars are recorded

Test Case Description: Ensure that all live sessions are recorded and available for learners to revisit and review.

Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

- Learner is logged into the platform
- Live session has concluded

Test Data: Live session details

Test Steps:

1. Navigate to the completed live sessions section
2. Open a previously conducted session
3. Verify if the session's recording is available

Postconditions:

- Learner is able to view the recorded session

Expected Result: The platform should make all session recordings available for later viewing.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLP004

Test Case Title: Verify that instructors can share resources

Test Case Description: Ensure that the platform allows instructors to share their screen and resources during live classes and webinars.

Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Live session is ongoing

Test Data: Resources to be shared (E.g. files, screen)

Test Steps:

1. Initiate a live session or webinar
2. Try to share screen or a resource

Postconditions:

- Resources or screen is shared to learners

Expected Result: The platform enables instructors to share resources like screen, files, etc. during sessions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP005

Test Case Title: Verify that notifications are sent before sessions start

Test Case Description: Ensure that the platform sends notifications to learners letting them know about upcoming live classes or webinars.

Test Suite: Live Classes and Webinars

Test Priority: High

Preconditions:

- Learner is logged into the platform
- Live session is scheduled for a future date

Test Data: Live session details

Test Steps:

1. Wait for the notification of the upcoming session
2. Check if the notification has been received
3. Verify the details of the upcoming session from the notification

Postconditions:

- Learner has gotten the notification of the upcoming session

Expected Result: The platform sends out notifications to learners about upcoming live classes or webinars.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: OLPM1

Test Case Title: Verify that the platform is accessible on smartphones and tablets

Test Case Description: This test case is intended to verify that the online learning platform is accessible and fully functional on both smartphones and tablets.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- User has a smartphone or tablet
- User is registered on the platform

Test Data: No test data needed

Test Steps:

1. Open the learning platform on a smartphone or tablet.
2. Login with valid credentials
3. Explore different features (Courses, Discussion Forums, etc.)

Postconditions:

- User is able to successfully access and use the platform on mobile devices

Expected Result: Online learning platform functions smoothly on both smartphones and tablets.

Severity: Major

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPM2

Test Case Title: Verify that the Mobile UI is optimized for touch navigation

Test Case Description: To ensure that the platform's UI is optimized for touch navigation, making it user-friendly for mobile users.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- User is logged into the platform via a smartphone or tablet

Test Data: No test data needed

Test Steps:

1. Navigate through different features
2. Attempt to interact with multiple elements on the platform
3. Verify touch response to the user input

Postconditions:

- User is able to navigate smoothly on the platform via touch

Expected Result: The platform's UI is touch-responsive and optimized for mobile devices.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPM3

Test Case Title: Verify that course materials are accessible and readable on mobile devices

Test Case Description: This test case checks if course materials can be effectively accessed and are readable on mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- User is logged into the platform via a smartphone or tablet
- Courses with materials are available on the platform

Test Data: Course Materials

Test Steps:

1. Access a course
2. Open course materials
3. Check the readability and accessibility of the materials

Postconditions:

- User can successfully access and read course materials on the mobile device

Expected Result: Course materials are accessible and readable on mobile devices.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPM4

Test Case Title: Verify that notifications are sent to mobile devices

Test Case Description: To confirm that the platform sends notifications to mobile devices upon updates or changes.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- User is logged into the platform via a smartphone or tablet
- Notifications have been activated on the user's device

Test Data: No test data needed

Test Steps:

1. Trigger an event that should lead to a notification (like assignment submission, forum reply, etc.)
2. Check for the notification on the mobile device

Postconditions:

- User receives notifications on their mobile devices regarding updates or changes

Expected Result: Upon any updates or changes, notifications are sent to mobile devices.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPM5

Test Case Title: Verify that the platform supports offline access to downloaded materials

Test Case Description: To ensure that users can access downloaded materials from the platform while offline.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- User has logged into the platform via a smartphone or tablet
- User has downloaded materials from the platform

Test Data: Downloaded Materials

Test Steps:

1. Disconnect device from the network
2. Access the downloaded materials

Postconditions:

- User can access downloaded materials while offline

Expected Result: Users can access downloaded materials from the learning platform while offline.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 57

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Resource Library

Feature description -

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides enriches the learning experience by providing additional insights and resources. Learners can access diverse content to deepen their understanding, explore related topics, and enhance their knowledge beyond the confines of the core curriculum.

Userstory title -

As a learner, I want access to a resource library so that I can explore additional materials and deepen my understanding.

Acceptance criteria -

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for offline access.', 'Instructors can recommend resources to learners.']

Generated Test Cases:

Test Case 1:

Test Case ID: RL01

Test Case Title: Verify that the resource library includes e-books, articles, and reference guides

Test Case Description: Ensure that the e-books, articles and reference guides are available in the resource library.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- Learner is logged in

- Resource library is active and accessible

Test Data: Sample collection of e-books, articles, and reference guide titles

Test Steps:

1. Navigate to Resource Library
2. Browse the available materials
3. Locate and open e-book, articles and reference guides

Postconditions:

- E-books, articles and reference guides were found

Expected Result: The system displays a variety of e-books, articles, and reference guides in the resource library.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: RL02

Test Case Title: Verify that learners can search for specific materials in the resource library

Test Case Description: Ensure that the search functionality is working properly on Resource Library

Test Suite: Resource Library

Test Priority: High

Preconditions:

- Learner is logged in
- The resource library contains materials

Test Data: Specific e-book, article, or reference guide title

Test Steps:

1. Navigate to Resource Library

2. Use the search function to find specific resources

Postconditions:

- Specific resources were found

Expected Result: The system displays the requested content when users search for specific resources in the resource library.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: RL03

Test Case Title: Verify that resources are categorized and tagged for easy navigation

Test Case Description: Ensure that the resources in the Resource Library are well organized by category and tag.

Test Suite: Resource Library

Test Priority: High

Preconditions:

- Learner is logged in
- The resource library contains categorized and tagged materials

Test Data: No test data needed

Test Steps:

1. Navigate to Resource Library
2. Browse resources by category and tags

Postconditions:

- User has browsed through different categories and tags

Expected Result: The system shows resources grouped by categories and tags for easy navigation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: RL04

Test Case Title: Verify that learners can download resources for offline access

Test Case Description: Ensure that resources in the library can be downloaded for offline use.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- Learner is logged in
- Resource library has downloadable content

Test Data: No test data needed

Test Steps:

1. Navigate to Resource Library
2. Select a resource item
3. Click the download button

Postconditions:

- Resource file is downloaded

Expected Result: The system allows users to download resource content for offline access.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: RL05

Test Case Title: Verify that instructors can recommend resources to learners

Test Case Description: Ensure that instructors have the capability to recommend resource materials to learners.

Test Suite: Resource Library

Test Priority: Medium

Preconditions:

- Instructor is logged in
- There are available resources to recommend

Test Data: Specific e-book, article, or reference guide title

Test Steps:

1. Navigate to Resource Library as an Instructor
2. Select a resource to recommend
3. Use the recommend function to suggest it to a learner

Postconditions:

- Recommended resource is sent to the learner

Expected Result: The system allows instructors to recommend resources to learners.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 58

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Peer Review

Feature description -

Learners actively engage in peer review processes, offering constructive feedback and insights on each other's work. Through collaborative assessment and review, learners develop critical evaluation skills, enhance their understanding of course concepts, and benefit from peer-to-peer learning exchanges.

Userstory title -

As a learner, I want to participate in peer review so that I can give and receive feedback on assignments.

Acceptance criteria -

['Learners can submit assignments for peer review.', 'Peers can provide feedback on submitted assignments.', 'A structured review process guides the feedback.', 'Learners can view feedback on their work.', 'Reviewers remain anonymous to ensure unbiased feedback.']

Generated Test Cases:

Test Case 1:

Test Case ID: OLPR01

Test Case Title: Verify that learners can submit assignments for peer review

Test Case Description: Ensure that the platform allows learners to submit their assignments for review by their peers.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in

- Assignment is completed

Test Data: Assignment submission details

Test Steps:

1. Go to "My Assignments" page
2. Select the completed assignment
3. Click on "Submit for Peer Review" button

Postconditions:

- Assignment is marked as "Under Review"

Expected Result: The system accepts the assignment submission and marks it as "Under Review".

Severity: Major

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPR02

Test Case Title: Verify that peers can provide feedback on submitted assignments

Test Case Description: Ensure that the peer-review functionality in the system allows other learners to provide feedback on submitted assignments.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in
- Assignments for review are available

Test Data: Review & feedback details

Test Steps:

1. Go to the "Review Assignments" page
2. Select an assignment
3. Write feedback and submit

Postconditions:

- Assignment review status is updated

Expected Result: The system allows learners to provide feedback on assignments and updates the review status.

Severity: Major

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPR03

Test Case Title: Verify that a structured review process guides the feedback

Test Case Description: The system must guide learners through a structured review process while providing feedback on assignments.

Test Suite: Peer Review

Test Priority: Medium

Preconditions:

- Learner is logged in
- Assignment for review is selected

Test Data: No test data needed

Test Steps:

1. Start the review process
2. Observe the prompts and instructions for providing feedback

Postconditions:

- Review is submitted within the guide of the structured process

Expected Result: The system guides the learner through a structured review process while providing feedback.

Severity: Major

Priority: Medium

Type of Testing: Usability

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPR04

Test Case Title: Verify that learners can view feedback on their work

Test Case Description: Ensure that the system allows learners to view the feedback received on their submissions.

Test Suite: Peer Review

Test Priority: High

Preconditions:

- Learner is logged in
- Feedback is available on assignments

Test Data: No test data needed

Test Steps:

1. Go to the "My Assignments" page
2. Select an assignment
3. Click on "View Feedback"

Postconditions:

- Feedback is displayed to the learner

Expected Result: The system displays the feedback received from peers on the assignments.

Severity: Major

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPR05

Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback

Test Case Description: The system must ensure that the identity of reviewers remain anonymous to ensure unbiased feedback.

Test Suite: Peer Review

Test Priority: Low

Preconditions:

- Learner is logged in
- Feedback on assignments is available

Test Data: No test data needed

Test Steps:

1. Go to the "My Assignments" page
2. Select an assignment
3. View feedback

Postconditions:

- No personal information about the reviewer is disclosed

Expected Result: The system does not disclose the identity of the reviewer, ensuring unbiased feedback.

Severity: Minor

Priority: Low

Type of Testing: Security

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 Case 1:

Test Case ID: TC-01

Test Case Title: Verify that learners earn certificates upon course completion

Test Case Description: This test case will verify if the system correctly awards certificates to learners upon successful completion of a course.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- The learner is registered and logged in
- The learner has successfully completed a course

Test Data: Course completion status, Certificate details

Test Steps:

1. Login as a learner
2. Navigate to completed courses
3. Verify if a certificate is awarded

Postconditions:

- Learner's profile is updated with a new certificate

Expected Result: The system updates the learner's profile with a new certificate upon successful completion of a course.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TC-02

Test Case Title: Verify that badges are awarded for specific achievements and milestones

Test Case Description: This test case will check if the system correctly awards badges to the learners for achieving specific milestones.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- The learner is logged in
- Learner has reached a specific milestone/achievement

Test Data: Achievement or milestone details, Badge details

Test Steps:

1. Login as a learner
2. Navigate to the achievements section
3. Verify if a badge is awarded

Postconditions:

- Learner's profile is updated with a new badge

Expected Result: The system updates the learner's profile with a new badge upon accomplishing a specific milestone or achievement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TC-03

Test Case Title: Verify that certificates and badges are accessible through learner profiles

Test Case Description: This test case verifies if the learners can access their certificates and badges through their learner profiles.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- The learner is logged in
- Learner has earned at least one badge or certificate

Test Data: No test data needed

Test Steps:

1. Login as a learner
2. Navigate to the profile section
3. Verify if earned certificates and badges are displayed

Postconditions:

- Learner views earned certificates and badges

Expected Result: The system displays the earned certificates and badges in the learner's profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TC-04

Test Case Title: Verify that learners can share their certificates and badges on social media

Test Case Description: This test case verifies if the system effectively enables learners to share their earned certificates and badges on social media platforms.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- The learner is logged in

- Learner has earned at least one badge or certificate

Test Data: Social media login credentials

Test Steps:

1. Login as a learner
2. Navigate to the profile section
3. Click on a certificate or badge
4. Select the "Share on social media" option
5. Choose a social media platform and share

Postconditions:

- Learner shares earned certificates or badges on selected social media platform

Expected Result: The system facilitates the sharing of the learner's achieved certificates and badges on their chosen social media platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TC-05

Test Case Title: Verify that instructors can issue custom badges for special achievements

Test Case Description: This test case will confirm if the system supports instructors in creating and issuing custom badges for learners' special achievements.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

- The instructor is logged in
- The instructor has identified a learner for a special achievement badge

Test Data: Badge details, Learner information

Test Steps:

1. Login as an instructor
2. Navigate to the learner's profile
3. Create a custom badge
4. Assign the custom badge to a learner

Postconditions:

- Learner's profile is updated with a new custom badge

Expected Result: The system correctly supports the creation of a custom badge by the instructor and updates the recipient learner's profile accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 60

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Gamification Elements

Feature description -

Gamification elements such as points, levels, and leaderboards are strategically integrated into the learning

experience to motivate learners and foster engagement. By incorporating game-like mechanics and incentives, the platform incentivizes active participation, goal attainment, and healthy competition, enhancing learner motivation and enjoyment.

Userstory title -

As a learner, I want gamification elements so that I can be motivated to achieve my learning goals.

Acceptance criteria -

['Learners earn points for completing activities.', 'Levels are achieved based on accumulated points.', 'Leaderboards display top-performing learners.', 'Badges are awarded for reaching milestones.', 'Learners can view their progress in a gamification dashboard.']

Generated Test Cases:

Test Case 1:

Test Case ID: GO1

Test Case Title: Verify that learners earn points for completing activities

Test Case Description: Ensure that the system awards learners points each time they complete an activity on the online learning platform.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is signed in
- Activities are available for the learner to complete

Test Data: Activity completion data

Test Steps:

1. Select an activity to complete
2. Complete the activity
3. Check the points balance

Postconditions:

- Points balance is increased

Expected Result: The system increases the learner's points balance upon completion of an activity.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: GO2

Test Case Title: Verify that levels are achieved based on accumulated points

Test Case Description: Ensure that the system promotes learners to a higher level based on the points they have accumulated.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is signed in
- Learner has completed activities and earned points

Test Data: Points balance data

Test Steps:

1. Check the current level of the learner
2. Complete activities to accumulate points
3. Check the new level of the learner

Postconditions:

- Learner level is increased

Expected Result: The system increases the learner's level when a specific points threshold is met.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: GO3

Test Case Title: Verify that leaderboards display top-performing learners.

Test Case Description: Ensure that the system correctly displays top-performing learners on a leaderboard according to their points and levels.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Multiple learners are signed in and active

Test Data: Learners points and levels data

Test Steps:

1. Review the leaderboard
2. Identify top-performing learners

Postconditions:

- Leaderboard is correctly displayed

Expected Result: The system correctly displays top-performing learners on the leaderboard.

Severity: Medium

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: GO4

Test Case Title: Verify that badges are awarded for reaching milestones

Test Case Description: Ensure that the system awards badges to learners when they achieve specific learning milestones.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

- Learner is signed in
- Milestones are defined in the learning path

Test Data: Milestone achievement data

Test Steps:

1. Select a milestone to achieve
2. Complete the steps to achieve the milestone
3. Check for awarded badge

Postconditions:

- Badges are rewarded

Expected Result: The system rewards badge to the learner when they achieve a milestone.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: GO5

Test Case Title: Verify that learners can view their progress in a gamification dashboard

Test Case Description: Ensure that the system provides a gamification dashboard where learners can view their progress, including points, levels, leaderboard status, and badges.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

- Learner is signed in

Test Data: No test data needed

Test Steps:

1. Access the gamification dashboard
2. View points, levels, leaderboard status, and badges

Postconditions:

- Dashboard display is correct

Expected Result: The system correctly displays all gamification elements and progress in the dashboard.

Severity: Medium

Type of Testing: Usability 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 Case 1:

Test Case ID: MUL01

Test Case Title: Verify that the platform interface is available in multiple languages.

Test Case Description: This test case is to ensure that the platform provides its interface in multiple languages.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- The platform is accessible and running.
- Languages are configured correctly in the system.

Test Data: No test data needed

Test Steps:

1. Log in to the platform.
2. Navigate to the settings or language option.
3. Check if there are multiple language options.

Postconditions:

- Interface language has been checked.

Expected Result: Multiple language options should be available for selection in the interface.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: MUL02

Test Case Title: Verify that course materials are available in different languages.

Test Case Description: This test case validates that course materials are offered in multiple languages on the platform.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:

- Courses are available on the platform.
- Courses have multilingual versions.

Test Data: Course details

Test Steps:

1. Log in to the platform.
2. Select a course.
3. Check if course materials are available in different languages.

Postconditions:

- Course materials checked for multiple languages.

Expected Result: The system provides course materials in multiple languages.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: MUL03

Test Case Title: Verify that learners can select their preferred language at any time.

Test Case Description: This test case ensures that the system enables learners to switch between languages at any given time.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- The platform is operational.
- The user is logged into the platform.

Test Data: User login credentials

Test Steps:

1. Log in to the platform.
2. Navigate to language settings.
3. Change to a different language.
4. Check if the change reflects instantly.

Postconditions:

- Language has been switched in settings.

Expected Result: The system should immediately reflect the language change in the interface.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: MUL04

Test Case Title: Verify that multilingual support includes translated help and support resources.

Test Case Description: This test case verifies that help and support resources are provided in multiple languages.

Test Suite: Multilingual Support

Test Priority: Medium

Preconditions:

- Help and support resources are available.
- Translated versions of resources are available.

Test Data: No test data needed

Test Steps:

1. Log in to the platform.
2. Navigate to the help or support section.
3. Check for availability of help resources in different languages.

Postconditions:

- Help and support resources checked for multilingual availability.

Expected Result: The system provides help and support resources in multiple languages.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: MUL05

Test Case Title: Verify that instructors can provide content in multiple languages.

Test Case Description: This test case ensures that instructors can offer course content in various languages.

Test Suite: Multilingual Support

Test Priority: Low

Preconditions:

- Instructor access is granted.
- Course content is available in multiple languages.

Test Data: Instructor login credentials, course details

Test Steps:

1. Log in as an instructor.
2. Choose a course to edit.
3. Upload course materials in different languages.
4. Save changes and check how it appears for learners.

Postconditions:

- Course content has been updated in multiple languages.

Expected Result: Instructors can successfully upload and present course materials in multiple languages.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 62

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Accessibility Features

Feature description -

The platform prioritizes accessibility by incorporating features such as screen reader compatibility, adjustable font sizes, and color contrast options. These accessibility features ensure that learners with disabilities can access and navigate the platform with ease, promoting inclusivity and equitable access to educational opportunities.

Userstory title -

As a learner with disabilities, I want accessibility features so that I can use the platform effectively.

Acceptance criteria -

['The platform is compatible with screen readers.', 'Learners can adjust font sizes for better readability.', 'Color contrast options are available for improved visibility.', 'Accessibility features are easy to activate and customize.', 'Support resources are available to assist learners with disabilities.']

Generated Test Cases:

Test Case 1:

Test Case ID: OLP01A

Test Case Title: Verify that the platform is compatible with screen readers

Test Case Description: Test the platform's compatibility with popular screen readers to verify the effectiveness of accessibility features for visually impaired learners.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in
- Screen reader software installed and activated

Test Data: No test data needed

Test Steps:

1. Activate the screen reader software
2. Navigate various areas of the platform (Courses, Lectures, etc.)

Postconditions:

- Platform content is readable by screen reader software

Expected Result: The platform is fully compatible and accessible with screen reader software.

Severity: Major

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLP02A

Test Case Title: Verify that learners can adjust the font sizes for better readability

Test Case Description: Test the functionality that allows users to adjust the font sizes on the platform, aiding readability for visually impaired learners.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in

Test Data: Various font sizes

Test Steps:

1. Navigate to a course or lecture page
2. Locate and adjust the font size setting
3. Verify the changes in font size on text content

Postconditions:

- Font size is changed to user specification

Expected Result: The font size adjusts according to user selection, improving readability.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLP03A

Test Case Title: Verify that color contrast options are available for improved visibility

Test Case Description: Test the available color contrast options on the platform to ensure improved visibility for learners with visual impairments and color blindness.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in

Test Data: Different color contrast settings

Test Steps:

1. Navigate to the platform settings
2. Locate and adjust the color contrast setting
3. Verify the changes in color contrast on the platform

Postconditions:

- Color contrast changes as per user selection

Expected Result: The platform provides different color contrast options for improved visibility.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLP04A

Test Case Title: Verify that accessibility features are easy to activate and customize

Test Case Description: Ensure that users can easily enable and customize accessibility features, providing a seamless user experience for learners with disabilities.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Navigate to the platform settings
2. Locate the accessibility features
3. Enable and customize different accessibility features

Postconditions:

- Accessibility features are activated and customized

Expected Result: Accessibility features are easy to activate and customize across the platform.

Severity: Major

Type of Testing: Usability Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP05A

Test Case Title: Verify that support resources are available to assist learners with disabilities

Test Case Description: Validate the availability and accessibility of support resources specifically designed

to assist learners with disabilities to navigate and utilize the platform effectively.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Look for the support or help section on the platform
2. Locate resources designed for users with disabilities
3. Review the contents and quality of the resources

Postconditions:

- Support resources for learners with disabilities are available

Expected Result: The platform offers easy-to-access and effective support resources for learners with disabilities.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 63

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Moderation

Feature description -

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

Userstory title -

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

Acceptance criteria -

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

Generated Test Cases:

Test Case 1:

Test Case ID: OLP01

Test Case Title: Verify that instructors can edit, delete, or highlight posts in discussions

Test Case Description: This test case aims to ensure that instructors can actively moderate discussions by editing, deleting, or highlighting posts.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in
- Discussion forum is available

- Posts are existing in the forum

Test Data: Discussion Forum Details, Post details

Test Steps:

1. Go to the respective discussion forum
2. Select a post to moderate
3. Conduct moderation actions (Edit, Delete, Highlight)

Postconditions:

- Moderation actions are applied to posts

Expected Result: The system allows the instructor to edit, delete, or highlight the selected posts in the forum.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLP02

Test Case Title: Verify that instructors can mute or ban disruptive users

Test Case Description: This test case checks the instructor's ability to mute or ban disruptive users in the discussion forum.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in
- Discussion forum with disruptive users is available

Test Data: User details in the discussion forum

Test Steps:

1. Go to the respective discussion forum

2. Identify a disruptive user
3. Proceed to mute or ban actions

Postconditions:

- Disruptive user is muted or banned

Expected Result: The system allows the instructor to mute or ban the disruptive users successfully.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLP03

Test Case Title: Verify that moderation actions are logged and reported

Test Case Description: This test case ensures that every moderation action taken by the instructor is logged and reported correctly.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- Instructor is logged
- Moderation actions have been made

Test Data: Moderation action data

Test Steps:

1. Perform some moderation actions
2. Go to moderator's log/report panel
3. Inspect if the actions are logged and reported

Postconditions:

- Moderation actions are logged and reported

Expected Result: The system accurately logs and reports all moderation actions taken by the instructor.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLP04

Test Case Title: Verify that community guidelines are visible to all participants

Test Case Description: This test case verifies that community guidelines are clearly visible to all participants in the discussion forum.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Participant is logged into the platform
- Accessed the discussion forum

Test Data: No test data needed.

Test Steps:

1. Enter the discussion forum
2. Look for the community guidelines

Postconditions:

- Community guidelines are seen by all users

Expected Result: The system ensures the community guidelines are visible in the appropriate area of the discussion forum to all the participants.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP05

Test Case Title: Verify that notifications are sent for moderated actions

Test Case Description: This test case verifies whether the system sends notifications regarding moderated actions made to relevant users.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in
- Moderation actions have been taken
- Relevant user's notification settings are turned on

Test Data: User details, notification details

Test Steps:

1. Perform moderation actions
2. Validate the delivery of the notification to the relevant user

Postconditions:

- Notification for moderated actions sent

Expected Result: The system sends appropriate notifications to the users related to the taken moderation actions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Learning Analytics

Feature description -

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement, performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning experience to meet learners' needs effectively.

Userstory title -

As an instructor, I want access to learning analytics so that I can understand learner engagement and performance.

Acceptance criteria -

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.', 'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.', 'Instructors can export analytics reports for further analysis.']

Generated Test Cases:

Test Case 1:

Test Case ID: LAD01

Test Case Title: Verify that learning analytics dashboards display engagement metrics

Test Case Description: Ensure that the learning analytics dashboard accurately displays the engagement

metrics for instructors to review.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Student interaction data is available

Test Data: Student interaction data (views, downloads, participation, completion, etc.)

Test Steps:

1. Log in as an instructor
2. Navigate to the learning analytics dashboard
3. Review the engagement metrics displayed

Postconditions:

- Engagement metrics are accurately reflected on the dashboard

Expected Result: The dashboard successfully displays accurate engagement metrics.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: LAD02

Test Case Title: Verify that performance trends are visualized over time

Test Case Description: Ensure that the learning analytics dashboards visualize the performance trends over time.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Student performance data is available over a certain period

Test Data: Student performance data (quiz scores, progress, completion, etc.)

Test Steps:

1. Log in as an instructor
2. Navigate to the learning analytics dashboard
3. Review the performance trends displayed over time

Postconditions:

- Performance trends over time are accurately visualized on the dashboard

Expected Result: The dashboard successfully visualizes accurate performance trends over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: LAD03

Test Case Title: Verify that instructors can access detailed reports on learner progress

Test Case Description: Ensure that instructors can access and review detailed reports on the progress of learners.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Detailed learner progress data is available

Test Data: Detailed individual learner progress data

Test Steps:

1. Log in as an instructor
2. Navigate to the learning analytics dashboard
3. Access and review the detailed learner progress reports

Postconditions:

- Instructor can access and review detailed learner progress reports

Expected Result: The instructor can successfully access and review detailed reports on learner progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: LAD04

Test Case Title: Verify that analytics data is updated in real-time

Test Case Description: Ensure that all analytics data displayed on the learning analytics dashboards is updated in real-time.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- There are recent learner interactions and progress data

Test Data: Learner interactions and progress data

Test Steps:

1. Log in as an instructor

2. Navigate to the learning analytics dashboard

3. Monitor the data updates in real-time as learners interact with the platform

Postconditions:

- Analytics data is successfully updated in real-time

Expected Result: The analytics data is successfully updated in real-time on the learning analytics dashboard.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: LAD05

Test Case Title: Verify that instructors can export analytics reports for further analysis

Test Case Description: Ensure that instructors can export the analytics reports for offline analysis and record keeping.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

- Instructor is logged in

Test Data: No test data needed

Test Steps:

1. Log in as an instructor

2. Navigate to the learning analytics dashboard

3. Access any report and click on export option

Postconditions:

- The selected report is successfully downloaded in the chosen format

Expected Result: Instructors can successfully export analytics reports for further offline analysis.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 65

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Learning Features

Feature description -

The platform integrates social learning features such as group projects, peer collaboration tools, and social sharing functionalities to foster community engagement and knowledge exchange. By facilitating interaction, networking, and collaboration among learners, social learning features enrich the learning experience and promote peer-to-peer support and mentorship.

Userstory title -

As a learner, I want social learning features so that I can collaborate with peers and engage in group projects.

Acceptance criteria -

['Learners can create and join groups for projects.', 'Collaboration tools are available for group communication.', 'Learners can share resources and feedback within groups.', 'Instructors can assign group projects and monitor progress.', 'Social sharing features allow learners to share achievements.']

Generated Test Cases:

Test Case 1:

Test Case ID: OL-01

Test Case Title: Verify that learners can create and join groups for projects

Test Case Description: Ensures the system allows learners to create new groups for projects and allows them to join these groups.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in
- Course is enrolled in.

Test Data: Learner's username, Group name

Test Steps:

1. Navigate to the group creation page
2. Enter the name and description of the group
3. Confirm group creation
4. Navigate to group list
5. Join created group

Postconditions:

- Group is created
- Learner joins group

Expected Result: Group is created and the learner is able to join the group.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OL-02

Test Case Title: Verify that collaboration tools are available for group communication

Test Case Description: Ensures that learners within a group can communicate using built-in collaboration tools.

Test Suite: Social Learning-Features

Test Priority: High

Preconditions:

- Learner is logged in
- Group is joined

Test Data: Test chat message

Test Steps:

1. Navigate to Group communication area
2. Send test chat message
3. Verify message display in chat history

Postconditions:

- Test message is sent and received

Expected Result: The collaboration tool allows for communication within the group.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: OL-03

Test Case Title: Verify that learners can share resources and feedback within groups

Test Case Description: Ensures that learners within a group can share learning resources and provide feedback to each other.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in
- Group is joined

Test Data: Sample study resource

Test Steps:

1. Navigate to Group resources area
2. Upload sample study resource
3. Navigate to Feedback section
4. Post feedback
5. Verify resource and feedback visibility within the group

Postconditions:

- Resource and feedback are shared within the group

Expected Result: The system allows learners to share resources and feedback within the group.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: OL-04

Test Case Title: Verify that instructors can assign group projects and monitor progress

Test Case Description: Ensures that instructors can assign tasks to groups and monitor their progress.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Instructor is logged in
- Course is created

Test Data: Task details

Test Steps:

1. Navigate to Course page
2. Assign project to a group
3. Navigate to Progress monitoring section
4. Verify task visibility and progress update for the group

Postconditions:

- Project is assigned to the group
- Instructor can monitor the progress of the group

Expected Result: Instructors can assign tasks to groups and monitor their progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: OL-05

Test Case Title: Verify that social sharing features allow learners to share achievements

Test Case Description: The system should allow learners to share their course completion certificates and achievements on social networking sites.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner has completed a course

Test Data: No test data needed

Test Steps:

1. Navigate to Achievements page
2. Click on the share button on the certificate or badge
3. Choose a social network to share to
4. Verify sharing functionality

Postconditions:

- Achievement is shared on the social platform

Expected Result: The sharing feature shows the shared certificate or badge on the selected social networking site.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 66

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Authoring Tools

Feature description -

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

Userstory title -

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

Acceptance criteria -

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

Generated Test Cases:

Test Case 1:

Test Case ID: CLK1

Test Case Title: Verify that instructors can create multimedia presentations.

Test Case Description: Ensure that content authoring tools enable instructors to create multimedia presentations, with features like integrating images, audio, video, and text.

Test Suite: Content Authoring Tools

Test Priority: High

Pre-conditions:

- Instructor account is set up and logged in
- Instructor is on content authoring tools platform

Test Data: Mock images, audio, video, and text

Test Steps:

1. Select create new presentation
2. Integrate images, audio, video, and text
3. Finalize presentation

Postconditions:

- A new presentation is created and saved in the instructor's account

Expected Result: The content authoring tools platform lets instructor to create a new multimedia presentation successfully.

Severity: Major

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: CLK2

Test Case Title: Verify that interactive modules can be developed and integrated.

Test Case Description: Validate that instructors can use content authoring tools to develop interactive learning modules, such as quizzes, assignments, etc.

Test Suite: Content Authoring Tools

Test Priority: High

Pre-conditions:

- Instructor account is setup and logged in
- Instructor is on content authoring tools platform

Test Data: Mock text for quizzes or assignments

Test Steps:

1. Select create interactive module
2. Develop quiz/assignment

3. Finalize interactive module

Postconditions:

- New interactive module is created and saved in the instructor's account

Expected Result: The content authoring tools platform allows instructor to create a new interactive module successfully.

Severity: Major

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: CLK3

Test Case Title: Verify that rich media resources, such as videos and animations, are supported.

Test Case Description: Ensure that the content authoring tools support integration of various types of rich media resources into course content.

Test Suite: Content Authoring Tools

Test Priority: High

Pre-conditions:

- Instructor account is setup and logged in
- Instructor is on content authoring tools platform

Test Data: Mock videos and animations

Test Steps:

1. Select to create rich media content
2. Upload video or animation
3. Integrate the video/animation into the course content

Postconditions:

- Rich media resource is integrated into the course content.

Expected Result: The content authoring tools platform allows instructor to integrate rich media resources into course content.

Severity: Major

Priority: High

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: CLK4

Test Case Title: Verify that content authoring tools are user-friendly and intuitive.

Test Case Description: Validate the user-friendliness and intuitiveness of the content authoring tools, by evaluating the user interface design and navigation experience.

Test Suite: Content Authoring Tools

Test Priority: Medium

Pre-conditions:

- Instructor account is setup and logged in
- Instructor is on content authoring tools platform

Test Data: No Test data needed

Test Steps:

- 1.Navigate through different tools
2. Assess the ease of use and intuitiveness

Postconditions:

- Instructor exits the content authoring tools platform

Expected Result: The content authoring tools platform should be easy to navigate and intuitive for

instructors to use.

Severity: Minor

Priority: Medium

Type of Testing: Usability

Test Case Approach: Positive

Test Case 5:

Test Case ID: CLK5

Test Case Title: Verify that instructors can preview and edit content before publishing.

Test Case Description: Ensure that the content authoring tools allow instructors to preview the course content before publishing, and edit if needed for perfection.

Test Suite: Content Authoring Tools

Test Priority: High

Pre-conditions:

- Instructor account is setup and logged in
- Instructor is on content authoring tools platform
- Course content is created

Test Data: Existing course content

Test Steps:

1. Click on preview before publish button
2. Review the content
3. Edit the content if required

Postconditions:

- Course content is reviewed and edited (if needed) before publishing

Expected Result: The content authoring tools platform lets instructor to preview and edit content before publishing.

Severity: Major

Priority: High

Type of Testing: Functional

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 Case 1:

Test Case ID: LPA001

Test Case Title: Verify that learning paths include sequential courses and modules

Test Case Description: Ensure that the platform provides learning paths comprising sequential courses and modules.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Course and module data is loaded on the platform
- Learning paths are defined

Test Data: Course and Module list for a given Learning Path

Test Steps:

1. Log in as a learner
2. Select a learning path
3. View the sequence of courses and modules

Postconditions:

- Learning path details are displayed

Expected Result: The system displays the sequential courses and modules in the selected learning path.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: LPA002

Test Case Title: Verify that curricula are aligned with specific learning objectives

Test Case Description: Ensure that the platform aligns its curricula with the specific learning objectives.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Curricula are defined and paired with respective learning objectives

Test Data: Learning objectives for a given curriculum

Test Steps:

1. Log in as a learner
2. Select a curriculum
3. View the associated learning objectives

Postconditions:

- Curriculum and associated learning objectives are displayed

Expected Result: The system displays the curriculum aligned with specified learning objectives.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: LPA003

Test Case Title: Verify that learners can track their progress within a learning path

Test Case Description: Ensure that the platform enables learners to track their progress in a selected learning path.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Learner has started a learning path

Test Data: Learner's progress details in a learning path

Test Steps:

1. Log in as a learner
2. Navigate to the learning path progression
3. Check the learning progress

Postconditions:

- Learner's progress within a learning path is displayed

Expected Result: The system allows learners to track their progress within the selected learning path.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: LPA004

Test Case Title: Verify that instructors can customize learning paths for different needs

Test Case Description: Ensure that the platform provides instructors with the functionality to customize learning paths based on various learner needs.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Instructor has permission to customize learning paths

Test Data: Customization settings for a learning path

Test Steps:

1. Log in as an instructor
2. Select a learning path to customize
3. Make desired changes and save

Postconditions:

- Customizations to the learning path are saved

Expected Result: The system allows instructors to customize learning paths based on different learning needs.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: LPA005

Test Case Title: Verify that the system sends notifications for upcoming courses or milestones

Test Case Description: Ensure that the platform notifies learners about upcoming courses or milestones in their learning paths.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

- Upcoming courses or milestones are defined in the learning paths

Test Data: Details of upcoming courses or milestones

Test Steps:

1. Log in as a learner
2. Check notifications

Postconditions:

- Notifications for upcoming courses or milestones displayed

Expected Result: The system sends notifications about upcoming courses or milestones to learners.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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: ATC01

Test Case Title: Verify that attendance is automatically recorded during live sessions.

Test Case Description: Ensure that the system automatically logs the student's entry, exit, and duration in a live session. This is crucial for tracking learner attendance.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- The learner is registered for a course.
- A live session is scheduled and is currently in progress.

Test Data: Learner account information, Live session data.

Test Steps:

1. Login to the platform as a learner.
2. Join a live session.
3. Exit the live session.

Postconditions:

- The learner's attendance record is updated.

Expected Result: The system automatically records the learner's attendance during the live session.

Severity: Major

Type of Testing: System Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: ATC02

Test Case Title: Verify that instructors can view attendance reports for each session.

Test Case Description: Check if the system provides teachers with detailed attendance reports for each live session.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Attendance data is available for a course session.

Test Data: Attendance records, Course session data.

Test Steps:

1. Login to the platform as an instructor.
2. Navigate to the course session details.
3. View the attendance report.

Postconditions:

- The instructor views the attendance report.

Expected Result: The system allows instructors to view detailed attendance reports for each session.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: ATC03

Test Case Title: Verify that learners can see their attendance records.

Test Case Description: Ensure that students have access to their attendance records for live sessions.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Learner's attendance data exists in the system.

Test Data: Learner account data, Attendance records.

Test Steps:

1. Login as a learner.
2. Navigate to course details.
3. Access the attendance history.

Postconditions:

- The learner viewed his attendance records.

Expected Result: The system allows learners to see their individual attendance records.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: ATC04

Test Case Title: Verify that attendance data is integrated with participation metrics.

Test Case Description: Verify that the system integrates attendance data with other participation metrics like quiz scores, forum interactions to give a complete view of student engagement.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

- Attendance data and other participation metrics are available for a student.

Test Data: Attendance records, Participation metrics.

Test Steps:

1. Login as an instructor.
2. Select a student's profile.
3. View the integrated data on participation.

Postconditions:

- The instructor viewed the student's participation data integrated with attendance data.

Expected Result: The system integrates attendance data with other participation metrics.

Severity: Major

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: ATC05

Test Case Title: Verify that notifications are sent for missed sessions.

Test Case Description: Ensure that both the learner and instructor receive notifications when a live session is missed by the learner.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

- A live session has passed, and the learner didn't attend.

Test Data: Learner account data, Live session information.

Test Steps:

1. Run the system's missed session alert function after the live session.
2. Log in as a learner or instructor.
3. Check notifications.

Postconditions:

- The sent notification is checked.

Expected Result: The system sends notifications for missed sessions to both learners and instructors.

Severity: Major

Type of Testing: Functional Testing

Data - 69

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Licensing and Rights Management

Feature description -

The platform implements robust content licensing and rights management features to protect intellectual property and ensure compliance with copyright laws. By managing content licenses, usage rights, and permissions, the platform safeguards the interests of content creators and providers, enabling secure and lawful distribution of educational materials.

Userstory title -

As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

Acceptance criteria -

['Content licenses are assigned to each resource.', 'Usage rights and permissions are clearly defined.', 'Content creators can set restrictions on usage and distribution.', 'Instructors and learners are informed of content licensing terms.', 'Notifications are sent for license expiration or renewal.']

Generated Test Cases:

Test Case 1:

Test Case ID: CLRM01

Test Case Title: Verify that content licenses are assigned to each resource

Test Case Description: This test case checks if the platform assigns content licenses correctly to each resource uploaded by the content creators.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Content creator is logged in
- Content is uploaded for publication

Test Data: Uploaded content (e.g., videos, documents, quizzes)

Test Steps:

1. Log in as a content creator
2. Upload a piece of content (e.g., a video lecture)
3. Go to the licensing settings for the uploaded content

Postconditions:

- License is assigned to the uploaded content

Expected Result: Upon checking the licensing settings, the platform should show that a license was assigned to the uploaded content.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: CLRM02

Test Case Title: Verify that usage rights and permissions are clearly defined

Test Case Description: This test case checks if the platform clearly defines the usage rights and permissions for a content creator's resources.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Content creator is logged in

Test Data: Usage rights and permissions settings for a resource

Test Steps:

1. Log in as a content creator
2. Go to the usage rights settings for a piece of content

Postconditions:

- Usage rights and permissions are defined

Expected Result: The platform should clearly list and elucidate the usage rights and permissions for the selected resource.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: CLRM03

Test Case Title: Verify that content creators can set restrictions on usage and distribution

Test Case Description: This test case checks if the platform allows content creators to set restrictions on how their resources are used and distributed.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Content creator is logged in

Test Data: Restrictions settings for a resource

Test Steps:

1. Log in as a content creator
2. Go to the restrictions settings for a uploaded resource
3. Set the desired restrictions

Postconditions:

- Restrictions on usage and distribution are set

Expected Result: The platform should apply the set restrictions to the appropriate resource.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: CLRM04

Test Case Title: Verify that instructors and learners are informed of content licensing terms

Test Case Description: This test case checks if the platform adequately informs both instructors and learners of the content licensing terms for a particular resource.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Instructor or learner is logged in

Test Data: Licensing terms for a resource

Test Steps:

1. Log in as an instructor or learner

2. Access a licensed resource

Postconditions:

- The user is informed of the licensing terms

Expected Result: Upon accessing a licensed resource, the platform should present the user with the relevant content licensing terms.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: CLRM05

Test Case Title: Verify that notifications are sent for license expiration or renewal

Test Case Description: This test case checks if the platform sends notifications to content creators concerning the expiration or renewal of content licenses.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

- Content creator is logged in
- A content license is close to expiration or due for renewal

Test Data: No test data needed

Test Steps:

1. Log in as a content creator
2. Check notification center or email

Postconditions:

- A notification for license expiration or renewal is received

Expected Result: The platform sends a notification to the content creator to inform about the impending

expiration or renewal of content licenses.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 70

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Versioning

Feature description -

The platform implements content versioning features, allowing instructors to manage and track multiple versions of course materials. By providing version control, the platform ensures that updates, revisions, and changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily revert to previous versions if needed.

Userstory title -

As an instructor, I want content versioning features so that I can manage and track changes to my course materials.

Acceptance criteria -

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.', 'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.',

'Notifications are sent for significant content updates.']

Generated Test Cases:

Test Case 1:

Test Case ID: OLPTCV1

Test Case Title: Verify that instructors can create and save multiple versions of content

Test Case Description: Ensure that the content versioning feature enables instructors to create, edit and save various versions of course materials.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in
- Course contents are available for editing

Test Data: Course content details

Test Steps:

1. Go to the course content management page
2. Select a course content to edit
3. Make changes to the content
4. Save the content as a new version

Postconditions:

- The new version of content is saved

Expected Result: The system allows the instructor to create and save multiple versions of a course content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPTCV2

Test Case Title: Verify that version history is maintained and accessible

Test Case Description: Ensure that the system records and displays a history of different versions of course content for instructors to review.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in
- Various versions of course content exist

Test Data: No test data needed

Test Steps:

1. Go to the course content management page
2. Select a course content with multiple versions
3. Access the version history

Postconditions:

- Version history of the content is accessible

Expected Result: The system maintains and displays the version history of course content for instructors.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPTCV3

Test Case Title: Verify that instructors can compare different versions of content

Test Case Description: Ensure that instructors can perform a comparison between different versions of a particular course content.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in
- At least two versions of a course content exist

Test Data: No test data needed

Test Steps:

1. Go to the course content management page
2. Select a course content with multiple versions
3. Select two versions to compare

Postconditions:

- Comparison result is generated

Expected Result: The system enables instructors to compare different versions of a course content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPTCV4

Test Case Title: Verify that previous versions can be restored if needed

Test Case Description: Ensure that instructors can restore to a previous version of content when necessary.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in
- At least two versions of a course content exist

Test Data: No test data needed

Test Steps:

1. Go to the course content management page
2. Select a course content with multiple versions
3. Select a previous version to restore

Postconditions:

- Content is restored to the chosen version

Expected Result: The system allows instructors to revert to a previous version of a course content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPTCV5

Test Case Title: Verify that notifications are sent for significant content updates

Test Case Description: Ensure that in case of significant changes to the course content, notifications are generated and sent to the related parties.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Significant changes are made to a course content

Test Data: No test data needed

Test Steps:

1. Go to the course content management page
2. Make significant changes to a course content
3. Save the changes

Postconditions:

- Notification is sent for the content update

Expected Result: The system sends notifications post significant updates on course content.

Severity: Major

Type of Testing: Functional

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 Case 1:

Test Case ID: API1

Test Case Title: Verify that APIs are available for key platform functionalities

Test Case Description: This test case is to verify that APIs are available for each of the key functionalities of the online learning platform.

Test Suite: API Integration

Test Priority: High

Preconditions:

- Platform is up and running
- Developer account exists and is functional

Test Data: API keys, Endpoints

Test Steps:

1. Log in to the developer account
2. Access the list of available APIs
3. Check for the availability of APIs for each key platform functionalities

Postconditions:

- API endpoints are checked
- Response from the system is recorded

Expected Result: APIs are available for each of the key functionalities of the online learning platform.

Severity: Major

Type of Testing: API Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: API2

Test Case Title: Verify that comprehensive API documentation is provided

Test Case Description: Verification of the availability and the quality of API documentation in assisting third-party application integration.

Test Suite: API Integration

Test Priority: High

Preconditions:

- Developer account exists and is functional

Test Data: No test data needed

Test Steps:

1. Log in to the developer account
2. Access the API documentation
3. Review the information provided in the API documentation

Postconditions:

- API documentation has been reviewed

Expected Result: The API documentation is comprehensive and provides clear instruction for third-party application integration.

Severity: Major

Type of Testing: Documentation Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: API3

Test Case Title: Verify that secure access protocols are implemented for APIs

Test Case Description: Ensure the API provides secure access protocols for third-party applications to maintain data integrity and privacy.

Test Suite: API Integration

Test Priority: High

Preconditions:

- The third-party application has proper authentication details

Test Data: API keys, Tokens, Credentials of third-party application

Test Steps:

1. Use API key/token to authenticate third-party application
2. Attempt to access the API
3. Observe the security measures while accessing API

Postconditions:

- Access attempt is made
- Security protocols for the API are observed

Expected Result: Secure access protocols are implemented correctly providing secure access to the API.

Severity: Critical

Type of Testing: Security Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: API4

Test Case Title: Verify that third-party applications can authenticate and access APIs

Test Case Description: To ensure that third-party applications are able to authenticate and have access to the platform's APIs successfully.

Test Suite: API Integration

Test Priority: High

Preconditions:

- Successful integration of third-party application
- Proper authentication details available

Test Data: API keys, Tokens, Credentials of third-party application

Test Steps:

1. Use API key/token to authenticate third-party application
2. Request for an API endpoint
3. Validate the successful response and data from API call

Postconditions:

- API call is made
- Response from the system is recorded

Expected Result: The third-party applications are able to authenticate and have successfully accessed the platform's APIs.

Severity: Critical

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: API5

Test Case Title: Verify that rate limiting and monitoring features are in place for APIs

Test Case Description: To verify that the API has appropriate rate limiting and monitoring features to prevent abuse and to track usage.

Test Suite: API Integration

Test Priority: Medium

Preconditions:

- Multiple API calls are ready for execution

Test Data: API keys, Endpoints

Test Steps:

1. Make multiple simultaneous requests to an API endpoint
2. Exceed the expected rate limit
3. Observe the system's response

Postconditions:

- Rate limiting and monitoring are checked
- System's response is recorded

Expected Result: The system properly enforces rate limits and has adequate monitoring in place for tracking API usage.

Severity: Major

Type of Testing: Performance Testing

Test Case Approach: Negative

Data - 72

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features, the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

Test Case 1:

Test Case ID: DA01

Test Case Title: Verify that data visualization tools are available for analysis.

Test Case Description: This test case verifies that the online learning platform makes data visualization tools available to administrators for analyzing learner performance.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in
- There are existing learner data in the system

Test Data: Learner performance data

Test Steps:

1. Navigate to the Data Analytics and Insights section
2. Open the data visualization tools
3. Apply different visualization techniques on learner data

Postconditions:

- Data visualizations are displayed

Expected Result: The system provides various visualization tools for administrators to analyze learner performance.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: DA02

Test Case Title: Verify that reports can be generated for learner performance metrics.

Test Case Description: This test case checks if the platform allows administrators to generate reports on learner performance metrics.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in
- Existing learner performance metrics in the system

Test Data: Learner performance metrics

Test Steps:

1. Navigate to Data Analytics and Insights section
2. Select learner performance metrics

3. Generate a report

Postconditions:

- The report is generated and can be viewed

Expected Result: The system allows administrators to generate reports on learner performance metrics.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: DA03

Test Case Title: Verify the provision of real-time analytics for engagement and outcomes.

Test Case Description: This test case validates that the system provides real-time analytics on learner engagement and outcomes to administrators.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator is logged in
- Learners are actively using the platform

Test Data: Real-time learner engagement and outcomes data

Test Steps:

1. Navigate to Data Analytics and Insights section
2. Access real-time analytics

Postconditions:

- Real-time analytics are displayed

Expected Result: The system provides real-time analytics on learner engagement and outcomes to administrators.

Severity: Major

Type of Testing: Performance Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: DA04

Test Case Title: Verify that data can be exported for further analysis.

Test Case Description: This test verifies that the system allows the export of data for administrators to conduct further analysis.

Test Suite: Data Analytics and Insights

Test Priority: Medium

Preconditions:

- Administrator logged in
- There is existing data to export

Test Data: Various datasets from the platform

Test Steps:

1. Navigate to Data Analytics and Insights section
2. Choose a dataset to export
3. Export the data in a chosen format

Postconditions:

- Data is exported successfully

Expected Result: The system successfully exports the selected data in the chosen format.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: DA05

Test Case Title: Verify that notifications are sent for significant data trends or anomalies.

Test Case Description: This test verifies that the system sends notifications to administrators when significant data trends or anomalies are detected.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

- Administrator logged in
- Significant data trends or anomalies detected

Test Data: No test data needed

Test Steps:

1. Navigate to Data Analytics and Insights section
2. Set triggers for anomalies
3. Check if notification is received when trigger is set off

Postconditions:

- Notification is received

Expected Result: The system sends a notification alerting the administrator of significant data trends or anomalies.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Roles and Permissions

Feature description -

The platform implements a user roles and permissions system, allowing administrators to define and manage different levels of access and functionality for various user groups. This feature ensures that users have appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing security and user experience.

Userstory title -

As an administrator, I want to manage user roles and permissions so that I can control access to platform features.

Acceptance criteria -

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are notified of their assigned roles and permissions.']

Generated Test Cases:

Test Case 1:

Test Case ID: TC001_OLP

Test Case Title: Verify that administrators can create and assign user roles.

Test Case Description: Ensure that the system allows administrators to create new user roles and assign these

roles to users.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- Administrator is logged on
- Users are registered in system

Test Data: User profile information, New role information

Test Steps:

1. Go to user management panel
2. Create a new user role
3. Assign the role to a user

Postconditions:

- New user role is created
- User is assigned the new role

Expected Result: The system creates the new user role, assigns it to the selected user and logs the changes.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: TC002_OLP

Test Case Title: Verify that permissions are customizable for each role.

Test Case Description: Ensure that the system allows administrators to customize permissions for each user role.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- Administrator is logged on
- User roles are already created

Test Data: Permission information

Test Steps:

1. Go to user management panel
2. Select a user role
3. Customize permissions for the role

Postconditions:

- Permissions are updated

Expected Result: The system updates the permissions for the selected role and logs the changes.

Severity: Major

Type of Testing: Security

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: TC003_OLP

Test Case Title: Verify that users are restricted to functionalities based on their roles.

Test Case Description: Determines if the system effectively controls access to functionalities based on user roles.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- Users are logged in with assigned roles

- User roles are assigned with permissions

Test Data: No test data needed

Test Steps:

1. Login as a user with a role
2. Navigate through the application
3. Try accessing a functionality not allowed for the role

Postconditions:

- Access to certain functionalities is restricted

Expected Result: The system restricts the user from accessing certain functionalities, according to their role.

Severity: Critical

Type of Testing: Security

Test Case Behaviour: Negative

Test Case 4:

Test Case ID: TC004_OLP

Test Case Title: Verify that changes to roles and permissions are logged.

Test Case Description: Ensure that the system logs all changes made to roles and permissions.

Test Suite: User Roles and Permissions

Test Priority: Low

Preconditions:

- Administrator is logged on
- Changes were made to user roles or permissions

Test Data: Change logs

Test Steps:

1. Go to change management panel

2. Check logs for changes to user roles and permissions

Postconditions:

- Updates are logged

Expected Result: The system logs the changes made to user roles and permissions.

Severity: Minor

Type of Testing: Data-Driven

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: TC005_OLP

Test Case Title: Verify that users are notified of their assigned roles and permissions.

Test Case Description: Check if the system notifies users when they are assigned roles and permissions.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- Users are assigned roles and permissions

Test Data: Notification settings

Test Steps:

1. Assign a role to a user
2. Check if user received a notification

Postconditions:

- Notifications are sent to the user

Expected Result: The system sends a notification to the user once they are assigned a role and its associated permissions.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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: 001

Test Case Title: Verify that learners can rate courses and instructors on a scale

Test Case Description: This test case verifies that learners can provide ratings to courses and instructors on a given scale.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- Learner is logged in.
- Learner has completed or is currently enrolled in a course.

Test Data: Course ID, Instructor ID, Rating scale (1-5)

Test Steps:

1. Open a course
2. Locate the rating system
3. Select a rating on the scale for the course and instructor
4. Save the rating

Postconditions:

- Rating is saved

Expected Result: The system should successfully save the rating for both the course and the instructor.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: 002

Test Case Title: Verify that feedback forms are available for detailed comments

Test Case Description: This test case verifies that feedback forms are accessible to learners for providing detailed comments.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

- Learner is logged in
- Learner has completed or is currently enrolled in a course

Test Data: No test data needed

Test Steps:

1. Open a course
2. Locate the feedback form
3. Fill in the feedback form with comments
4. Submit the feedback form

Postconditions:

- Feedback form is submitted

Expected Result: The system should successfully submit the detailed feedback provided by the learner.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: 003

Test Case Title: Verify that instructors can view and respond to feedback

Test Case Description: This test case verifies that instructors can view feedback and have the functionality to respond.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- Instructor is logged in

Test Data: Feedback comments

Test Steps:

1. Open the instructor dashboard
2. Locate the feedback section
3. Open a piece of feedback
4. Respond to feedback
5. Save response

Postconditions:

- Response to feedback is saved

Expected Result: The system should enable instructors to view and respond to feedback successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: 004

Test Case Title: Verify that aggregated ratings are displayed publicly

Test Case Description: This test case verifies that the aggregated ratings for courses and instructors are visible to all users.

Test Suite: User Feedback and Rating System

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Aggregated ratings data

Test Steps:

1. Select a course or instructor
2. Locate the rating section
3. Check the aggregated ratings

Postconditions:

- Ratings are viewable

Expected Result: The system should accurately display the aggregated ratings for all users to see.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: 005

Test Case Title: Verify that anonymous feedback options are available

Test Case Description: This test case checks if users have the option to submit feedback anonymously.

Test Suite: User Feedback and Rating System

Test Priority: Low

Preconditions:

- Learner is logged in
- Learner has completed or is currently enrolled in a course

Test Data: Availability of anonymous checkbox or feature

Test Steps:

1. Open a course
2. Locate the feedback form

3. Check the option for anonymous feedback

4. Submit the feedback form

Postconditions:

- Anonymous feedback is submitted

Expected Result: The system should offer an option to submit feedback anonymously and successfully receive such feedback.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 75

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Social Media Integration

Feature description -

The platform supports social media integration, allowing learners and instructors to share their achievements, courses, and updates on social media platforms. This feature promotes engagement and community building, enhancing the visibility of the platform's offerings.

Userstory title -

As a learner, I want to share my achievements on social media so that I can showcase my progress and connect

with peers.

Acceptance criteria -

['Learners can connect their social media accounts to the platform.', 'Achievements and course completions can be shared on social media.', 'Privacy settings control the visibility of shared content.', 'Instructors can promote courses and updates through social media.', 'Analytics track the reach and engagement of shared content.']

Generated Test Cases:

Test Case 1:

Test Case ID: SMI_001

Test Case Title: Verify that learners can connect their social media accounts to the platform

Test Case Description: This test case is designed to confirm that the platform supports social media integration for learners, allowing them to connect their accounts successfully.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- Learner has a valid account on the platform and on the social media platform
- Learner is logged in

Test Data: Learner's social media account details

Test Steps:

1. Navigate to profile settings.
2. Click on the 'Connect Social Media' option.
3. Enter the appropriate social media account details.
4. Click on the 'Connect' button.

Postconditions:

- Learner's account is linked to social media account

Expected Result: Social media account is successfully linked to the learner's account on the platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMI_002

Test Case Title: Verify that achievements and course completions can be shared on social media

Test Case Description: This test case verifies that the platform allows learners to share their achievements and course completions on their linked social media accounts.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- Learner has finished a course or earned an achievement
- Social media account is linked

Test Data: Details of the course/achievement

Test Steps:

1. Navigate to the 'Achievements' section.
2. Select the achievement to be shared.
3. Click on the 'Share' button and select the social media platform.

Postconditions:

- Achievement/course completion details are shared on social media

Expected Result: The platform allows sharing of achievements/course completions on the selected social media platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3:

Test Case ID: SMI_003

Test Case Title: Verify that privacy settings control the visibility of shared content

Test Case Description: This test case ensures that the platform allows learners to control the visibility of content (achievements and course completions) shared on social media via their privacy settings.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- Social media account is linked
- There is content shared on social media

Test Data: No test data needed

Test Steps:

1. Go to profile settings.
2. Navigate to privacy settings of shared content.
3. Change the visibility settings and save.

Postconditions:

- The visibility of shared content is updated according to privacy settings

Expected Result: The system updates the visibility of shared content based on the new privacy settings.

Severity: Major

Type of Testing: Security Testing

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMI_004

Test Case Title: Verify that instructors can promote courses and updates through social media

Test Case Description: This test case ensures that instructors can share updates about their courses and promote them on their linked social media accounts.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- Instructor has a valid account on the platform and on the social media platform
- Instructor's social media accounts are linked

Test Data: Course details

Test Steps:

1. Navigate to the 'Courses' section.
2. Select the course to promote.
3. Click on the 'Share' button and select the social media platform.

Postconditions:

- The course is shared on social media

Expected Result: The platform allows instructors to share and promote their courses on the connected social media platform.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMI_005

Test Case Title: Verify that analytics track the reach and engagement of shared content

Test Case Description: This test case checks if the platform tracks and provides analytics data related to the reach and engagement of content shared on social media.

Test Suite: Social Media Integration

Test Priority: Low

Preconditions:

- Content is shared on social media

Test Data: No test data needed

Test Steps:

1. Navigate to the 'Analytics' tab.
2. Check the analytics data related to shared content, including reach and engagement metrics.

Postconditions:

- Analytics data for shared content is displayed

Expected Result: The platform accurately tracks and displays the analytics related to reach and engagement of shared content.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Data - 76

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Activity Tracking

Feature description -

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity

levels and progress towards fitness goals.

Userstory title -

As a user, I want to track my activities so that I can monitor my daily physical activity levels.

Acceptance criteria -

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']

Generated Test Cases:

Test Case 1:

Test Case ID: AT001

Test Case Title: Verify that the apps record data on distance, duration, pace, and calories burned

Test Case Description: This test case is designed to verify that the Activity Tracking feature properly records the necessary statistics during the user's workout.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

- User is logged in
- User is ready to start a workout
- Location services are enabled

Test Data: User's workout data

Test Steps:

1. Select the 'Start Workout' button
2. Perform a predefined workout
3. Save the workout

Postconditions:

- Workout data is recorded

Expected Result: The app records and displays data on distance, duration, pace, and calories burned.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: AT002

Test Case Title: Verify that users can view activity summaries

Test Case Description: This test is to ensure that post-workout, users can view a summary of their activity.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has completed a workout

Test Data: No test data needed

Test Steps:

1. Select the 'Workout History' button
2. Select a specific workout

Postconditions:

- Workout summary is displayed

Expected Result: The app displays a summary of the user's workout.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: AT003

Test Case Title: Verify that the app stores data for historical analysis

Test Case Description: This test is to verify the app's ability to store and display historical workout data.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has completed multiple workouts

Test Data: Workout history data

Test Steps:

1. Select the 'Workout History' button

Postconditions:

- Workout history is displayed

Expected Result: The app displays a log of all past workouts.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: AT004

Test Case Title: Verify that activity tracking can be paused and resumed

Test Case Description: This test case checks if users can pause and resume tracking during the middle of a workout.

Test Suite: Activity Tracking

Test Priority: High

Preconditions:

- User is logged in
- User is in the middle of a workout

Test Data: User's workout data

Test Steps:

1. Select the 'Pause' button during a workout
2. Select the 'Resume' button

Postconditions:

- Workout tracking is paused and resumed

Expected Result: The app can pause and resume tracking the workout without errors.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: AT005

Test Case Title: Verify that user receive notifications for milestones

Test Case Description: This test case tests if the app sends notifications to users once they have reached a milestone in their fitness journey.

Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has completed a workout
- User has reached a milestone

Test Data: No test data needed

Test Steps:

1. Check for a notification after reaching a milestone

Postconditions:

- Milestone notification is received

Expected Result: The app sends a notification when milestones are reached.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 77

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

GPS Tracking

Feature description -

Integrated GPS tracking allows users to accurately track outdoor activities and routes, providing detailed maps, elevation profiles, and location-based metrics. Users can visualize their workout routes, analyze performance data, and explore new routes for added variety and challenge.

Userstory title -

As a user, I want to use GPS tracking so that I can accurately track my outdoor activities and routes.

Acceptance criteria -

['GPS tracking accurately records route data.', 'Users can view detailed maps and elevation profiles.', 'Performance data includes location-based metrics.', 'Routes can be saved and revisited.', 'Notifications are provided for deviations from planned routes.']

Generated Test Cases:

Test Case 1:

Test Case ID: GPS001

Test Case Title: Verify that GPS tracking accurately records route data

Test Case Description: Ensure that the Fitness Tracker App's GPS tracking function accurately records the user's route data during outdoor activities.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- User is logged in
- GPS is enabled on the device

Test Data: User's outdoor activity data

Test Steps:

1. Initiate an outdoor activity like running or cycling
2. Allow the app to track the route using GPS
3. End the activity and review the generated route data

Postconditions:

- Route data is recorded and saved

Expected Result: The Fitness Tracker App accurately records and displays the user's route data.

Severity: Major

Type of Testing: Functional, Integration

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: GPS002

Test Case Title: Verify that users can view detailed maps and elevation profiles

Test Case Description: Ensure that the Fitness Tracker App provides detailed maps and elevation profiles based on the tracked GPS data.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- User is logged in
- GPS is enabled on the device
- GPS data from a previous workout is available

Test Data: User's workout data with GPS

Test Steps:

1. Open the app and go to the Workouts history section
2. Open a workout with GPS tracking
3. Review the presented map and elevation profile

Postconditions:

- Detailed map and elevation profile are displayed

Expected Result: The Fitness Tracker App successfully presents a detailed map and elevation profile.

Severity: Major

Type of Testing: Functional, Usability

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: GPS003

Test Case Title: Verify that performance data includes location-based metrics

Test Case Description: Ensure that the Fitness Tracker App integrates location-based metrics into the performance data provided to the user.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- User is logged in
- GPS is enabled on the device

Test Data: User's workout data with GPS

Test Steps:

1. Initiate an outdoor activity with GPS tracking
2. Complete the activity and review the location-based metrics in the performance data

Postconditions:

- The performance data includes location-based metrics

Expected Result: The Fitness Tracker App successfully incorporates location-based metrics into the performance data.

Severity: Major

Type of Testing: Functional, Integration

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: GPS004

Test Case Title: Verify that routes can be saved and revisited

Test Case Description: Ensure that the Fitness Tracker App allows the user to save workout routes and revisit them at a later time.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- GPS is enabled on the device
- A workout with GPS tracking has been completed

Test Data: Saved workout route data

Test Steps:

1. Open the app and go to the Workouts history section
2. Open a previously saved workout with GPS tracking
3. Review the saved route data

Postconditions:

- User revisits the saved route

Expected Result: The Fitness Tracker App successfully retrieves and displays saved route data.

Severity: Major

Type of Testing: Functional, Performance

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: GPS005

Test Case Title: Verify that notifications are provided for deviations from planned routes

Test Case Description: Ensure that the Fitness Tracker App is able to detect when the user deviates from a planned route and sends a notification about the same.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- GPS is enabled on the device
- A workout with a planned route is setup

Test Data: Planned route data

Test Steps:

1. Start the workout with a planned route
2. Deviate from the planned route during the workout
3. Wait for the deviation notification from the app

Postconditions:

- Notification for deviation is sent

Expected Result: The Fitness Tracker App detects the deviation from the planned route and sends a notification.

Severity: Minor

Type of Testing: Functional, Integration

Test Case Behaviour: Positive

Data - 78

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Heart Rate Monitoring

Feature description -

The app monitors heart rate in real-time using built-in sensors or compatible wearable devices, providing insights into exercise intensity, recovery, and cardiovascular health. Heart rate data helps users optimize their workouts, maintain target heart rate zones, and track improvements in fitness over time.

Userstory title -

As a user, I want to monitor my heart rate so that I can optimize my workouts and track my cardiovascular health.

Acceptance criteria -

['Heart rate data is recorded in real-time.', 'Users can set and monitor target heart rate zones.', 'Historical heart rate data is stored and viewable.', 'Alerts are provided for abnormal heart rate readings.', 'Data is compatible with various wearable devices.']

Generated Test Cases:

Test Case 1:

Test Case ID: HRM001

Test Case Title: Verify that heart rate data is recorded in real-time

Test Case Description: Ensure that the system accurately records and updates heart rate data in real time when a user's heart rate changes.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in
- Wearable device is connected and heart rate monitor is activated

Test Data: User's heart rate data

Test Steps:

1. Start heart rate monitoring
2. Verify that heart rate data displays on the app
3. Raise physical activity level to increase heart rate (ex: begin jogging)
4. Verify that the app reflects the change in heart rate

Postconditions:

- Heart rate data is updated real-time

Expected Result: The system records and reflects user's heart rate data in real-time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: HRM002

Test Case Title: Verify that users can set and monitor target heart rate zones

Test Case Description: Ensure that the system allows users to set and monitor their cardio zones, keeping them informed about their current workout intensity.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in

Test Data: Target heart rate zones

Test Steps:

1. Access heart rate settings
2. Set a target heart rate zone
3. Save changes
4. Conduct a workout to verify if heart rate zone is displayed

Postconditions:

- Target heart rate zone is set

Expected Result: The system allows users to set target heart rate zones and displays the information during workouts.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: HRM003

Test Case Title: Verify that historical heart rate data is stored and viewable

Test Case Description: Ensure that the system stores past heart rate data and allows users to view historical data trends.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

- User has conducted previous workouts with the app
- User is logged in

Test Data: Historic heart rate data

Test Steps:

1. Access user's heart rate history
2. Scroll through different time periods and verify data is displayed

Postconditions:

- Heart rate history is viewable

Expected Result: The system displays accurate historical heart rate data.

Severity: Major

Type of Testing: Functionality

Test Case Approach: Positive

Test Case 4:

Test Case ID: HRM004

Test Case Title: Verify that alerts are provided for abnormal heart rate readings

Test Case Description: To ensure that the system generates and displays alerts when heart rate readings fall outside the normal range.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

- User is logged in
- Heart rate is being monitored

Test Data: Heart rate data

Test Steps:

1. Simulate an abnormally high or low heart rate reading
2. Check if the app generates an alert

Postconditions:

- Abnormal heart rate alert is generated

Expected Result: The system generates an alert for abnormal heart rate readings.

Severity: Critical

Type of Testing: Negative Testing

Test Case Approach: Negative

Test Case 5:

Test Case ID: HRM005

Test Case Title: Verify that data is compatible with various wearable devices

Test Case Description: Ensure that the app correctly gathers heart rate data from a variety of wearable devices.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

- User owns different wearable devices
- User is logged in

Test Data: Heart rate data from various wearable devices

Test Steps:

1. Connect a wearable device to the app
2. Start a workout and verify that heart rate data is correctly displayed
3. Switch to a different wearable device and repeat the test

Postconditions:

- Heart rate data is correctly displayed from various devices

Expected Result: The system correctly collects and displays heart rate data from various compatible

wearable devices.

Severity: Major

Type of Testing: Compatibility Testing

Test Case Approach: Positive

Data - 79

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Calorie Tracking

Feature description -

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical activities, helping them maintain a healthy balance between energy intake and expenditure. This feature promotes mindful eating habits and supports weight management goals.

Userstory title -

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and expenditure.

Acceptance criteria -

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie

information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.')

Generated Test Cases:

Test Case 1:

Test Case ID: CT1

Test Case Title: Verify that users can log daily food intake and calorie consumption

Test Case Description: The test aims to ensure that the users have the ability to keep a record of their daily food consumption and their corresponding calorie intake within the app.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged into the Fitness Tracker App
- User has consumed food

Test Data: Food consumption details along with their calorific values

Test Steps:

1. Click on the Calorie Tracking Feature
2. Enter the details of the consumed food items
3. Enter the respective calorific values
4. Click 'Save'

Postconditions:

- The food intake entry is saved

Expected Result: The app successfully logs the user's food intake and calorie consumption.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: CT2

Test Case Title: Verify that the app calculates and displays total calories consumed and burned

Test Case Description: This test is designed to ensure that the app accurately calculates and displays the total calories consumed and expended by the user.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has logged food intake and physical activities

Test Data: User's food and physical activity logs

Test Steps:

1. Click on 'Calorie Summary'
2. Verify the total calories consumed and burned

Postconditions:

- The total calories consumed and burned should be displayed

Expected Result: The app correctly calculates and displays the total calories consumed and burnt.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: CT3

Test Case Title: Verify that users can set daily calorie goals and track progress

Test Case Description: This test checks whether the app allows users to set daily calorie goals and monitor

their progress towards achieving these goals.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Daily calorie goal value

Test Steps:

1. Go to 'Settings'
2. Enter daily calorie goal
3. Save changes
4. Navigate to 'Calorie Summary'
5. Check the progress of the calorie goal

Postconditions:

- Calorie goal is updated

Expected Result: The app correctly allows the user to set a daily calorie goal and track their progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: CT4

Test Case Title: Verify that food items and calorie information are searchable in the database

Test Case Description: This test ensures that users can search for specific food items or calorie data within the app's database.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Food item name or calorific value

Test Steps:

1. Navigate to 'Food Database'
2. Enter the food item name or calorific value in the 'Search' field
3. Click the 'Search' button

Postconditions:

- List of matching food items and/or calorific values appears

Expected Result: The app correctly displays items from the database that match the user's search criteria.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: CT5

Test Case Title: Verify that users receive alerts for exceeding calorie goals

Test Case Description: This test aims to ensure that the app alerts the user when their daily calorie consumption exceeds the set goal.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has set a daily calorie goal
- User's calorie consumption for the day exceeds the set goal

Test Data: Dietary data that results in daily caloric intake exceeding the set goal

Test Steps:

1. Navigate to the 'Food Database'
2. Add food items such that total daily caloric intake exceeds the set goal
3. Wait for the alert from the app

Postconditions:

- Alert notification for exceeding calorie goal is received

Expected Result: The app alerts the user when their daily calorie consumption exceeds the set goal.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Data - 80

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and

recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.', 'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

Test Case 1:

Test Case ID: ST01

Test Case Title: Verify that the app can record sleep data accurately

Test Case Description: Ensure that the app records accurate sleep data including duration and quality.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has a compatible device
- The device has enough battery life for overnight usage

Test Data: User sleep data

Test Steps:

1. User goes to sleep with the device on
2. User wakes up and checks the sleep data on the app

Postconditions:

- Sleep data is saved in the app history

Expected Result: Accurate sleep data for duration and quality is displayed

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: ST02

Test Case Title: Verify that users can view detailed sleep stages and patterns

Test Case Description: Ensure that the app can provide detailed visuals of sleep stages and patterns

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

- User has sleep data recorded from previous night

Test Data: User sleep data

Test Steps:

1. User opens the app
2. User goes to sleep history tab
3. User checks the detailed sleep stage and pattern chart

Postconditions:

- The sleep stage and pattern chart is updated

Expected Result: The app provides a consistent and accurate chart of sleep stages and patterns

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: ST03

Test Case Title: Verify that app provides insights into sleep disturbances

Test Case Description: Ensure that the app highlights any potential sleep disturbances and issues

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- User has sleep data recorded from previous night

Test Data: Sleep data

Test Steps:

1. User opens app
2. User checks the sleep analysis tab
3. Check for any highlighted sleep disturbances

Postconditions:

- Sleep disturbance data is updated

Expected Result: The app provides accurate information on sleep disturbances

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: ST04

Test Case Title: Verify that users receive recommendations for improving sleep quality

Test Case Description: Ensure that the app provides actionable suggestions for better sleep quality

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- User has sleep data recorded from previous night

Test Data: Sleep data

Test Steps:

1. User opens app
2. User navigates to recommendations tab
3. User reads through the provided sleep-quality improvement recommendations

Postconditions:

- Recommendations are provided based on sleep data

Expected Result: The app provides beneficial recommendations for better sleep quality

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: ST05

Test Case Title: Verify that sleep data can be viewed in daily, weekly, and monthly summaries

Test Case Description: Ensure that the app aggregates and displays sleep data in daily, weekly, and monthly formats

Test Suite: Sleep Tracking

Test Priority: Medium

Preconditions:

- User has sleep data recorded for more than a week

Test Data: Sleep data for multiple days/weeks

Test Steps:

1. User opens app

2. User goes to sleep history tab
3. User checks daily, weekly, and monthly summaries

Postconditions:

- Summarized data is displayed

Expected Result: The app cleanly aggregates sleep data into daily, weekly, and monthly summaries

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 81

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Logging

Feature description -

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify trends in performance and adherence to training programs.

Userstory title -

As a user, I want to log my workouts so that I can track my progress and improve my performance.

Acceptance criteria -

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.', 'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can filter logs by date and workout type.']

Generated Test Cases:

Test Case 1:

Test Case ID: WLog01

Test Case Title: Verify that users can log workouts with detailed information.

Test Case Description: The test case verifies that users can log individual workouts, specifying the exercise type, duration, intensity, and notes or comments.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User logged into the fitness tracker app
- User has completed a workout

Test Data: Workout details (Type of workout, duration, intensity, and any additional notes or comments)

Test Steps:

1. Open the workout logging feature
2. Click on "Log a workout"
3. Enter all the required workout details
4. Save the workout log

Postconditions:

- Workout log is created

Expected Result: The system allows users to log workouts, saving all entered details accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: WLog02

Test Case Title: Verify that users can edit workout entries.

Test Case Description: The test case verifies that users can edit previously logged workouts with updated details.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User logged into the fitness tracker app
- User has logged a workout

Test Data: Updated workout details

Test Steps:

1. Open the workout log
2. Select a workout to edit
3. Update the workout details
4. Save the changes

Postconditions:

- Workout log is updated

Expected Result: The system should allow users to edit workout logs and save the updated details successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: WLog03

Test Case Title: Verify that users can delete workout entries.

Test Case Description: The test case verifies that users can delete any previously logged workouts.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User logged into the fitness tracker app
- User has logged a workout

Test Data: No test data needed

Test Steps:

1. Open the workout log
2. Select a workout to delete
3. Confirm deletion

Postconditions:

- Selected workout log is deleted

Expected Result: The system should successfully delete the selected workout log.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: WLog04

Test Case Title: Verify that logged workout data is stored and viewable for progress-tracking.

Test Case Description: The test case verifies that logged workouts are stored and users can view them to track progress.

Test Suite: Workout Logging

Test Priority: High

Preconditions:

- User logged into the fitness tracker app
- User has logged a workout

Test Data: No test data needed

Test Steps:

1. Open the workout log
2. Browse through different logged workouts

Postconditions:

- User is able to view stored workout logs

Expected Result: The system should store all logged workouts and make them accessible for tracking.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: WLog05

Test Case Title: Verify that users can filter logs by date and workout type.

Test Case Description: The test case verifies that users can filter workout logs based on date and workout type for easier access.

Test Suite: Workout Logging

Test Priority: Medium

Preconditions:

- User is logged into the fitness tracker app
- User has logged a workout

Test Data: Date and Workout Type

Test Steps:

1. Open the workout log
2. Apply filters for date and workout type
3. Verify the logs displays accordingly

Postconditions:

- Workout logs are filtered and displayed based on selected criteria

Expected Result: The system should allow users to filter workout logs by date and workout type.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 82

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Goal Setting

Feature description -

Users can set personalized fitness goals based on activity level, weight loss targets, performance milestones, or health objectives. The app provides progress tracking and motivational feedback to help users stay focused and committed to achieving their goals.

Userstory title -

As a user, I want to set fitness goals so that I can stay motivated and track my progress.

Acceptance criteria -

['Users can set goals for various fitness metrics.', 'Progress tracking is available for each goal.', 'Users receive motivational feedback and reminders.', 'Goals can be adjusted or updated as needed.', 'The app provides a summary of goal attainment.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT001

Test Case Title: Verify that users can set goals for various fitness metrics

Test Case Description: Ensure that the app allows users to set personalized fitness goals based on different fitness metrics like steps taken, calories burned, distance traveled, and heart rate.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User is registered and logged in to the app
- User has basic knowledge about fitness metrics

Test Data: Fitness goals data (e.g., 10,000 steps a day, burn 500 calories a day)

Test Steps:

1. Log in to the app
2. Navigate to "Goal Setting" section
3. Enter desired fitness metrics goals

4. Save changes

Postconditions:

- Fitness metrics goals are saved in the system

Expected Result: The app saves the fitness goals and updates them in the user's profile.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT002

Test Case Title: Verify that progress tracking is available for each goal

Test Case Description: Ensure that the Fitness Tracker App provides users with a mechanism to track their progress towards each of their set fitness goals.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User has set fitness goals

Test Data: No test data needed

Test Steps:

1. Open the app
2. Go to the "Goal Setting" section
3. View the progress tracking for each goal

Postconditions:

- Fitness goals' progress is measurable and visible

Expected Result: The app shows the progress of each fitness goal.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT003

Test Case Title: Verify that users receive motivational feedback and reminders

Test Case Description: Ensure that the app generates motivational feedback and reminders based on the user's progress in achieving their fitness goals.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User has set fitness goals

Test Data: No test data needed

Test Steps:

1. Open the app
2. Go to the "Goal Setting" section
3. Check for motivational feedback and reminders

Postconditions:

- User receives feedback and reminders

Expected Result: The app sends motivational feedback and reminders based on the progress of fitness goals.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT004

Test Case Title: Verify that goals can be adjusted or updated as needed

Test Case Description: Ensure that the app allows users to adjust or update their fitness goals as needed to adapt to their changing fitness levels and objectives.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User has set fitness goals

Test Data: Updated fitness goals data

Test Steps:

1. Open the app
2. Go to the "Goal Setting" section
3. Choose a goal to adjust or update
4. Make changes and Save

Postconditions:

- Updated goals are saved in the system

Expected Result: The app allows the user to adjust or update their fitness goals as needed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT005

Test Case Title: Verify that the app provides a summary of goal attainment

Test Case Description: Ensure that the Fitness Tracker App provides a summary of goal attainment which shows an overview of users' fitness goal achievements.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

- User has set fitness goals

Test Data: No test data needed

Test Steps:

1. Open the app
2. Go to the "Goal Setting" section
3. Check for the summary of goal attainment

Postconditions:

- User can view the summary of goal attainment

Expected Result: The app displays a summary of goal attainment showing the achievement status of each fitness goal.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 83

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Workout Plans

Feature description -

The app offers pre-designed workout plans and training programs tailored to different fitness levels, goals, and preferences. Users can follow structured workout routines, receive guidance on exercise selection and progression, and track their adherence to the plan.

Userstory title -

As a user, I want to follow structured workout plans so that I can achieve my fitness goals with guided training.

Acceptance criteria -

['Users can select workout plans based on fitness levels.', 'Plans include detailed exercise routines and schedules.', 'Users can track adherence and progress within the plan.', 'The app provides notifications and reminders for workouts.', 'Users can adjust plans based on progress and preferences.']

Generated Test Cases:

Test Case 1:

Test Case ID: FTW1

Test Case Title: Verify that users can select workout plans based on fitness levels.

Test Case Description: This test case checks whether the app allows users to select workout plans tailored to their specific fitness levels.

Test Suite: Workout Plans

Test Priority: High

Precondition:

- User is registered and logged in
- Users have completed the fitness assessment

Test Data: User's fitness level data

Test Steps:

1. Open the app and log in
2. Navigate to the workout plans section
3. Check the available workout plans categorized by fitness levels
4. Choose a workout plan that matches the user's fitness level

Postconditions:

- User has selected the workout plan

Expected Result: The app allows the user to select a workout plan based on their fitness level.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: FTW2

Test Case Title: Verify that plans include detailed exercise routines and schedules.

Test Case Description: This test case verifies if the fitness plans in the app provide a well-detailed list of exercise routines including schedule, duration, repetition, and rest periods.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in
- User has selected a workout plan

Test Data: No test data needed

Test Steps:

1. Log in to the app
2. Go to the workout plans section

3. Open the selected workout plan
4. View the details of the exercise routines and schedules

Postconditions:

- User have viewed the details of exercise routines and schedules

Expected Result: The app provides detailed exercise routines and schedules as part of the selected workout plan.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: FTW3

Test Case Title: Verify that users can track adherence and progress within the plan.

Test Case Description: This test case tests the app's ability to allow users to track their workout completion, adherence to the schedule, and progress in their fitness goals.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is registered and logged in
- User has started a workout plan

Test Data: User's workout data, workout plan details

Test Steps:

1. Log in to the app
2. Access the tracking/progress section
3. Check the user's adherence and progress towards the selected workout plan

Postconditions:

- User has checked their adherence and progress within the plan

Expected Result: The app allows the user to track adherence and progress within the chosen workout plan.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: FTW4

Test Case Title: Verify that the app provides notifications and reminders for workouts.

Test Case Description: This test case checks if the app sends timely notifications and reminders related to the user's workout schedule.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is registered and logged in
- User has scheduled a workout

Test Data: User's workout schedule

Test Steps:

1. Log in to the app
2. Set up a workout schedule
3. Check for notifications and reminders related to the scheduled workout

Postconditions:

- User has received notifications and reminders for workouts

Expected Result: The app sends notifications and reminders for workouts based on the user's scheduled workout plan.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: FTW5

Test Case Title: Verify that users can adjust plans based on progress and preferences.

Test Case Description: This test case confirms if the app allows users to adjust their workout plans based on their progress and preferences.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User is registered and logged in
- User has an active workout plan

Test Data: User's fitness progress data, user preferences data

Test Steps:

1. Log in to the app
2. Access the active workout plan
3. Make adjustments to the workout plan based on progress and preferences
4. Save changes

Postconditions:

- User has adjusted the workout plan

Expected Result: The app allows the user to adjust their workout plan based on their progress and preferences.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 84

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Community Support

Feature description -

Users can connect with a community of like-minded individuals, share progress updates, participate in challenges, and offer support and encouragement to fellow users. Community support fosters accountability, motivation, and a sense of camaraderie among users striving towards similar fitness goals.

Userstory title -

As a user, I want to connect with a community so that I can share my progress and stay motivated.

Acceptance criteria -

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.', 'The app facilitates participation in community challenges.', 'Users can offer and receive support and encouragement.', 'Community activity and engagement are tracked.']

Generated Test Cases:

Test Case 1

Test Case ID: UTC-1

Test Case Title: Verify that users can join and participate in community groups

Test Case Description: Validation of the feature's capability to allow a user to join and participate in community groups

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is signed in
- Community group(s) is/are available

Test Data: User credentials and community information

Test Steps:

1. Navigate to community section
2. Browse available community groups
3. Select a group and join
4. Post a comment or participate in the group's activity

Postconditions:

- User joins the group and can interact within the group.

Expected Result: The user is able to join the selected community group and participate in activities

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: UTC-2

Test Case Title: Verify that users can share their progress updates and achievements

Test Case Description: Ensure that the application allows users to post their progress updates and completed achievements within the community

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- User is signed in
- User belongs to a community group

Test Data: User progress update input, achievements

Test Steps:

1. Navigate to a community group
2. Post a progress update or achievement

Postconditions:

- User progress update or achievement is visible to community

Expected Result: The posted progress update or achievement is successfully shared within the community group

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: UTC-3

Test Case Title: Verify that the app facilitates participation in community challenges

Test Case Description: Check if the system is capable of managing and promoting community challenges and allowing users to participate

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is signed in

- Community group(s) is/are available
- Challenges are available in the chosen community

Test Data: Challenge details

Test Steps:

1. Navigate to community section
2. Browse available challenges
3. Select a challenge and participate

Postconditions:

- User participation in the challenge is recorded

Expected Result: The user is able to participate in selected community challenges

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: UTC-4

Test Case Title: Verify that users can offer and receive support/encouragement

Test Case Description: To check if the system is capable of providing functionality to offer and receive support/encouragement

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- User is signed in
- User belongs to a community group

Test Data: Specific text for support or encouragement

Test Steps:

1. Navigate to a community group
2. Choose a user post
3. Post a supportive or encouraging comment

Postconditions:

- Supplementary or encouraging comment is visible under the selected post

Expected Result: The posted comment is posted successfully and is visible underneath the chosen post

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: UTC-5

Test Case Title: Verify that community activity and engagement are successfully tracked

Test Case Description: To ensure that the application successfully records and displays user engagement and community activities

Test Suite: Community Support

Test Priority: High

Preconditions:

- User is signed in
- User belongs to a community group

Test Data: No test data needed

Test Steps:

1. Navigate to profile/account settings
2. View activity or engagement metrics

Postconditions:

- User can see updated engagement metrics

Expected Result: The system successfully tracks and displays user engagement metrics accurately

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 85

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Wearable Devices

Feature description -

The app seamlessly integrates with popular wearable devices such as fitness trackers, smartwatches, and heart rate monitors, allowing users to sync activity data and access advanced features. Integration with wearables enhances data accuracy, usability, and convenience for users tracking their fitness activities on the go.

Userstory title -

As a user, I want to integrate my wearable devices so that I can sync my fitness data seamlessly.

Acceptance criteria -

['The app supports integration with multiple wearable devices.', 'Users can sync activity data from their wearables.', 'The app provides real-time data updates from wearables.', 'Wearable data includes steps, heart rate, and calories burned.', 'Users can troubleshoot connectivity issues with wearables.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT01

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: Verify the compatibility of the fitness tracker app with different brands and models of wearable devices.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User is logged in
- Have multiple wearable devices on hand (e.g., Fitbit, Apple Watch)

Test Data: Various wearable devices

Test Steps:

1. Go to the device setting of the app
2. Choose to pair a device
3. Select different brands or models

Postconditions:

- Wearable device is connected
- User's fitness data coming from wearable device is displayed in the app

Expected Result: The app supports integration with all tested wearable devices.

Severity: Major

Type of Testing: Compatibility

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: FT02

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: Testing the seamless synchronisation of user fitness data from wearable devices to the fitness tracker app.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User's wearable device is connected to the app

Test Data: Activity data from wearable device (e.g., steps taken, heart rate)

Test Steps:

1. Perform physical activities with the wearable device on
2. Open the app
3. Check if the data on the wearable device matches that on the app

Postconditions:

- Fitness data is synced

Expected Result: User's fitness data from wearable device is correctly and seamlessly synchronised to fitness tracker app.

Severity: Critical

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: FT03

Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: Ensure that the fitness tracker app provides real-time updates from the connected

wearable device.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User's wearable device is connected to the app

Test Data: Live activity data from wearable device (e.g., current heart rate)

Test Steps:

1. Perform physical activities with the wearable device on
2. Open the app
3. Check for real-time data updates on the app

Postconditions:

- Real-time data is provided

Expected Result: The fitness tracker app provides real-time updates from the wearable device.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: FT04

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: Test if the fitness tracker app correctly displays data such as steps taken, heart rate, and calories burned from the connected wearable device.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User's wearable device is connected to the app

Test Data: Activity data from the wearable device

Test Steps:

1. Perform physical activities with the wearable device on
2. Open the app
3. Verify if data on steps taken, heart rate and calories burned are displayed on the app

Postconditions:

- Steps, heart rate, and calories burned data are displayed

Expected Result: The fitness tracker app correctly displays steps, heart rate, and calories burned from the wearable device.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: FT05

Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: Test if the app provides troubleshooting options when connectivity issues arise with the wearable devices.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions:

- User's wearable device is connected to the app

Test Data: No test data needed

Test Steps:

1. Go to the device setting of the app
2. Disconnect the wearable device

3. Attempt to re-connect the wearable device

4. Check the app for troubleshooting options

Postconditions:

- Troubleshooting options are provided

Expected Result: The app provides troubleshooting options to users when they experience connectivity issues with their wearable devices.

Severity: Minor

Type of Testing: Usability

Test Case Behaviour: Positive

Data - 86

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Water Intake Tracking

Feature description -

Users can monitor their daily water intake and hydration levels, setting hydration goals and receiving reminders to drink water throughout the day. Proper hydration is essential for overall health and performance, and this feature helps users stay hydrated and maintain optimal fluid balance.

Userstory title -

As a user, I want to track my water intake so that I can stay hydrated and maintain optimal health.

Acceptance criteria -

['Users can log their daily water intake.', 'The app calculates total water consumption for the day.', 'Users can set daily hydration goals.', 'Reminders are provided to encourage regular water intake.', 'Users can view hydration progress and trends.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT001

Test Case Title: Verify that users can log their daily water intake

Test Case Description: Ensuring that the system allows users to record the amount of water they consume each day.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has access to water intake tracking feature

Test Data: Volume of water intake for the day

Test Steps:

1. Open the fitness tracker app
2. Navigate to the water intake tracking feature
3. Input the volume of water consumed for the day

Postconditions:

- Water intake for the day is recorded

Expected Result: The system records the daily water intake of the user accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT002

Test Case Title: Verify that the app calculates total water consumption for the day

Test Case Description: Ensuring that the system accurately calculates the total volume of water consumed by the user throughout the day.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User's water intake data for the day is available

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app
2. Navigate to the water intake tracking feature
3. Review the calculated total water intake for the day

Postconditions:

- Total water consumption for the day is calculated

Expected Result: The system accurately calculates and displays the user's total water consumption for the day.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT003

Test Case Title: Verify that users can set daily hydration goals

Test Case Description: Testing if users can set their individual daily hydration targets according to their needs.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: Desired hydration goal

Test Steps:

1. Open the fitness tracker app
2. Navigate to the water intake tracking feature
3. Command the app to set a hydration goal

Postconditions:

- User's daily hydration goal is set

Expected Result: The system enables users to set their daily hydration goals.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT004

Test Case Title: Verify that the app provides reminders to encourage regular water intake

Test Case Description: Ensuring the app sends regular reminders and alerts to users to drink water and achieve their hydration goals.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

- User is logged in
- Hydration goal is set

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app
2. Set a hydration goal
3. Wait for a reminder to drink water

Postconditions:

- User is reminded to drink water

Expected Result: The system sends timely reminders to the user to drink water and make progress towards their hydration goal.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT005

Test Case Title: Verify that users can view progress and trends on their hydration

Test Case Description: Testing if users can easily view and monitor their progress and trends regarding their water intake and hydration.

Test Suite: Water Intake Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Available data on user's water intake

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app
2. Navigate to the water intake tracking feature
3. Command the system to display progress and trends

Postconditions:

- Hydration progress and trends are displayed

Expected Result: The system accurately displays the progress and trends on the user's hydration levels based on recorded water intake.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 87

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Nutrition Tracking

Feature description -

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

Userstory title -

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

Acceptance criteria -

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT_Nt_Case1

Test Case Title: Verify that users can log their daily meals and snacks

Test Case Description: The app should allow users to input the meals and snacks they consume each day for tracking purposes.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has consumed a meal or snack

Test Data: Food intake details

Test Steps:

1. Click on the food diary tab
2. Select "Add meal/snack"

3. Input meal/snack details and submit

Postconditions:

- Meal or snack data is logged into the system

Expected Result: The system logs the meal/snack and displays it in the user's food diary.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT_Nt_Case2

Test Case Title: Verify that the app calculates macronutrient intake

Test Case Description: The system should calculate the user's macronutrient intake (carbs, fats, proteins) based on the logged meals and snacks.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User has logged the meal/snack

Test Data: Macronutrient content of meal/snack

Test Steps:

1. Click on nutrition tab
2. Verify the displayed macronutrient intake against expected values

Postconditions:

- Macronutrient intake is calculated and displayed

Expected Result: The system accurately calculates and displays the user's macronutrient intake.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT_Nt_Case3

Test Case Title: Verify that Users can set nutritional goals and track the progress

Test Case Description: The user should be able to set nutritional goals and have progress tracked against the set goals.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

- User is logged in

Test Data: Nutritional goals

Test Steps:

1. Click on the goals tab
2. Set nutritional goals
3. Return after a period of time to verify progress against the goals

Postconditions:

- Nutritional goals are set
- Progress is tracked against the set goals

Expected Result: The system allows setting of goals and accurately tracks progress against these goals.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT_Nt_Case4

Test Case Title: Verify that the app provides nutritional information and suggestions

Test Case Description: The system should provide users with nutritional information and personalized suggestions based on their logged meals and snacks.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

- User has logged the meal/snack

Test Data: Nutritional value of meals/snacks

Test Steps:

1. Input sample meal/snack
2. Log into the system
3. Verify if appropriate nutritional information and suggestions are provided by the system

Postconditions:

- Nutritional information is provided
- Personalized suggestions are given based on user's data

Expected Result: The system provides accurate nutritional information and appropriate suggestions based on user's data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT_Nt_Case5

Test Case Title: Verify that users can view their nutritional history and trends

Test Case Description: The user should be able to view their nutritional history and identify trends over

time.

Test Suite: Nutrition Tracking

Test Priority: Low

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Click on nutrition history tab
2. Verify if nutritional history and trends are displayed

Postconditions:

- Nutritional history and trends are displayed

Expected Result: The system accurately displays the user's nutritional history and identifiable trends over time.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 88

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Progress Reports

Feature description -

Users receive regular progress reports and summaries summarizing their activity levels, workout performance, and goal attainment. Progress reports highlight achievements, trends, and areas for improvement, empowering users to make informed decisions and adjustments to their fitness routines.

Userstory title -

As a user, I want to receive progress reports so that I can track my achievements and identify areas for improvement.

Acceptance criteria -

['Users receive weekly and monthly progress reports.', 'Reports include activity levels, workout performance, and goal attainment.', 'Users can view detailed breakdowns of achievements and trends.', 'The app provides recommendations based on report findings.', 'Users can customize the frequency and content of reports.']

Generated Test Cases:

Test Case 1:

Test Case ID: PROG_001

Test Case Title: Verify that users receive weekly and monthly progress reports

Test Case Description: Ensure that the system generates and sends weekly and monthly progress reports to users

Test Suite: Progress Reports

Test Priority: High

Preconditions:

- User is registered and logged in
- User has recorded physical activity data

Test Data: User's physical activity history

Test Steps:

1. Log in as a return user
2. Check the mailbox for received progress reports
3. Validate the frequency (weekly and monthly) of these reports

Postconditions:

- Weekly and monthly reports are present in the mailbox

Expected Result: The system sends weekly and monthly progress reports to users with relevant data

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: PROG_002

Test Case Title: Verify that reports include activity levels, workout performance, and goal attainment

Test Case Description: Ensure that individual progress reports contain information about user's activity levels, workout performance, and goal attainment

Test Suite: Progress Reports

Test Priority: High

Preconditions:

- User is logged in
- Progress reports are available

Test Data: User's progress report

Test Steps:

1. Open a received progress report
2. Check if the report includes data about activity levels, workout performance, and goal attainment

Postconditions:

- Report details are viewed and validated

Expected Result: The progress report includes activity levels, workout performance, and goal attainment

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: PROG_003

Test Case Title: Verify that users can view detailed breakdowns of achievements and trends

Test Case Description: Ensure that the progress reports allow users to see a detailed breakdown of their achievements and trends

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in
- Progress reports are available

Test Data: User's progress report

Test Steps:

1. Open a received progress report
2. Check for a detailed section of achievements and trends
3. Open and view the details in that section

Postconditions:

- Detailed breakdown is viewed

Expected Result: The progress report includes a detailed breakdown of achievements and trends

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: PROG_004

Test Case Title: Verify that the app provides recommendations based on report findings

Test Case Description: Ensure that the system generates recommendations for the user based on their report findings

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

- User is logged in
- Progress reports are available
- User has report findings

Test Data: User's report findings

Test Steps:

1. Open a received progress report
2. View the recommendations section
3. Check that recommendations align with report findings

Postconditions:

- Recommendations are viewed and validated

Expected Result: The progress report provides recommendations based on report findings

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: PROG_005

Test Case Title: Verify that users can customize the frequency and content of reports

Test Case Description: Ensure that users can modify the frequency and the content of their received progress reports

Test Suite: Progress Reports

Test Priority: Low

Preconditions:

- User is logged in
- User navigates to settings

Test Data: No test data needed

Test Steps:

1. Navigate to report customization settings
2. Change the frequency and content of the progress reports
3. Save changes
4. Confirm changes in next received report

Postconditions:

- Changes to report frequency and content are saved

Expected Result: Users can successfully customize the frequency and content of reports

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 89

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

In-App Challenges

Feature description -

The app offers in-app challenges and competitions, encouraging users to set personal records, compete with friends, and earn rewards for achieving fitness milestones. In-app challenges add an element of fun, motivation, and friendly competition to the fitness tracking experience.

Userstory title -

As a user, I want to participate in in-app challenges so that I can stay motivated and achieve my fitness goals.

Acceptance criteria -

['Users can join various in-app challenges.', 'Challenges have clear objectives and rules.', 'Users can track their progress within the challenge.', 'Rewards are provided for achieving challenge milestones.', 'Users can compete with friends and view leaderboards.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT001

Test Case Title: Verify that users can join various in-app challenges.

Test Case Description: Testing the functionality that lets users join different challenges offered in the app.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is logged in
- App is installed and operational
- Internet connection is stable

Test Data: Account details and challenge details

Test Steps:

1. Open the Challenges section of the app
2. Browse the available challenges
3. Select and join a challenge

Postconditions:

- User is joined to the selected challenge

Expected Result: The app allows the user to join the selected challenge successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT002

Test Case Title: Verify that the Challenges have clear objectives and rules.

Test Case Description: Ensure that each challenge in the app comes with clear objectives and rules.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the Challenges section of the app

Test Data: No test data needed

Test Steps:

1. Select a challenge
2. View the objectives and rules of the challenge

Postconditions:

- Challenge objectives and rules are displayed

Expected Result: Challenge details, including objectives and rules, are displayed clearly.

Severity: Minor

Type of Testing: Usability

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT003

Test Case Title: Verify that users can track their progress within the challenge.

Test Case Description: Testing if the app allows users to keep track of their progress in the challenge.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User is participating in a challenge

Test Data: User activity data

Test Steps:

1. Go to the active challenge
2. View progress details

Postconditions:

- Challenge progress is updated

Expected Result: The app shows the user's updated progress in the challenge.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT004

Test Case Title: Verify that rewards are provided for achieving challenge milestones.

Test Case Description: Checking that the app is facilitating rewards for users who reach specific milestones in challenges.

Test Suite: In-App Challenges

Test Priority: Medium

Preconditions:

- User has achieved a milestone in a challenge

Test Data: Challenge progress data

Test Steps:

1. Achieve a milestone within a challenge
2. Check for reward notification
3. Confirm reward receipt

Postconditions:

- Reward is assigned and received

Expected Result: The app triggers the appropriate reward for the user for reaching the challenge milestone.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT005

Test Case Title: Verify that users can compete with friends and view leaderboards.

Test Case Description: Ensure that the app allows users to compete with friends in challenges and view their standings on leaderboards.

Test Suite: In-App Challenges

Test Priority: High

Preconditions:

- User and friends are joined in a challenge

Test Data: Challenge data and friend list

Test Steps:

1. View leaderboard for the challenge
2. Check user's position and friend's position

Postconditions:

- Leaderboard is updated

Expected Result: The app displays the leaderboard showing the current positions of the user and their friends in the challenge.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Customizable Dashboards

Feature description -

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

Userstory title -

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Acceptance criteria -

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

Generated Test Cases:

Test Case 01:

Test Case ID: FT1

Test Case Title: Verify that users can select which metrics to display on their dashboard

Test Case Description: Test to verify that the Fitness Tracker App allows users to select specific fitness

metrics to be displayed on their custom dashboard.

Test Suite: Customizable Dashboard Feature

Test Priority: High

Preconditions:

- User is logged in
- Data for various metrics available

Test Data: Fitness metrics data (e.g., steps, calories, distance, heart rate)

Test Steps:

1. Access the dashboard customization setting
2. Choose a desired fitness metric
3. Confirm selection

Postconditions:

- Selected fitness metric is displayed on the dashboard

Expected Result: The selected fitness metric shows up in the user's dashboard and updates in real time.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 02:

Test Case ID: FT2

Test Case Title: Verify that the dashboard layout and theme can be customized

Test Case Description: Ensure that the Fitness Tracker App allows users to customize the layout and theme of their dashboard.

Test Suite: Customizable Dashboard Feature

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

1. Go to dashboard settings
2. Select layout/theme customization options
3. Make changes and save

Postconditions:

- Dashboard theme and layout are updated

Expected Result: The system updates the dashboard layout and theme according to the user's preferences.

Severity: Minor

Type of Testing: Usability

Test Case Behaviour: Positive

Test Case 03:

Test Case ID: FT3

Test Case Title: Verify that user's dashboard settings can be saved and updated

Test Case Description: Ensure that the Fitness Tracker App saves and updates the changes made in the dashboard settings.

Test Suite: Customizable Dashboard Feature

Test Priority: High

Preconditions:

- User is logged in
- User has made changes in the dashboard settings

Test Data: No test data needed

Test Steps:

1. After making changes in the dashboard settings, click on "Save"

2. Logout and log back into the system

3. Check if the changes are retained

Postconditions:

- Dashboard settings are retained post logout

Expected Result: The changes in dashboard settings must be saved and retained even after logout and re-login.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 04:

Test Case ID: FT4

Test Case Title: Verify if dashboard displays update in real-time

Test Case Description: Ensure that the dashboard in the Fitness Tracker App dynamically updates with real-time data.

Test Suite: Customizable Dashboard Feature

Test Priority: High

Preconditions:

- User is logged in
- Change in fitness data has occurred

Test Data: Real-time fitness data

Test Steps:

1. Perform a fitness activity tracked by the app
2. Load the dashboard to check if it has updated with the latest data

Postconditions:

- Dashboard shows the latest fitness data

Expected Result: The system accurately displays real-time fitness data in the user's dashboard.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 05:

Test Case ID: FT5

Test Case Title: Verify that users can reset the dashboard to default settings

Test Case Description: Ensure that the Fitness Tracker App allows users to reset their customized dashboard to the default settings.

Test Suite: Customizable Dashboard Feature

Test Priority: Medium

Preconditions:

- User is logged in
- User has previously customized their dashboard

Test Data: No Test data needed

Test Steps:

1. Go to dashboard settings
2. Select "Reset to Defaults"
3. Confirm action

Postconditions:

- Dashboard reverts to default settings

Expected Result: The system resets the user's dashboard to the default settings

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: 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 Case 1:

Test Case ID: FT01_SocShare

Test Case Title: Verify that users can share workout summaries and achievements on social media.

Test Case Description: Test case seeks to confirm the application's capability to allow users to share their track records and achievements on social media platforms.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- User is logged into the application
- User has workout history or achievements to share

Test Data: User's workout summaries and achievements

Test Steps:

1. Open the application
2. Go to the workout summary or achievements section
3. Select the share button
4. Choose the desired social media platform
5. Share the workout

Postconditions:

- Sharing is successful

Expected Result: The application allows the user to share their workout records and achievements on the selected social media platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT02_SocShare

Test Case Title: Verify that the application supports sharing on multiple social media platforms.

Test Case Description: This test case will ensure that the fitness tracker application can support several social networking sites.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- User is logged into the application
- At least one social media platform account exists

Test Data: No test data needed

Test Steps:

1. Log into the application
2. Access the share feature
3. Identify and select multiple social media platforms
4. Verify the functionality on each platform

Postconditions:

- The share feature is successful on multiple platforms

Expected Result: The application can support sharing on several social media platforms.

Severity: Major

Type of Testing: Compatibility

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT03_SocShare

Test Case Title: Verify that users can customize the format and content of shared posts.

Test Case Description: This test case will verify that the application allows users to adjust the format and data of the posts they share on social media.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- User is logged into the application

Test Data: No test data needed

Test Steps:

1. Open the application
2. Access the share feature
3. Attempt to alter the format and content of the workout/achievement post
4. Share the customized post to social media

Postconditions:

- The shared post appears on social media as customized

Expected Result: The application allows customization of format and content for shared posts.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT04_SocShare

Test Case Title: Verify that sharing features are integrated seamlessly with social media.

Test Case Description: This test case is designed to check that sharing features are seamlessly integrated with social media platforms.

Test Suite: Social Sharing

Test Priority: High

Preconditions:

- User is logged into the application
- User has a valid social media account linked to the app.

Test Data: No test data needed

Test Steps:

1. Open the application
2. Go to share feature
3. Share a post to a social media platform

Postconditions:

- Post is shared on social media

Expected Result: Sharing features are seamlessly integrated, and the post is visible on the linked social media account.

Severity: Major

Type of Testing: Integration

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT05_SocShare

Test Case Title: Verify that users can track the engagement of their shared posts.

Test Case Description: This test case checks that application users can track the engagements (likes, comments, shares) on their shared fitness activity posts.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

- User has shared a post through the application

Test Data: User's shared fitness activity posts

Test Steps:

1. Open the application
2. Go to the section that shows shared posts

3. Check the engagement metrics for a particular post

Postconditions:

- User is able to see the engagements on the post

Expected Result: Users are able to see the engagements (likes, comments, shares) on their shared posts.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 92

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Rest and Recovery Tracking

Feature description -

The app includes features for tracking rest days, recovery activities, and recovery metrics such as resting heart rate and sleep quality. Rest and recovery tracking helps users prioritize recovery, prevent overtraining, and optimize their overall health and performance.

Userstory title -

As a user, I want to track my rest and recovery so that I can optimize my fitness performance and prevent overtraining.

Acceptance criteria -

['Users can log rest days and recovery activities.', 'The app tracks metrics such as resting heart rate and sleep quality.', 'Users receive recommendations for recovery based on their activities.', 'The app provides alerts for insufficient recovery time.', 'Users can view trends and history of their recovery data.']

Generated Test Cases:

Test Case 1

Test Case ID: R&R001

Test Case Title: Verify that users can log rest days and recovery activities

Test Case Description: Ensure that the system allows users to enter and save details about their rest days and recovery activities.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in
- Fitness data is already recorded

Test Data: Rest days information, details of recovery activities

Test Steps:

1. Navigate to recovery tracking section
2. Click 'Add' to log a rest day or recovery activity
3. Input the required details
4. Save the entries

Postconditions:

- Rest day and recovery activity is logged

Expected Result: The system successfully logs the rest day or recovery activity and reflects it in the user's overall recovery schedule.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: R&R002

Test Case Title: Verify that the app tracks metrics such as resting heart rate and sleep quality

Test Case Description: Confirm that the system accurately calculates and records metrics like resting heart rate and sleep quality.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in
- Metrics devices are connected (if applicable)

Test Data: Heart rate measurements, sleep cycle data

Test Steps:

1. Go to the metrics tracking section
2. Review recorded data such as resting heart rate and sleep quality
3. Cross-verify with metrics devices

Postconditions:

- Metrics are recorded

Expected Result: The system accurately records and displays metrics for resting heart rate and sleep quality.

Severity: Critical

Type of Testing: Performance

Test Case Approach: Positive

Test Case 3

Test Case ID: R&R003

Test Case Title: Verify that users receive recovery recommendations based on their activities

Test Case Description: Verify that the system generates recovery advice tailored to the user's recorded activity and rest data.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Recent fitness activity data is available

Test Data: User fitness activity records

Test Steps:

1. Go to the recommendations section
2. Review provided recovery suggestions
3. Contrast suggestions with recent fitness activities

Postconditions:

- Recommendations are provided

Expected Result: The system generates accurate and beneficial recovery suggestions based on the user's recent fitness activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: R&R004

Test Case Title: Verify that the app provides alerts for insufficient recovery time

Test Case Description: Confirm that the application sends warning notifications when the user hasn't recorded sufficient recovery time according to their activities.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in
- Recent fitness activity data is available

Test Data: User profile and activity records

Test Steps:

1. Skip logging recovery time or log less than the recommended recovery time
2. Review the notifications received

Postconditions:

- Alerts are sent out

Expected Result: The system sends alert notifications when insufficient recovery time is logged in relation to the users activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: R&R005

Test Case Title: Verify that users can view trends and history of their recovery data

Test Case Description: Check that the system precisely shows the history and trends of the user's recorded recovery data.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Previous recovery data is recorded

Test Data: Historical recovery data

Test Steps:

1. Navigate to the history section
2. Review the historical data and generated trends
3. Cross-verify with the original data

Postconditions:

- Recovery trends and history are displayed

Expected Result: The system accurately displays the trends and history of the user's recovery data.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 93

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Audio Coaching

Feature description -

Users can access audio-guided workouts, coaching cues, and motivational messages during their workouts, enhancing engagement and adherence. Audio coaching provides real-time feedback, encouragement, and guidance, simulating the experience of having a personal trainer or workout buddy.

Userstory title -

As a user, I want to access audio coaching during workouts so that I can stay motivated and receive real-time guidance.

Acceptance criteria -

['Users can access a variety of audio-guided workouts.', 'Audio coaching includes motivational messages and coaching cues.', 'Users can choose coaching based on workout type and intensity.', 'Audio coaching integrates seamlessly with other app features.', 'Users can control audio settings and preferences.']

Generated Test Cases:

Test Case 1:

Test Case ID: FTAC01

Test Case Title: Verify that users can access a variety of audio-guided workouts

Test Case Description: This test case is designed to verify whether users can access a variety of audio-guided workouts through the 'Audio Coaching' feature in the Fitness Tracker App.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

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

Test Data: No test data needed

Test Steps:

1. Locate and click on the 'Audio Coaching' feature
2. Browse through the available audio-guided workouts
3. Select one and start the workout

Postconditions:

- Audio-guided workout is initiated

Expected Result: The system streams the selected audio-guided workout seamlessly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: FTAC02

Test Case Title: Verify that audio coaching includes motivational messages and coaching cues

Test Case Description: This test case checks the provision of motivational messages and coaching cues during the audio-guided workouts.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in
- User is within an audio-guided workout

Test Data: No test data needed

Test Steps:

1. Start an audio-guided workout
2. Listen for motivational messages and coaching cues
3. Note their frequency and relevance

Postconditions:

- The workout ends

Expected Result: The user receives appropriate motivational messages and coaching cues throughout the workout.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: FTAC03

Test Case Title: Verify that users can choose coaching based on workout type and intensity

Test Case Description: This test case will validate if users can select appropriate audio coaching based on the type and intensity level of their workout.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in
- User is on the 'Audio Coaching' page

Test Data: Different types of workout and their corresponding intensities

Test Steps:

1. Choose a type of workout and intensity level
2. Browse the available audio-guided workouts for that type and intensity
3. Select one and begin the workout

Postconditions:

- The selected workout begins with the chosen audio coaching

Expected Result: The system allows the user to select and start an appropriate audio-guided workout based on the workout type and intensity.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: FTAC04

Test Case Title: Verify that audio coaching integrates seamlessly with other app features

Test Case Description: This test aims to ensure the smooth integration of the 'Audio Coaching' feature with other application features such as progress tracking, goal setting, etc.

Test Suite: Audio Coaching

Test Priority: High

Preconditions:

- User is logged in
- User is using the 'Audio Coaching' feature

Test Data: No test data needed

Test Steps:

1. Start an audio-guided workout
2. Access other features of the app (e.g., progress tracking) without interrupting the ongoing workout
3. Note the changes and synchronization with other features

Postconditions:

- Audio-guided workout continues running alongside other functions

Expected Result: All app features work seamlessly alongside continuous audio-guided workouts.

Severity: Major

Type of Testing: Integration

Test Case Approach: Positive

Test Case 5:

Test Case ID: FTAC05

Test Case Title: Verify that users can control audio settings and preferences

Test Case Description: This test case ensures that users can adjust the audio settings and preferences for the 'Audio Coaching' feature to suit their workout needs.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the 'Audio Coaching' settings page

Test Data: Different audio setting preferences

Test Steps:

1. Navigate to 'Audio Coaching' settings
2. Adjust different audio settings (e.g., volume, speed)
3. Save changes and return to the workout

Postconditions:

- Changes in audio settings and preferences are saved

Expected Result: Users can successfully change and save their audio settings and preferences for audio-guided workouts.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Health Apps

Feature description -

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

Userstory title -

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive view of my well-being.

Acceptance criteria -

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.', 'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from multiple apps.', 'Integration settings are easy to manage and configure.']

Generated Test Cases:

Test Case 1:

Test Case ID: FTC001

Test Case Title: Verify that the app integrates with popular health and wellness apps.

Test Case Description: To ensure that the fitness tracker app can successfully integrate with widely used health and wellness applications.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User has the fitness tracker app installed
- User has other health/wellness apps installed

Test Data: Existing health metric data in the health/wellness apps

Test Steps:

1. Open the fitness tracker app
2. Navigate to the 'Integration' section
3. Select a health/wellness app to integrate

Postconditions:

- Fitness tracker app is connected with the selected health and wellness app

Expected Result: The system successfully integrates the fitness tracker app with the selected health and wellness app.

Severity: Major

Type of Testing: Integration Testing

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: FTC002

Test Case Title: Verify that users can sync data from connected health apps.

Test Case Description: Test the ability of users to synchronize health metric data from the connected health/wellness apps with the fitness tracker app.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User has integrated at least one health/wellness app with the fitness tracker app
- There is new health metric data available for synchronization in the connected apps

Test Data: Updated health metric data in the health/wellness apps

Test Steps:

1. Open the integrated health/wellness app
2. Navigate to the 'Data Sync' section of the fitness tracker app
3. Activate the sync process

Postconditions:

- The health metric data from the connected health/wellness app is updated within the fitness tracker app

Expected Result: The system successfully syncs the health metrics data from the connected health and wellness app into the fitness tracker app.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: FTC003

Test Case Title: Verify that the app provides a consolidated view of health metrics.

Test Case Description: Test for the display of comprehensive health metric data from all integrated apps in a consolidated view within the fitness tracker app.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User has integrated at least one health/wellness app with the fitness tracker app

- There is health metric data available from the connected apps

Test Data: Health metric data from the health/wellness apps

Test Steps:

1. Open the fitness tracker app
2. Navigate to the 'Consolidated View' section

Postconditions:

- The consolidated view of health metrics is visible

Expected Result: The system successfully presents a comprehensive consolidated view of health metrics from all integrated health and wellness apps.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: FTC004

Test Case Title: Verify that users receive insights based on combined data from multiple apps.

Test Case Description: Test for generation of insights or recommendations based on the combined health metric data from all integrated health and wellness apps.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User has integrated at least one health/wellness app with the fitness tracker app
- There is health metric data available from the connected apps

Test Data: Health metric data from the health/wellness apps

Test Steps:

1. Open the fitness tracker app

2. Navigate to the 'Insights' section

Postconditions:

- Insights based on combined data are visible

Expected Result: The system generates and presents insights based on the combined health metrics from all integrated health and wellness apps.

Severity: Major

Type of Testing: Functional Testing

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: FTC005

Test Case Title: Verify that integration settings are easy to manage and configure.

Test Case Description: To ensure that the user can easily manage and configure the settings for the integration of health and wellness apps with the fitness tracker app.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- User has the fitness tracker app installed

Test Data: No test data needed

Test Steps:

1. Open the fitness tracker app
2. Navigate to the 'Integration Settings' section
3. Attempt to manage and configure the integration settings

Postconditions:

- Changes to the integration settings are saved

Expected Result: The system allows the user to successfully manage and configure the integration settings

with ease.

Severity: Minor

Type of Testing: Usability Testing

Test Case Behaviour: Positive

Data - 95

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Body Composition Tracking

Feature description -

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and the effectiveness of exercise and nutrition interventions.

Userstory title -

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed decisions about my training and nutrition.

Acceptance criteria -

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition

changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set goals for body composition metrics.']

Generated Test Cases:

Test Case 1:

Test Case ID: FT001

Test Case Title: Verify that users can log body composition metrics

Test Case Description: Ensure that the system allows users to log body composition metrics such as weight, body fat percentage, muscle mass.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has body composition data to log

Test Data: User's body composition data

Test Steps:

1. Go to the Body Composition Tracking section
2. Click on add new entry button
3. Enter Body composition data
4. Save it

Postconditions:

- Body composition for the user is logged

Expected Result: The system saves the user's body composition data and displays it in the user's account.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: FT002

Test Case Title: Verify BMI calculation

Test Case Description: Ensure that the app calculates and displays the Body Mass Index (BMI) correctly based on the logged body composition data.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has logged body composition data

Test Data: User's body composition data

Test Steps:

1. Go to the Body Composition Tracking section
2. Check the BMI value displayed

Postconditions:

- None

Expected Result: The BMI value correctly calculated based on the logged weight and height is displayed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: FT003

Test Case Title: Verify body composition changes history

Test Case Description: Ensure that the system allows users to view trends and history of body composition changes.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- User has logged body composition data on multiple occasions

Test Data: No test data needed

Test Steps:

1. Go to the Body Composition Tracking section
2. Click on View history button

Postconditions:

- User has viewed the body composition history

Expected Result: The system displays trends and history of body composition changes.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: FT004

Test Case Title: Verify app insights and recommendation system

Test Case Description: Ensure the app is providing insights and recommendations based on the user's body composition data.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has logged their body composition data

Test Data: User's body composition data

Test Steps:

1. Go to the Body Composition Tracking section
2. Check the insights and recommendations provided by the app

Postconditions:

- User has reviewed insights and recommendations

Expected Result: Based on the user's body composition history, the app provides insights and personalized recommendations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: FT005

Test Case Title: Verify goal setting for body composition metrics

Test Case Description: Ensure the app allows users to set goals for body composition metrics and tracks progress towards those goals.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: User's body composition goals

Test Steps:

1. Go to the Goals section
2. Enter goals for body composition metrics
3. Save goals

Postconditions:

- User's body composition goals are saved in the system

Expected Result: User is able to set goals for body composition metrics and the system saves these goals.

Severity: Major

Type of Testing: Functional

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:

Test Case 1:

Test Case ID: FT001

Test Case Title: Verify that users can view detailed performance metrics for each workout.

Test Case Description: Ensure that the app provides in-depth performance metrics for each workout such as duration, calories burned, heart rate, etc.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User has completed at least one workout
- User is logged into the app

Test Data: User's workout details

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the workout analysis section.

3. Select a specific workout.

Postconditions:

- Detailed performance metrics for the selected workout are displayed.

Expected Result: The app displays the comprehensive performance metrics of the selected workout effectively.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: FT002

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration.

Test Case Description: Ensure that the app generates and displays trends and insights regarding the intensity and duration of the user's workouts.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User has completed multiple workouts
- User is logged into the app

Test Data: User's workout details for a considerable period

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the workout analysis section.
3. Access the trends and insights feature.

Postconditions:

- Workout trends and insights are displayed.

Expected Result: The app accurately generates and displays trends and insights into the user's workout intensity and duration.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: FT003

Test Case Title: Verify that users can track calorie burn and other health metrics.

Test Case Description: Ensure that the app effectively tracks and displays calorie burn and various other health metrics such as heart rate, steps taken, distance traveled, etc.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

- User has completed at least one workout
- User is logged into the app

Test Data: User's workout details

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the health metrics section.
3. Select the 'calorie burn' or other health metrics for an overview.

Postconditions:

- Calorie burn and health metrics are displayed.

Expected Result: The app accurately tracks and reflects the calorie burn and other vital health metrics.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: FT004

Test Case Title: Verify that the app offers recommendations based on workout analysis.

Test Case Description: Ensure that the app provides personalized workout recommendations to users based on their past workout analyses.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- User has completed multiple workouts
- User is logged into the app

Test Data: User's workout details for a considerable period

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the workout analysis section.
3. Access the 'Recommendations' feature.

Postconditions:

- Personalized workout recommendations based on past workouts are displayed.

Expected Result: The app accurately generates and displays personalized workout recommendations based on the user's workout analysis.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: FT005

Test Case Title: Verify that users can compare current workouts with past performance.

Test Case Description: Ensure that the app correctly facilitates comparison between a user's current workout performance and their past performance metrics.

Test Suite: Workout Analysis

Test Priority: Medium

Preconditions:

- User has completed multiple workouts
- User is logged into the app

Test Data: User's workout details for a considerable period

Test Steps:

1. Open the Fitness Tracker App.
2. Navigate to the workout analysis section.
3. Use the 'Compare Workouts' feature.

Postconditions:

- Past and current workout performance metrics are displayed for comparison.

Expected Result: The app accurately compares and displays a user's current workout performance alongside their past performance metrics.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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: W1

Test Case Title: Verify that current weather conditions and forecasts are displayed

Test Case Description: Assess whether the system displays current weather conditions and forecasts.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- Internet connection is stable

Test Data: Various weather conditions and forecasts

Test Steps:

1. Open the Fitness Tracker app
2. Navigate to the weather forecast section
3. Refresh/check the weather updates

Postconditions:

- Weather updates are displayed

Expected Result: The app displays current weather conditions and forecasts successfully

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: W2

Test Case Title: Verify that users can view weather details for different locations

Test Case Description: Ensure the system provides weather details for various selected locations.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- Internet connection is stable

Test Data: Different locations

Test Steps:

1. Open the Fitness Tracker app
2. Navigate to the weather forecast section
3. Input/select different locations
4. Check weather details

Postconditions:

- Weather details for selected location are displayed

Expected Result: The app displays weather details for the chosen location accurately

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: W3

Test Case Title: Verify that weather data is updated in real-time

Test Case Description: Evaluate whether the system updates weather data in real-time.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- Internet connection is stable

Test Data: Real-time weather updates

Test Steps:

1. Open the Fitness Tracker app
2. Navigate to the weather forecast section
3. Refresh feed over a fixed time interval

Postconditions:

- Real-time weather updates are displayed

Expected Result: The app updates weather data in real-time

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: W4

Test Case Title: Verify that the app offers recommendations based on weather conditions

Test Case Description: Validate whether the system offers workout recommendations based on current weather conditions.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- Internet connection is stable

Test Data: Various weather conditions

Test Steps:

1. Open the Fitness Tracker app
2. Navigate to the weather forecast section
3. Check for workout recommendations

Postconditions:

- Recommendations are displayed based on current weather conditions

Expected Result: The app offers valid workout recommendations according to current weather conditions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: W5

Test Case Title: Verify that users can receive weather alerts for extreme conditions

Test Case Description: Confirm whether the system sends weather alerts for extreme conditions.

Test Suite: Weather Integration

Test Priority: High

Preconditions:

- User is logged in
- Internet connection is stable

Test Data: Extreme weather conditions

Test Steps:

1. Open the Fitness Tracker app
2. Navigate to the weather forecast section
3. Input a location with extreme weather conditions
4. Check for weather alerts

Postconditions:

- Alerts for extreme weather conditions are displayed

Expected Result: The app sends weather alerts for extreme conditions accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Mood Tracking

Feature description -

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

Userstory title -

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the impact of exercise on my mental well-being.

Acceptance criteria -

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and energy levels.']

Generated Test Cases:

Test Case 1:

Test Case ID: FTAC1

Test Case Title: Verify that users can log their mood, energy levels, and motivation.

Test Case Description: This test case tests the ability of the feature to allow users to log their mood, energy levels, motivation, and insights experienced during workouts.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

- User is logged in
- User has just completed a workout

Test Data: User mood data

Test Steps:

1. Navigate to the Mood Tracking feature
2. Input mood, energy, and motivation data
3. Save the data

Postconditions:

- Mood, energy, and motivation data is logged for the workout

Expected Result: User is able to log their mood, energy levels, and motivation after a workout.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: FTAC2

Test Case Title: Verify that the app provides insights into mood and energy trends over time.

Test Case Description: This test case tests the feature's ability to generate insights into the user's mood and energy trends as per the logged exercise experiences.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

- User has been regularly logging mood data after workouts over a certain period of time

Test Steps:

1. Navigate to the Mood Tracking feature
2. View generated insights on the mood and energy over time

Postconditions:

- Insights into mood and energy trends have been reviewed by the user

Expected Result: The user can view progressive insights into their mood and energy trends based on their exercise experiences.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: FTAC3

Test Case Title: Verify that users receive recommendations based on their mood and energy data.

Test Case Description: This test case tests the feature's ability to give the user personalised exercise recommendations based on their mood and energy data.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User has logged mood, energy and motivation data

Test Steps:

1. Navigate to the Mood Tracking feature
2. Review recommendations based on mood and energy data

Postconditions:

- User has received recommendations to improve mood and energy levels

Expected Result: The user receives recommendations based on their logged mood and energy data.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: FTAC4

Test Case Title: Verify that the app supports journaling for personal reflections on mood and workouts.

Test Case Description: This test case tests if the feature provides a journaling function for the user to record personal reflections on their mood and workouts.

Test Suite: Mood Tracking

Test Priority: Low

Preconditions:

- User has completed a workout

Test Data: User's mood and energy data, along with personal reflections

Test Steps:

1. Navigate to the Mood Tracking feature
2. Input personal reflections

Postconditions:

- User has logged their personal reflections

Expected Result: The user is able to journal their personal reflections on their mood and workouts.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: FTAC5

Test Case Title: Verify that users can set goals for improving mood and energy levels.

Test Case Description: This test case tests the ability of the feature to allow users to set goals for their mood and energy levels.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User is logged in

Test Data: User's goals for mood and energy levels

Test Steps:

1. Navigate to the Mood Tracking feature
2. Input mood and energy goals
3. Save the goals

Postconditions:

- User's goals for mood and energy levels have been set

Expected Result: Users are able to set goals for their mood and energy levels.

Severity: Critical

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 99

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Creation

Feature description -

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's recipe database.

Userstory title -

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

Acceptance criteria -

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.', 'Recipes are saved in the user's profile.', 'Recipes are accessible to other users on the platform.', 'Users receive a confirmation upon successful recipe submission.']

Generated Test Cases:

Test Case 1:

Test Case ID: RC01

Test Case Title: Verify that users can input and save ingredients and cooking instructions.

Test Case Description: This test case ensures that the system allows users to input and save the ingredients and cooking instructions for their recipes.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in
- User is on the create recipe page

Test Data: Recipe details (Ingredients and cooking instructions)

Test Steps:

1. Click on the 'Create New Recipe' button
2. Input ingredients and cooking instructions
3. Click on Save button

Postconditions:

- Recipe is saved in the user's profile

Expected Result: The system allows the user to input and save the ingredients and cooking instructions, and stores the recipe in the user's profile.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2:

Test Case ID: RC02

Test Case Title: Verify that users can upload photos for their recipes.

Test Case Description: This test case verifies that users can successfully upload photos for their recipes.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the 'Create New Recipe' Page

Test Data: Recipe Photo

Test Steps:

1. Click on the 'Upload Recipe Photo' button
2. Select a photo
3. Click on 'Save Photo'

Postconditions:

- Photo is saved with the recipe in user's profile

Expected Result: The system enables the user to upload a photo, and it is successfully saved alongside the recipe.

Severity: Medium

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3:

Test Case ID: RC03

Test Case Title: Verify that the created recipes are saved in the user's profile.

Test Case Description: This test case ensures that the recipes created by the user are indeed saved in their respective user profiles.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

- User is logged in
- User has created a recipe

Test Data: No test data needed

Test Steps:

1. Create a new recipe
2. Navigate to user's profile

3. Check the saved recipes section

Postconditions:

- The recently created recipe is visible in the user's profile

Expected Result: The system successfully saves the created recipe in the user's profile.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4:

Test Case ID: RC04

Test Case Title: Verify that recipes are accessible to other users on the platform.

Test Case Description: This test case guarantees that the recipes saved on a user's profile are accessible to other users on the platform.

Test Suite: Recipe Sharing

Test Priority: High

Preconditions:

- Different users are logged in
- A recipe has been saved on User A's profile

Test Data: User A's profile, Saved recipe

Test Steps:

1. Login as User B
2. Navigate to User A's profile
3. Access the saved recipes

Postconditions:

- User B can view User A's recipe

Expected Result: The system ensures that User A's recipe is accessible to User B.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5:

Test Case ID: RC05

Test Case Title: Verify that users receive a confirmation upon successful recipe submission.

Test Case Description: This test case tests whether the system sends a confirmation message to users after they successfully submit a recipe.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

- User is logged in
- User is on the 'Create New Recipe' Page

Test Data: Recipe details - ingredients, cooking instructions, optional photo

Test Steps:

1. Complete the recipe form
2. Click on 'Submit Recipe'

Postconditions:

- User receives a confirmation message

Expected Result: The system sends a confirmation message to the user after a successful recipe submission.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Data - 100

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Discovery

Feature description -

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

Userstory title -

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance criteria -

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

Generated Test Cases:

Test Case 1:

Test Case ID: RSP001

Test Case Title: Verify that users can search for recipes using keywords and ingredients

Test Case Description: This test case is designed to verify the functionality that allows users to discover new recipes by searching for keywords or ingredients.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- Users must be logged in to their account
- Database must contain a good variety of recipes

Test Data: Keyword or ingredient (e.g., "vegan", "chocolate", "chicken")

Test Steps:

1. Log in to the account
2. Enter the keyword or ingredient in the search bar
3. Click the search button

Postconditions:

- Search results are displayed

Expected Result: A list of recipes that match the entered keyword or ingredient is displayed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: RSP002

Test Case Title: Verify that search results accurately match the search criteria

Test Case Description: This test case is designed to confirm the relevance and accuracy of search results when users search for recipes using keywords or ingredients.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User must be logged in to their account
- User has performed a search

Test Data: Previous search result data

Test Steps:

1. Inspect the list of returned recipes
2. Verify that each displayed recipe matches the search criteria

Postconditions:

- Detailed information about the recipe is available when clicked

Expected Result: All recipes displayed match the entered keyword or ingredient.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: RSP003

Test Case Title: Verify that users can filter search results by cuisine type, dietary preferences, and other criteria

Test Case Description: This test case checks the functionality that allows users to filter the discovered recipes based on their preferences.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User must be logged in to their account
- User has performed a search

Test Data: Filters (e.g., "Italian", "Vegetarian", "Under 30 minutes")

Test Steps:

1. Perform a keyword or ingredient search
2. Select filters of choice
3. Apply filters

Postconditions:

- Filtered search results are displayed

Expected Result: The returned recipes match both the original search and the applied filters.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: RSP004

Test Case Title: Verify that recipes in the search results are detailed and accessible

Test Case Description: This test case ensures the recipes returned from the search are detailed and accessible to enhance user satisfaction.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User must be logged in to their account
- User has performed a search

Test Data: No test data needed

Test Steps:

1. Perform a keyword or ingredient search
2. Select a recipe from the search result

3. Inspect the details of the recipe

Postconditions:

- Selected recipe details are displayed

Expected Result: The user is able to view a detailed view of the selected recipe including ingredients, preparation steps, and images.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: RSP005

Test Case Title: Verify that users can save search results for future reference

Test Case Description: This test case ensures that the system allows users to save their favorite recipes from the search results for future reference.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User must be logged in to their account
- User has performed a search

Test Data: No test data needed

Test Steps:

1. Perform a keyword or ingredient search
2. From the results, select a recipe to save
3. Click the "Save" or "Add to Favorites" option

Postconditions:

- The selected recipe is saved in the user's favorites list

Expected Result: The user is able to save recipes from the search results to their account for later use.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive
