Generated Test Cases via llama3-70b-8192 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:

Here are the 5 test cases for the User Profiles feature:

Test Case 1:

Test Case ID: UP_01

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

Test Case Description: Ensure that users can add and edit their name in their profile, and the changes are saved

successfully.

Test Suite: User Profiles

Test Priority: High

Preconditions:

- User is logged in

- Profile creation page is accessible

Test Data: No test data needed

Test Steps:

1. Go to the profile creation/edit page

- 2. Enter a new name
- 3. Click save changes
- 4. Verify that the name is updated

Postconditions:

- Name is updated in the profile

Expected Result: The system allows users to add and edit their name, and the changes are saved successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: UP_02

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

Test Case Description: Ensure that the system accepts and displays standard image formats (e.g., JPEG, PNG
GIF) for profile pictures.
Test Suite: User Profiles
Test Priority: Medium
Preconditions:
- User is logged in
- Profile creation page is accessible
Test Data: Sample images in different formats (JPEG, PNG, GIF)
Test Steps:
1. Go to the profile creation/edit page
2. Upload a profile picture in each format (JPEG, PNG, GIF)
3. Verify that the image is displayed correctly
Postconditions:
- Profile picture is displayed correctly
Expected Result: The system accepts and displays standard image formats for profile pictures.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: UP_03
Test Case Title: Verify that users can set privacy settings for each field
Test Case Description: Ensure that users can set privacy settings for each field (name, bio, profile picture,
location, interests) and the settings are saved successfully.
Test Suite: User Profiles
Test Priority: High
Preconditions:

- User is logged in - Profile creation page is accessible Test Data: No test data needed Test Steps: 1. Go to the profile creation/edit page 2. Set privacy settings for each field (e.g., public, friends only, private) 3. Click save changes 4. Verify that the privacy settings are updated Postconditions: - Privacy settings are updated for each field Expected Result: The system allows users to set privacy settings for each field, and the settings are saved successfully. Severity: Critical Type of Testing: Security Test Case Approach: Positive **Test Case 4:** Test Case ID: UP_04 Test Case Title: Verify that the profile is saved successfully Test Case Description: Ensure that the system saves the profile changes successfully and displays a confirmation message. Test Suite: User Profiles Test Priority: High Preconditions: - User is logged in - Profile creation page is accessible

Test Data: No test data needed

Test Steps:
1. Go to the profile creation/edit page
2. Make changes to the profile (e.g., update bio, add interests)
3. Click save changes
4. Verify that a confirmation message is displayed
Postconditions:
- Profile changes are saved successfully
Expected Result: The system saves the profile changes successfully and displays a confirmation message.
Severity: Blocker
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5:
Test Case ID: UP_05
Test Case Title: Verify that users receive a confirmation upon saving their profile
Test Case Description: Ensure that the system sends a confirmation notification to the user upon saving their
profile changes.
Test Suite: User Profiles
Test Priority: Medium
Preconditions:
- User is logged in
- Profile creation page is accessible
Test Data: No test data needed
Test Steps:
1. Go to the profile creation/edit page
2. Make changes to the profile (e.g., update bio, add interests)
3. Click save changes

4. Verify that a confirmation notification is sent to the user

Postconditions:

- Confirmation notification is sent to the user

Expected Result: The system sends a confirmation notification to the user upon saving their profile changes.

Severity: Major

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:

Here are the 5 detailed test cases for the News Feed feature:

Test Case ID: NF001

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

Test Case Description: Ensure that the news feed displays posts from accounts that the user has followed,

including textual, image, and video content.

Test Suite: News Feed

Test Priority: High

Preconditions:

- * User is logged in
- * User has followed at least one account
- * Accounts have posted content

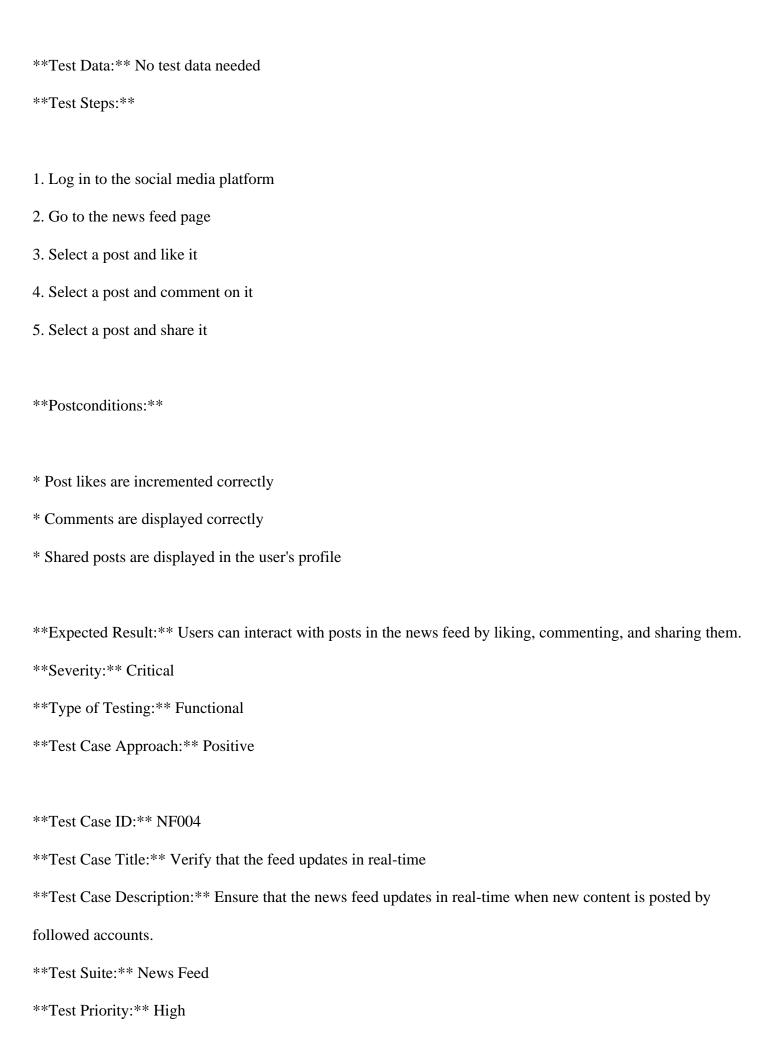
Test Data: No test data needed

Test Steps:

- 1. Log in to the social media platform
- 2. Go to the news feed page
- 3. Verify that posts from followed accounts are displayed
- 4. Check the content types (text, image, video) of the displayed posts

Postconditions:
* Posts from followed accounts are displayed
* Content types are correctly rendered
Expected Result: The news feed displays posts from followed accounts, including textual, image, and
video content.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case ID: NF002
Test Case Title: Verify that content is shown in chronological or algorithmic order
Test Case Description: Ensure that the news feed displays content in either chronological or
algorithmically determined order, depending on the user's settings.
Test Suite: News Feed
Test Priority: Medium
Preconditions:
* User is logged in
* User has followed at least one account
* Accounts have posted content
Test Data: No test data needed
Test Steps:

1. Log in to the social media platform
2. Go to the news feed page
3. Verify that content is displayed in either chronological or algorithmic order
4. Change the sorting order and verify that the content is re-sorted correctly
Postconditions:
* Content is displayed in either chronological or algorithmic order
* Sorting order changes correctly
Expected Result: The news feed displays content in either chronological or algorithmic order, depending
on the user's settings.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case ID: NF003
Test Case Title: Verify that users can interact with posts (like, comment, share)
Test Case Description: Ensure that users can interact with posts in the news feed by liking, commenting,
and sharing them.
Test Suite: News Feed
Test Priority: High
Preconditions:
* User is logged in
* User has followed at least one account
* Accounts have posted content



Preconditions:
* User is logged in
* User has followed at least one account
* Accounts have posted content
Test Data: No test data needed
Test Steps:
1. Log in to the social media platform
2. Go to the news feed page
3. Post a new update from a followed account
4. Verify that the update is displayed in the news feed in real-time
Postconditions:
* New content is displayed in the news feed in real-time
* Content is correctly rendered
Expected Result: The news feed updates in real-time when new content is posted by followed accounts
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case ID: NF005
Test Case Title: Verify that users can filter content based on type or topic
Test Case Description: Ensure that users can filter content in the news feed based on type (e.g., video,

image, text) or topic (e.g., sports, entertainment, pointics).
Test Suite: News Feed
Test Priority: Medium
Preconditions:
* User is logged in
* User has followed at least one account
* Accounts have posted content with different types and topics
Test Data: No test data needed
Test Steps:
1. Log in to the social media platform
2. Go to the news feed page
3. Select a filter option (type or topic)
4. Verify that the content is filtered correctly
Postconditions:
* Content is filtered correctly based on the selected option
* Filter options are correctly displayed
Expected Result: Users can filter content in the news feed based on 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:

Here are the 5 detailed test cases for the Messaging feature:

Test Case 1:

Test Case ID: MSG001
Test Case Title: Verify that users can send text messages
Test Case Description: Ensure that users can send text messages to their friends or in group chats successfully.
Test Suite: Messaging
Test Priority: High
Preconditions:
* User is logged in
* User has at least one friend or is part of a group chat
Test Data: Text message content
Test Steps:
1. Log in to the social media platform
2. Select a friend or group chat
3. Type and send a text message
4. Verify the message is sent successfully
Postconditions:
* Message is displayed in the conversation thread
Expected Result: The system sends the text message successfully to the selected friend or group chat.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case 2:

Test Case ID: MSG002

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

Test Case Description: Ensure that users can send images and videos to their friends or in group chats
successfully.
Test Suite: Messaging
Test Priority: High
Preconditions:
* User is logged in
* User has at least one friend or is part of a group chat
Test Data: Image and video files
Test Steps:
1. Log in to the social media platform
2. Select a friend or group chat
3. Select an image or video file to send
4. Verify the file is sent successfully
Postconditions:
* File is displayed in the conversation thread
Expected Result: The system sends the image or video file successfully to the selected friend or group chat.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: MSG003
Test Case Title: Verify that messages can be sent in real-time
Test Case Description: Ensure that messages are delivered in real-time to the recipient.

Test Suite: Messaging
Test Priority: High
Preconditions:
* User is logged in
* User has at least one friend online
Test Data: No test data needed
Test Steps:
1. Log in to the social media platform
2. Select a friend who is online
3. Send a text message
4. Verify the message is delivered immediately to the recipient
Postconditions:
* Message is displayed in the recipient's conversation thread
Expected Result: The system delivers the message in real-time to the recipient.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4:
Test Case ID: MSG004
Test Case Title: Verify that users can start private conversations or group chats
Test Case Description: Ensure that users can initiate private conversations or group chats successfully.
Test Suite: Messaging
Test Priority: Medium

Preconditions:
* User is logged in
* User has at least one friend or multiple friends
Test Data: Friend or friend list
Test Steps:
1. Log in to the social media platform
2. Select a friend or multiple friends
3. Initiate a private conversation or group chat
4. Verify the conversation or chat is created successfully
Postconditions:
* Conversation or chat is displayed in the messaging interface
Expected Result: The system creates a private conversation or group chat successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5:
Test Case ID: MSG005
Test Case Title: Verify that messages show read receipts and users are notified of new messages
Test Case Description: Ensure that messages display read receipts and users receive notifications of new
messages.
Test Suite: Messaging
Test Priority: Medium
Preconditions:

* User has at least one friend or is part of a group chat
Test Data: No test data needed
Test Steps:
1. Log in to the social media platform
2. Select a friend or group chat
3. Send a text message
4. Verify the message shows a read receipt when the recipient reads it
5. Verify the recipient receives a notification of the new message
Postconditions:
* Read receipt is displayed in the conversation thread
* Notification is displayed in the recipient's notification center
Expected Result: The system displays read receipts and sends notifications of new messages successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Data - 4
Project name -
Social Media Platform
Project description -
Develop a comprehensive social media platform where users can create profiles, connect with friends, share

* User is logged in

posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending

topics and communities. The platform should include features like news feed, messaging, user profiles,

privacy settings, notifications, and analytics.

Feature name -

Content Sharing

Feature description -

Users can share various types of content with their followers or the public, including text posts, photos,

videos, links, articles, and documents. Content sharing fosters engagement, creativity, and self-expression

among users, allowing them to share their thoughts, experiences, and interests with others.

Userstory title -

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

Acceptance criteria -

['Users can create and share text posts, photos, and videos.', 'Content can be shared with followers or

publicly.', 'Users can edit or delete their posts.', 'Posts should support hashtags and tagging.', 'Users should

receive feedback through likes, comments, and shares.']

Generated Test Cases:

Here are the 5 test cases for the Content Sharing feature:

Test Case 1

Test Case ID: TC001

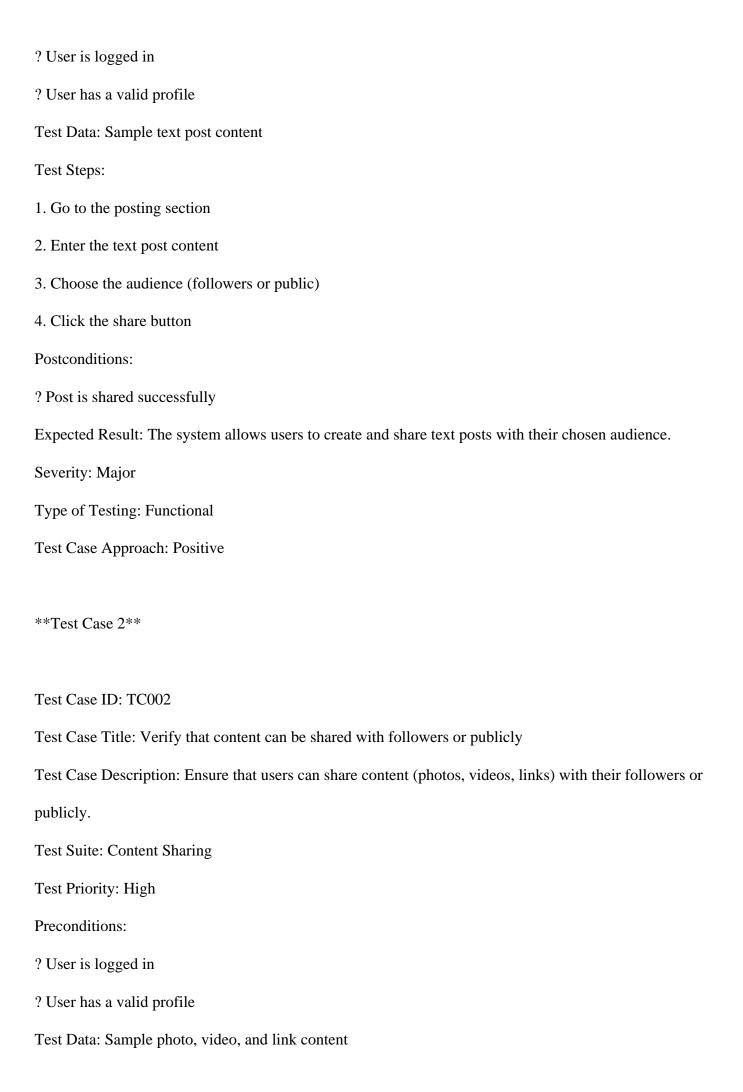
Test Case Title: Verify that users can create and share text posts

Test Case Description: Ensure that users can create and share text posts with their followers or publicly.

Test Suite: Content Sharing

Test Priority: High

Preconditions:



Test Steps:
1. Go to the posting section
2. Choose the content type (photo, video, link)
3. Upload the content
4. Choose the audience (followers or public)
5. Click the share button
Postconditions:
? Content is shared successfully
Expected Result: The system allows users to share content with their chosen audience.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3
Test Case ID: TC003
Test Case Title: Verify that users can edit or delete their posts
Test Case Description: Ensure that users can edit or delete their shared posts.
Test Suite: Content Sharing
Test Priority: Medium
Preconditions:
? User is logged in
? User has a valid profile
? User has a shared post
Test Data: Sample post content
Test Steps:
1. Go to the post

2. Click the edit button 3. Make changes to the post 4. Click the save button 5. Alternatively, click the delete button Postconditions: ? Post is edited or deleted successfully Expected Result: The system allows users to edit or delete their shared posts. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 4** Test Case ID: TC004 Test Case Title: Verify that posts support hashtags and tagging Test Case Description: Ensure that users can add hashtags and tag other users in their posts. Test Suite: Content Sharing Test Priority: Medium Preconditions: ? User is logged in ? User has a valid profile Test Data: Sample post content with hashtags and tag Test Steps: 1. Go to the posting section 2. Enter the post content 3. Add hashtags

4. Tag another user

5. Click the share button
Postconditions:
? Post is shared successfully with hashtags and tag
Expected Result: The system supports hashtags and tagging in posts.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5
Test Case ID: TC005
Test Case Title: Verify that users receive feedback through likes, comments, and shares
Test Case Description: Ensure that users receive feedback on their shared posts through likes, comments, and
shares.
Test Suite: Content Sharing
Test Priority: High
Preconditions:
? User is logged in
? User has a valid profile
? User has a shared post
Test Data: Sample post content
Test Steps:
1. Go to the post
2. Like the post
3. Comment on the post
4. Share the post
Postconditions:

? Feedback is updated successfully

Expected Result: The system updates the feedback (likes, comments, shares) for the shared post.

Severity: Major

Type of Testing: Functional

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:
Here are the 5 detailed test cases for the provided input data:
Test Case 1: Verify that users can set their profile to public
Test Case ID: PRV001
Test Case Title: Verify that users can set their profile to public
Test Case Description: Ensure that the system allows users to set their profile to public, making it accessible to
all users on the platform.
Test Suite: Privacy Settings
Test Priority: High
Preconditions:
- User is logged in
- Profile is created
Test Data: No test data needed
Test Steps:
1. Go to profile settings
2. Select privacy settings
3. Set profile to public
4. Save changes
Postconditions:
- Profile privacy is updated
Expected Result: The system sets the profile to public, allowing all users to view the profile.
Severity: Major
Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that users can change privacy settings at any time

Test Case ID: PRV002

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

Test Case Description: Ensure that the system allows users to change their privacy settings at any time, and

the changes are reflected immediately.

Test Suite: Privacy Settings

Test Priority: Medium

Preconditions:

- User is logged in

- Profile is created

Test Data: No test data needed

Test Steps:

- 1. Go to profile settings
- 2. Select privacy settings
- 3. Change privacy settings to private
- 4. Save changes
- 5. Verify that the changes are reflected on the profile

Postconditions:

- Profile privacy is updated

Expected Result: The system allows users to change their privacy settings at any time, and the changes are

reflected immediately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

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

Test Case ID: PRV003

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

Test Case Description: Ensure that the system allows users to control who can view individual posts, and the

visibility settings are respected.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in

- Post is created

Test Data: Post details

Test Steps:

1. Create a new post

2. Set visibility to restricted

3. Share the post with specific users

4. Verify that only authorized users can view the post

Postconditions:

- Post visibility is updated

Expected Result: The system allows users to control who can view individual posts, and the visibility settings

are respected.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

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

Test Case ID: PRV004

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

Test Case Description: Ensure that the system respects the privacy settings immediately, and any changes are

reflected in real-time.

Test Suite: Privacy Settings

Test Priority: High

Preconditions:

- User is logged in

- Profile is created

Test Data: No test data needed

Test Steps:

1. Go to profile settings

2. Select privacy settings

3. Set profile to private

4. Verify that the changes are reflected on the profile in real-time

Postconditions:

- Profile privacy is updated

Expected Result: The system respects the privacy settings immediately, and any changes are reflected in

real-time.

Severity: Blocker

Type of Testing: Functional

Test Case Approach: Positive

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

Test Case ID: PRV005

Test Case Title: Verify that users are notified of any changes in privacy settings
Test Case Description: Ensure that the system notifies users of any changes made to their privacy settings,
keeping them informed of any updates.
Test Suite: Privacy Settings
Test Priority: Medium
Preconditions:
- User is logged in
- Profile is created
Test Data: No test data needed
Test Steps:
1. Go to profile settings
2. Select privacy settings
3. Change privacy settings to restricted
4. Verify that the user receives a notification of the changes
Postconditions:
- Notification is sent to the user
Expected Result: The system notifies users of any changes made to their privacy settings, keeping them
informed of any updates.
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:

Here are the 5 detailed test cases for the Notifications feature of the Social Media Platform:

Test Case 1: Verify that users receive notifications for likes, comments, and shares

Test Case ID: NTFC001

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

Test Case Description: Ensure that the system sends notifications to users when their posts receive likes,

comments, or shares.

Test Suite: Notifications
Test Priority: High
Preconditions:
* User is logged in
* User has posted content (e.g., photo, video, text)
Test Data: No test data needed
Test Steps:
1. Post content (e.g., photo, video, text)
2. Have another user like, comment, or share the post
3. Verify that the original user receives a notification
Postconditions:
* Notification is displayed in real-time
* Notification includes a link to the relevant activity
Expected Result: The system sends a notification to the original user when their post receives a like, commen
or share.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that users are notified of new friend requests and mentions

Test Case ID: NTFC002
Test Case Title: Verify that users are notified of new friend requests and mentions
Test Case Description: Ensure that the system sends notifications to users when they receive friend requests or
are mentioned in a post.
Test Suite: Notifications
Test Priority: High
Preconditions:
* User is logged in
* Another user sends a friend request or mentions the user in a post
Test Data: No test data needed
Test Steps:
1. Send a friend request to the user
2. Mention the user in a post
3. Verify that the user receives a notification for both actions
Postconditions:
* Notification is displayed in real-time
* Notification includes a link to the relevant activity
Expected Result: The system sends a notification to the user when they receive a friend request or are
mentioned in a post.
Severity: Major

Type of Testing: Functional

Test Case Approach: Positive
Test Case 3: Verify that notifications are displayed in real-time
Test Case ID: NTFC003
Test Case Title: Verify that notifications are displayed in real-time
Test Case Description: Ensure that the system displays notifications in real-time when a user interacts with
their content.
Test Suite: Notifications
Test Priority: High
Preconditions:
* User is logged in
* User has posted content (e.g., photo, video, text)
Test Data: No test data needed
Test Steps:
1. Post content (e.g., photo, video, text)
2. Have another user interact with the post (e.g., like, comment, share)
3. Verify that the notification is displayed in real-time
Postconditions:
* Notification is displayed instantly
* Notification includes a link to the relevant activity

Expected Result: The system displays notifications in real-time when a user interacts with their content.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that users can customize notification preferences
Test Case ID: NTFC004
Test Case Title: Verify that users can customize notification preferences
Test Case Description: Ensure that the system allows users to customize their notification preferences.
Test Suite: Notifications
Test Priority: Medium
Preconditions:
* User is logged in
* User has accessed their notification settings
Test Data: No test data needed
Test Steps:
1. Access notification settings
2. Customize notification preferences (e.g., turn off notifications for likes, turn on notifications for comments)
3. Verify that the system saves the changes
Postconditions:
* Notification preferences are updated

* System honors the user's notification preferences
Expected Result: The system allows users to customize their notification preferences.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that notifications include a link to the relevant activity
Test Case ID: NTFC005
Test Case Title: Verify that notifications include a link to the relevant activity
Test Case Description: Ensure that the system includes a link to the relevant activity in notifications.
Test Suite: Notifications
Test Priority: High
Preconditions:
* User is logged in
* User has posted content (e.g., photo, video, text)
Test Data: No test data needed
Test Steps:
1. Post content (e.g., photo, video, text)
2. Have another user interact with the post (e.g., like, comment, share)
3. Verify that the notification includes a link to the relevant activity
Postconditions:

* Notification includes a link to the relevant activity

* Link redirects to the correct activity page

Expected Result: The system includes a link to the relevant activity in notifications.

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:

Here are the 5 test cases generated for the given input data:

Test Case 1:

Test Case ID: SFT-001

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

Test Case Description: Ensure that the search functionality allows users to find relevant content, profiles, and topics on the platform using keywords or hashtags.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in
- Search functionality is available on the platform

Test Data: Keywords or hashtags (e.g., #socialmedia, "social media", etc.)

Test Steps:

- 1. Log in to the platform
- 2. Enter a keyword or hashtag in the search bar
- 3. Press the search button
- 4. Observe the search results

Postconditions:

- Search results are displayed

Expected Result: The search functionality returns relevant results containing profiles, posts, and topics related

to the searched keyword or hashtag. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2:** Test Case ID: SFT-002 Test Case Title: Verify that search results include profiles, posts, and topics Test Case Description: Ensure that the search results display a comprehensive list of relevant content, including profiles, posts, and topics, when a user searches for a keyword or hashtag. Test Suite: Search Functionality Test Priority: High Preconditions: - User is logged in - Search functionality is available on the platform Test Data: Keywords or hashtags (e.g., #technology, "tech news", etc.) Test Steps: 1. Log in to the platform 2. Enter a keyword or hashtag in the search bar 3. Press the search button 4. Observe the search results 5. Verify that profiles, posts, and topics are included in the results

Expected Result: The search results display a comprehensive list of relevant content, including profiles, posts,

Postconditions:

Severity: Major

- Search results are displayed

and topics related to the searched keyword or hashtag.

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: SFT-003

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

Test Case Description: Ensure that the search results are sorted in order of relevance to the searched keyword

or hashtag, providing users with the most relevant content at the top.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in

- Search functionality is available on the platform

Test Data: Keywords or hashtags (e.g., #music, "rock music", etc.)

Test Steps:

1. Log in to the platform

- 2. Enter a keyword or hashtag in the search bar
- 3. Press the search button
- 4. Observe the search results
- 5. Verify that the results are sorted in order of relevance

Postconditions:

- Search results are displayed

Expected Result: The search results are displayed in order of relevance, with the most relevant content at the

top.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: SFT-004

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

Test Case Description: Ensure that users can filter search results by content type (e.g., profiles, posts, topics)

to narrow down their search results.

Test Suite: Search Functionality

Test Priority: Medium

Preconditions:

- User is logged in
- Search functionality is available on the platform

Test Data: Keywords or hashtags (e.g., #movies, "action movies", etc.)

Test Steps:

- 1. Log in to the platform
- 2. Enter a keyword or hashtag in the search bar
- 3. Press the search button
- 4. Observe the search results
- 5. Filter search results by content type (e.g., profiles, posts, topics)
- 6. Verify that the filtered results are displayed correctly

Postconditions:

- Filtered search results are displayed

Expected Result: The search results are filtered correctly by content type, providing users with relevant results.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: SFT-005

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

Test Case Description: Ensure that the search functionality provides relevant suggestions as the user types

their search query, facilitating a more efficient search experience.

Test Suite: Search Functionality

Test Priority: High

Preconditions:

- User is logged in

- Search functionality is available on the platform

Test Data: Keywords or hashtags (e.g., "soci", "tech", etc.)

Test Steps:

1. Log in to the platform

2. Start typing a keyword or hashtag in the search bar

3. Observe the search suggestions

4. Verify that the suggestions are relevant to the typed query

Postconditions:

- Search suggestions are displayed

Expected Result: The search functionality provides relevant suggestions as the user types their search query,

facilitating a more efficient search experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 8

Project name -

Social Media Platform

Project description -

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

posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending

topics and communities. The platform should include features like news feed, messaging, user profiles,

privacy settings, notifications, and analytics.

Feature name -

Hashtags and Tagging

Feature description -

Hashtags and tagging allow users to categorize their content and make it more discoverable to others. Users

can add relevant hashtags to their posts or tag other users, brands, or locations, increasing the visibility and

reach of their content.

Userstory title -

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

Acceptance criteria -

['Users can add hashtags to their posts.', 'Users can tag other profiles, brands, or locations.', 'Hashtags and tags

are clickable and lead to related content.', 'Users can search for content using hashtags.', 'The system should

suggest popular hashtags as users type.']

Generated Test Cases:

Here are the test cases for the Hashtags and Tagging feature:

Test Case 1:

Test Case ID: HTT001

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

Test Case Description: Ensure that users can add relevant hashtags to their posts, making them discoverable

by others.

Test Suite: Hashtags and Tagging
Test Priority: High
Preconditions:
* User is logged in
* User has created a post
Test Data: Post content, hashtag text
Test Steps:
1. Create a new post
2. Type a hashtag in the post content
3. Click the post button
4. Verify the hashtag is displayed in the post
Postconditions:
* Hashtag is displayed in the post
Expected Result: The system allows users to add hashtags to their posts, making them discoverable by others.
Severity: Major
Type of Testing: Functional
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: Ensure that users can tag other profiles, brands, or locations in their posts, making
them discoverable by others.
Test Suite: Hashtags and Tagging
Test Priority: High
Preconditions:
* User is logged in

* User has created a post

Test Data: Post content, profile/brand/location to tag

Test Steps:

1. Create a new post

2. Type the "@" symbol followed by the profile/brand/location name

3. Select the profile/brand/location from the suggestions

4. Click the post button

5. Verify the tagged profile/brand/location is displayed in the post

Postconditions:

* Tagged profile/brand/location is displayed in the post

Expected Result: The system allows users to tag other profiles, brands, or locations in their posts, making

them discoverable by others.

Severity: Major

Type of Testing: Functional

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: Ensure that hashtags and tags are clickable and lead to related content, enhancing user

engagement.

Test Suite: Hashtags and Tagging

Test Priority: High

Preconditions:

* User is logged in

* A post with a hashtag or tag exists

Test Data: Post with hashtag or tag

Test Steps:
1. Go to a post with a hashtag or tag
2. Click on the hashtag or tag
3. Verify the system redirects to related content
Postconditions:
* System redirects to related content
Expected Result: The system allows users to click on hashtags and tags, leading to related content and
enhancing user engagement.
Severity: Major
Type of Testing: Functional
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: Ensure that users can search for content using hashtags, making it easier to discover
related posts.
Test Suite: Hashtags and Tagging
Test Priority: High
Preconditions:
* User is logged in
* Posts with hashtags exist
Test Data: Hashtag text
Test Data: Hashtag text Test Steps:

2. Type a hashtag

3. Click the search button

4. Verify the system displays related posts Postconditions: * System displays related posts Expected Result: The system allows users to search for content using hashtags, making it easier to discover related posts. Severity: Major Type of Testing: Functional 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: Ensure that the system suggests popular hashtags as users type, enhancing user experience and encouraging engagement. Test Suite: Hashtags and Tagging Test Priority: Medium Preconditions: * User is logged in * User is creating a new post Test Data: Hashtag text Test Steps: 1. Start typing a hashtag in the post content 2. Verify the system suggests popular hashtags

- 3. Select a suggested hashtag
- 4. Verify the hashtag is added to the post content

Postconditions:

* System suggests popular hashtags

Expected Result: The system suggests popular hashtags as users type, enhancing user experience and encouraging engagement.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 9

Project name -

Social Media Platform

Project description -

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

Feature name -

Emojis and Reactions

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

remove or change their reactions.']

Generated Test Cases:

Here are the 5 test cases for the Emojis and Reactions feature:

Test Case 1:

Test Case ID: EMOJI001

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

Test Case Description: This test case ensures that users can react to posts using different emojis, enhancing their expression and engagement on the platform.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

* User is logged in

* Post is available on the news feed

Test Data: No test data needed

Test Steps:

- 1. Go to the news feed
- 2. Select a post
- 3. Click on the reaction button
- 4. Choose an emoji from the available options
- 5. Verify that the reaction is applied to the post

Postconditions:

* Reaction is displayed on the post

Expected Result: The system allows users to react to posts with a variety of emojis, and the reaction is displayed on the post.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: EMOJI002

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

Test Case Description: This test case ensures that users can use emojis in comments and messages to convey

emotions and enhance communication.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

* User is logged in

* Post is available on the news feed

Test Data: No test data needed

Test Steps:

- 1. Go to the news feed
- 2. Select a post
- 3. Click on the comment button
- 4. Type a comment with an emoji
- 5. Verify that the comment with the emoji is posted
- 6. Repeat steps 3-5 for a message

Postconditions:

* Comment and message with emojis are posted

Expected Result: The system allows users to use emojis in comments and messages, and they are displayed

correctly.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: EMOJI003

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

Test Case Description: This test case ensures that users can view the count of reactions on posts, providing

social proof and encouraging engagement.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

* User is logged in

* Post is available on the news feed with at least one reaction

Test Data: No test data needed

Test Steps:

- 1. Go to the news feed
- 2. Select a post with reactions
- 3. Verify that the reaction count is displayed
- 4. Click on the reaction count to view the reaction list
- 5. Verify that the reaction list is displayed correctly

Postconditions:

* Reaction count and list are displayed correctly

Expected Result: The system displays the count of reactions on posts, and users can view the reaction list by

clicking on the count.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: EMOJI004

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

Test Case Description: This test case ensures that the platform supports a diverse range of emojis, catering to

users from different cultures and languages.

Test Suite: Emojis and Reactions

Test Priority: Medium

Preconditions:

* User is logged in

Test Data: List of emojis from different categories (e.g., Smileys, Animals, Flags)

Test Steps:

- 1. Go to the emoji picker
- 2. Verify that the platform supports a wide range of emojis from different categories
- 3. Select an emoji from each category
- 4. Verify that the selected emojis are displayed correctly

Postconditions:

* Emojis from different categories are displayed correctly

Expected Result: The system supports a wide range of emojis from different categories, and they are displayed

correctly.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: EMOJI005

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

Test Case Description: This test case ensures that users can modify or remove their reactions, giving them

control over their interactions on the platform.

Test Suite: Emojis and Reactions

Test Priority: High

Preconditions:

* User is logged in

* Post is available on the news feed with a reaction from the user

Test Data: No test data needed

Test Steps:

1. Go to the news feed

- 2. Select a post with a reaction from the user
- 3. Click on the reaction button
- 4. Verify that the reaction can be removed or changed
- 5. Remove or change the reaction
- 6. Verify that the reaction is updated correctly

Postconditions:

* Reaction is updated correctly

Expected Result: The system allows users to remove or change their reactions, and the reaction is updated

correctly.

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:

Here are five detailed test cases for the Live Streaming feature of the Social Media Platform:

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

Test Case ID: LS001

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

Test Case Description: Ensure that users can successfully initiate and terminate live streams, allowing them to

control their real-time video content.

Test Suite: Live Streaming

Test Priority: High

Preconditions:

* User is logged in

* User has necessary permissions to create live streams

Test Data: No test data needed

Test Steps:

- 1. Log in to the social media platform
- 2. Navigate to the live streaming feature
- 3. Click the "Go Live" button
- 4. Verify that the live stream is started successfully
- 5. Click the "Stop Streaming" button
- 6. Verify that the live stream is stopped successfully

Postconditions:

* Live stream is started and stopped successfully

Expected Result: The user can start and stop live streams without any issues.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that live streams can be public or limited to followers
Test Case ID: LS002

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

Test Case Description: Ensure that users have the option to control the visibility of their live streams, allowing them to choose between public or followers-only access.

Test Suite: Live Streaming

Test Priority: Medium

Preconditions:

- * User is logged in
- * User has necessary permissions to create live streams

Test Data: No test data needed

Test Steps:

- 1. Log in to the social media platform
- 2. Navigate to the live streaming feature
- 3. Click the "Go Live" button
- 4. Select the "Public" option
- 5. Verify that the live stream is accessible to all users
- 6. Select the "Followers-only" option
- 7. Verify that the live stream is only accessible to followers

Postconditions:

* Live stream visibility is updated successfully

Expected Result: The user can choose between public or followers-only live streams.

Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that viewers can interact with the streamer through comments and reactions** Test Case ID: LS003 Test Case Title: Verify that viewers can interact with the streamer through comments and reactions Test Case Description: Ensure that viewers can engage with the streamer in real-time using comments and reactions, enhancing the overall live streaming experience. Test Suite: Live Streaming Test Priority: Medium Preconditions: * User is logged in * User has necessary permissions to view live streams Test Data: No test data needed Test Steps: 1. Log in to the social media platform 2. Navigate to a live stream 3. Enter a comment in the live chat

4. Verify that the comment is displayed in the live chat

5. Click on a reaction button (e.g., "Like" or "Love")

6. Verify that the reaction is displayed in the live chat

Postconditions:

* Viewer's comment and reaction are displayed in the live chat
Expected Result: Viewers can interact with the streamer through comments and reactions.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that the stream displays viewer count and live comments
Test Case ID: LS004
Test Case Title: Verify that the stream displays viewer count and live comments
Test Case Description: Ensure that the live stream displays the viewer count and live comments in real-time,
providing the streamer with a seamless and engaging experience.
Test Suite: Live Streaming
Test Priority: Medium
Preconditions:
* User is logged in
* User has necessary permissions to view live streams
Test Data: No test data needed
Test Steps:
1. Log in to the social media platform
2. Navigate to a live stream
3. Verify that the viewer count is displayed
4. Verify that live comments are displayed in the live chat
Postconditions:

* Viewer count and live comments are displayed in the live stream Expected Result: The stream displays the viewer count and live comments. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 5: Verify that users can save live streams for later viewing** Test Case ID: LS005 Test Case Title: Verify that users can save live streams for later viewing Test Case Description: Ensure that users can save live streams for later viewing, allowing them to access the content at a later time. Test Suite: Live Streaming Test Priority: Medium Preconditions: * User is logged in * User has necessary permissions to view live streams Test Data: No test data needed Test Steps: 1. Log in to the social media platform 2. Navigate to a live stream 3. Click the "Save" button 4. Verify that the live stream is saved successfully 5. Navigate to the saved live streams section

6. Verify that the saved live stream is accessible

Postconditions:

* Live stream is saved successfully

Expected Result: The user can save live streams for later viewing.

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:

Here are the 5 test cases for the Stories feature:

Test Case 1: Verify that users can create stories with photos

Test Case ID: STO-TC-001

Test Case Title: Verify that users can create stories with photos

Test Case Description: Ensure that the system allows users to create stories with photos and upload them

successfully.

Test Suite: Stories

Test Priority: High

Preconditions:

- User is logged in
- User has a profile picture

Test Data: Test photo

Test Steps:

- 1. Go to the story creation page
- 2. Select the photo option
- 3. Upload a test photo
- 4. Add a caption (optional)
- 5. Click on the "Post" button

Postconditions:

- Story is created with the uploaded photo

Expected Result: The system creates a story with the uploaded photo and displays it in the user's story feed.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that stories disappear after 24 hours

Test Case ID: STO-TC-002

Test Case Title: Verify that stories disappear after 24 hours

Test Case Description: Ensure that the system automatically removes stories after a 24-hour period.

Test Suite: Stories

Test Priority: High

Preconditions:

- User has created a story

- Story is at least 24 hours old

Test Data: No test data needed

Test Steps:

- 1. Wait for 24 hours after story creation
- 2. Log in to the user's account
- 3. Check the story feed

Postconditions:

- Story is no longer visible in the story feed

Expected Result: The system automatically removes the story from the user's story feed after 24 hours.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can see who viewed their stories

Test Case ID: STO-TC-003

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

Test Case Description: Ensure that the system displays the list of users who viewed the story.

Test Suite: Stories

Test Priority: Medium

Preconditions:

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

Test Data: No test data needed

Test Steps:

- 1. Log in to the user's account
- 2. Go to the story feed
- 3. Click on the "Viewers" option

Postconditions:

- List of users who viewed the story is displayed

Expected Result: The system displays the list of users who viewed the story.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that stories can be shared publicly or with selected followers

Test Case ID: STO-TC-004

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

Test Case Description: Ensure that the system allows users to control the visibility of their stories.

Test Suite: Stories
Test Priority: High
Preconditions:
- User has created a story
Test Data: No test data needed
Test Steps:
1. Log in to the user's account
2. Go to the story creation page
3. Select the audience option
4. Choose to share the story publicly or with selected followers
5. Click on the "Post" button
Postconditions:
- Story is shared with the selected audience
Expected Result: The system shares the story with the selected audience (publicly or with selected followers).
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that users can add stickers and filters to their stories
Test Case ID: STO-TC-005
Test Case Title: Verify that users can add stickers and filters to their stories
Test Case Description: Ensure that the system allows users to customize their stories with stickers and filters.
Test Suite: Stories
Test Priority: Medium
Preconditions:
- User has created a story

Test Data: No test data needed

Test Steps:

1. Log in to the user's account

2. Go to the story creation page

3. Select the sticker or filter option

4. Choose a sticker or filter to add

5. Click on the "Post" button

Postconditions:

- Story is created with the added sticker or filter

Expected Result: The system allows users to add stickers and filters to their stories.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 12

Project name -

Social Media Platform

Project description -

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

Feature name -

Analytics and Insights

Feature description -

Analytics and insights provide users with data and metrics about their activity, audience engagement, and

content performance on the platform. Users can access information such as follower demographics, post reach,

engagement metrics, and audience growth trends, helping them optimize their presence and strategy on the

platform.

Userstory title -

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

content.

Acceptance criteria -

['Users can view data on follower demographics.', 'Post reach and engagement metrics are available.', 'Users

can track audience growth trends.', 'Analytics are displayed in a user-friendly format.', 'Data can be filtered by

time range and content type.']

Generated Test Cases:

Here are the 5 test cases for the feature "Analytics and Insights" of the Social Media Platform project:

Test Case 1:

Test Case ID: analytics_1

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

Test Case Description: Ensure that the analytics feature displays accurate and up-to-date information about a

user's follower demographics, including age, location, and interests.

Test Suite: Analytics and Insights

Test Priority: High

Preconditions:

* User is logged in

* User has followers

Test Data: No test data needed

Test Steps:

1. Go to the analytics dashboard 2. Click on the "Follower Demographics" tab 3. Verify that the demographics data is displayed accurately Postconditions: * Follower demographics data is updated Expected Result: The analytics feature displays accurate and up-to-date information about a user's follower demographics. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2:** Test Case ID: analytics_2 Test Case Title: Verify that post reach and engagement metrics are available Test Case Description: Ensure that the analytics feature provides users with accurate and up-to-date information about their post reach and engagement metrics, including likes, comments, and shares. Test Suite: Analytics and Insights Test Priority: High Preconditions: * User is logged in * User has published a post Test Data: No test data needed Test Steps:

rest attention

- 1. Go to the analytics dashboard
- 2. Click on the "Post Analytics" tab
- 3. Verify that the post reach and engagement metrics are displayed accurately

Postconditions:

* Post analytics data is updated

Expected Result: The analytics feature provides users with accurate and up-to-date information about their

post reach and engagement metrics.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: analytics 3

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

Test Case Description: Ensure that the analytics feature allows users to view and track their audience growth

trends over time, including follower acquisition and loss rates.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

* User is logged in

* User has followers

Test Data: No test data needed

Test Steps:

1. Go to the analytics dashboard

2. Click on the "Audience Growth" tab

3. Verify that the audience growth trends are displayed accurately

Postconditions:

* Audience growth trends data is updated

Expected Result: The analytics feature allows users to view and track their audience growth trends over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: analytics 4

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

Test Case Description: Ensure that the analytics feature presents data in a clear, concise, and user-friendly

format, including visualizations and graphs.

Test Suite: Analytics and Insights

Test Priority: Medium

Preconditions:

* User is logged in

Test Data: No test data needed

Test Steps:

- 1. Go to the analytics dashboard
- 2. Verify that the analytics data is displayed in a user-friendly format
- 3. Verify that visualizations and graphs are used to present data

Postconditions:

* Analytics data is up-to-date

Expected Result: The analytics feature presents data in a clear, concise, and user-friendly format.

Severity: Minor

Type of Testing: Usability

Test Case Approach: Positive

Test Case 5:

Test Case ID: analytics_5

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

Test Case Description: Ensure that the analytics feature allows users to filter data by time range and content

Test Suite: Analytics and Insights
Test Priority: Medium
Preconditions:
* User is logged in
* User has published multiple posts
Test Data: No test data needed
Test Steps:
1. Go to the analytics dashboard
2. Click on the "Post Analytics" tab
3. Select a specific time range and content type
4. Verify that the analytics data is filtered accurately
Postconditions:
* Filtered analytics data is updated
Expected Result: The analytics feature allows users to filter data by time range and content type.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Data - 13
Project name -
Social Media Platform
Project description -
Develop a comprehensive social media platform where users can create profiles, connect with friends, share

posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending

type, enabling them to gain insights into specific periods and content performance.

topics and communities. The platform should include features like news feed, messaging, user profiles,

privacy settings, notifications, and analytics.

Feature name -

Events and Groups

Feature description -

Events and groups enable users to organize and participate in events, communities, or interest-based groups on

the platform. Users can create events, RSVP to invitations, join groups, and participate in discussions,

fostering community engagement and connection around shared interests.

Userstory title -

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

Acceptance criteria -

['Users can create and join events.', 'Users can join interest-based groups.', 'Group discussions can be

organized by topic.', 'Event details include date, time, location, and description.', 'Users can RSVP to events

and see the list of attendees.']

Generated Test Cases:

Here are the 5 test cases for the input data:

Test Case 1:

Test Case ID: EG001

Test Case Title: Verify that users can create events

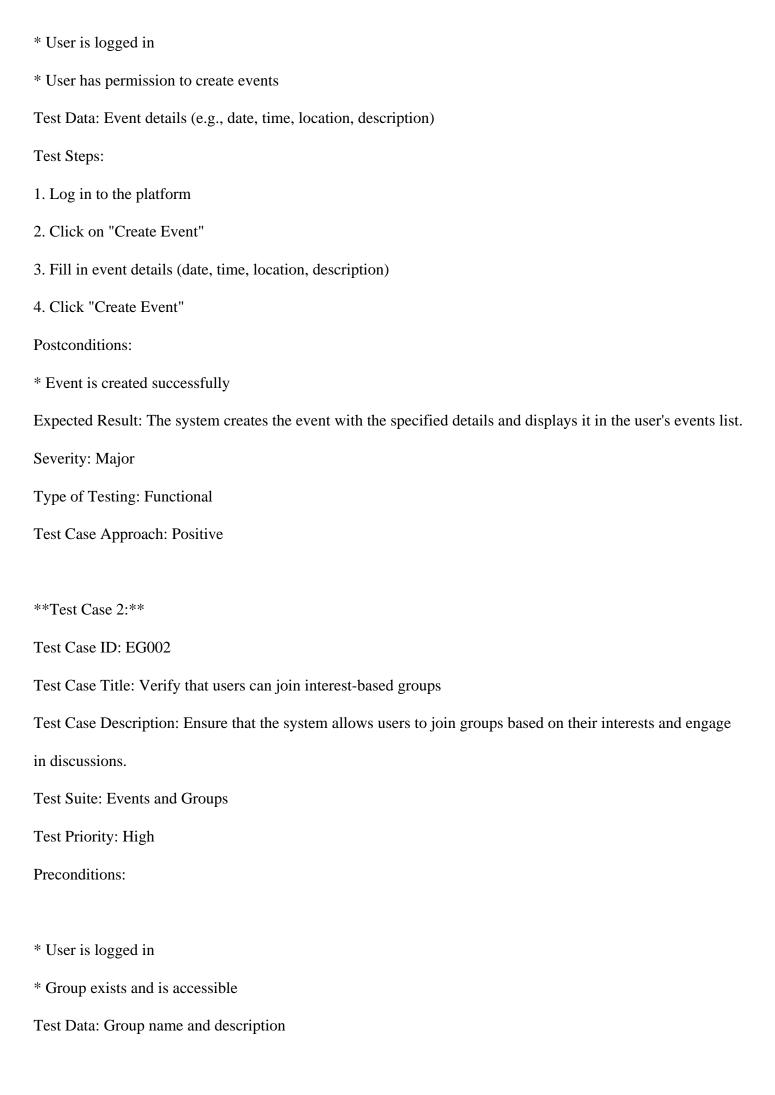
Test Case Description: Ensure that the system allows users to create events with required details such as date,

time, location, and description.

Test Suite: Events and Groups

Test Priority: High

Preconditions:



Test Steps:
1. Log in to the platform
2. Search for a group by interest
3. Click "Join Group"
4. Participate in a discussion topic
Postconditions:
* User is added to the group
Expected Result: The system adds the user to the group and allows them to participate in discussions.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: EG003
Test Case Title: Verify that group discussions can be organized by topic
Test Case Description: Ensure that the system allows users to create and participate in topic-based discussions
within groups.
Test Suite: Events and Groups
Test Priority: Medium
Preconditions:
* User is logged in
* User is a member of a group
Test Data: Discussion topic and message
Test Steps:
1. Log in to the platform
2. Go to a group discussion

3. Create a new topic or respond to an existing one 4. Post a message Postconditions: * Topic is created or updated Expected Result: The system organizes the discussion by topic and displays user posts accordingly. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 4:** Test Case ID: EG004 Test Case Title: Verify that event details include date, time, location, and description Test Case Description: Ensure that the system displays required event details to users. Test Suite: Events and Groups Test Priority: Medium Preconditions: * User is logged in * Event exists and is accessible Test Data: Event ID Test Steps: 1. Log in to the platform 2. Go to an event page 3. Verify event details (date, time, location, description) Postconditions: * Event details are displayed correctly Expected Result: The system displays the event details accurately, including date, time, location, and

description.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5:
Test Case ID: EG005
Test Case Title: Verify that users can RSVP to events and see the list of attendees
Test Case Description: Ensure that the system allows users to respond to event invitations and view the list of
attendees.
Test Suite: Events and Groups
Test Priority: High
Preconditions:
* User is logged in
* Event exists and is accessible
Test Data: Event ID
Test Steps:
1. Log in to the platform
2. Go to an event page
3. RSVP to the event
4. View the list of attendees
Postconditions:
* User is added to the attendee list
Expected Result: The system updates the attendee list and allows users to view it.
Severity: Major
Type of Testing: Functional

Test Case Approach: Positive

Data - 14

Project name -

Social Media Platform

Project description -

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

Feature name -

Monetization Options

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are the 5 detailed test cases for the Monetization Options feature:
Test Case 1: Verify that users can apply for monetization options
Test Case ID: MHgnR1
Test Case Title: Verify that users can apply for monetization options
Test Case Description: Ensure that users can successfully apply for monetization options on the platform.
Test Suite: Monetization Options
Test Priority: High
Preconditions:
* User is logged in
* User has a completed profile
Test Data: No test data needed
Test Steps:
1. Go to the monetization options page
2. Click on the "Apply Now" button
3. Fill out the application form with valid data
4. Submit the application
Postconditions:
* Application is submitted successfully
Expected Result: The system accepts the user's application for monetization options and sends a confirmation
notification.
Severity: Major

Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that monetization includes advertising revenue and sponsored content
rest case 2. Verify that monetization includes advertising revenue and sponsored content
Test Case ID: MHgnR2
Test Case Title: Verify that monetization includes advertising revenue and sponsored content
Test Case Description: Ensure that monetization options include advertising revenue and sponsored content
partnerships.
Test Suite: Monetization Options
Test Priority: High
Preconditions:
* User has been approved for monetization
* User has a published post with sufficient engagement
Test Data: Advertising revenue and sponsored content details
Test Steps:
1. Go to the monetization dashboard
2. Select a published post with sufficient engagement
3. Check for available advertising revenue options
4. Check for available sponsored content partnerships
Postconditions:
* Advertising revenue options are available

* Sponsored content partnerships are available

Expected Result: The system displays available advertising revenue options and sponsored content
partnerships for the selected post.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that users can receive payments through the platform
Test Case ID: MHgnR3
Test Case Title: Verify that users can receive payments through the platform
Test Case Description: Ensure that users can receive payments for their monetized content through the
platform.
Test Suite: Monetization Options
Test Priority: High
Preconditions:
* User has been approved for monetization
* User has earned sufficient revenue from monetized content
Test Data: Payment details (bank account, PayPal, etc.)
Test Steps:
1. Go to the monetization dashboard
2. Check for available payment options
3. Select a payment method (e.g., bank transfer)

4. Enter payment details

5. Confirm payment

Postconditions:
* Payment is processed successfully
Expected Result: The system processes the payment and updates the user's earnings report.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that detailed earnings reports are available
Test Case ID: MHgnR4
Test Case Title: Verify that detailed earnings reports are available
Test Case Description: Ensure that users can access detailed earnings reports for their monetized content.
Test Suite: Monetization Options
Test Priority: Medium
Preconditions:
* User has been approved for monetization
* User has earned revenue from monetized content
Test Data: No test data needed
Test Steps:
1. Go to the monetization dashboard
2. Click on the "Earnings Report" button
3. Select a date range for the report
4. Check for detailed earnings information (e.g., revenue, ad clicks, etc.)

Postconditions:
* Detailed earnings report is generated
Expected Result: The system generates a detailed earnings report for the selected date range, including
revenue, ad clicks, and other relevant information.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that monetization eligibility requirements are clearly stated
Test Case ID: MHgnR5
Test Case Title: Verify that monetization eligibility requirements are clearly stated
Test Case Description: Ensure that the platform clearly states the eligibility requirements for monetization
options.
Test Suite: Monetization Options
Test Priority: Medium
Preconditions:
* User is logged in
* User is on the monetization options page
Test Data: No test data needed
Test Steps:
1. Go to the monetization options page
2. Check for eligibility requirements section

3. Verify that the requirements are clearly stated

Postconditions:

* Eligibility requirements are clearly stated

Expected Result: The system displays clear and concise eligibility requirements for monetization options on

the platform.

Severity: Minor

Type of Testing: Usability

Test Case Approach: Positive

Data - 15

Project name -

Social Media Platform

Project description -

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

Feature name -

Explore or Discover Tab

Feature description -

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

Userstory title -

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

topics.

Acceptance criteria -

['The explore tab shows trending and popular content.', 'Content recommendations are personalized based on

user interests.', 'Users can filter content by category or topic.', 'The tab updates regularly with new content.',

'Users can save or follow content from the explore tab.']

Generated Test Cases:

Based on the provided input data, I will generate exactly 5 detailed test cases to cover all the acceptance

criteria for the Explore or Discover Tab feature.

Test Case 1:

Test Case ID: Explore1

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

Test Case Description: Ensure that the explore tab displays trending and popular content from across the

platform, helping users discover new accounts, topics, and trends.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

? User is logged in

? Explore tab is accessible

Test Data: No test data needed

Test Steps:

1. Log in to the platform

2. Navigate to the explore tab

3. Verify that trending and popular content is displayed

Postconditions:

? Trending and popular content is updated regularly

Expected Result: The explore tab displays trending and popular content from across the platform.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: Explore2

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

Test Case Description: Ensure that the explore tab provides personalized content recommendations based on

users' interests, engagement history, and social connections.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

? User has engaged with content on the platform

? User has connections and interests set up

Test Data: User profile data with interests and engagement history

Test Steps:

- 1. Log in to the platform
- 2. Navigate to the explore tab
- 3. Verify that content recommendations match user interests and engagement history

Postconditions:

? Personalized content recommendations are updated regularly

Expected Result: The explore tab provides personalized content recommendations based on user interests and

engagement history.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: Explore3

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

Test Case Description: Ensure that the explore tab allows users to filter content by category or topic,

facilitating content discovery and exploration.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:

? User is logged in

? Explore tab is accessible

Test Data: No test data needed

Test Steps:

1. Log in to the platform

2. Navigate to the explore tab

3. Select a category or topic filter

4. Verify that content is filtered accordingly

Postconditions:

? Filtered content is updated regularly

Expected Result: The explore tab allows users to filter content by category or topic.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: Explore4

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

Test Case Description: Ensure that the explore tab updates regularly with new content, keeping users engaged

and interested in the platform.

Test Suite: Explore Tab

Test Priority: High

Preconditions:

? User is logged in

? Explore tab is accessible

Test Data: No test data needed

Test Steps:

1. Log in to the platform

2. Navigate to the explore tab

3. Verify that new content is displayed regularly

Postconditions:

? New content is updated in real-time

Expected Result: The explore tab updates regularly with new content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: Explore5

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

Test Case Description: Ensure that the explore tab allows users to save or follow content, enabling easy access

and engagement with discovered content.

Test Suite: Explore Tab

Test Priority: Medium

Preconditions:
? User is logged in
? Explore tab is accessible
Test Data: No test data needed
Test Steps:
1. Log in to the platform
2. Navigate to the explore tab
3. Select content to save or follow
4. Verify that content is saved or followed successfully
Postconditions:
? Saved or followed content is accessible in the user's profile
Expected Result: The explore tab allows users to save or follow content.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Each test case thoroughly covers one or more of the acceptance criteria, ensuring comprehensive coverage and potential defect identification.
Data - 16
Project name -
Social Media Platform
Project description -
Develop a comprehensive social media platform where users can create profiles, connect with friends, share

posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending

topics and communities. The platform should include features like news feed, messaging, user profiles,

privacy settings, notifications, and analytics.

Feature name -

Accessibility Features

Feature description -

Accessibility features ensure that the platform is inclusive and accessible to users with disabilities. This may

include features such as alt text for images, screen reader compatibility, keyboard shortcuts, color contrast

options, and adjustable font sizes, enhancing usability and accessibility for all users.

Userstory title -

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

Acceptance criteria -

['The platform provides alt text for images.', 'Screen reader compatibility is available.', 'Keyboard shortcuts

can be used for navigation.', 'Color contrast and font size options are provided.', 'Accessibility features are

clearly documented.']

Generated Test Cases:

Here are the 5 test cases for the Accessibility Features of the Social Media Platform:

Test Case 1:

Test Case ID: Acc1

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

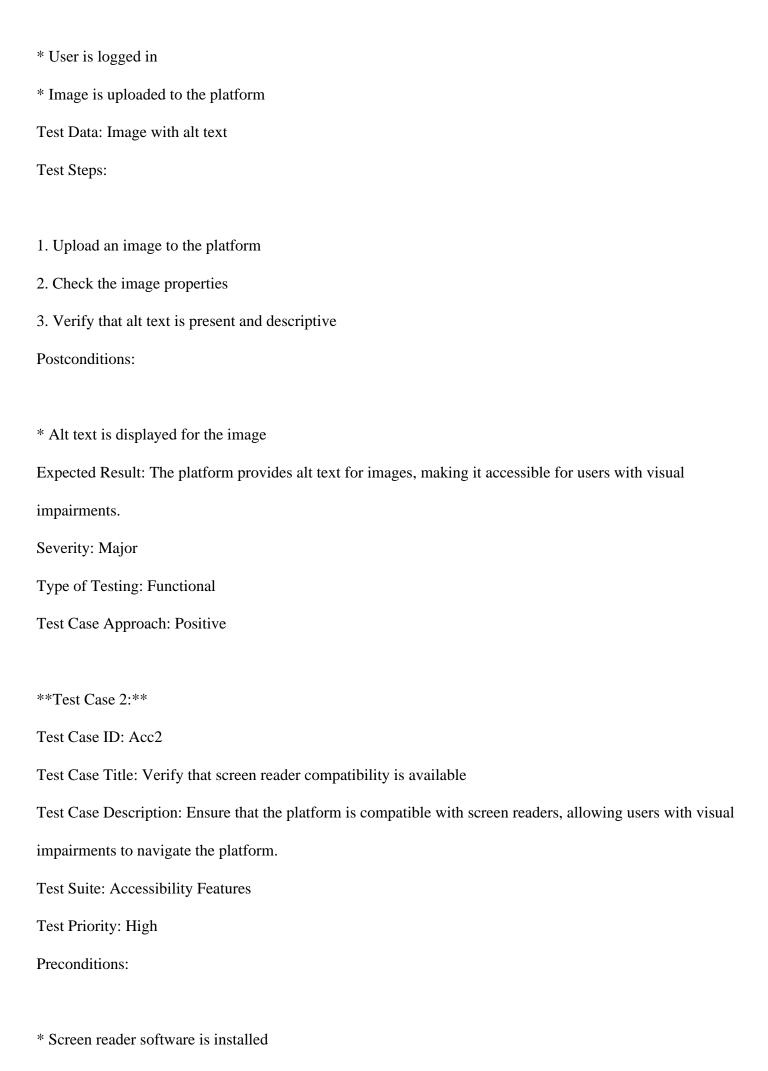
Test Case Description: Ensure that the platform provides alt text for images, making it accessible for users

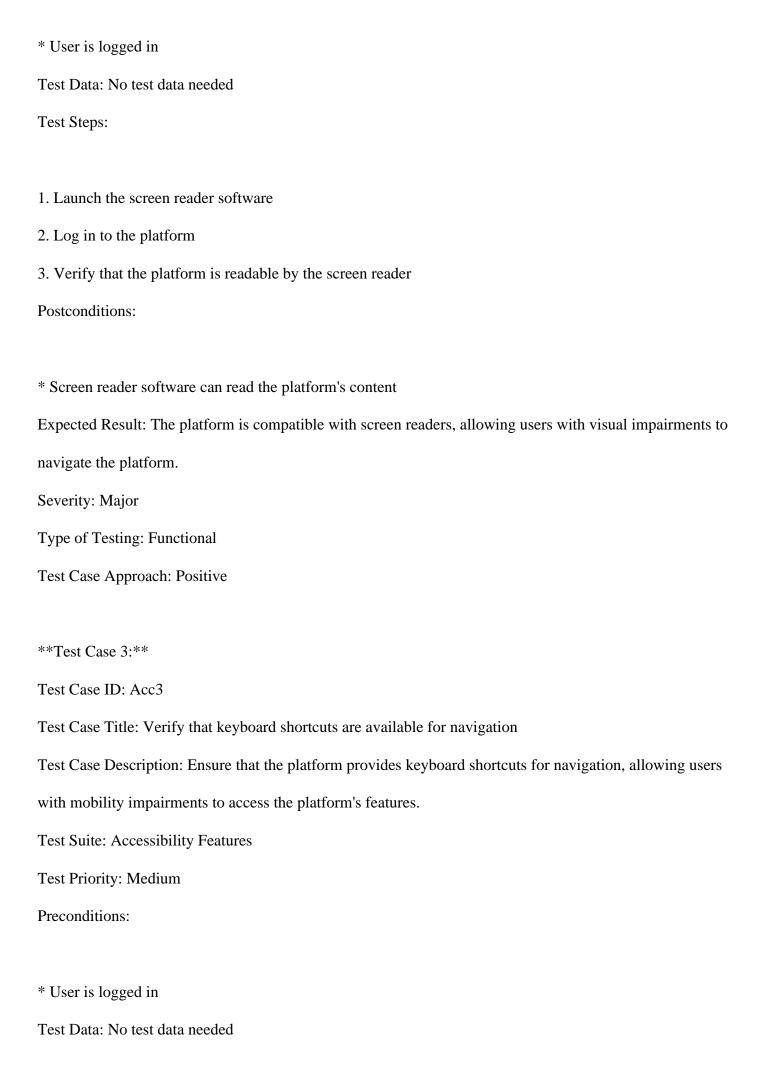
with visual impairments.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:





Test Steps:
1. Log in to the platform
2. Press common keyboard shortcuts (e.g., Ctrl + Home, Ctrl + End)
3. Verify that the platform navigates accordingly
Postconditions:
* Keyboard shortcuts navigate the platform correctly
Expected Result: The platform provides keyboard shortcuts for navigation, allowing users with mobility
impairments to access the platform's features.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4:
Test Case ID: Acc4
Test Case Title: Verify that color contrast and font size options are provided
Test Case Description: Ensure that the platform provides options for adjusting color contrast and font size,
allowing users with visual impairments to customize their experience.
Test Suite: Accessibility Features
Test Priority: Medium
Preconditions:
* User is logged in
Test Data: No test data needed
Test Steps:

1. Log in to the platform
2. Go to settings
3. Verify that color contrast and font size options are available
4. Adjust the options
5. Verify that the platform's appearance changes accordingly
Postconditions:
* Color contrast and font size options are available and functional
Expected Result: The platform provides options for adjusting color contrast and font size, allowing users with
visual impairments to customize their experience.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5:
Test Case ID: Acc5
Test Case Title: Verify that accessibility features are clearly documented
Test Case Description: Ensure that the platform provides clear documentation on its accessibility features,
allowing users with disabilities to understand how to use them.
Test Suite: Accessibility Features
Test Priority: Low
Preconditions:
* User is logged in
Test Data: No test data needed
Test Steps:

- 1. Log in to the platform
- 2. Go to help or support section
- 3. Search for accessibility features documentation
- 4. Verify that the documentation is clear and concise

Postconditions:

* Accessibility features documentation is clear and concise

Expected Result: The platform provides clear documentation on its accessibility features, allowing users with disabilities to understand how to use them.

Severity: Minor

Type of Testing: Usability

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:

Here are the 5 test cases for the Safety and Moderation Tools feature:

Test Case 1:

Test Case ID: SMPT-001

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

Test Case Description: Ensure that the system accurately identifies and filters out harmful or offensive

content, such as profanity, hate speech, or explicit material, from user-generated posts and comments.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in

- Inappropriate content is posted on the platform

Test Data: Sample posts/comments with inappropriate content

Test Steps:

1. Post/content is created with harmful/offensive language

2. Content moderation algorithm is triggered

3. System filters out the inappropriate content Postconditions:

- Inappropriate content is removed from the platform

Expected Result: The system filters out inappropriate content, ensuring a safe and positive experience for users.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: SMPT-002

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

Test Case Description: Ensure that users can easily report harmful or offensive content, which will then be reviewed by moderators for appropriate action.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- Inappropriate content is posted on the platform

Test Data: Sample posts/comments with inappropriate content

Test Steps:

- 1. User identifies inappropriate content
- 2. User reports the content using the reporting feature
- 3. System generates a notification for moderators

Postconditions:

- Moderators review the reported content

Expected Result: Users can report harmful or inappropriate content, which is then reviewed by moderators for

appropriate action. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3:** Test Case ID: SMPT-003 Test Case Title: Verify that blocking capabilities are available to users Test Case Description: Ensure that users can block other users who engage in harmful or inappropriate behavior, preventing future interactions. Test Suite: Safety and Moderation Tools Test Priority: Medium Preconditions: - User is logged in - User is interacting with another user who is engaging in harmful behavior Test Data: User IDs for testing blocking functionality Test Steps: 1. User interacts with another user who is engaging in harmful behavior 2. User blocks the offending user

3. System restricts future interactions between the two users

Postconditions:

- Blocked user cannot interact with the user who blocked them

Expected Result: Users can block other users who engage in harmful or inappropriate behavior, ensuring a safer experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: SMPT-004

Test Case Title: Verify that comment filters prevent offensive language

Test Case Description: Ensure that the system filters out comments that contain offensive language,

preventing harmful or inappropriate interactions.

Test Suite: Safety and Moderation Tools

Test Priority: High

Preconditions:

- User is logged in
- Comment with offensive language is posted

Test Data: Sample comments with offensive language

Test Steps:

- 1. Comment with offensive language is posted
- 2. System filters out the comment
- 3. User is notified that their comment was not posted due to offensive language

Postconditions:

- Comment is not visible on the platform

Expected Result: The system filters out comments with offensive language, maintaining a safe and positive

environment.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: SMPT-005

Test Case Title: Verify that community guidelines are enforced consistently

Test Case Description: Ensure that the system consistently enforces community guidelines, including
consequences for users who violate the guidelines.
Test Suite: Safety and Moderation Tools
Test Priority: High
Preconditions:
- User is logged in
- User violates community guidelines
Test Data: User IDs for testing community guidelines enforcement
Test Steps:
1. User violates community guidelines
2. System flags the violation
3. Consequences are applied (e.g., warning, suspension, ban)
Postconditions:
- User is notified of the consequences
Expected Result: The system consistently enforces community guidelines, maintaining a safe and positive
environment.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Data - 18
Project name -

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

Social Media Platform

Project description -

posts, photos, and videos, engage with content through likes, comments, and shares, and explore trending

topics and communities. The platform should include features like news feed, messaging, user profiles,

privacy settings, notifications, and analytics.

Feature name -

Multi-Language Support

Feature description -

Multi-language support allows the platform to cater to users from diverse linguistic backgrounds by offering

interface localization and content translation features. Users can select their preferred language for the

interface, and posts or messages can be translated into different languages, enhancing accessibility and user

experience across global audiences.

Userstory title -

As a user, I want multi-language support so that I can use the platform in my preferred language.

Acceptance criteria -

['Users can select their preferred interface language.', 'The platform offers content translation features.', 'Posts

and messages can be translated into different languages.', 'Language preferences are saved and applied

consistently.', 'The platform supports a wide range of languages.']

Generated Test Cases:

Here are the 5 test cases for the Multi-Language Support feature:

Test Case 1: Verify that users can select their preferred interface language

Test Case ID: ML1

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

Test Case Description: This test case ensures that users can select their preferred language for the interface,

and the platform applies the changes consistently.

Test Suite: Multi-Language Support

Test Priority: High
Preconditions:
- User is logged in
- User is on the settings page
Test Data: List of supported languages (e.g., English, Spanish, French, Arabic)
Test Steps:
1. Go to the settings page
2. Click on the language dropdown
3. Select a new language (e.g., Spanish)
4. Save changes
Postconditions:
- The interface language is updated
Expected Result: The system updates the interface language to the user's selected preference, and all elements
are translated correctly.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that the platform offers content translation features
Test Case ID: ML2
Test Case Title: Verify that the platform offers content translation features
Test Case Description: This test case ensures that the platform provides content translation features, enabling
users to translate posts and messages into different languages.
Test Suite: Multi-Language Support
Test Priority: High
Preconditions:

- User is logged in
- User has created a post or sent a message

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

Test Steps:

- 1. Create a new post or send a message
- 2. Click on the translate button
- 3. Select a target language (e.g., Arabic)
- 4. Verify the translated content

Postconditions:

- The content is translated correctly

Expected Result: The system provides accurate translations of the post or message content into the selected

language.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that posts and messages can be translated into different languages

Test Case ID: ML3

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

Test Case Description: This test case ensures that posts and messages can be translated into multiple

languages, enhancing accessibility and user experience across global audiences.

Test Suite: Multi-Language Support

Test Priority: High

Preconditions:

- User is logged in
- User has created a post or sent a message

Test Data: List of supported languages (e.g., English, Spanish, French, Arabic) Test Steps: 1. Create a new post or send a message 2. Click on the translate button 3. Select multiple target languages (e.g., Spanish, French, Arabic) 4. Verify the translated content for each language Postconditions: - The content is translated correctly for each language Expected Result: The system translates the post or message content accurately into multiple languages. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 4: Verify that language preferences are saved and applied consistently** Test Case ID: ML4 Test Case Title: Verify that language preferences are saved and applied consistently Test Case Description: This test case ensures that the platform saves and applies user language preferences consistently across different sessions and devices. Test Suite: Multi-Language Support Test Priority: Medium Preconditions: - User is logged in

- User has selected a preferred language

Test Data: No test data needed

1. Log out of the platform

Test Steps:

- 2. Log back in on a different device or browser
- 3. Verify the interface language is still set to the user's preferred language

Postconditions:

- The interface language is still set to the user's preferred language

Expected Result: The system maintains the user's language preferences across different sessions and devices.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that the platform supports a wide range of languages

Test Case ID: ML5

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

Test Case Description: This test case ensures that the platform supports a diverse set of languages, catering to users from various linguistic backgrounds.

Test Suite: Multi-Language Support

Test Priority: Low

Preconditions:

- User is logged in

Test Data: List of supported languages (e.g., English, Spanish, French, Arabic, Chinese, Japanese, Korean)

Test Steps:

- 1. Go to the settings page
- 2. Click on the language dropdown
- 3. Verify the availability of multiple languages
- 4. Select a language and verify the interface translation

Postconditions:

- The interface is translated correctly for the selected language

Expected Result: The system supports a wide range of languages, and the interface is translated accurately for

each language.

Severity: Minor

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:

Here are the 5 detailed test cases for the Cross-Platform Integration feature:

Test Case 1: Verify that users can link their accounts with other social media platforms

Test Case ID: CPI-TC-001

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

Test Case Description: Ensure that users can connect their accounts with other social media platforms,

enabling seamless cross-platform integration.

Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

* User is logged in to the social media platform

* Other social media platforms are available for linking (e.g., Facebook, Twitter, Instagram)

Test Data: No test data needed

Test Steps:

- 1. Go to account settings
- 2. Click on "Link Account" button
- 3. Select another social media platform to link (e.g., Facebook)
- 4. Enter login credentials for the selected platform
- 5. Confirm account linking

Postconditions:
* Account is linked with the selected social media platform
* User can access features of the linked platform
Expected Result: The system allows users to successfully link their accounts with other social media
platforms.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that content can be cross-posted to other platforms
Test Case ID: CPI-TC-002
Test Case Title: Verify that content can be cross-posted to other platforms
Test Case Description: Ensure that users can share content across multiple platforms, increasing content reach
and engagement.
Test Suite: Cross-Platform Integration
Test Priority: High
Preconditions:
* User is logged in to the social media platform
* Account is linked with at least one other social media platform
Test Data: Sample post content (e.g., text, image, video)
Test Steps:

1. Create a new post on the social media platform 2. Select the option to cross-post to other linked platforms 3. Choose the platforms to share the post on (e.g., Facebook, Twitter) 4. Confirm cross-posting Postconditions: * Post is successfully shared on the selected platforms * Post is visible to followers on the selected platforms Expected Result: The system allows users to cross-post content to other linked social media platforms. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that users can sign in using single sign-on (SSO) capabilities** Test Case ID: CPI-TC-003 Test Case Title: Verify that users can sign in using single sign-on (SSO) capabilities Test Case Description: Ensure that users can utilize single sign-on functionality to access their accounts across multiple platforms. Test Suite: Cross-Platform Integration

Test Priority: High

Preconditions:

^{*} User is not logged in to the social media platform

* Account is linked with at least one other social media platform
Test Data: No test data needed
Test Steps:
1. Go to the login page
2. Select the option to use SSO
3. Choose the platform to use for SSO (e.g., Facebook)
4. Enter login credentials for the selected platform
5. Confirm login
Postconditions:
* User is successfully logged in to the social media platform
* User can access features of the platform
Expected Result: The system allows users to utilize SSO capabilities to access their accounts across multiple
platforms.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that integration with external media sources is available
Test Case ID: CPI-TC-004
Test Case Title: Verify that integration with external media sources is available

Test Case Description: Ensure that users can access and share content from external media sources, such as

YouTube or Vimeo.
Test Suite: Cross-Platform Integration
Test Priority: Medium
Preconditions:
* User is logged in to the social media platform
Test Data: Sample external media source content (e.g., YouTube video)
Test Steps:
1. Go to the content creation page
2. Select the option to share content from an external media source
3. Choose the external media source (e.g., YouTube)
4. Enter the content URL or search for the content
5. Confirm sharing
Postconditions:
* Content from the external media source is successfully shared
* Content is visible to followers on the social media platform
Expected Result: The system allows users to integrate and share content from external media sources.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that content sharing across platforms is seamless

Test Case ID: CPI-TC-005 Test Case Title: Verify that content sharing across platforms is seamless Test Case Description: Ensure that users can share content across multiple platforms without encountering any issues or errors. Test Suite: Cross-Platform Integration Test Priority: High Preconditions: * User is logged in to the social media platform * Account is linked with at least one other social media platform Test Data: Sample post content (e.g., text, image, video) Test Steps: 1. Create a new post on the social media platform 2. Select the option to share the post on other linked platforms 3. Confirm sharing 4. Verify that the post is visible on the selected platforms 5. Repeat the process with different content types (e.g., video, image) Postconditions:

* Post is successfully shared on the selected platforms

* Post is visible to followers on the selected platforms

Expected Result: The system allows seamless content sharing across multiple platforms.

Severity: Major

Type of Testing: Functional

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:

Here are the 5 test cases for the Gamification and Rewards feature:

Test Case 1: Earn Points and Badges

Test Case ID: Gam1

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

Test Case Description: Ensure that the system awards points and badges to users for their activities, such as posting, commenting, and sharing content.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User is logged in
- User has created at least one post

Test Data: No test data needed

Test Steps:

- 1. Log in as a user
- 2. Create a new post
- 3. Comment on the post
- 4. Share the post
- 5. Verify that points and badges are awarded

Postconditions:

- Points and badges are updated in the user's profile

Expected Result: The system awards points and badges to the user for their activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Leaderboard Display

Test Case ID: Gam2

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

Test Case Description: Ensure that the system displays a leaderboard showcasing top users based on their

points or achievements.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- Multiple users have earned points and badges

Test Data: No test data needed

Test Steps:

1. Log in as a user

2. Navigate to the leaderboard page

3. Verify that the leaderboard displays top users based on points or achievements

4. Verify that the user's ranking is updated correctly

Postconditions:

- Leaderboard is updated with the latest user rankings

Expected Result: The system displays a leaderboard showcasing top users based on their points or

achievements.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Challenges and Rewards

Test Case ID: Gam3

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

Test Case Description: Ensure that the system provides challenges for users to complete and earn rewards,

such as virtual goods, discounts, or exclusive content.

Test Suite: Gamification and Rewards

Test Priority: High

Preconditions:

- User is logged in

Test Data: No test data needed

Test Steps:

- 1. Log in as a user
- 2. Navigate to the challenges page
- 3. Select a challenge to complete
- 4. Complete the challenge
- 5. Verify that the reward is awarded

Postconditions:

- User's progress is updated in the challenges page

Expected Result: The system provides challenges for users to complete and earn rewards.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: View Progress and Achievements

Test Case ID: Gam4

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

Test Case Description: Ensure that the system provides a dedicated section for users to view their progress and

achievements, including earned points, badges, and rewards.

Test Suite: Gamification and Rewards

Test Priority: Medium

Preconditions:

- User is logged in

- User has earned points, badges, and rewards

Test Data: No test data needed

Test Steps:

1. Log in as a user

2. Navigate to the user's profile page

3. Verify that the progress and achievements section is displayed

4. Verify that the section displays earned points, badges, and rewards

Postconditions:

- User's progress and achievements section is updated

Expected Result: The system provides a dedicated section for users to view their progress and achievements.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Invalid Reward Redemption

Test Case ID: Gam5

Test Case Title: Verify that the system prevents invalid reward redemption

Test Case Description: Ensure that the system prevents users from redeeming rewards that are not earned or

are invalid.

Test Suite: Gamification and Rewards Test Priority: Medium Preconditions: - User is logged in - User has not earned a specific reward Test Data: No test data needed Test Steps: 1. Log in as a user 2. Attempt to redeem a reward that was not earned 3. Verify that the system prevents the redemption Postconditions: - User's reward balance is not updated Expected Result: The system prevents users from redeeming rewards that are not earned or are invalid. Severity: Minor Type of Testing: Functional

Data - 21

Project name -

Task Management Tool

Test Case Approach: Negative

Project description -

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

Feature name -

Task Creation

Feature description -

Users can create tasks by providing detailed information such as the task title, description, due date, priority

level, and assigning it to specific team members. This feature serves as the foundation for organizing and

managing tasks within the tool.

Userstory title -

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

Acceptance criteria -

['Users can enter a task title, description, due date, and priority level.', 'Tasks can be assigned to specific team

members.', 'The system should save and display created tasks.', 'Users can edit task details after creation.', 'The

system should validate required fields (e.g., task title).']

Generated Test Cases:

Here are the 5 detailed test cases for the Task Management Tool's Task Creation feature:

Test Case 1: Verify that users can enter task title, description, due date, and priority level

Test Case ID: TC001

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

Test Case Description: Ensure that users can create tasks by providing detailed information such as task title,

description, due date, and priority level.

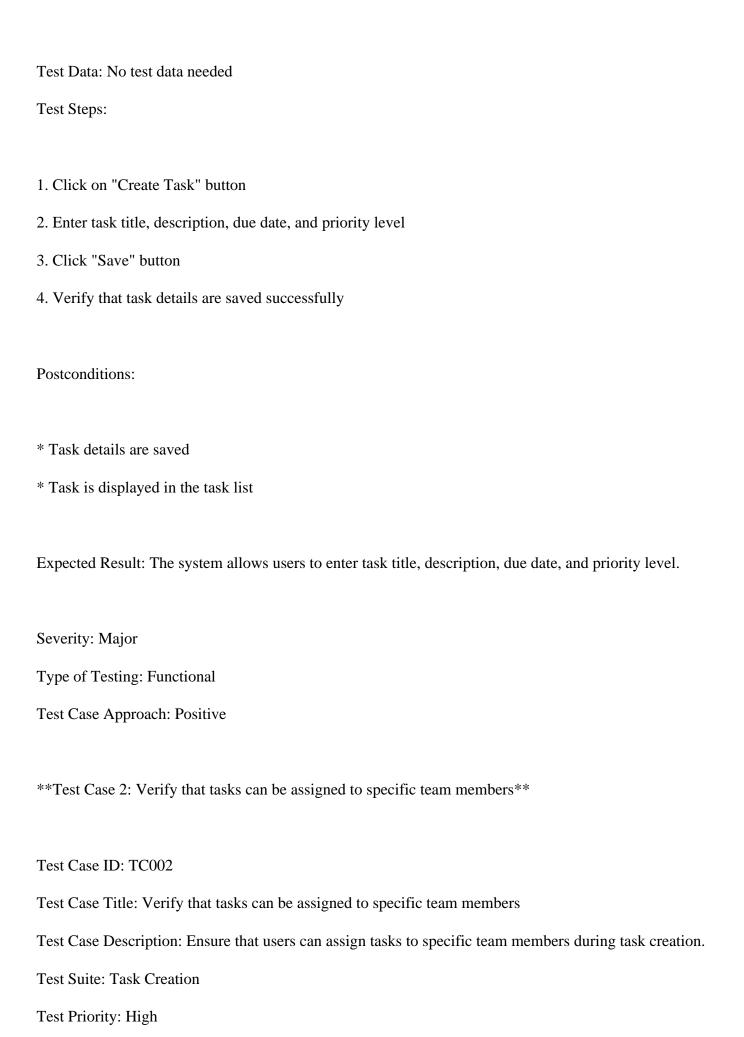
Test Suite: Task Creation

Test Priority: High

Preconditions:

* User is logged in

* Access to Task Creation feature



Preconditions:
* User is logged in
* Access to Task Creation feature
* Team members are available for assignment
Test Data: List of team members
Test Steps:
1. Click on "Create Task" button
2. Enter task title, description, due date, and priority level
3. Select a team member from the dropdown list
4. Click "Save" button
5. Verify that task is assigned to the selected team member
Postconditions:
* Task is assigned to the selected team member
* Task is displayed in the task list with assigned team member's name
Expected Result: The system allows tasks to be assigned to specific team members.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that the system saves and displays created tasks

Test Case ID: TC003
Test Case Title: Verify that the system saves and displays created tasks
Test Case Description: Ensure that the system saves and displays created tasks in the task list.
Test Suite: Task Creation
Test Priority: High
Preconditions:
* User is logged in
* Access to Task Creation feature
Test Data: No test data needed
Test Steps:
1. Click on "Create Task" button
2. Enter task title, description, due date, and priority level
3. Click "Save" button
4. Verify that task is displayed in the task list
Postconditions:
* Task is displayed in the task list
* Task details are displayed correctly
Expected Result: The system saves and displays created tasks in the task list.
Severity: Critical

Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that users can edit task details after creation
Test Case ID: TC004
Test Case Title: Verify that users can edit task details after creation
Test Case Description: Ensure that users can edit task details after task creation.
Test Suite: Task Creation
Test Priority: Medium
Preconditions:
* User is logged in
* Access to Task Creation feature
* Task is created and saved
Test Data: Updated task details
Test Steps:
1. Click on the created task
2. Click on "Edit" button
3. Update task title, description, due date, and priority level
4. Click "Save" button
5. Verify that task details are updated successfully

Postconditions:

* Task details are updated
* Updated task details are displayed in the task list
Expected Result: The system allows users to edit task details after creation.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that the system validates required fields
Test Case ID: TC005
Test Case Title: Verify that the system validates required fields
Test Case Description: Ensure that the system validates required fields such as task title during task creation.
Test Suite: Task Creation
Test Priority: High
Preconditions:
* User is logged in
* Access to Task Creation feature
Test Data: No test data needed
Test Steps:
1. Click on "Create Task" button
2. Leave task title blank
3. Click "Save" button

4. Verify that error message is displayed for required field
Postconditions:
* Error message is displayed for required field
* Task is not saved
Expected Result: The system validates required fields and displays an error message if not filled.
Severity: Blocker
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:

Here are the 5 detailed test cases for the Task Management Tool's Task Assignment feature:

Test Case 1: Verify that users can select team members to assign tasks

Test Case ID: TA001

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

Test Case Description: Ensure that users can select team members from the available list to assign tasks, and

the system correctly assigns the task to the selected team member.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

* User is logged in

* A task is created

* Team members are available

Test Data: No test data needed

Test Steps:

1. Go to the task details page

- 2. Click on the 'Assign' button
- 3. Select a team member from the dropdown list
- 4. Click 'Assign' to save changes

Postconditions:

* Task is assigned to the selected team member

Expected Result: The system allows users to select team members to assign tasks and correctly assigns the task to the selected team member.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that assignees receive notifications about their tasks

Test Case ID: TA002

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

Test Case Description: Ensure that when a task is assigned to a team member, the system sends a notification to the assignee about the new task assignment.

Test Suite: Task Assignment

Test Priority: High

Preconditions:

- * User is logged in
- * A task is created
- * Team member is available

Test Data: No test data needed

Test Steps:

- 1. Assign a task to a team member
- 2. Verify that the team member receives a notification about the new task assignment

3. Check the notification content for accuracy Postconditions: * Notification is sent to the assignee Expected Result: The system sends a notification to the assignee about the new task assignment, and the notification content is accurate. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that assigned tasks are displayed in the assignee's task list** Test Case ID: TA003 Test Case Title: Verify that assigned tasks are displayed in the assignee's task list Test Case Description: Ensure that when a task is assigned to a team member, the system displays the task in the assignee's task list. Test Suite: Task Assignment Test Priority: High Preconditions: * User is logged in * A task is created * Team member is available

Test Data: No test data needed

1. Assign a task to a team member

3. Verify that the assigned task is displayed in the assignee's task list

2. Log in as the assignee

Test Steps:

Postconditions:

* Assigned task is displayed in the assignee's task list

Expected Result: The system displays the assigned task in the assignee's task list, and the task details are

accurate.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that users can reassign tasks to different team members

Test Case ID: TA004

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

Test Case Description: Ensure that users can reassign tasks to different team members, and the system

correctly updates the task assignment and notification.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

* User is logged in

* A task is created

* Team members are available

Test Data: No test data needed

Test Steps:

1. Assign a task to a team member

2. Reassign the task to a different team member

3. Verify that the task is reassigned to the new team member

4. Verify that the original assignee receives a notification about the task reassignment

Postconditions:

* Task is reassigned to the new team member

Expected Result: The system allows users to reassign tasks to different team members, correctly updates the task assignment, and sends notifications to both the original and new assignees.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that task assignment history is tracked

Test Case ID: TA005

Test Case Title: Verify that task assignment history is tracked

Test Case Description: Ensure that the system maintains a record of task assignment history, including

changes to task assignments and notifications.

Test Suite: Task Assignment

Test Priority: Medium

Preconditions:

* User is logged in

* A task is created

* Team members are available

Test Data: No test data needed

Test Steps:

- 1. Assign a task to a team member
- 2. Reassign the task to a different team member
- 3. Verify that the task assignment history is updated to reflect the changes
- 4. Verify that the task assignment history includes notifications sent to assignees

Postconditions:

* Task assignment history is updated

Expected Result: The system maintains a record of task assignment history, including changes to task

assignments and notifications, and this history is accurate and up-to-date.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 23

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Deadline Reminders

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are the 5 detailed test cases for the Task Deadline Reminders feature:
Test Case 1: Verify that users can set reminders for tasks based on deadlines
Test Case ID: TDRL1
Test Case Title: Verify that users can set reminders for tasks based on deadlines
Test Case Description: Ensure that users can configure reminders for tasks based on deadlines, allowing them
to stay on track and meet task completion dates.
Test Suite: Task Deadline Reminders
Test Priority: High
Preconditions:
* User is logged in
* Task is created with a deadline
Test Data: Task details with deadline
Test Steps:
1. Open the task details page
2. Click on the "Reminders" tab
3. Select the reminder option based on the deadline
4. Choose the desired reminder timing (e.g., 1 day, 1 week)
5. Save changes
Postconditions:
* Reminder is set for the task

Expected Result: The system allows users to set reminders for tasks based on deadlines, enabling them to receive timely notifications.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that reminders can be customized by urgency level

Test Case ID: TDRL2

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

Test Case Description: Ensure that users can adjust reminder settings based on the task's urgency level,

ensuring they receive notifications aligned with the task's priority.

Test Suite: Task Deadline Reminders

Test Priority: Medium

Preconditions:

- * User is logged in
- * Task is created with a deadline
- * Multiple urgency levels are available (e.g., low, medium, high)

Test Data: Task details with deadline and multiple urgency levels

Test Steps:

- 1. Open the task details page
- 2. Click on the "Reminders" tab
- 3. Select the reminder option based on the deadline
- 4. Choose the desired urgency level (e.g., high, medium)

6. Verify that the reminder settings are updated accordingly
Postconditions:
* Reminder settings are updated based on the selected urgency level
Expected Result: The system allows users to customize reminders by urgency level, ensuring they receive
notifications that align with the task's priority.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that the system sends notifications for upcoming deadlines
Test Case ID: TDRL3
Test Case Title: Verify that the system sends notifications for upcoming deadlines
Test Case Description: Ensure that the system sends notifications to users for upcoming deadlines, keeping
them informed and on track to complete tasks on time.
Test Suite: Task Deadline Reminders
Test Priority: High
Preconditions:
* User is logged in
* Task is created with a deadline
* Reminder is set for the task
Test Data: Task details with deadline and reminder set

5. Save changes

Test Steps:
1. Wait for the reminder trigger (e.g., 1 day before deadline)
2. Verify that the notification is sent to the user (e.g., email, in-app)
3. Check the notification content (e.g., task name, deadline, urgency level)
Postconditions:
* Notification is sent to the user
Expected Result: The system sends timely notifications to users for upcoming deadlines, keeping them
informed and on track to complete tasks on time.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that users can choose the delivery method for reminders
Test Case ID: TDRL4
Test Case Title: Verify that users can choose the delivery method for reminders
Test Case Description: Ensure that users can select their preferred delivery method for reminders, allowing
them to receive notifications in a way that suits them best.
Test Suite: Task Deadline Reminders
Test Priority: Medium
Preconditions:
* User is logged in
* Task is created with a deadline

* Multiple delivery methods are available (e.g., email, in-app, SMS)
Test Data: Task details with deadline and multiple delivery methods Test Steps:
1. Open the task details page
2. Click on the "Reminders" tab
3. Select the reminder option based on the deadline
4. Choose the desired delivery method (e.g., email, in-app)
5. Save changes
Postconditions:
* Reminder delivery method is updated
Expected Result: The system allows users to choose their preferred delivery method for reminders, ensuring
they receive notifications in a way that suits them best.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that reminder settings can be adjusted or disabled
Test Case ID: TDRL5
Test Case Title: Verify that reminder settings can be adjusted or disabled
Test Case Description: Ensure that users can modify or disable reminder settings as needed, giving them
control over their notification preferences.
Test Suite: Task Deadline Reminders

Test Priority: Medium
Preconditions:
* User is logged in
* Task is created with a deadline
* Reminder is set for the task
Test Data: Task details with deadline and reminder set
Test Steps:
1. Open the task details page
2. Click on the "Reminders" tab
3. Modify the reminder settings (e.g., change reminder timing, disable reminders)
4. Save changes
Postconditions:
* Reminder settings are updated or disabled
Expected Result: The system allows users to adjust or disable reminder settings, giving them control over their
notification preferences.
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 prioritize tasks based on their urgency and importance, allowing them to focus on high-priority

tasks and manage their workload effectively. Tasks can be categorized into different priority levels, such as

high, medium, or low, based on their importance to the project or team.

Userstory title -

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

Acceptance criteria -

['Users can set priority levels for tasks (High, Medium, Low).', 'Tasks can be sorted based on priority levels.',

'Users can change the priority level of a task.', 'The system visually distinguishes tasks based on priority.',

'Tasks without a specified priority default to Medium.']

Generated Test Cases:

Here are five detailed test cases for the Task Prioritization feature:

Test Case 1: Verify that users can set priority levels for tasks

Test Case ID: TP-001

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

Test Case Description: This test case ensures that users can set priority levels for tasks as High, Medium, or

Low.
Test Suite: Task Prioritization
Test Priority: High
Preconditions:
- User is logged in
- Task is created
Test Data: No test data needed
Test Steps:
1. Go to the task list
2. Select a task
3. Click on the priority dropdown
4. Select a priority level (High, Medium, or Low)
5. Save changes
Postconditions:
- Task priority is updated
Expected Result: The system allows users to set priority levels for tasks as High, Medium, or Low, and
displays the updated priority.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case 2: Verify that tasks can be sorted based on priority levels

Test Case ID: TP-002

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

Test Case Description: This test case ensures that tasks can be sorted based on their priority levels.

Test Suite: Task Prioritization

Test Priority: Medium Preconditions: - User is logged in - Multiple tasks with different priority levels are created Test Data: No test data needed Test Steps: 1. Go to the task list 2. Click on the priority column header 3. Select the sorting option (e.g., High to Low or Low to High) 4. Verify the task list is sorted correctly Postconditions: - Task list is sorted based on priority levels Expected Result: The system sorts tasks based on their priority levels, displaying High priority tasks first, followed by Medium and then Low priority tasks. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that users can change the priority level of a task**

Test Case ID: TP-003

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 priority level of a task.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- User is logged in

- Task is created with an initial priority level Test Data: No test data needed Test Steps: 1. Go to the task list 2. Select a task 3. Click on the priority dropdown 4. Select a new priority level (different from the initial priority) 5. Save changes Postconditions: - Task priority is updated Expected Result: The system allows users to change the priority level of a task, and displays the updated priority. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 4: Verify that the system visually distinguishes tasks based on priority** Test Case ID: TP-004 Test Case Title: Verify that the system visually distinguishes tasks based on priority Test Case Description: This test case ensures that the system visually distinguishes tasks based on their priority levels. Test Suite: Task Prioritization Test Priority: Medium Preconditions: - User is logged in

- Multiple tasks with different priority levels are created

Test Data: No test data needed

Test Steps:

1. Go to the task list

2. Verify that tasks with different priority levels have distinct visual indicators (e.g., icons, colors, or labels)

Postconditions:

- Tasks are visually distinguished based on priority levels

Expected Result: The system visually distinguishes tasks based on their priority levels, providing a clear visual

indication of task priority.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that tasks without a specified priority default to Medium

Test Case ID: TP-005

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

Test Case Description: This test case ensures that tasks without a specified priority level default to Medium.

Test Suite: Task Prioritization

Test Priority: Low

Preconditions:

- User is logged in

- Task is created without specifying a priority level

Test Data: No test data needed

Test Steps:

1. Go to the task list

2. Verify that the task is assigned a Medium priority level

Postconditions:

- Task priority is set to Medium

Expected Result: The system defaults tasks without a specified priority level to Medium.

Severity: Trivial

Type of Testing: Functional

Test Case Approach: Positive

Data - 25

Project name -

Task Management Tool

Project description -

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

Feature name -

Subtasks

Feature description -

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

Userstory title -

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

Acceptance criteria -

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

Generated Test Cases:

Here are	the 5	test	cases	for the	e given	input	data:

Test Case 1: Verify that users can create subtasks under main tasks

Test Case ID: TC001

Test Case Description: Ensure that users can create subtasks under main tasks, enabling them to break down complex tasks into manageable parts.

Test Suite: Subtasks

Test Priority: High

Preconditions:

- User is logged in
- Main task is created

Test Data: No test data needed

Test Steps:

- 1. Go to the main task
- 2. Click on the "Create Subtask" button
- 3. Enter subtask details (title, description, due date, assignee)
- 4. Save the subtask

Postconditions:

- Subtask is created under the main task

Expected Result: The system allows users to create subtasks under main tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: TC002
Test Case Description: Ensure that subtasks can have their own due dates and assignees, providing flexibility
in task execution.
Test Suite: Subtasks
Test Priority: Medium
Preconditions:
- User is logged in
- Main task is created
- Subtask is created
Test Data: No test data needed
Test Steps:
1. Go to the subtask
2. Update the due date
3. Assign a team member to the subtask
4. Save the changes
Postconditions:
- Subtask due date and assignee are updated
Expected Result: The system allows users to set independent due dates and assignees for subtasks.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that the system tracks the progress of each subtask

Test Case ID: TC003

Test Case 2: Verify that subtasks can have their own due dates and assignees

Test Case Description: Ensure that the system tracks the progress of each subtask, enabling users to monitor
task execution effectively.
Test Suite: Subtasks
Test Priority: High
Preconditions:
- User is logged in
- Main task is created
- Subtask is created
Test Data: No test data needed
Test Steps:
1. Go to the subtask
2. Update the subtask status (e.g., "In Progress")
3. Verify the updated status is reflected in the main task
Postconditions:
- Subtask status is updated
Expected Result: The system tracks the progress of each subtask and updates the main task accordingly.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that subtasks can be marked as complete independently of the main task
Test Case ID: TC004
Test Case Description: Ensure that subtasks can be marked as complete independently of the main task,
allowing users to track progress accurately.
Test Suite: Subtasks

Test Priority: Medium

Preconditions:
- User is logged in
- Main task is created
- Subtask is created
Test Data: No test data needed
Test Steps:
1. Go to the subtask
2. Mark the subtask as complete
3. Verify the subtask status is updated
Postconditions:
- Subtask is marked as complete
Expected Result: The system allows users to mark subtasks as complete independently of the main task.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that dependencies between subtasks can be set
Test Case ID: TC005
Test Case Description: Ensure that dependencies between subtasks can be set, enabling users to define task
execution order effectively.
Test Suite: Subtasks
Test Priority: High
Preconditions:
- User is logged in
- Main task is created

- Multiple subtasks are created

Test Data: No test data needed

Test Steps:

1. Go to the subtask list

2. Set a dependency between two subtasks

3. Verify the dependency is reflected in the subtask list

Postconditions:

- Dependency between subtasks is set

Expected Result: The system allows users to set dependencies between subtasks, defining the task execution order.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 26

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Dependencies

Feature description -

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

Userstory title -

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

Acceptance criteria -

['Users can set dependencies between tasks.', 'The system displays dependencies in the task details.', 'Tasks

cannot be marked as complete if dependent tasks are not completed.', 'Users can update or remove

dependencies as needed.', 'The system provides alerts or warnings if a dependent task is not completed.']

Generated Test Cases:

Here are the five test cases for the Task Dependencies feature:

Test Case 1:

Test Case ID: TD1

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

Test Case Description: Ensure that the system allows users to define dependencies between tasks, specifying

the order in which tasks should be completed.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in

- At least two tasks are created

Test Data: Task A and Task B

Test Steps:

1. Go to Task A

2. Click on the "Dependencies" tab

3. Add Task B as a dependent task

4. Save changes Postconditions: - Task A depends on Task B Expected Result: The system successfully sets the dependency between Task A and Task B, displaying the dependency in the task details. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2:** Test Case ID: TD2 Test Case Title: Verify that the system displays dependencies in the task details Test Case Description: Ensure that the system displays the dependencies set by the user in the task details section. Test Suite: Task Dependencies Test Priority: Medium Preconditions: - User is logged in - Task dependencies are set Test Data: No test data needed Test Steps: 1. Go to Task A 2. Click on the "Details" tab 3. Verify that the dependencies are displayed Postconditions:

- Dependencies are displayed

Expected Result: The system correctly displays the dependencies set by the user in the task details section.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TD3

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

Test Case Description: Ensure that the system prevents users from marking a task as complete if its dependent

tasks are not completed.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in
- Task dependencies are set
- Dependent task is not completed

Test Data: Task A and its dependent task

Test Steps:

- 1. Go to Task A
- 2. Try to mark Task A as complete
- 3. Verify that the system prevents completion

Postconditions:

- Task A is not marked as complete

Expected Result: The system prevents the user from marking Task A as complete if its dependent task is not

completed.

Severity: Major

Type of Testing: Functional Test Case Approach: Negative **Test Case 4:** Test Case ID: TD4 Test Case Title: Verify that users can update or remove dependencies as needed Test Case Description: Ensure that the system allows users to update or remove dependencies after they are set. Test Suite: Task Dependencies Test Priority: Medium Preconditions: - User is logged in - Task dependencies are set Test Data: Task A and its dependent task Test Steps: 1. Go to Task A 2. Update the dependency (e.g., add a new dependent task) 3. Save changes 4. Verify that the dependency is updated Postconditions: - Dependency is updated Expected Result: The system successfully updates the dependency, and the changes are reflected in the task details section. Severity: Minor Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TD5

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

Test Case Description: Ensure that the system provides alerts or warnings to users if a dependent task is not

completed, preventing potential workflow disruptions.

Test Suite: Task Dependencies

Test Priority: High

Preconditions:

- User is logged in

- Task dependencies are set

- Dependent task is not completed

Test Data: Task A and its dependent task

Test Steps:

1. Go to Task A

2. Verify that an alert or warning is displayed

3. Verify that the alert or warning indicates the dependent task is not completed

Postconditions:

- Alert or warning is displayed

Expected Result: The system provides an alert or warning to the user, indicating that a dependent task is not

completed, and preventing potential workflow disruptions.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 27

Project name -

Task Management Tool

Project description -

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

should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Comments and Collaboration

Feature description -

Users can collaborate on tasks by leaving comments, attaching files, and engaging in discussions directly

within the task interface. This feature facilitates communication and teamwork, allowing team members to

provide updates, share feedback, and ask questions related to the task.

Userstory title -

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

Acceptance criteria -

['Users can add comments to tasks.', 'Comments can include text, attachments, and links.', 'Users can reply to

and edit their comments.', 'The system notifies users of new comments.', 'Comments are displayed in

chronological order.']

Generated Test Cases:

Here are the 5 detailed test cases for the Task Comments and Collaboration feature:

Test Case 1: Verify that users can add comments to tasks

Test Case ID: TC001

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

Test Case Description: Ensure that users can add comments to tasks, enabling collaboration and discussion.

Test Suite: Task Comments and Collaboration

Test Priority: High

Preconditions:

? User is logged in

? Task is created

Test Data: No test data needed

Test Steps:

1. Log in to the application

2. Create a new task or select an existing one

3. Click on the comment button

4. Enter a comment and submit

Postconditions:

? Comment is added to the task

Expected Result: The system allows users to add comments to tasks.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that comments can include text, attachments, and links

Test Case ID: TC002

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

Test Case Description: Ensure that comments can include various media types, enhancing collaboration and

discussion.

Test Suite: Task Comments and Collaboration

Preconditions:
? User is logged in
? Task is created
Test Data:
? Sample text
? Sample attachment (e.g., image, document)
? Sample link
Test Steps:
1. Log in to the application
2. Create a new task or select an existing one
3. Click on the comment button
4. Enter a comment with text, attachment, and link
5. Submit the comment
Postconditions:
? Comment is added to the task with included media
Expected Result: The system allows comments to include text, attachments, and links.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that users can reply to and edit their comments
Test Case ID: TC003
Test Case Title: Verify that users can reply to and edit their comments
Test Case Description: Ensure that users can engage in discussions by replying to and editing their comments

Test Priority: High

Test Suite: Task Comments and Collaboration

Preconditions:
? User is logged in
? Task is created with a comment
Test Data:
? Sample comment reply
? Sample edit to the comment
Test Steps:
1. Log in to the application
2. Select a task with an existing comment
3. Click on the reply button
4. Enter a reply and submit
5. Edit the original comment
6. Save the changes
Postconditions:
? Reply is added to the comment thread
? Edited comment is updated
Expected Result: The system allows users to reply to and edit their comments.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that the system notifies users of new comments
Test Case ID: TC004
Test Case Title: Verify that the system notifies users of new comments

Test Case Description: Ensure that the system alerts users to new comments, keeping them informed and

Test Priority: High

Test Suite: Task Comments and Collaboration
Test Priority: High
Preconditions:
? User is logged in
? Task is created with a comment
Test Data:
? Sample new comment
Test Steps:
1. Log in to the application
2. Select a task with an existing comment
3. Add a new comment as another user
4. Verify notification receipt
Postconditions:
? Notification is sent to the user
Expected Result: The system notifies users of new comments.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that comments are displayed in chronological order
Test Case ID: TC005
Test Case Title: Verify that comments are displayed in chronological order
Test Case Description: Ensure that comments are organized chronologically, facilitating easy tracking of
discussions.
Test Suite: Task Comments and Collaboration

engaged.

Test Priority: High
Preconditions:
? User is logged in
? Task is created with multiple comments
Test Data:
? Sample multiple comments with different timestamps
Test Steps:
1. Log in to the application
2. Select a task with multiple comments
3. Verify comment order
Postconditions:
? Comments are displayed in chronological order
Expected Result: The system displays comments in chronological order.
Severity: Major
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:

Here are the 5 detailed test cases for the File Attachments feature:

Test Case 1: Verify that multiple files can be attached to a task

Test Case ID: FM1

Test Case Title: Verify that multiple files can be attached to a task

Test Case Description: Ensure that users can attach multiple files to a task, including documents, images, and

other resources.

Test Suite: File Attachments

Test Priority: High

Preconditions:

- User is logged in

- Task is created

Test Data: Three files of different types (e.g., docx, png, pdf)

Test Steps:
1. Go to the task details page
2. Click on the "Attach file" button
3. Select three different files to attach
4. Click "Upload"
Postconditions:
- Multiple files are attached to the task
Expected Result: The system allows users to attach multiple files to a task.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that attachments can include various file types
Test Case ID: FM2
Test Case Title: Verify that attachments can include various file types
Test Case Description: Ensure that users can attach files of different types, including documents, images,
spreadsheets, and links.
Test Suite: File Attachments
Test Priority: High
Preconditions:
- User is logged in
- Task is created
Test Data: Files of different types (e.g., docx, png, xls, pdf, url)
Test Steps:
1. Go to the task details page
2. Click on the "Attach file" button

3. Select a file of each type to attach 4. Click "Upload" Postconditions: - Files of different types are attached to the task Expected Result: The system allows users to attach files of various types. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that users can view and download attachments** Test Case ID: FM3 Test Case Title: Verify that users can view and download attachments Test Case Description: Ensure that users can view and download attached files from the task details page. Test Suite: File Attachments Test Priority: High Preconditions: - User is logged in - Task is created with attached files Test Data: No test data needed Test Steps: 1. Go to the task details page 2. Click on an attached file 3. Verify that the file can be viewed or downloaded Postconditions: - File is downloaded or viewed successfully

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

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that the system limits the size of attachments

Test Case ID: FM4

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

Test Case Description: Ensure that the system restricts the size of attachments to prevent data overload.

Test Suite: File Attachments

Test Priority: Medium

Preconditions:

- User is logged in

- Task is created

Test Data: Large file (>10MB)

Test Steps:

1. Go to the task details page

2. Click on the "Attach file" button

3. Select a large file to attach

4. Verify that the system restricts the upload

Postconditions:

- System restricts the upload of large files

Expected Result: The system limits the size of attachments to prevent data overload.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Negative

Test Case 5: Verify that users can remove attachments
Γest Case ID: FM5
Γest Case Title: Verify that users can remove attachments
Γest Case Description: Ensure that users can remove attached files from the task details page.
Γest Suite: File Attachments
Γest Priority: Medium
Preconditions:
User is logged in
Task is created with attached files
Γest Data: No test data needed
Γest Steps:
. Go to the task details page
2. Click on an attached file
3. Click on the "Remove" button
4. Verify that the file is removed
Postconditions:
Attachment is removed from the task
Expected Result: The system allows users to remove attached files.
Severity: Minor
Γype 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:

Here are the 5 test cases for the Recurring Tasks feature:

Test Case 1: Verify that users can set tasks to recur at specified intervals

Test Case ID: RT001

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

Test Case Description: Ensure that users can set tasks to recur at daily, weekly, or monthly intervals

successfully.
Test Suite: Recurring Tasks
Test Priority: High
Preconditions:
* User is logged in
* Task is created
Test Data: Task details with recurrence interval (e.g., daily, weekly, monthly)
Test Steps:
1. Create a new task
2. Set the recurrence interval (daily, weekly, or monthly)
3. Save the task
Postconditions:
* Task is saved with recurrence interval
Expected Result: The system allows users to set tasks to recur at specified intervals correctly.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that the system generates instances of recurring tasks automatically
Test Case ID: RT002
Test Case Title: Verify that the system generates instances of recurring tasks automatically
Test Case Description: Ensure that the system automatically generates instances of recurring tasks based on
the set recurrence interval.

Test Suite: Recurring Tasks
Test Priority: High
Preconditions:
* User is logged in
* Task is created with recurrence interval
Test Data: No test data needed
Test Steps:
1. Wait for the recurrence interval to pass (e.g., 1 day, 1 week, 1 month)
2. Check the task list for new instances of the recurring task
Postconditions:
* New instances of the recurring task are generated
Expected Result: The system generates instances of recurring tasks automatically based on the set recurrence
interval.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that users can modify the recurrence pattern or cancel the recurrence
Test Case ID: RT003
Test Case Title: Verify that users can modify the recurrence pattern or cancel the recurrence
Test Case Description: Ensure that users can modify the recurrence pattern or cancel the recurrence of a task
successfully.
Test Suite: Recurring Tasks

Test Priority: Medium
Preconditions:
* User is logged in
* Task is created with recurrence interval
Test Data: Task details with modified recurrence interval or cancellation
Test Steps:
1. Edit the task and modify the recurrence interval or cancel the recurrence
2. Save the changes
Postconditions:
* Task recurrence is updated or cancelled
Expected Result: The system allows users to modify the recurrence pattern or cancel the recurrence correctly.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that the system notifies users of upcoming instances of recurring tasks
Test Case ID: RT004
Test Case Title: Verify that the system notifies users of upcoming instances of recurring tasks
Test Case Description: Ensure that the system sends notifications to users about upcoming instances of
recurring tasks.
Test Suite: Recurring Tasks
Test Priority: Medium
Preconditions:

* User is logged in
* Task is created with recurrence interval and notification settings
Test Data: No test data needed
Test Steps:
1. Wait for the recurrence interval to approach (e.g., 1 day, 1 week, 1 month)
2. Check the user's notification settings
Postconditions:
* User receives notification about the upcoming instance of the recurring task
Expected Result: The system notifies users of upcoming instances of recurring tasks correctly.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that users can view a list of all instances of a recurring task
Test Case ID: RT005
Test Case Title: Verify that users can view a list of all instances of a recurring task
Test Case Description: Ensure that users can view a list of all instances of a recurring task, including past and
future instances.
Test Suite: Recurring Tasks
Test Priority: Medium
Preconditions:
* User is logged in

* Task is created with recurrence interval Test Data: Task details with recurrence interval Test Steps: 1. Go to the task list 2. Filter tasks by recurrence 3. Check the list of instances of the recurring task Postconditions: * List of instances of the recurring task is displayed Expected Result: The system allows users to view a list of all instances of a recurring task correctly. 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:

Here are the 5 test cases generated based on the input data:

Test Case 1:

Test Case ID: TKL123

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

Test Case Description: Ensure that the system allows users to add tags and labels to tasks, enabling

categorization and filtering.

Test Suite: Task Tags and Labels

Test Priority: High

Preconditions:

- User is logged in

- Task is created

Test Data: Tag and label names

Test Steps:

1. Go to task details page

2. Click on "Add Tag" button

3. Enter tag name and label name4. Click "Save" buttonPostconditions:Tag and label are displayed on the task list

Expected Result: The system allows users to add tags and labels to tasks successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TKL124

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

Test Case Description: Ensure that the system allows users to customize and edit tags, enabling flexible categorization.

Test Suite: Task Tags and Labels

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created
- Tag is added to the task

Test Data: Updated tag name

Test Steps:

- 1. Go to task details page
- 2. Click on the three-dot menu next to the tag
- 3. Select "Edit" option
- 4. Update the tag name
- 5. Click "Save" button

Postconditions:

- Tag is updated on the task list

Expected Result: The system allows users to customize and edit tags successfully.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TKL125

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

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

enabling efficient task management.

Test Suite: Task Tags and Labels

Test Priority: High

Preconditions:

- User is logged in

- Multiple tasks are created with different tags and labels

Test Data: Filter and sort criteria (e.g., tag, label, priority)

Test Steps:

- 1. Go to task list page
- 2. Select a filter criteria (e.g., tag, label)
- 3. Choose a sorting option (e.g., alphabetical, priority)
- 4. Apply the filter and sort

Postconditions:

- Tasks are filtered and sorted accordingly

Expected Result: The system allows users to filter and sort tasks based on tags and labels successfully.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TKL126

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

Test Case Description: Ensure that the system displays tags and labels clearly in the task list, enabling easy

task identification.

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 task list page

2. Verify that tags and labels are displayed next to each task

Postconditions:

- Tags and labels are visible on the task list

Expected Result: The system displays tags and labels prominently in the task list.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TKL127

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

Test Case Description: Ensure that the system allows users to remove tags and labels from tasks, enabling
flexibility in task categorization.
Test Suite: Task Tags and Labels
Test Priority: Medium
Preconditions:
- User is logged in
- Task is created with a tag and label
Test Data: No test data needed
Test Steps:
1. Go to task details page
2. Click on the three-dot menu next to the tag or label
3. Select "Remove" option
Postconditions:
- Tag or label is removed from the task list
Expected Result: The system allows users to remove tags and labels from tasks successfully.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Data - 31

Project name -

Task Management Tool

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Status Tracking

Feature description -

Users can track the status of tasks, including their progress, completion percentage, and current stage. Task

status updates provide visibility into task lifecycle and project progression, helping users monitor task

progress and identify bottlenecks.

Userstory title -

As a user, I want to track the status of tasks so that I can monitor progress and identify bottlenecks.

Acceptance criteria -

['Users can update the status of tasks (e.g., To Do, In Progress, Completed).', 'The system displays task status

prominently in the task list.', 'Users can view the progress percentage of tasks.', 'The system provides

notifications for status changes.', 'Task status can be filtered and sorted.']

Generated Test Cases:

Here are the five detailed test cases for the Task Status Tracking feature:

Test Case 1: Verify that users can update the status of tasks

Test Case ID: TST-1

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

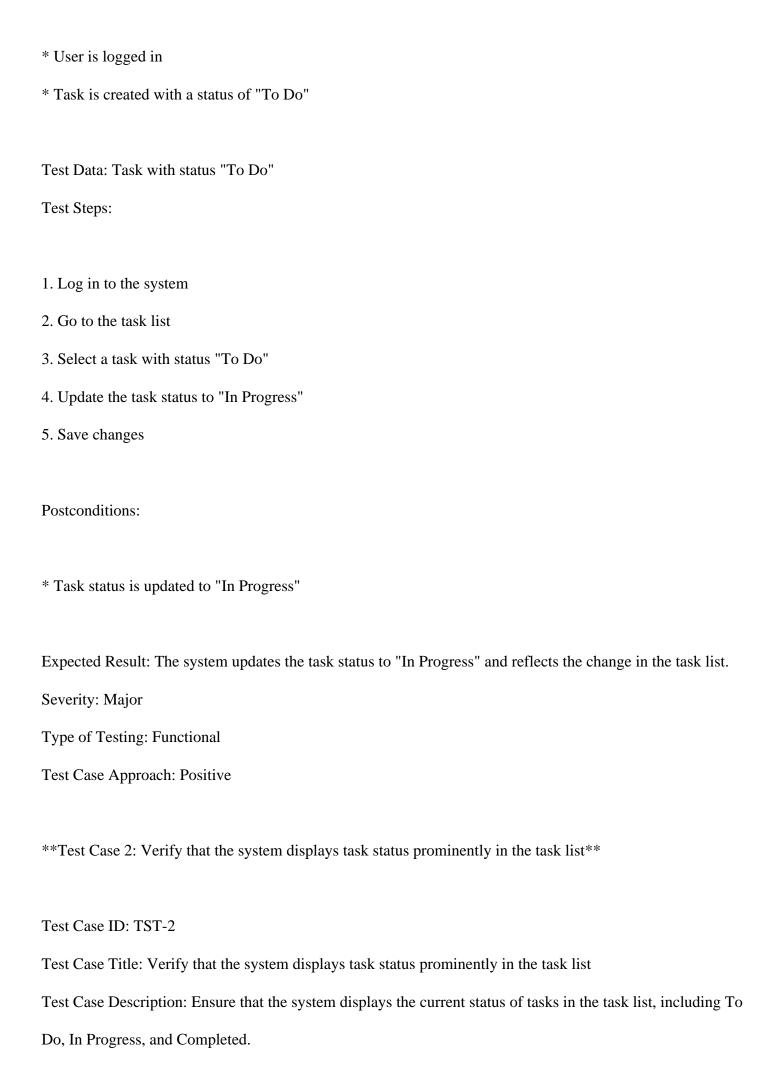
Test Case Description: Ensure that the system allows users to update the status of tasks to To Do, In Progress,

or Completed.

Test Suite: Task Status Tracking

Test Priority: High

Preconditions:

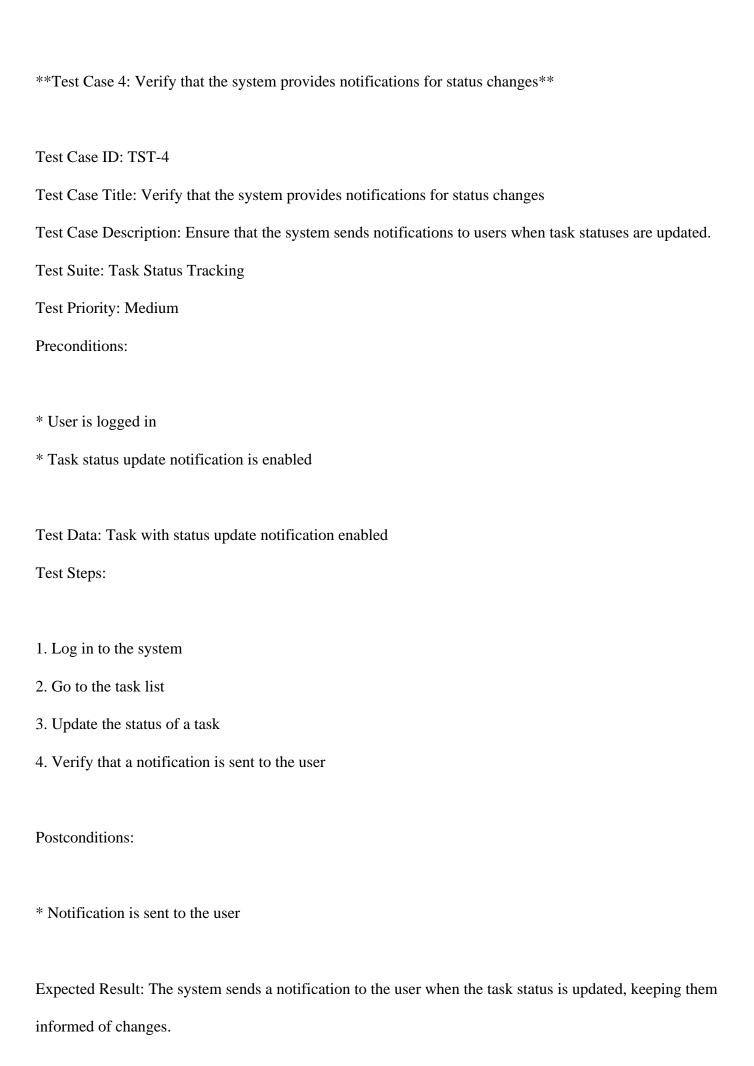


Test Suite: Task Status Tracking
Test Priority: High
Preconditions:
* User is logged in
* Multiple tasks with different statuses are created
Test Data: Multiple tasks with different statuses
Test Steps:
1. Log in to the system
2. Go to the task list
3. Verify that each task displays its current status (To Do, In Progress, or Completed)
Postconditions:
* Task list displays current status of each task
Expected Result: The system displays the current status of each task in the task list, allowing users to quickly
identify task progress.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that users can view the progress percentage of tasks

Test Case ID: TST-3

Test Case Title: Verify that users can view the progress percentage of tasks Test Case Description: Ensure that the system displays the progress percentage of tasks, allowing users to track task completion. Test Suite: Task Status Tracking Test Priority: Medium Preconditions: * User is logged in * Task with progress percentage is created Test Data: Task with progress percentage Test Steps: 1. Log in to the system 2. Go to the task list 3. Select a task with progress percentage 4. Verify that the progress percentage is displayed Postconditions: * Progress percentage is displayed for the selected task Expected Result: The system displays the progress percentage of the task, enabling users to monitor task progress. Severity: Minor Type of Testing: Functional

Test Case Approach: Positive



Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that task status can be filtered and sorted
Test Case ID: TST-5
Test Case Title: Verify that task status can be filtered and sorted
Test Case Description: Ensure that the system allows users to filter and sort tasks by their status (To Do, In
Progress, or Completed).
Test Suite: Task Status Tracking
Test Priority: Low
Preconditions:
* User is logged in
* Multiple tasks with different statuses are created
Test Data: Multiple tasks with different statuses
Test Steps:
1. Log in to the system
2. Go to the task list
3. Filter tasks by status (e.g., "To Do")
4. Verify that only tasks with the filtered status are displayed
5. Sort tasks by status (e.g., "In Progress" first)
6. Verify that tasks are sorted correctly

Postconditions:

* Tasks are filtered and sorted correctly

Expected Result: The system allows users to filter and sort tasks by their status, making it easier to manage

and track tasks.

Severity: Trivial

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:

Here are the 5 test cases for the Task Management Tool feature "Task Filtering and Sorting":

Test Case 1: Filter Tasks by Due Date

Test Case ID: FTS_FILT_DUE_DATE

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

Test Case Description: Ensure that users can filter tasks based on their due dates, allowing them to focus on

urgent tasks.

Test Suite: Task Filtering and Sorting

Test Priority: High

Preconditions:

* User is logged in

* Multiple tasks with different due dates are created

Test Data: No test data needed

Test Steps:

1. Go to the task list

2. Click on the filter button

3. Select the "Due Date" filter option

4. Choose a specific date or range
5. Apply the filter
Postconditions:
* Only tasks with the selected due date or range are displayed
Expected Result: The system filters tasks by due date, allowing users to view relevant tasks.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Sort Tasks by Priority
Test Case ID: FTS_SORT_PRIOR
Test Case Title: Verify that tasks can be sorted by priority
Test Case Description: Ensure that tasks can be sorted by their priority, enabling users to prioritize their work.
Test Suite: Task Filtering and Sorting
Test Priority: Medium
Preconditions:
* User is logged in
* Multiple tasks with different priorities are created
Test Data: No test data needed
Test Steps:
1. Go to the task list
2. Click on the sort button
3. Select the "Priority" sort option

4. Choose a specific sort order (e.g., high to low)
5. Apply the sort
Postconditions:
* Tasks are sorted by priority in the selected order
Expected Result: The system sorts tasks by priority, allowing users to focus on high-priority tasks.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Apply Multiple Filter Options
Test Case ID: FTS_MULT_FILT
Test Case Title: Verify that users can apply multiple filter options
Test Case Description: Ensure that users can apply multiple filter options to narrow down their task list.
Test Suite: Task Filtering and Sorting
Test Priority: High
Preconditions:
* User is logged in
* Multiple tasks with different attributes (e.g., due date, priority, assignee) are created
Test Data: No test data needed
Test Steps:
1. Go to the task list
2. Click on the filter button
3. Select multiple filter options (e.g., due date, priority, assignee)

4. Apply the filters
Postconditions:
* Tasks are filtered by all selected options
Expected Result: The system applies multiple filter options, allowing users to view specific tasks.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Save and Apply Custom Filter and Sort Settings
Test Case ID: FTS_CUST_FILT_SORT
Test Case Title: Verify that users can save and apply custom filter and sort settings
Test Case Description: Ensure that users can save their preferred filter and sort settings for future use.
Test Suite: Task Filtering and Sorting
Test Priority: Medium
Preconditions:
* User is logged in
* Multiple tasks with different attributes are created
Test Data: No test data needed
Test Steps:
1. Go to the task list
2. Click on the filter and sort buttons
3. Select custom filter and sort options
4. Save the settings

5. Log out and log back in
6. Go to the task list
7. Apply the saved settings
Postconditions:
* Tasks are filtered and sorted according to the saved settings
Expected Result: The system saves and applies custom filter and sort settings, allowing users to quickly access
their preferred view.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Filter and Sort Options Do Not Affect Actual Data
Test Case ID: FTS_FILT_SORT_NODATA_CHANGE
Test Case Title: Verify that filter and sort options do not affect actual task data
Test Case Description: Ensure that applying filter and sort options does not alter the underlying task data.
Test Suite: Task Filtering and Sorting
Test Priority: High
Preconditions:
* User is logged in
* Multiple tasks with different attributes are created
Test Data: No test data needed
Test Steps:
1. Go to the task list

2. Apply a filter or sort option

3. Verify that the task data remains unchanged

Postconditions:

* Task data remains unchanged

Expected Result: The system does not modify the actual task data when applying filter and sort options.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Negative

Data - 33

Project name -

Task Management Tool

Project description -

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

Feature name -

Time Tracking

Feature description -

Users can track the time spent on individual tasks, enabling accurate time management, project billing, and resource allocation. Time tracking features can include timers, timesheets, and reports that capture billable hours and productivity metrics.

Userstory title -

As a user, I want to track the time spent on tasks so that I can manage my time efficiently.

Acceptance criteria -

['Users can start and stop a timer for each task.', 'The system records and displays total time spent on tasks.', 'Users can manually enter or adjust time entries.', 'Time tracking data is stored and can be exported.', 'The system provides reports on time spent per task or project.']

Generated Test Cases:

Here are the 5 test cases for the Time Tracking feature:

Test Case 1:

Test Case ID: TT001

Test Case Title: Verify that users can start and stop a timer for each task

Test Case Description: This test case ensures that users can accurately track time spent on individual tasks by starting and stopping a timer.

Test Suite: Time Tracking

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: No test data needed

Test Steps:

- 1. Go to the task details page
- 2. Click on the "Start Timer" button
- 3. Perform some work on the task
- 4. Click on the "Stop Timer" button

Postconditions:

- Timer is stopped

Expected Result: The system records the time spent on the task accurately.

Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2:** Test Case ID: TT002 Test Case Title: Verify that the system records and displays total time spent on tasks Test Case Description: This test case ensures that the system accurately records and displays the total time spent on tasks, including multiple sessions. Test Suite: Time Tracking Test Priority: High Preconditions: - User is logged in - Task is created - Timer is started and stopped multiple times Test Data: No test data needed Test Steps: 1. Go to the task details page 2. Check the total time spent on the task 3. Verify that the total time is correct Postconditions:

- Total time is updated

Expected Result: The system displays the total time spent on the task accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TT003

Test Case Title: Verify that users can manually enter or adjust time entries

Test Case Description: This test case ensures that users can manually enter or adjust time entries for tasks, and

the system updates the total time spent accurately.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

- User is logged in

- Task is created

Test Data: Manual time entry data (e.g., 2 hours, 30 minutes)

Test Steps:

1. Go to the task details page

2. Click on the "Log Time" button

3. Enter manual time entry

4. Verify that the total time is updated

Postconditions:

- Total time is updated

Expected Result: The system updates the total time spent on the task accurately based on the manual time

entry.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TT004

Test Case Title: Verify that time tracking data is stored and can be exported

Test Case Description: This test case ensures that the system stores time tracking data and allows users to export it for further analysis or reporting.

Test Suite: Time Tracking

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created
- Time tracking data is generated

Test Data: No test data needed

Test Steps:

- 1. Go to the time tracking reports page
- 2. Click on the "Export" button
- 3. Verify that the time tracking data is exported successfully

Postconditions:

- Time tracking data is exported

Expected Result: The system exports the time tracking data in a readable format (e.g., CSV, PDF).

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TT005

Test Case Title: Verify that the system provides reports on time spent per task or project

Test Case Description: This test case ensures that the system generates reports on time spent per task or

project, providing insights into productivity and resource allocation.

Test Suite: Time Tracking

Test Priority: High

Preconditions:
- User is logged in
- Task is created
- Time tracking data is generated
Test Data: No test data needed
Test Steps:
1. Go to the time tracking reports page
2. Select a task or project
3. Verify that the report shows the time spent on the selected task or project
Postconditions:
- Report is generated
Expected Result: The system provides a detailed report on time spent per task or project, including charts and
graphs.
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:

Here are the 5 detailed test cases for the Task Templates feature:

Test Case 1:

Test Case ID: TT001

Test Case Title: Verify that users can create and save task templates

Test Case Description: Ensure that users can create task templates with predefined structures, descriptions,

assignees, and due dates, and save them for future use.

Test Suite: Task Templates

Test Priority: High

Preconditions:

* User is logged in

* User has necessary permissions to create templates

Test Data: Template details (e.g., template name, description, assignee, due date)

Test Steps:

- 1. Access the task template creation page
- 2. Enter template details (e.g., name, description, assignee, due date)
- 3. Save the template

Postconditions:

* Template is created and saved successfully

Expected Result: The system allows users to create and save task templates with required details.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TT002

Test Case Title: Verify that templates can include predefined task structures, descriptions, assignees, and due

dates

Test Case Description: Ensure that task templates can include predefined task structures, descriptions,

assignees, and due dates, which can be applied to new tasks or projects.

Test Suite: Task Templates

Test Priority: High

Preconditions:

- * User is logged in
- * User has necessary permissions to create templates

Test Data: Template details (e.g., template name, description, assignee, due date, task structure)

Test Steps:

- 1. Access the task template creation page
- 2. Enter template details (e.g., name, description, assignee, due date, task structure)
- 3. Save the template
- 4. Apply the template to a new task or project

Postconditions:

* Template is applied successfully to the new task or project

Expected Result: The system allows templates to include predefined task structures, descriptions, assignees,

and due dates, which can be applied to new tasks or projects.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TT003

Test Case Title: Verify that users can apply templates to new tasks or projects

Test Case Description: Ensure that users can apply saved templates to new tasks or projects, streamlining task

creation and ensuring consistency.

Test Suite: Task Templates

Test Priority: High

Preconditions:

* User is logged in

* User has necessary permissions to create tasks or projects

* Template is created and saved

Test Data: New task or project details

Test Steps:

1. Access the task or project creation page

2. Select a saved template

3. Apply the template to the new task or project

Postconditions:

* Template is applied successfully to the new task or project

Expected Result: The system allows users to apply saved templates to new tasks or projects, streamlining task

creation and ensuring consistency. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 4:** Test Case ID: TT004 Test Case Title: Verify that templates can be edited or deleted as needed Test Case Description: Ensure that users can edit or delete saved task templates as needed, allowing for flexibility and maintainability. Test Suite: Task Templates Test Priority: Medium Preconditions: * User is logged in * User has necessary permissions to edit or delete templates * Template is created and saved Test Data: Template details (e.g., template name, description, assignee, due date) Test Steps: 1. Access the task template management page 2. Select a saved template 3. Edit or delete the template Postconditions: * Template is edited or deleted successfully

Expected Result: The system allows users to edit or delete saved task templates as needed, allowing for

flexibility and maintainability.

Type of Testing: Functional

Severity: Minor

Test Case Approach: Positive
Test Case 5:
Test Case ID: TT005
Test Case Title: Verify that the system displays a list of available templates for selection
Test Case Description: Ensure that the system provides a list of available task templates for users to select and
apply to new tasks or projects.
Test Suite: Task Templates
Test Priority: Medium
Preconditions:
* User is logged in
* User has necessary permissions to access task templates
Test Data: No test data needed
Test Steps:
1. Access the task template selection page
2. Verify the list of available templates
Postconditions:
* List of available templates is displayed
Expected Result: The system displays a list of available task templates for users to select and apply to new
tasks or projects.
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 -

Calendar Integration

Feature description -

Tasks can be synced with users' calendars, providing a unified view of task deadlines and commitments

alongside other scheduled events and appointments. Calendar integration helps users manage their time

effectively and avoid scheduling conflicts.

Userstory title -

As a user, I want to sync tasks with my calendar so that I can manage my time and avoid scheduling conflicts.

Acceptance criteria -

['Users can sync tasks with external calendars (e.g., Google Calendar, Outlook).', 'The system displays task

deadlines in the calendar view.', 'Users can set reminders for tasks via the calendar.', 'Calendar events update

automatically when tasks are changed.', 'Users can manage calendar sync settings within the app.']

Generated Test Cases:

Here are the 5 test cases for the Calendar Integration feature:

Test Case 1: Verify that users can sync tasks with external calendars

Test Case ID: TCT001

Test Case Title: Verify that users can sync tasks with external calendars

Test Case Description: Ensure that the system allows users to synchronize tasks with external calendars, such

as Google Calendar or Outlook.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User has a valid account with an external calendar service (e.g., Google Calendar)

- User has created a task with a deadline in the Task Management Tool

Test Data: Task details with deadline

Test Steps:

1. Log in to the Task Management Tool

2. Navigate to the Calendar Integration settings

3. Select the external calendar service to sync with

4. Authenticate with the external calendar service

5. Verify that the task is synced with the external calendar

Postconditions:

- Task is displayed in the external calendar

Expected Result: The system successfully syncs tasks with external calendars.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that system displays task deadlines in the calendar view

Test Case ID: TCT002

Test Case Title: Verify that system displays task deadlines in the calendar view

Test Case Description: Ensure that the system displays task deadlines in the calendar view, allowing users to

visualize their commitments.

Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User has synced tasks with an external calendar

- User has created a task with a deadline in the Task Management Tool

Test Data: Task details with deadline

Test Steps:

1. Log in to the Task Management Tool

2. Navigate to the calendar view

3. Verify that task deadlines are displayed in the calendar

4. Click on a task deadline to view task details

Postconditions:

- Task deadline is displayed in the calendar view

Expected Result: The system displays task deadlines in the calendar view.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can set reminders for tasks via the calendar

Test Case ID: TCT003

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 via the calendar, ensuring

they stay on track with their commitments.

Test Suite: Calendar Integration

Test Priority: Medium

Preconditions:

- User has synced tasks with an external calendar

- User has created a task with a deadline in the Task Management Tool Test Data: Task details with deadline Test Steps: 1. Log in to the Task Management Tool 2. Navigate to the calendar view 3. Click on a task deadline to view task details 4. Set a reminder for the task 5. Verify that the reminder is displayed in the external calendar Postconditions: - Reminder is set for the task Expected Result: The system allows users to set reminders for tasks via the calendar. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 4: Verify that calendar events update automatically when tasks are changed** Test Case ID: TCT004 Test Case Title: Verify that calendar events update automatically when tasks are changed Test Case Description: Ensure that the system updates calendar events automatically when tasks are changed, ensuring that users' calendars remain up-to-date. Test Suite: Calendar Integration

Test Priority: High

Preconditions:

- User has synced tasks with an external calendar
- User has created a task with a deadline in the Task Management Tool

Test Data: Task details with deadline

Test Steps:
1. Log in to the Task Management Tool
2. Navigate to the task list
3. Update the task deadline
4. Verify that the calendar event is updated in the external calendar
Postconditions:
- Calendar event is updated automatically
Expected Result: The system updates calendar events automatically when tasks are changed.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that users can manage calendar sync settings within the app
Test Case ID: TCT005
Test Case Title: Verify that users can manage calendar sync settings within the app
Test Case Description: Ensure that the system allows users to manage calendar sync settings within the app,
providing flexibility and control over calendar integration.
Test Suite: Calendar Integration
Test Priority: Medium
Preconditions:
- User has created an account in the Task Management Tool
Test Data: No test data needed
Test Steps:

3. Verify that calendar sync options are available (e.g., sync frequency, calendar selection)

1. Log in to the Task Management Tool

2. Navigate to the calendar settings

4. Update calendar sync settings

5. Verify that changes are saved successfully

Postconditions:

- Calendar sync settings are updated

Expected Result: The system allows users to manage calendar sync settings within the app.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 36

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Duplication

Feature description -

Users can duplicate tasks or task templates, saving time and effort when creating similar tasks for different projects or recurring activities. Task duplication allows users to reuse existing task structures and configurations, maintaining consistency and reducing manual effort.

Userstory title -

As a user, I want to duplicate tasks so that I can save time when creating similar tasks.

Acceptance criteria -

['Users can duplicate tasks or task templates.', 'The duplicated task retains all original details, including descriptions, assignees, and deadlines.', 'Users can modify the duplicated task if needed.', 'The system clearly

indicates that a task has been duplicated.', 'Duplicated tasks are saved and displayed in the task list.']

Generated Test Cases:

Here are the 5 detailed test cases for the Task Duplication feature:

Test Case 1: Verify that users can duplicate tasks

Test Case ID: TCTD1

Test Case Title: Verify that users can duplicate tasks

Test Case Description: Ensure that users can duplicate tasks, saving time and effort when creating similar

tasks for different projects or recurring activities.

Test Suite: Task Duplication

Test Priority: High

Preconditions:

- User is logged in

- At least one task exists in the task list

Test Data: Task details

Test Steps:

1. Select a task to duplicate

2. Click on the duplicate button

3. Confirm the duplication

4. Verify the duplicated task in the task list

Postconditions:

- The duplicated task is saved and displayed in the task list

Expected Result: The system allows users to duplicate tasks and saves the duplicated task in the task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that duplicated tasks retain original details

Test Case ID: TCTD2

Test Case Title: Verify that duplicated tasks retain 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: High

Preconditions:

- User is logged in

- At least one task exists in the task list

Test Data: Task details

Test Steps:

1. Select a task to duplicate

2. Click on the duplicate button

3. Verify that the duplicated task has the same description, assignees, and deadlines as the original task

Postconditions:

- The duplicated task has the same details as the original task

Expected Result: The duplicated task retains all original details, including descriptions, assignees, and

deadlines.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can modify duplicated tasks

Test Case ID: TCTD3

Test Case Title: Verify that users can modify duplicated tasks

Test Case Description: Ensure that users can modify the duplicated task if needed, allowing flexibility and

customization.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

- User is logged in

- At least one task exists in the task list

Test Data: Task details

Test Steps:

1. Select a task to duplicate

2. Click on the duplicate button

3. Modify the duplicated task (e.g., change description, assignee, or deadline)

4. Verify that the changes are saved

Postconditions:

- The duplicated task is updated with the modified details

Expected Result: The system allows users to modify the duplicated task and saves the changes.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that duplicated tasks are clearly indicated

Test Case ID: TCTD4

Test Case Title: Verify that duplicated tasks are clearly indicated

Test Case Description: Ensure that the system clearly indicates that a task has been duplicated, reducing confusion and ensuring transparency.

Test Suite: Task Duplication

Test Priority: Medium

Preconditions:

- User is logged in

- At least one task exists in the task list

Test Data: Task details

Test Steps:

1. Select a task to duplicate

2. Click on the duplicate button

3. Verify that the duplicated task is marked as "Duplicated" or has a distinct icon/label

Postconditions:

- The duplicated task is clearly indicated as such

Expected Result: The system clearly indicates that a task has been duplicated.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that duplicated tasks are saved and displayed correctly

Test Case ID: TCTD5

Test Case Title: Verify that duplicated tasks are saved and displayed correctly

Test Case Description: Ensure that duplicated tasks are saved and displayed in the task list, allowing users to access and manage them easily.

Test Suite: Task Dupilcation
Test Priority: High
Preconditions:
- User is logged in
- At least one task exists in the task list
Test Data: Task details
Test Steps:
1. Select a task to duplicate
2. Click on the duplicate button
3. Verify that the duplicated task is saved and displayed in the task list
4. Verify that the duplicated task is accessible and editable
Postconditions:
- The duplicated task is saved and displayed in the task list
Expected Result: The system saves and displays the duplicated task in the task list, allowing users to access
and manage it.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Data - 37
Project name -
Task Management Tool
2 don 1.2mm50v.it 1 001

Project description -

Build a task management tool that allows users to create, assign, prioritize, and track tasks and projects. Users should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task History

Feature description -

Users can view the history of changes and updates made to tasks, tracking modifications, comments, and other

relevant activity. Task history provides a comprehensive audit trail of task-related actions, facilitating

accountability, and transparency in task management.

Userstory title -

As a user, I want to view the history of changes to tasks so that I can track updates and modifications.

Acceptance criteria -

['Users can view a detailed history of changes to tasks.', 'The history includes modifications, comments, and

other relevant activity.', 'Users can filter the history by type of change or date.', 'The system provides

timestamps for all changes.', 'Task history is accessible from the task details page.']

Generated Test Cases:

Here are the 5 detailed test cases for the Task History feature:

Test Case 1:

Test Case ID: TH001

Test Case Title: Verify that users can view a detailed history of changes to tasks

Test Case Description: Ensure that the system allows users to view a comprehensive history of changes made

to tasks, including modifications, comments, and other relevant activity.

Test Suite: Task History

Test Priority: High

Preconditions:

* User is logged in

* Task is created with modifications and comments

Test Data: No test data needed

Test Steps:

1. Go to the task details page

2. Click on the task history tab

3. Verify that the task history is displayed with all changes, modifications, and comments

Postconditions:

* Task history is updated

Expected Result: The system displays a detailed history of changes to tasks, including modifications,

comments, and other relevant activity.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: TH002

Test Case Title: Verify that task history includes timestamps for all changes

Test Case Description: Ensure that the system provides timestamps for all changes made to tasks, facilitating

accountability and transparency in task management.

Test Suite: Task History

Test Priority: High

Preconditions:

* User is logged in

* Task is created with modifications and comments

Test Data: No test data needed

Test Steps:

- 1. Go to the task details page
- 2. Click on the task history tab
- 3. Verify that timestamps are displayed for each change, modification, and comment

Postconditions:

* Task history is updated

Expected Result: The system provides timestamps for all changes made to tasks, including modifications, comments, and other relevant activity.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: TH003

Test Case Title: Verify that users can filter task history by type of change or date

Test Case Description: Ensure that the system allows users to filter the task history by type of change or date, making it easier to track specific updates and modifications.

Test Suite: Task History

Test Priority: Medium

Preconditions:

- * User is logged in
- * Task is created with modifications and comments

Test Data: No test data needed

Test Steps:

- 1. Go to the task details page
- 2. Click on the task history tab
- 3. Select a filter option (e.g., by type of change or date)

4. Verify that the filtered task history is displayed

Postconditions:

* Task history is updated

Expected Result: The system allows users to filter the task history by type of change or date, displaying only

the relevant updates and modifications.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: TH004

Test Case Title: Verify that task history is accessible from the task details page

Test Case Description: Ensure that the system provides easy access to the task history from the task details

page, allowing users to quickly view updates and modifications.

Test Suite: Task History

Test Priority: High

Preconditions:

* User is logged in

* Task is created with modifications and comments

Test Data: No test data needed

Test Steps:

1. Go to the task details page

2. Verify that the task history tab is visible

3. Click on the task history tab

4. Verify that the task history is displayed

Postconditions:

* Task history is updated

Expected Result: The system provides easy access to the task history from the task details page, allowing users

to quickly view updates and modifications.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: TH005

Test Case Title: Verify that task history displays all relevant activity

Test Case Description: Ensure that the system displays all relevant activity in the task history, including

comments, modifications, and other updates.

Test Suite: Task History

Test Priority: High

Preconditions:

* User is logged in

* Task is created with modifications and comments

Test Data: No test data needed

Test Steps:

1. Go to the task details page

2. Click on the task history tab

3. Verify that all relevant activity is displayed, including comments, modifications, and other updates

Postconditions:

* Task history is updated

Expected Result: The system displays all relevant activity in the task history, including comments,

modifications, and other updates.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Each of these test cases thoroughly covers one or more of the acceptance criteria, ensuring comprehensive coverage of the Task History feature.

Data - 38

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Notifications

Feature description -

Users receive notifications for task updates, comments, mentions, and other relevant activity, keeping them informed and engaged with ongoing task progress. Notifications can be delivered via email, mobile push notifications, or in-app alerts, ensuring timely communication and collaboration.

Userstory title -

As a user, I want to receive notifications for task updates so that I can stay informed about ongoing task progress.

Acceptance criteria -

['Users receive notifications for task updates, comments, mentions, and other relevant activities.',

'Notifications can be delivered via email, mobile push notifications, or in-app alerts.', 'Users can customize

notification settings based on preferences.', 'Notifications provide detailed information about the activity.',

'The system logs notification history for future reference.']

Generated Test Cases:

Here are the 5 test cases for the Task Notifications feature:

Test Case 1: Verify that users receive notifications for task updates

Test Case ID: TN001

Test Case Title: Verify that users receive notifications for task updates

Test Case Description: Ensure that users receive notifications when task updates occur, keeping them

informed about ongoing task progress.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

* User is logged in

* Task is created and assigned to the user

* Notification settings are set to receive updates

Test Data: No test data needed

Test Steps:

1. Update a task by adding a comment

2. Verify that a notification is sent to the assigned user

3. Verify that the notification contains detailed information about the task update

Postconditions:
* Notification is sent to the user
* Notification contains task update details
Expected Result: The system sends a notification to the assigned user with detailed information about the task
update.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that notifications can be delivered via multiple channels
Test Case ID: TN002
Test Case Title: Verify that notifications can be delivered via multiple channels
Test Case Description: Ensure that notifications can be delivered via email, mobile push notifications, or
in-app alerts, catering to different user preferences.
Test Suite: Task Notifications
Test Priority: Medium
Preconditions:
* User is logged in
* Notification settings are set to receive updates via multiple channels
Test Data: No test data needed
Test Steps:

- 1. Update a task by adding a comment
- 2. Verify that a notification is sent to the user via email
- 3. Verify that a notification is sent to the user via mobile push notification
- 4. Verify that an in-app alert is displayed

Postconditions:

- * Notifications are sent via multiple channels
- * User receives notifications via all selected channels

Expected Result: The system sends notifications via multiple channels, allowing users to receive updates according to their preferences.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can customize notification settings

Test Case ID: TN003

Test Case Title: Verify that users can customize notification settings

Test Case Description: Ensure that users can customize their notification settings based on their preferences,

controlling the types of notifications they receive.

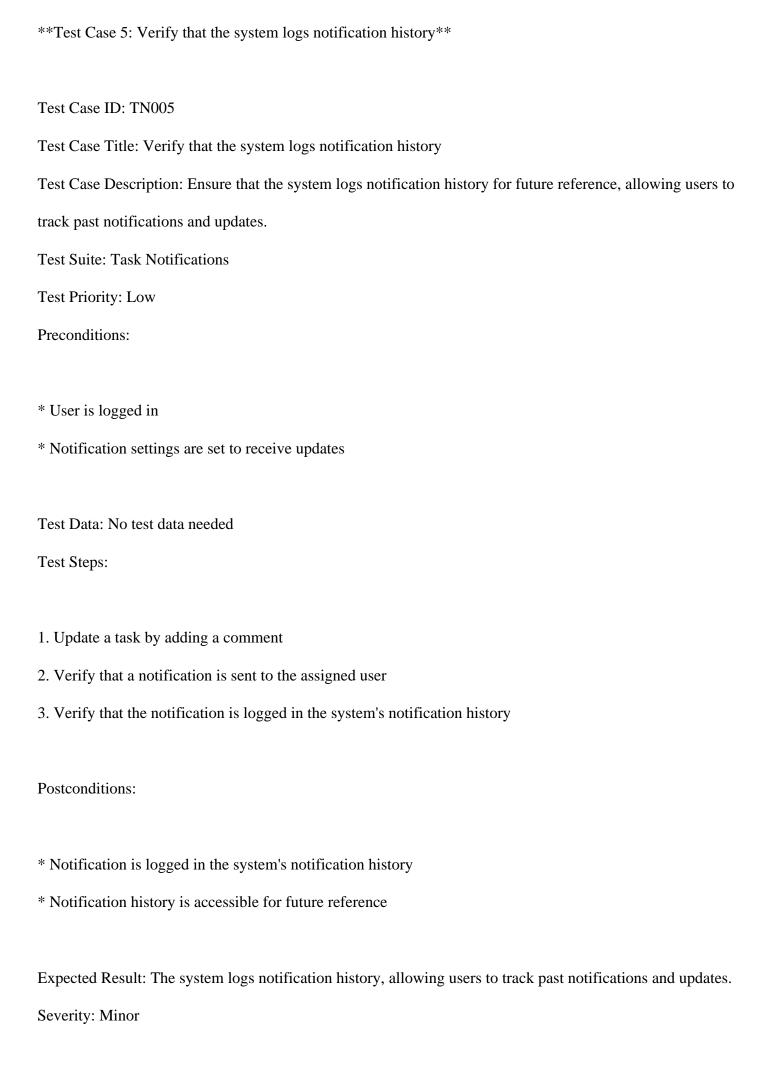
Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

* User is logged in
* Notification settings are accessible
Test Data: No test data needed
Test Steps:
1. Access notification settings
2. Customize notification settings to receive only specific types of updates
3. Verify that notifications are sent according to customized settings
Postconditions:
* Notification settings are updated
* User receives notifications according to customized settings
Expected Result: The system allows users to customize their notification settings, filtering out unwanted
updates.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that notifications provide detailed information
Test Case ID: TN004
Test Case Title: Verify that notifications provide detailed information
Test Case Description: Ensure that notifications provide detailed information about the activity, keeping users
informed about ongoing task progress.

Test Suite: Task Notifications
Test Priority: High
Preconditions:
* User is logged in
* Task is created and assigned to the user
* Notification settings are set to receive updates
Test Data: No test data needed
Test Steps:
1. Update a task by adding a comment
2. Verify that a notification is sent to the assigned user
3. Verify that the notification contains detailed information about the task update (e.g., task name, update
description, updated fields)
Postconditions:
* Notification is sent to the user
* Notification contains detailed information about the task update
Expected Result: The system sends notifications with detailed information about the task update, keeping
users informed.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive



Type of Testing: Functional

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:

Here are the 5 test cases for the Task Archiving feature:

Test Case 1: Verify that users can archive completed tasks Test Case ID: TA-001 Test Case Title: Verify that users can archive completed tasks Test Case Description: Ensure that users can archive completed tasks, moving them to a separate section for easy access. This test case verifies that the feature allows users to declutter their task list and focus on active tasks. Test Suite: Task Archiving Test Priority: High Preconditions: * User is logged in * Completed task exists in the task list Test Data: No test data needed Test Steps: 1. Log in to the task management tool 2. Select a completed task 3. Click the "Archive" button 4. Confirm the archiving of the task Postconditions: * Task is moved to the archived tasks section * Task is no longer visible in the active task list

Expected Result: The system allows users to archive completed tasks, moving them to a separate section for easy access. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2: Verify that archived tasks are moved to a separate section** Test Case ID: TA-002 Test Case Title: Verify that archived tasks are moved to a separate section Test Case Description: Ensure that archived tasks are moved to a separate section, making it easy for users to access and manage their archived tasks. This test case verifies that the feature provides a convenient way to access historical data. Test Suite: Task Archiving Test Priority: High Preconditions: * User is logged in * Task is archived Test Data: No test data needed Test Steps:

- 1. Log in to the task management tool
- 2. Go to the archived tasks section
- 3. Verify that the archived task is present in the section

Postconditions:
* Archived task is visible in the archived tasks section
Expected Result: The system moves archived tasks to a separate section, making it easy for users to access and manage their archived tasks.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that users can search and view archived tasks
Test Case ID: TA-003
Test Case Title: Verify that users can search and view archived tasks
Test Case Description: Ensure that users can search and view archived tasks, allowing them to access
important historical data. This test case verifies that the feature provides a convenient way to retrieve archived
tasks.
Test Suite: Task Archiving
Test Priority: Medium
Preconditions:
* User is logged in
* Task is archived
Test Data: Archived task details (e.g., task name, description)
Test Steps:

1. Log in to the task management tool

2. Search for an archived task using the search bar

3. Verify that the archived task is displayed in the search results

4. Click on the archived task to view its details

Postconditions:

* Archived task details are displayed

Expected Result: The system allows users to search and view archived tasks, providing access to important

historical data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that the system provides an option to restore archived tasks

Test Case ID: TA-004

Test Case Title: Verify that the system provides an option to restore archived tasks

Test Case Description: Ensure that the system provides an option to restore archived tasks to the active list,

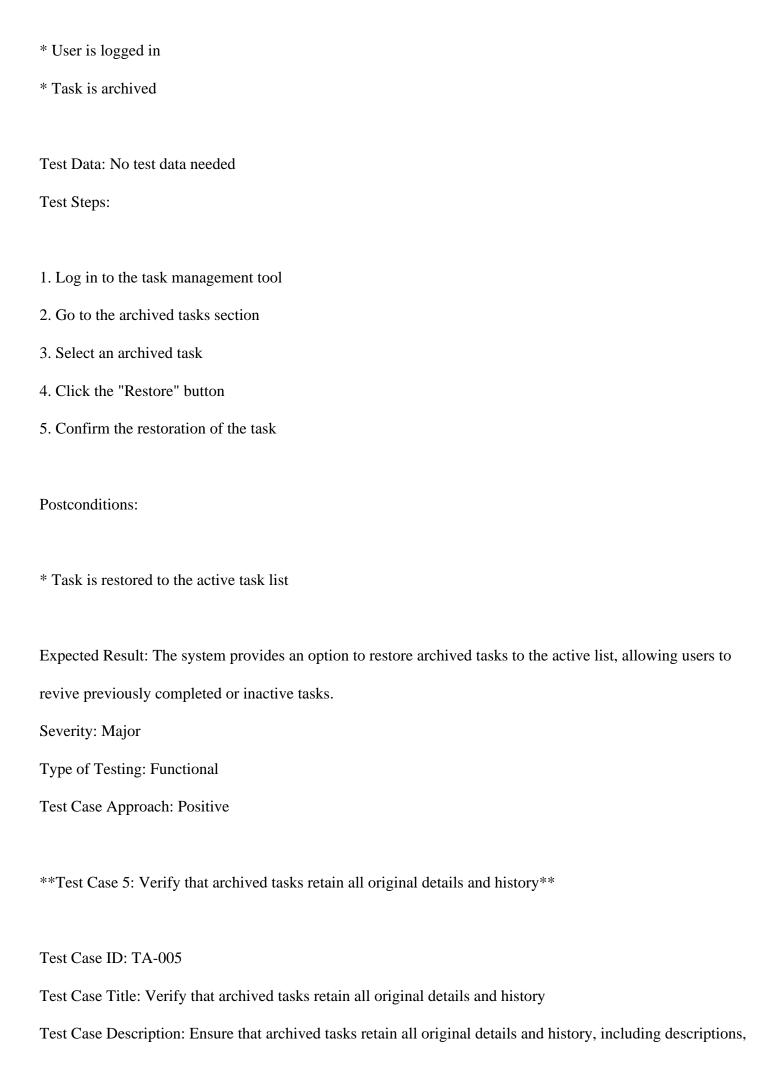
allowing users to revive previously completed or inactive tasks. This test case verifies that the feature allows

users to easily restore tasks to their original state.

Test Suite: Task Archiving

Test Priority: Medium

Preconditions:



attachments, and comments. This test case verifies that the feature preserves task data when archiving tasks.
Test Suite: Task Archiving
Test Priority: Medium
Preconditions:
* User is logged in
* Task is archived
Test Data: Archived task details (e.g., task name, description, attachments, comments)
Test Steps:
1. Log in to the task management tool
2. Go to the archived tasks section
3. Select an archived task
4. Verify that the task retains all original details and history
Postconditions:
* Archived task retains all original details and history
Expected Result: The system retains all original details and history of archived tasks, preserving task data.
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 -

Guest Access

Feature description -

Users can grant limited access to external collaborators or clients, allowing them to view and collaborate on

specific tasks or projects without full platform access. Guest access features provide flexibility in sharing task

information and involving external stakeholders in project discussions and updates.

Userstory title -

As a user, I want to grant guest access to external collaborators so that they can view and collaborate on

specific tasks.

Acceptance criteria -

['Users can invite external collaborators as guests with limited access.', 'Guests can view and collaborate on

specific tasks or projects.', 'The system provides clear distinctions between guest and full access roles.', 'Users

can manage and revoke guest access at any time.', 'Guests cannot access sensitive or restricted areas of the

platform.']

Generated Test Cases:

Here are the 5 detailed test cases for the Guest Access feature:

Test Case 1: Verify that users can invite external collaborators as guests with limited access

Test Case ID: GT-001

Test Case Description: Ensure that users can invite external collaborators as guests with limited access to view
and collaborate on specific tasks or projects.
Test Suite: Guest Access
Test Priority: High
Preconditions:
* User is logged in
* User has a project or task to invite a guest to
Test Data: No test data needed
Test Steps:
1. Go to the project or task page
2. Click on the "Invite" button
3. Enter the guest's email address
4. Set the guest's role as "Guest"
5. Click "Invite"
Postconditions:
* Guest receives an invitation email
* Guest can log in and access the invited project or task
Expected Result: The system sends an invitation email to the guest, and the guest can log in and access the
invited project or task with limited access.
Severity: Major
Type of Testing: Functional

Test Case Approach: Positive
Test Case 2: Verify that guests can view and collaborate on specific tasks or projects
Test Case ID: GT-002
Test Case Description: Ensure that guests can view and collaborate on specific tasks or projects they are
invited to.
Test Suite: Guest Access
Test Priority: High
Preconditions:
* Guest has accepted the invitation
* Guest has logged in
Test Data: Task or project details
Test Steps:
1. Log in as a guest
2. Go to the invited project or task page
3. Verify that the guest can view task or project details
4. Verify that the guest can add comments or attachments
5. Verify that the guest can update task or project status
Postconditions:
* Guest can view and collaborate on the task or project

Expected Result: The system allows the guest to view and collaborate on the invited project or task, with
limited access.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that the system provides clear distinctions between guest and full access roles
Test Case ID: GT-003
Test Case Description: Ensure that the system clearly distinguishes between guest and full access roles, with
clear labels and restrictions.
Test Suite: Guest Access
Test Priority: Medium
Preconditions:
* User is logged in
* Guest is logged in
Test Data: No test data needed
Test Steps:
Test Steps: 1. Log in as a user
1. Log in as a user
 Log in as a user Verify that the user has full access to all features

Postconditions:
* Clear distinctions between guest and full access roles are visible
Expected Result: The system clearly distinguishes between guest and full access roles, with clear labels and
restrictions.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that users can manage and revoke guest access at any time
Test Case ID: GT-004
Test Case Description: Ensure that users can manage and revoke guest access at any time, removing access to
invited projects or tasks.
Test Suite: Guest Access
Test Priority: High
Preconditions:
* User has invited a guest
* Guest has accepted the invitation
Test Data: No test data needed
Test Steps:
1. Log in as a user
2. Go to the project or task page

3. Click on the "Manage Guests" button
4. Revoke the guest's access
5. Verify that the guest no longer has access to the project or task
Postconditions:
* Guest access is revoked
* Guest can no longer access the project or task
Expected Result: The system allows users to manage and revoke guest access at any time, removing access to
invited projects or tasks.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that guests cannot access sensitive or restricted areas of the platform
Test Case ID: GT-005
Test Case Description: Ensure that guests cannot access sensitive or restricted areas of the platform, such as
account settings or billing information.
Test Suite: Guest Access
Test Priority: High
Preconditions:
* Guest has accepted the invitation
* Guest has logged in

Test Data: No test data needed
Test Steps:
1. Log in as a guest
2. Attempt to access account settings
3. Attempt to access billing information
4. Verify that access is denied
5. Verify that an error message is displayed
Postconditions:
* Guest access is denied
Expected Result: The system restricts guest access to sensitive or restricted areas of the platform, with an error
message displayed.
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:

Here are the 5 detailed test cases for the Task Import/Export feature:

Test Case 1: Verify that users can import tasks from external sources in various formats

Test Case ID: TI001

Test Case Title: Verify that users can import tasks from external sources in various formats

Test Case Description: Ensure that the system allows users to import tasks from external sources in different

file formats such as CSV, Excel, etc.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:
* User is logged in
* External file in CSV or Excel format is available
Test Data: Sample CSV/Excel file with task data
Test Steps:
1. Go to the task import page
2. Select the file format (CSV or Excel)
3. Upload the sample file
4. Map the fields correctly
5. Preview the imported data
Test Postconditions:
* Imported tasks are displayed in the task list
Expected Result: The system successfully imports tasks from the external file in the selected format.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that users can export tasks to other task management tools or file formats
Test Case ID: TE002
Test Case Title: Verify that users can export tasks to other task management tools or file formats

Test Case Description: Ensure that the system allows users to export tasks to other task management tools or

file formats such as CSV, Excel, etc.

Test Suite: Task Import/Export

Test Priority: High

Preconditions:

* User is logged in

* Tasks are available in the task list Test Data: No test data needed Test Steps: 1. Go to the task export page 2. Select the export format (CSV or Excel) 3. Choose the tasks to export 4. Export the tasks 5. Verify the exported file Test Postconditions: * Exported file is generated Expected Result: The system successfully exports tasks to the selected file format. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that the system provides clear mapping of fields during import/export** Test Case ID: TM003 Test Case Title: Verify that the system provides clear mapping of fields during import/export Test Case Description: Ensure that the system provides a clear and intuitive mapping of fields during the import and export process. Test Suite: Task Import/Export Test Priority: Medium Preconditions: * User is logged in

* External file in CSV or Excel format is available

Test Data: Sample CSV/Excel file with task data

Test Steps:

- 1. Go to the task import page
- 2. Select the file format (CSV or Excel)
- 3. Upload the sample file
- 4. Verify the field mapping options
- 5. Preview the imported data

Test Postconditions:

* Field mapping options are displayed correctly

Expected Result: The system provides clear and intuitive field mapping options during the import and export process.

Severity: Minor

Type of Testing: Usability

Test Case Approach: Positive

Test Case 4: Verify that users can preview imported data before finalizing the process

Test Case ID: TP004

Test Case Title: Verify that users can preview imported data before finalizing the process

Test Case Description: Ensure that the system allows users to preview the imported data before finalizing the

import process.

Test Suite: Task Import/Export

Test Priority: Medium

Preconditions:

* User is logged in

* External file in CSV or Excel format is available

Test Data: Sample CSV/Excel file with task data

Test Steps:

1. Go to the task import page 2. Select the file format (CSV or Excel) 3. Upload the sample file 4. Preview the imported data 5. Verify the data is correct **Test Postconditions:** * Preview of imported data is displayed Expected Result: The system allows users to preview the imported data before finalizing the import process. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 5: Verify that the system logs import/export activities for future reference** Test Case ID: TL005 Test Case Title: Verify that the system logs import/export activities for future reference Test Case Description: Ensure that the system logs all import and export activities for future reference and auditing purposes. Test Suite: Task Import/Export Test Priority: Low Preconditions: * User is logged in * Tasks are available in the task list Test Data: No test data needed Test Steps: 1. Go to the task import/export page

2. Perform an import or export operation

3. Verify the activity log

Test Postconditions:

* Import/export activity is logged

Expected Result: The system logs all import and export activities for future reference and auditing purposes.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 42

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Prioritization

Feature description -

Users can set and adjust task priorities, helping to prioritize work and manage deadlines effectively. Task prioritization features can include visual indicators, sorting options, and priority-based notifications, providing a clear view of the most critical tasks.

Userstory title -

As a user, I want to set and adjust task priorities so that I can manage my work and deadlines effectively.

Acceptance criteria -

['Users can set and adjust task priorities.', 'Tasks can be sorted and filtered based on priority levels.', 'The

system provides visual indicators of task priority.', 'Users receive notifications for high-priority tasks.',

'Priority changes are logged and tracked.']

Generated Test Cases:

Here are the 5 test cases for the Task Prioritization feature of the Task Management Tool:

Test Case 1: Verify that users can set task priorities

Test Case ID: tp001

Test Case Title: Verify that users can set task priorities

Test Case Description: This test case ensures that users can set task priorities successfully, enabling them to

manage their work and deadlines effectively.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in
- Task is created

Test Data: No test data needed

Test Steps:

- 1. Go to the task details page
- 2. Click on the priority dropdown menu
- 3. Select a priority level (e.g., High, Medium, Low)
- 4. Save changes

Postconditions:

- Task priority is updated

Expected Result: The system allows users to set task priorities, and the priority level is displayed correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that tasks can be sorted and filtered by priority levels

Test Case ID: tp002

Test Case Title: Verify that tasks can be sorted and filtered by priority levels

Test Case Description: This test case ensures that tasks can be sorted and filtered based on priority levels,

enabling users to focus on high-priority tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- Multiple tasks with different priority levels are created

Test Data: No test data needed

Test Steps:

- 1. Go to the task list page
- 2. Click on the sort/filter button
- 3. Select a priority level (e.g., High, Medium, Low)
- 4. Verify the sorted and filtered task list

Postconditions:

- Task list is sorted and filtered by priority level

Expected Result: The system allows users to sort and filter tasks based on priority levels, and the task list is

updated correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that the system provides visual indicators of task priority

Test Case ID: tp003

Test Case Title: Verify that the system provides visual indicators of task priority

Test Case Description: This test case ensures that the system provides visual indicators of task priority,

enabling users to quickly identify high-priority tasks.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- Tasks with different priority levels are created

Test Data: No test data needed

Test Steps:

1. Go to the task list page

2. Verify the visual indicators (e.g., icons, colors, badges) for each task

3. Check that the visual indicators match the priority level

Postconditions:

- Visual indicators are displayed correctly

Expected Result: The system provides visual indicators of task priority, and they are displayed correctly for

each task.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that users receive notifications for high-priority tasks

Test Case ID: tp004

Test Case Title: Verify that users receive notifications for high-priority tasks

Test Case Description: This test case ensures that users receive notifications for high-priority tasks, enabling

them to stay on top of critical tasks.

Test Suite: Task Prioritization

Test Priority: High

Preconditions:

- User is logged in

- High-priority task is created with a due date

Test Data: No test data needed

Test Steps:

1. Set up notification preferences

2. Wait for the task due date to approach

3. Verify that a notification is received

Postconditions:

- Notification is received for the high-priority task

Expected Result: The system sends notifications for high-priority tasks, and users receive them correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that priority changes are logged and tracked

Test Case ID: tp005

Test Case Title: Verify that priority changes are logged and tracked

Test Case Description: This test case ensures that priority changes are logged and tracked, enabling users to

maintain a record of changes and track task history.

Test Suite: Task Prioritization

Test Priority: Medium

Preconditions:

- Task is created with a priority level

Test Data: No test data needed

Test Steps:

1. Change the task priority level

2. Verify the change in the task history

3. Check that the priority change is logged correctly

Postconditions:

- Priority change is logged and tracked

Expected Result: The system logs and tracks priority changes, and the task history is updated correctly.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 43

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks with team members, including assigning tasks to others, sharing updates, and

communicating through comments or messages. Collaboration features aim to enhance teamwork and

coordination on tasks and projects.

Userstory title -

As a user, I want to collaborate on tasks with my team so that we can work together more effectively and stay

updated on task progress.

Acceptance criteria -

['Users can assign tasks to team members.', 'Users can share updates and communicate through comments or

messages.', 'Task updates and comments are visible to assigned team members.', 'Users receive notifications

for task updates and comments.', 'The system provides a clear view of task collaboration history.']

Generated Test Cases:

Here are the 5 detailed test cases for the Task Collaboration feature:

Test Case 1:

Test Case ID: TC001

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

Test Case Description: Ensure that the system allows users to assign tasks to team members, enabling

collaboration and task delegation.

Test Suite: Task Collaboration

Test Priority: High

Preconditions:

* User is logged in

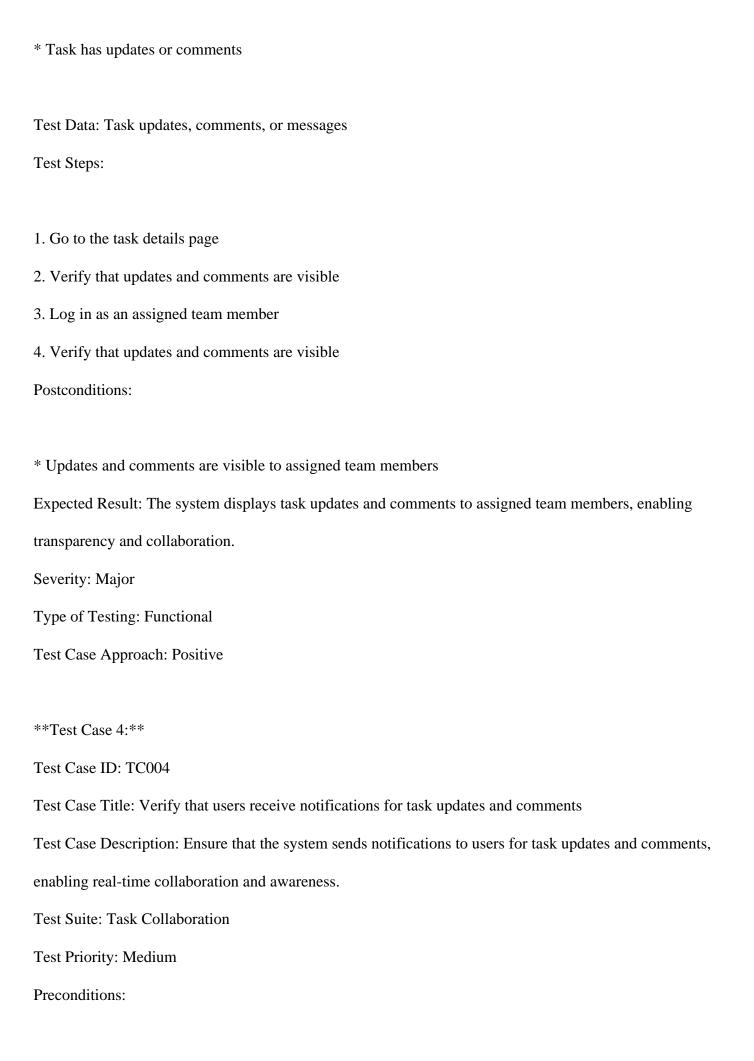
* Task is created

* Team members are available

Test Data: Task details, team member information

Test Steps:
1. Go to the task details page
2. Click on the "Assign to" button
3. Select a team member from the list
4. Click "Assign"
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 status.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2:
Test Case ID: TC002
Test Case Title: Verify that users can share updates and communicate through comments or messages
Test Case Description: Ensure that the system allows users to share updates and communicate with team
members through comments or messages, enabling real-time collaboration.
Test Suite: Task Collaboration
Test Priority: High
Preconditions:
* User is logged in
* Task is assigned to a team member
* Task has comments or messages

Test Data: Task updates, comments, or messages
Test Steps:
1. Go to the task details page
2. Click on the "Comment" or "Message" button
3. Enter a comment or message
4. Click "Post"
Postconditions:
* Comment or message is visible to assigned team members
Expected Result: The system allows users to share updates and communicate with team members through
comments or messages, which are visible to assigned team members.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: TC003
Test Case Title: Verify that task updates and comments are visible to assigned team members
Test Case Description: Ensure that the system displays task updates and comments to assigned team members
enabling transparency and collaboration.
Test Suite: Task Collaboration
Test Priority: High
Preconditions:
* User is logged in
* Task is assigned to a team member



* User is logged in * Task is assigned to a team member * Notifications are enabled Test Data: Task updates, comments, or messages Test Steps: 1. Go to the task details page 2. Update the task or add a comment 3. Verify that a notification is sent to the assigned team member Postconditions: * Notification is sent to the assigned team member Expected Result: The system sends notifications to users for task updates and comments, enabling real-time collaboration and awareness. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 5:** Test Case ID: TC005 Test Case Title: Verify that the system provides a clear view of task collaboration history Test Case Description: Ensure that the system displays a clear view of task collaboration history, enabling users to track task progress and updates. Test Suite: Task Collaboration Test Priority: Medium Preconditions:

* User is logged in
* Task has collaboration history
Test Data: Task collaboration history
Test Steps:
1. Go to the task details page
2. Verify that the collaboration history is displayed
3. Verify that the history includes updates, comments, and assignments
Postconditions:
* Collaboration history is displayed
Expected Result: The system provides a clear view of task collaboration history, enabling users to track task
progress and updates.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Data - 44
Project name -

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

Task Management Tool

Project description -

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:

Here are the 5 detailed test cases for the Task Tracking feature:

Test Case 1: Verify that users can view task statuses

Test Case ID: TT001

Test Case Title: Verify that users can view task statuses

Test Case Description: Ensure that the system displays the current status of tasks, allowing users to monitor

their progress.

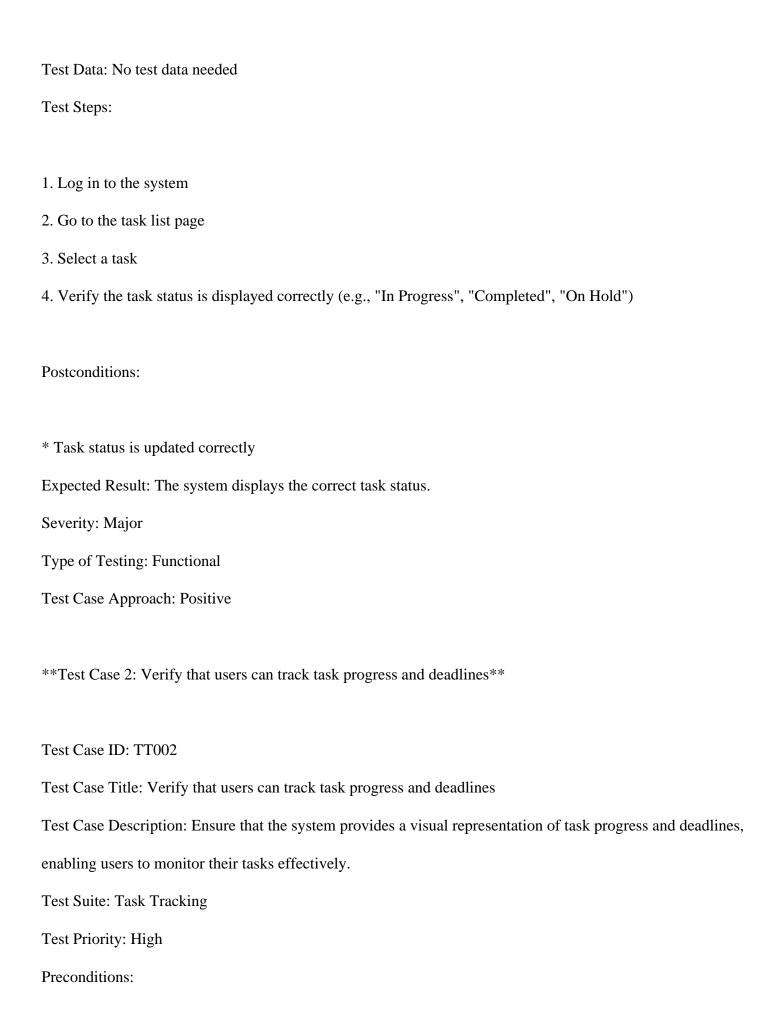
Test Suite: Task Tracking

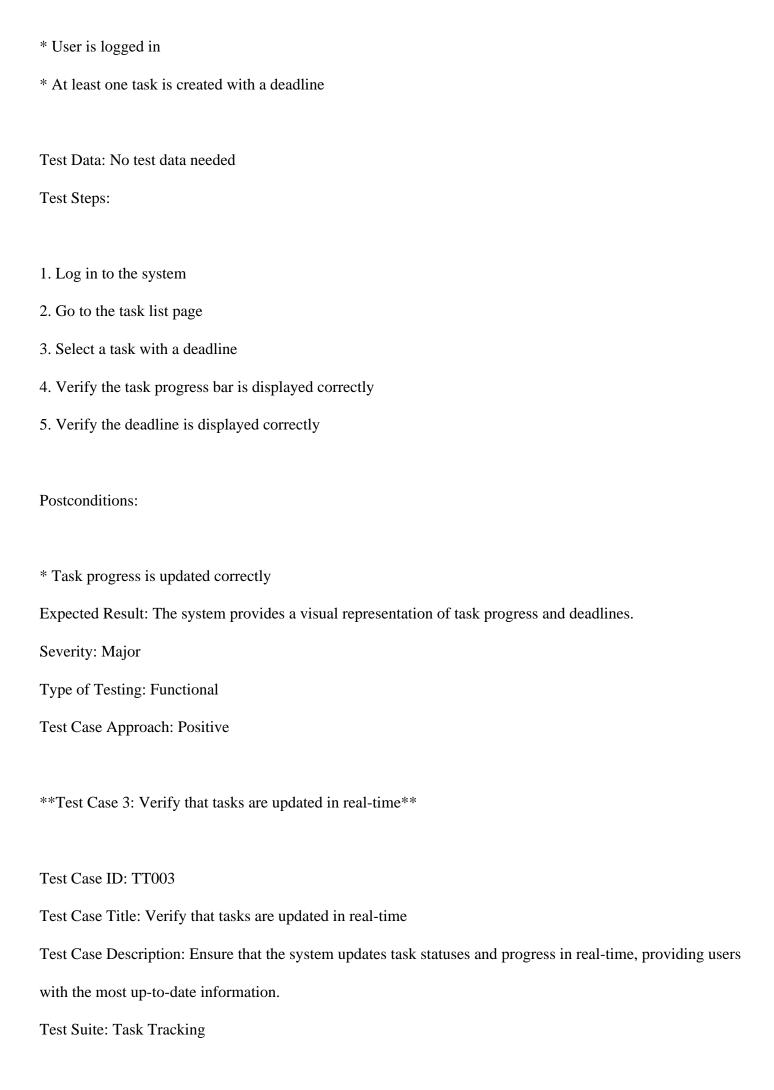
Test Priority: High

Preconditions:

* User is logged in

* At least one task is created





Test Priority: High
Preconditions:
* Two users are logged in
* One user creates a task and assigns it to the other user
Test Data: No test data needed
Test Steps:
1. User 1 creates a task and assigns it to User 2
2. User 2 logs in and updates the task status
3. Verify the task status is updated in real-time for User 1
4. Verify the task progress is updated correctly for User 1
Postconditions:
* Task status is updated correctly
Expected Result: The system updates tasks in real-time, displaying the most up-to-date information.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that system provides visual indicators for task progress
Test Case ID: TT004
Test Case Title: Verify that system provides visual indicators for task progress

Test Case Description: Ensure that the system uses visual indicators (e.g., colors, icons) to represent task

progress, making it easy for users to monitor their tasks.
Test Suite: Task Tracking
Test Priority: Medium
Preconditions:
* User is logged in
* At least one task is created with progress
Test Data: No test data needed
Test Steps:
1. Log in to the system
2. Go to the task list page
3. Select a task with progress
4. Verify visual indicators (e.g., colors, icons) are displayed correctly
Postconditions:
* Visual indicators are displayed correctly
Expected Result: The system provides visual indicators for task progress, making it easy for users to monitor
their tasks.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that users can track multiple tasks with different deadlines

Test Case ID: TT005 Test Case Title: Verify that users can track multiple tasks with different deadlines Test Case Description: Ensure that the system allows users to track multiple tasks with different deadlines, providing a comprehensive view of their task list. Test Suite: Task Tracking Test Priority: High Preconditions: * User is logged in * At least three tasks are created with different deadlines Test Data: No test data needed Test Steps: 1. Log in to the system 2. Go to the task list page 3. Verify all tasks are displayed correctly with their respective deadlines 4. Verify task progress and deadlines are updated correctly for each task Postconditions: * Task list is updated correctly Expected Result: The system allows users to track multiple tasks with different deadlines, providing a comprehensive view of their task list.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 45

Project name -

Task Management Tool

Project description -

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

should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Notifications

Feature description -

Users receive notifications for task-related events, such as new tasks, updates, comments, and upcoming

deadlines. The notifications aim to keep users informed and engaged with their tasks.

Userstory title -

As a user, I want to receive notifications for task-related events so that I can stay informed and manage my

tasks effectively.

Acceptance criteria -

['Users receive notifications for new tasks.', 'Users receive notifications for task updates and comments.',

'Users receive reminders for upcoming deadlines.', 'Notification settings can be customized by users.', "]

Generated Test Cases:

Here are the 5 detailed test cases for the Task Management Tool's Task Notifications feature:

Test Case 1: Verify that users receive notifications for new tasks

Test Case ID: NT1

Test Case Title: Verify that users receive notifications for new tasks

Test Case Description: Ensure that the system sends notifications to users when a new task is created and

assigned to them.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in

- New task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Create a new task and assign it to the user

2. Verify that the user receives a notification for the new task

3. Check the notification content to ensure it includes task details

Postconditions:

- Notification is sent to the user

Expected Result: The system sends a notification to the user for the new task, including task details.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that users receive notifications for task updates and comments

Test Case ID: NT2

Test Case Title: Verify that users receive notifications for task updates and comments

Test Case Description: Ensure that the system sends notifications to users when a task is updated or

commented on.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in

- Task is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Update a task and add a comment

2. Verify that the user receives a notification for the task update and comment

3. Check the notification content to ensure it includes task update and comment details

Postconditions:

- Notification is sent to the user

Expected Result: The system sends a notification to the user for the task update and comment, including task update and comment details.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users receive reminders for upcoming deadlines

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.

Test Suite: Task Notifications

Test Priority: High

Preconditions:

- User is logged in

- Task with upcoming deadline is created and assigned to the user

Test Data: No test data needed

Test Steps:

1. Create a task with an upcoming deadline

2. Verify that the user receives a reminder notification for the task deadline

3. Check the notification content to ensure it includes task deadline details

Postconditions:

- Reminder notification is sent to the user

Expected Result: The system sends a reminder notification to the user for the task deadline, including task deadline details.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that notification settings can be customized by users

Test Case ID: NT4

Test Case Title: Verify that notification settings can be customized by users

Test Case Description: Ensure that users can customize notification settings to control the types of

notifications they receive.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in

- Notification settings page is accessible

Test Data: No test data needed

Test Steps:

1. Access the notification settings page

- 2. Customize notification settings to receive specific types of notifications
- 3. Verify that the system respects the customized notification settings

Postconditions:

- Notification settings are updated

Expected Result: The system allows users to customize notification settings and respects the updated settings.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that users do not receive notifications for tasks they are not assigned to

Test Case ID: NT5

Test Case Title: Verify that users do not receive notifications for tasks they are not assigned to

Test Case Description: Ensure that users do not receive notifications for tasks they are not assigned to, even if

they are updated or commented on.

Test Suite: Task Notifications

Test Priority: Medium

Preconditions:

- User is logged in
- Task is created and assigned to another user

Test Data: No test data needed

Test Steps:

- 1. Update a task assigned to another user
- 2. Verify that the user does not receive a notification for the task update
- 3. Check the notification inbox to ensure no notification is received

Postconditions:

- No notification is sent to the user

Expected Result: The system does not send notifications to users for tasks they are not assigned to, even if the task is updated or commented on.

Severity: Minor

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.', "]

Here are the 5 test cases for the Task Categorization feature:
Test Case 1: Verify that users can create and assign tasks to different categories or projects
Test Case ID: TC001
Test Case Title: Verify that users can create and assign tasks to different categories or projects
Test Case Description: Ensure that users can create new categories or projects and assign tasks to them for
better organization and management.
Test Suite: Task Categorization
Test Priority: High
Preconditions:
* User is logged in
* Task management tool is accessible
Test Data: No test data needed
Test Steps:
1. Log in to the task management tool
2. Create a new category or project
3. Create a new task
4. Assign the task to the newly created category or project
5. Save the task
Postconditions:

Generated Test Cases:

* Task is successfully assigned to the category or project
* Category or project information is updated
Expected Result: The system allows users to create new categories or projects and assign tasks to them.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that users can view tasks grouped by categories or projects
Test Case ID: TC002
Test Case Title: Verify that users can view tasks grouped by categories or projects
Test Case Description: Ensure that users can view tasks organized by categories or projects to easily track and
manage their tasks.
Test Suite: Task Categorization
Test Priority: Medium
Preconditions:
* User is logged in
* Tasks are assigned to categories or projects
Test Data: No test data needed
Test Steps:
1. Log in to the task management tool
2. Go to the task list



Test Steps:
1. Log in to the task management tool
2. Select a task
3. Move the task to a different category or project
4. Save the changes
Postconditions:
* Task is successfully moved to the new category or project
* Task information is updated
Expected Result: The system allows users to move tasks between categories or projects.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that category or project information is accurately reflected in task views
Test Case ID: TC004
Test Case Title: Verify that category or project information is accurately reflected in task views
Test Case Description: Ensure that category or project information is correctly displayed in task views to
provide accurate task information.
Test Suite: Task Categorization
Test Priority: Low
Preconditions:

* User is logged in
* Task is assigned to a category or project
Test Data: No test data needed
Test Steps:
1. Log in to the task management tool
2. Go to a task view
3. Verify that category or project information is displayed
4. Verify that the information is accurate
Postconditions:
* Category or project information is displayed correctly
* Task information is accurate
Expected Result: The system accurately displays category or project information in task views.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
**Test Case 5: Verify that error messages are displayed when attempting to assign a task to a non-existent
category or project**
Test Case ID: TC005
Test Case Title: Verify that error messages are displayed when attempting to assign a task to a non-existent
category or project

Test Case Description: Ensure that the system prevents users from assigning tasks to non-existent categories
or projects and displays an error message.
Test Suite: Task Categorization
Test Priority: Low
Preconditions:
* User is logged in
* Task is not assigned to a category or project
Test Data: No test data needed
Test Steps:
1. Log in to the task management tool
2. Create a new task
3. Attempt to assign the task to a non-existent category or project
4. Verify that an error message is displayed
Postconditions:
* Error message is displayed
* Task is not assigned to the non-existent category or project
Expected Result: The system prevents users from assigning tasks to non-existent categories or projects and
displays an error message.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Negative

Data - 47

Project name -

Task Management Tool

Project description -

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

should be able to set deadlines, add descriptions and attachments to tasks, organize tasks into categories or

projects, collaborate with team members, and receive notifications and reminders for upcoming deadlines.

Feature name -

Task Search

Feature description -

Users can search for tasks based on various criteria, such as task name, description, category, priority, or due

date. The task search feature aims to help users quickly find specific tasks within their project or task list.

Userstory title -

As a user, I want to search for tasks based on various criteria so that I can quickly find specific tasks within

my project or task list.

Acceptance criteria -

['Users can search for tasks by name.', 'Users can search for tasks by description.', 'Users can filter search

results by category, priority, or due date.', 'Search results are accurate and relevant.', "]

Generated Test Cases:

Here are the 5 detailed test cases for the Task Search feature:

Test Case 1: Verify that users can search for tasks by name

Test Case ID: TS001

Test Case Title: Verify that users can search for tasks by name
Test Case Description: Ensure that users can search for tasks by task name and retrieve relevant results.
Test Suite: Task Search
Test Priority: High
Preconditions:
* User is logged in
* At least one task is created with a unique name
Test Data: No test data needed
Test Steps:
1. Log in to the Task Management Tool
2. Navigate to the Task Search page
3. Enter a task name in the search bar
4. Click the search button
Postconditions:
* Search results are displayed
Expected Result: The system retrieves and displays tasks with the searched name.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that users can search for tasks by description

Test Case Title: Verify that users can search for tasks by description
Test Case Description: Ensure that users can search for tasks by task description and retrieve relevant results.
Test Suite: Task Search
Test Priority: High
Preconditions:
* User is logged in
* At least one task is created with a unique description
Test Data: No test data needed
Test Steps:
1. Log in to the Task Management Tool
2. Navigate to the Task Search page
3. Enter a task description in the search bar
4. Click the search button
Postconditions:
* Search results are displayed
Expected Result: The system retrieves and displays tasks with the searched description.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case 3: Verify that users can filter search results by category

Test Case ID: TS002

Test Case ID: TS003
Test Case Title: Verify that users can filter search results by category
Test Case Description: Ensure that users can filter search results by task category and retrieve relevant results.
Test Suite: Task Search
Test Priority: Medium
Preconditions:
* User is logged in
* At least one task is created with a category assigned
Test Data: No test data needed
Test Steps:
1. Log in to the Task Management Tool
2. Navigate to the Task Search page
3. Enter a search query
4. Click the search button
5. Select a category from the filter options
6. Click the apply filter button
Postconditions:
* Search results are updated and filtered by category
Expected Result: The system retrieves and displays tasks that match the searched query and category.
Severity: Major
Type of Testing: Functional

Test Case Approach: Positive
Test Case 4: Verify that users can filter search results by priority
Test Case ID: TS004
Test Case Title: Verify that users can filter search results by priority
Test Case Description: Ensure that users can filter search results by task priority and retrieve relevant results
Test Suite: Task Search
Test Priority: Medium
Preconditions:
* User is logged in
* At least one task is created with a priority assigned
Test Data: No test data needed
Test Steps:
1. Log in to the Task Management Tool
2. Navigate to the Task Search page
3. Enter a search query
4. Click the search button
5. Select a priority from the filter options
6. Click the apply filter button
Postconditions:
* Search results are updated and filtered by priority

Expected Result: The system retrieves and displays tasks that match the searched query and priority.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that search results are accurate and relevant
Test Case ID: TS005
Test Case Title: Verify that search results are accurate and relevant
Test Case Description: Ensure that search results are accurate and relevant to the search query and filters
applied.
Test Suite: Task Search
Test Priority: High
Preconditions:
* User is logged in
* At least one task is created with various attributes (name, description, category, priority, due date)
Test Data: No test data needed
Test Steps:
1. Log in to the Task Management Tool
2. Navigate to the Task Search page
3. Enter a search query
4. Click the search button
5. Apply various filters (category, priority, due date)

6. Verify the search results

Postconditions:

* Search results are accurate and relevant

Expected Result: The system retrieves and displays tasks that match the searched query and filters applied.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Data - 48

Project name -

Task Management Tool

Project description -

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

Feature name -

Task Collaboration

Feature description -

Users can collaborate on tasks by adding comments, attaching files, and sharing tasks with other users. Task collaboration aims to enhance team communication and productivity.

Userstory title -

As a user, I want to collaborate on tasks by adding comments, attaching files, and sharing tasks with other users so that I can work more effectively with my team.

Acceptance criteria -

['Users can add comments to tasks.', 'Users can attach files to tasks.', 'Users can share tasks with other users.', 'Task collaboration features work seamlessly and enhance team communication.', "] **Generated Test Cases:** Here are the 5 test cases to cover the acceptance criteria for the Task Collaboration feature: **Test Case 1: Verify that users can add comments to tasks** Test Case ID: tct-001 Test Case Title: Verify that users can add comments to tasks Test Case Description: Ensure that the system allows users to add comments to tasks, enabling effective team communication and collaboration. Test Suite: Task Collaboration Test Priority: High Preconditions: * User is logged in * Task is created and assigned to the user * Comment section is available on the task page Test Data: No test data needed

Test Steps:

- 1. Go to the task page
- 2. Click on the comment section
- 3. Enter a comment and press enter
- 4. Verify that the comment is saved and displayed on the task page

Postconditions:
* Comment is added to the task
* Comment is visible to other users with access to the task
Expected Result: The system allows users to add comments to tasks, and the comment is displayed on the task
page for other users to see.
Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive
Test Case 2: Verify that users can attach files to tasks
Test Case ID: tct-002
Test Case Title: Verify that users can attach files to tasks
Test Case Description: Ensure that the system allows users to attach files to tasks, enabling effective document
sharing and collaboration.
Test Suite: Task Collaboration
Test Priority: High
Preconditions:
* User is logged in
* Task is created and assigned to the user
* File attachment section is available on the task page
Test Data: Sample file (e.g., PDF, Word document)

Test Steps:
1. Go to the task page
2. Click on the file attachment section
3. Select a file to attach and upload it
4. Verify that the file is attached and displayed on the task page
Postconditions:
* File is attached to the task
* File is visible to other users with access to the task
Expected Result: The system allows users to attach files to tasks, and the file is displayed on the task page for
other users to access.
Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive
Test Case 3: Verify that users can share tasks with other users
Test Case ID: tct-003
Test Case Title: Verify that users can share tasks with other users
Test Case Description: Ensure that the system allows users to share tasks with other users, enabling effective
collaboration and task assignment.
Test Suite: Task Collaboration
Test Priority: High
Preconditions:

* User is logged in
* Task is created and assigned to the user
* Sharing feature is available on the task page
Test Data: No test data needed
Test Steps:
1. Go to the task page
2. Click on the sharing feature
3. Enter the username or email of the user to share with
4. Verify that the task is shared and the other user has access to the task
Postconditions:
* Task is shared with the other user
* Other user has access to the task and can view its details
Expected Result: The system allows users to share tasks with other users, and the task is accessible to the
shared users.
Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive
Test Case 4: Verify that task collaboration features work seamlessly and enhance team communication
Test Case ID: tct-004

Test Case Title: Verify that task collaboration features work seamlessly and enhance team communication
Test Case Description: Ensure that the system's task collaboration features work together seamlessly, enabling
efficient team communication and collaboration.
Test Suite: Task Collaboration
Test Priority: High
Preconditions:
* User is logged in
* Task is created and assigned to the user
* Comment, file attachment, and sharing features are available on the task page
Test Data: No test data needed
Test Steps:
1. Go to the task page
2. Add a comment and attach a file to the task
3. Share the task with another user
4. Verify that the comment, file, and sharing features work together seamlessly and enhance team
communication
Postconditions:
* Task collaboration features work together seamlessly
* Team communication and collaboration are enhanced
Expected Result: The system's task collaboration features work together seamlessly, enabling efficient team
communication and collaboration.

Severity: Critical
Type of Testing: Functional
Test Case Behaviour: Positive
Test Case 5: Verify that users cannot collaborate on tasks without proper access permissions
Test Case ID: tct-005
Test Case Title: Verify that users cannot collaborate on tasks without proper access permissions
Test Case Description: Ensure that the system restricts access to task collaboration features based on user
permissions, preventing unauthorized access and ensuring data security.
Test Suite: Task Collaboration
Test Priority: Medium
Preconditions:
* User is logged in
* Task is created and assigned to another user
* Current user does not have access permissions to the task
Test Data: No test data needed
Test Steps:
1. Go to the task page
2. Try to add a comment, attach a file, or share the task
3. Verify that the system restricts access to the task collaboration features
Postconditions:

* User is unable to access task collaboration features

* System displays an error message or permission denied message

Expected Result: The system restricts access to task collaboration features based on user permissions, ensuring

data security and preventing unauthorized access.

Severity: Major

Type of Testing: Security

Test Case Behaviour: Negative

Data - 49

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Course Creation Tools

Feature description -

Instructors are equipped with intuitive tools to develop comprehensive courses encompassing various

multimedia elements like videos, presentations, quizzes, and assignments. These tools empower instructors to

structure content effectively, ensuring an engaging and informative learning experience for students.

Userstory title -

As an instructor, I want intuitive course creation tools so that I can easily develop comprehensive courses.

Acceptance criteria -

['Instructors can create courses using videos, presentations, quizzes, and assignments.', 'The tools are user-friendly and require minimal technical skills.', 'Courses can be previewed before publishing.', 'Instructors can save courses as drafts.', 'Courses can be edited post-publishing.']

Generated Test Cases:

Here are the 5 test cases for the Online Learning Platform's Course Creation Tools feature:

Test Case 1:

Test Case ID: OLP-CCT-1

Test Case Title: Verify that instructors can create courses using videos, presentations, quizzes, and assignments.

Test Case Description: Ensure that the course creation tools allow instructors to add various multimedia elements to their courses, enhancing the learning experience for students.

Test Suite: Course Creation Tools

Test Priority: High

Preconditions:

- * Instructor is logged in
- * Course creation tools are accessible

Test Data: Sample course content (video, presentation, quiz, assignment)

Test Steps:

- 1. Log in as an instructor
- 2. Access the course creation tools
- 3. Create a new course
- 4. Add a video lecture to the course

5. Add a presentation to the course 6. Add a quiz to the course 7. Add an assignment to the course Postconditions: * Course is created with multimedia elements Expected Result: The system allows instructors to create courses with videos, presentations, quizzes, and assignments, enhancing the learning experience for students. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2:** Test Case ID: OLP-CCT-2 Test Case Title: Verify that the course creation tools are user-friendly and require minimal technical skills. Test Case Description: Ensure that the course creation tools are intuitive and easy to use, allowing instructors to focus on creating engaging content without requiring extensive technical expertise. Test Suite: Course Creation Tools Test Priority: Medium Preconditions: * Instructor is logged in * Course creation tools are accessible Test Data: No test data needed

Test Steps:

- 1. Log in as an instructor 2. Access the course creation tools 3. Create a new course 4. Observe the user interface and available features 5. Attempt to create a simple course without technical assistance Postconditions: * Instructor can create a course without technical issues Expected Result: The system provides user-friendly course creation tools that require minimal technical skills, allowing instructors to easily create courses. Severity: Minor Type of Testing: Usability Test Case Approach: Positive **Test Case 3:** Test Case ID: OLP-CCT-3 Test Case Title: Verify that courses can be previewed before publishing. Test Case Description: Ensure that instructors can review and refine their courses before making them available to students, ensuring high-quality course content. Test Suite: Course Creation Tools Test Priority: High Preconditions: * Instructor is logged in
- * Course creation tools are accessible
- * A course is created with multimedia elements

Test Data: Sample course content (video, presentation, quiz, assignment)
Test Steps:
1. Log in as an instructor
2. Access the course creation tools
3. Create a new course
4. Add multimedia elements to the course
5. Preview the course
6. Verify that the course can be reviewed and refined before publishing
Postconditions:
* Course can be previewed before publishing
Expected Result: The system allows instructors to preview and refine their courses before making them
available to students, ensuring high-quality course content.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4:
Test Case ID: OLP-CCT-4
Test Case Title: Verify that instructors can save courses as drafts.
Test Case Description: Ensure that instructors can save their courses as drafts, allowing them to resume course
creation at a later time and reducing the risk of lost work.
Test Suite: Course Creation Tools
Test Priority: Medium
Preconditions:

* Instructor is logged in * Course creation tools are accessible * A course is created with multimedia elements Test Data: Sample course content (video, presentation, quiz, assignment) Test Steps: 1. Log in as an instructor 2. Access the course creation tools 3. Create a new course 4. Add multimedia elements to the course 5. Save the course as a draft 6. Verify that the course is saved and can be resumed later Postconditions: * Course is saved as a draft Expected Result: The system allows instructors to save their courses as drafts, enabling them to resume course creation at a later time and reducing the risk of lost work. Severity: Minor Type of Testing: Functional Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP-CCT-5

Test Case Title: Verify that courses can be edited post-publishing.

Test Case Description: Ensure that instructors can make changes to their published courses, allowing them to update and improve course content without affecting student access.

Test Suite: Course Creation Tools
Test Priority: High
Preconditions:
* Instructor is logged in
* Course creation tools are accessible
* A course is published with multimedia elements
Test Data: Sample course content (video, presentation, quiz, assignment)
Test Steps:
1. Log in as an instructor
2. Access the course creation tools
3. Select a published course
4. Make changes to the course content
5. Verify that the changes are saved and reflected in the published course
Postconditions:
* Course is updated and reflected in the published version
Expected Result: The system allows instructors to edit and update their published courses, enabling them to
improve course content without affecting student access.
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 guizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Interactive Lessons

Feature description -

Lessons are thoughtfully designed to foster learner engagement through interactive components such as

quizzes, polls, simulations, and discussions. By encouraging active participation and critical thinking, these

interactive elements enhance comprehension and retention of course material.

Userstory title -

As a learner, I want interactive lessons so that I can engage with the content more effectively.

Acceptance criteria -

['Lessons include quizzes, polls, simulations, and discussions.', 'Interactive elements are integrated seamlessly

into the lesson content.', 'Learners can track their progress through these activities.', 'Feedback is provided

after each interactive component.', 'Interactive lessons can be accessed on various devices.']

Generated Test Cases:

Here are the 5 test cases for the Interactive Lessons feature:

Test Case 1

Test Case ID: IL001

Test Case Title: Verify that lessons include quizzes, polls, simulations, and discussions.

Test Case Description: Ensure that interactive elements are included in lessons to facilitate engagement.

Test Suite: Interactive Lessons

Test Priority: High
Preconditions:
* Learner is logged in
* Lesson is available
Test Data: No test data needed
Test Steps:
1. Access a lesson
2. Verify the presence of quizzes, polls, simulations, and discussions
3. Attempt to interact with each element
Postconditions:
* Interactive elements are visible and functional
Expected Result: Lessons include quizzes, polls, simulations, and discussions that learners can interact with
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: Ensure that interactive elements do not interrupt the lesson flow or cause errors.
Test Suite: Interactive Lessons
Test Priority: High
Preconditions:
* Learner is logged in
* Lesson is available
Test Data: No test data needed
Test Steps:

1. Access a lesson 2. Interact with an interactive element (e.g., quiz, poll) 3. Verify that the lesson continues without errors or interruptions Postconditions: * Lesson flow is not interrupted Expected Result: Interactive elements are integrated seamlessly into the lesson content, without errors or interruptions. 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 interactive activities. Test Case Description: Ensure that learners can view their progress and receive feedback on interactive activities. Test Suite: Interactive Lessons Test Priority: High Preconditions: * Learner is logged in * Lesson is available Test Data: No test data needed Test Steps: 1. Access a lesson 2. Complete an interactive activity (e.g., quiz, poll) 3. Verify that progress is tracked and feedback is provided

Postconditions:

* Progress is tracked

Expected Result: Learners can track their progress through interactive activities and receive feedback.

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: Ensure that learners receive feedback on their performance in interactive activities.

Test Suite: Interactive Lessons

Test Priority: Medium

Preconditions:

* Learner is logged in

* Lesson is available

Test Data: No test data needed

Test Steps:

- 1. Access a lesson
- 2. Complete an interactive activity (e.g., quiz, poll)
- 3. Verify that feedback is provided

Postconditions:

* Feedback is displayed

Expected Result: Feedback is provided after each interactive component, helping learners understand their

performance.

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: Ensure that interactive lessons are compatible with different devices and browsers.
Test Suite: Interactive Lessons
Test Priority: Medium
Preconditions:
* Learner is logged in
* Lesson is available
Test Data: No test data needed
Test Steps:
1. Access a lesson on different devices (e.g., desktop, laptop, tablet, mobile)
2. Verify that interactive elements are functional on each device
3. Verify that the lesson is compatible with different browsers (e.g., Chrome, Firefox, Safari)
Postconditions:
* Interactive lessons are accessible on various devices and browsers
Expected Result: Interactive lessons can be accessed on various devices and browsers without issues.
Severity: Minor
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:

Here are the 5 detailed test cases for the Personalized Learning Paths feature:

Test Case 1:

Test Case ID: PLC1

Test Case Title: Verify that learners can choose courses and modules based on their interests

Test Case Description: Ensure that learners can select courses and modules that align with their unique

interests, preferences, and learning objectives.

Test Suite: Personalized Learning Paths Test Priority: High Preconditions: - Learner is logged in - Multiple courses and modules are available Test Data: No test data needed Test Steps: 1. Go to the course catalog 2. Filter courses by interest area (e.g., programming, marketing) 3. Select a course and its modules 4. Verify that the selected courses and modules are added to the learner's personalized learning path Postconditions: - Selected courses and modules are reflected in the learner's dashboard Expected Result: Learners can choose courses and modules that match their interests and preferences. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2:** Test Case ID: PLC2 Test Case Title: Verify that personalized learning paths adjust dynamically based on learner progress Test Case Description: Ensure that the personalized learning path updates dynamically as learners complete courses, modules, and activities. Test Suite: Personalized Learning Paths Test Priority: High

Preconditions:

- Learner is logged in

- Learner has an existing personalized learning path

Test Data: No test data needed

Test Steps:

1. Complete a course or module in the personalized learning path

2. Verify that the personalized learning path updates to reflect the learner's progress

3. Verify that new course or module recommendations are provided based on the learner's progress

Postconditions:

- Personalized learning path is updated to reflect learner progress

Expected Result: Personalized learning paths adjust dynamically based on learner progress, providing new

recommendations.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: PLC3

Test Case Title: Verify that learners receive recommendations for new courses and modules

Test Case Description: Ensure that learners receive personalized course and module recommendations based

on their interests, preferences, and learning objectives.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- Learner is logged in

- Learner has an existing personalized learning path

Test Data: No test data needed

Test Steps:

1. Access the learner's dashboard

- 2. Verify that personalized course and module recommendations are displayed
- 3. Verify that recommendations align with the learner's interests, preferences, and learning objectives

Postconditions:

- Personalized course and module recommendations are displayed

Expected Result: Learners receive relevant and personalized course and module recommendations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: PLC4

Test Case Title: Verify that progress tracking is personalized to each learner's path

Test Case Description: Ensure that progress tracking is tailored to each learner's unique learning path,

reflecting their completed and in-progress courses, modules, and activities.

Test Suite: Personalized Learning Paths

Test Priority: High

Preconditions:

- Learner is logged in
- Learner has an existing personalized learning path

Test Data: No test data needed

Test Steps:

- 1. Access the learner's dashboard
- 2. Verify that progress tracking reflects the learner's completed and in-progress courses, modules, and

activities

3. Verify that progress tracking is updated dynamically as the learner completes courses, modules, and

activities

Postconditions:

- Progress tracking is updated to reflect learner progress

Expected Result: Progress tracking is personalized to each learner's unique learning path.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: PLC5

Test Case Title: Verify that learners can adjust their learning paths at any time

Test Case Description: Ensure that learners can modify their personalized learning paths by adding, removing,

or reordering courses, modules, and activities at any time.

Test Suite: Personalized Learning Paths

Test Priority: Medium

Preconditions:

- Learner is logged in

- Learner has an existing personalized learning path

Test Data: No test data needed

Test Steps:

1. Access the learner's dashboard

2. Modify the personalized learning path by adding, removing, or reordering courses, modules, and activities

3. Verify that the modified personalized learning path is saved and reflected in the learner's dashboard

Postconditions:

- Modified personalized learning path is saved and reflected in the learner's dashboard

Expected Result: Learners can adjust their learning paths at any time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 52

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Progress Tracking

Feature description -

Learners benefit from comprehensive progress tracking features that provide detailed insights into their advancement through courses, modules, and tasks. By monitoring completed activities, grades, and achievements, learners can gauge their performance and identify areas for improvement.

Userstory title -

As a learner, I want detailed progress tracking so that I can monitor my advancement and identify areas for improvement.

Acceptance criteria -

['Progress tracking shows completed activities and grades.', 'Learners can view detailed reports on their progress.', 'Achievements are highlighted in the progress tracker.', 'Learners receive notifications about their progress.', 'Progress tracking can be exported as reports.']

Generated Test Cases:

Here are the five test cases for the Progress Tracking feature:

Test Case ID: PT-001

Test Case Title: Verify that progress tracking shows completed activities and grades

Test Case Description: Ensure that the progress tracking feature accurately displays completed activities and

corresponding grades for learners.

Test Suite: Progress Tracking

Test Priority: High

Preconditions:

? Learner is logged in

? Learner has completed at least one activity

Test Data: No test data needed

Test Steps:

1. Log in as a learner

2. Navigate to the progress tracking page

3. Verify that completed activities are listed with corresponding grades

Postconditions:

? Completed activities are displayed

Expected Result: The progress tracking page shows completed activities and grades accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: PT-002

Test Case Title: Verify that learners can view detailed reports on their progress

Test Case Description: Ensure that learners can access detailed reports on their progress, including completed

activities, grades, and achievements.

Test Suite: Progress Tracking

Test Priority: High
Preconditions:
? Learner is logged in
? Learner has completed at least one activity
Test Data: No test data needed
Test Steps:
1. Log in as a learner
2. Navigate to the progress tracking page
3. Click on the "View Detailed Report" button
4. Verify that the detailed report is displayed with completed activities, grades, and achievements
Postconditions:
? Detailed report is generated
Expected Result: The detailed report accurately displays learner's progress, including completed activities,
grades, and achievements.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: PT-003
Test Case Title: Verify that achievements are highlighted in the progress tracker
Test Case Description: Ensure that achievements earned by learners are highlighted in the progress tracking
feature.
Test Suite: Progress Tracking
Test Priority: Medium
Preconditions:
? Learner is logged in

? Learner has earned at least one achievement Test Data: No test data needed Test Steps: 1. Log in as a learner 2. Navigate to the progress tracking page 3. Verify that earned achievements are highlighted and clearly visible Postconditions: ? Achievements are highlighted Expected Result: The progress tracking page highlights achievements earned by learners. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 4:** Test Case ID: PT-004 Test Case Title: Verify that learners receive notifications about their progress Test Case Description: Ensure that learners receive timely notifications about their progress, including completed activities, grades, and achievements. Test Suite: Progress Tracking Test Priority: Medium Preconditions: ? Learner is logged in ? Learner has completed at least one activity Test Data: No test data needed Test Steps: 1. Log in as a learner 2. Complete an activity

3. Verify that a notification is sent to the learner about their progress
Postconditions:
? Notification is sent
Expected Result: Learners receive timely notifications about their progress, including completed activities
grades, and achievements.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5:
Test Case ID: PT-005
Test Case Title: Verify that progress tracking can be exported as reports
Test Case Description: Ensure that learners can export their progress tracking data as reports, including
completed activities, grades, and achievements.
Test Suite: Progress Tracking
Test Priority: Medium
Preconditions:
? Learner is logged in
? Learner has completed at least one activity
Test Data: No test data needed
Test Steps:
1. Log in as a learner
2. Navigate to the progress tracking page
3. Click on the "Export as Report" button
4. Verify that the report is generated and downloadable
Postconditions:
? Report is generated

Expected Result: Learners can export their progress tracking data as reports, including completed activities, grades, and achievements.

Severity: Major

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:

Here are the 5 detailed test cases for the Assessment and Grading feature:

Test Case 1: Verify that instructors can create quizzes

Test Case ID: TG8R9p

Test Case Title: Verify that instructors can create quizzes

Test Case Description: Ensure that instructors can create quizzes as a type of assessment on the online learning

platform.

Test Suite: Assessment and Grading

Test Priority: High

Preconditions:

* Instructor is logged in

* Course is created

* Quiz template is available

Test Data: Quiz details (title, description, questions)

Test Steps:

- 1. Go to the course page
- 2. Click on the "Create Assessment" button
- 3. Select "Quiz" as the assessment type
- 4. Fill in quiz details and questions

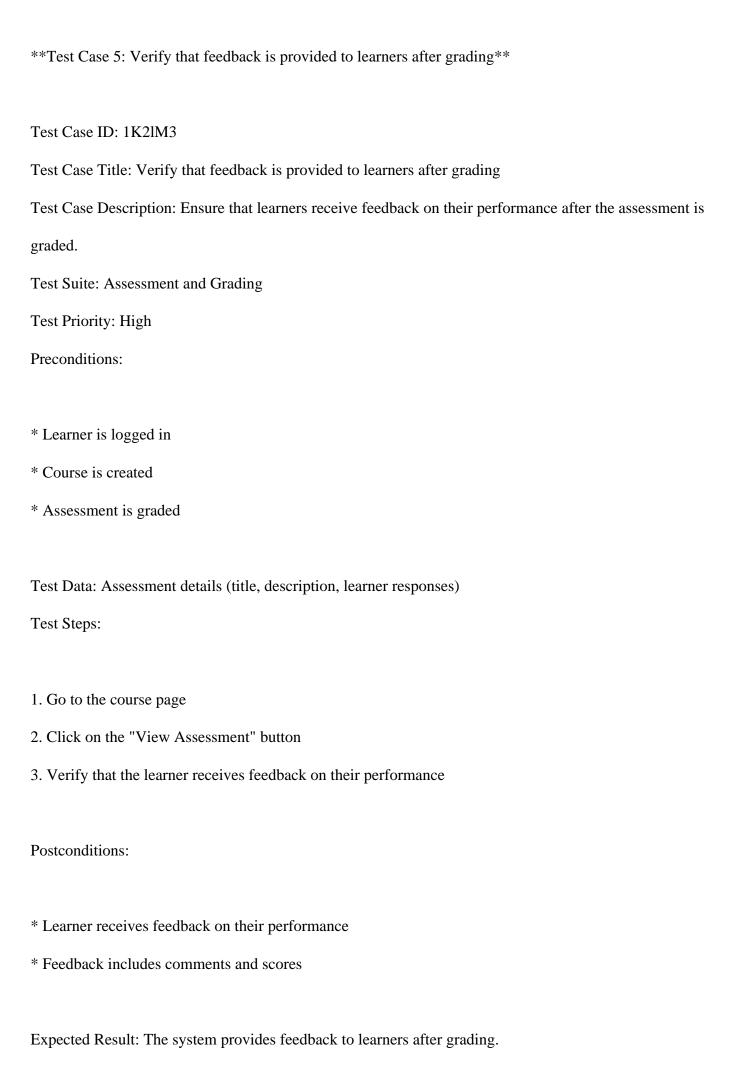
5. Save the quiz
Postconditions:
* Quiz is created and listed under course assessments
* Quiz is available for learners to take
Expected Result: The system allows instructors to create quizzes as a type of assessment on the online
learning platform.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that assessments can include various question types
Test Case ID: 3T5gH7
Test Case Title: Verify that assessments can include various question types
Test Case Description: Ensure that assessments can include multiple question types, such as MCQs, short
answers, and essays.
Test Suite: Assessment and Grading
Test Priority: High
Preconditions:
* Instructor is logged in
* Course is created
* Assessment template is available

Test Data: Assessment details (title, description, questions with various types)
Test Steps:
1. Go to the course page
2. Click on the "Create Assessment" button
3. Select "Quiz" or "Test" as the assessment type
4. Add questions with different types (MCQ, short answer, essay)
5. Save the assessment
Postconditions:
* Assessment is created and listed under course assessments
* Assessment includes questions with various types
Expected Result: The system allows assessments to include various question types.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that automatic grading is available for objective questions
Test Case ID: 7J4dS1
Test Case Title: Verify that automatic grading is available for objective questions
Test Case Description: Ensure that the system can automatically grade objective questions, such as MCQs.
Test Suite: Assessment and Grading
Test Priority: High
Preconditions:

* Course is created
* Assessment with objective questions is available
Test Data: Assessment details (title, description, objective questions with correct answers)
Test Steps:
1. Go to the course page
2. Click on the "Grade Assessment" button
3. Select an assessment with objective questions
4. Verify that the system automatically grades the questions
Postconditions:
* Assessment is graded automatically
* Correct answers are marked as correct
Expected Result: The system automatically grades objective questions correctly.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that manual grading is available for subjective questions
Test Case ID: 4T9eR2
Test Case Title: Verify that manual grading is available for subjective questions

* Instructor is logged in

Test Case Description: Ensure that instructors can manually grade subjective questions, such as essays.
Test Suite: Assessment and Grading
Test Priority: High
Preconditions:
* Instructor is logged in
* Course is created
* Assessment with subjective questions is available
Test Data: Assessment details (title, description, subjective questions)
Test Steps:
1. Go to the course page
2. Click on the "Grade Assessment" button
3. Select an assessment with subjective questions
4. Verify that the instructor can manually grade the questions
Postconditions:
* Assessment is graded manually
* Instructor feedback is provided
Expected Result: The system allows instructors to manually grade subjective questions.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive



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:

Here are the 5 test cases for the Online Learning Platform's Discussion Forums feature:

Test Case 1: Verify that learners can post questions and responses in forums

Test Case ID: OL_DF_001

Test Case Title: Verify that learners can post questions and responses in forums

Test Case Description: Ensure that learners can create new posts and respond to existing ones in discussion

forums.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

* Learner is logged in

* Course is enrolled

* Forum is accessible

Test Data: No test data needed

Test Steps:

- 1. Navigate to a course forum
- 2. Click on the "New Post" button
- 3. Enter a question or response and submit
- 4. Verify the post is created successfully

Postconditions:

* New post is created

* Post is visible to other learners and instructors

Expected Result: Learners can successfully post questions and responses in forums.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that forums support threaded discussions

Test Case ID: OL_DF_002

Test Case Title: Verify that forums support threaded discussions

Test Case Description: Ensure that learners can create threaded discussions by replying to existing posts.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

- * Learner is logged in
- * Course is enrolled
- * Forum is accessible

Test Data: No test data needed

Test Steps:

- 1. Navigate to a course forum
- 2. Create a new post
- 3. Reply to the post as a different learner
- 4. Verify the reply is threaded under the original post

Postconditions:

- * Threaded discussion is created
- * Reply is visible under the original post

Expected Result: Forums support threaded discussions, allowing learners to engage in meaningful

conversations.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that instructors can moderate discussions

Test Case ID: OL_DF_003

Test Case Title: Verify that instructors can moderate discussions

Test Case Description: Ensure that instructors can manage discussion forums by editing, deleting, or pinning

posts.

Test Suite: Discussion Forums

Test Priority: High

Preconditions:

* Instructor is logged in

* Course is created

* Forum is accessible

Test Data: No test data needed

Test Steps:

- 1. Navigate to a course forum as an instructor
- 2. Create a new post as a learner
- 3. Edit the post as an instructor
- 4. Verify the changes are reflected

Postconditions:

* Post is edited successfully

* Instructor moderation actions are reflected

Expected Result: Instructors can moderate discussions by editing, deleting, or pinning posts.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that learners receive notifications for new posts

Test Case ID: OL_DF_004

Test Case Title: Verify that learners receive notifications for new posts

Test Case Description: Ensure that learners receive notifications when new posts are created in discussion

forums.

Test Suite: Discussion Forums

Test Priority: Medium

Preconditions:

* Learner is logged in

* Course is enrolled

* Forum is accessible

Test Data: No test data needed

Test Steps:

1. Navigate to a course forum

2. Create a new post

3. Verify notification is sent to learners subscribed to the forum

Postconditions:

* Notification is sent to learners

* Learners are aware of new posts

Expected Result: Learners receive notifications for new posts in discussion forums.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that forums are accessible on all devices

Test Case ID: OL_DF_005

Test Case Title: Verify that forums are accessible on all devices Test Case Description: Ensure that discussion forums are accessible and functional on various devices and browsers. **Test Suite: Discussion Forums** Test Priority: High Preconditions: * Learner is logged in * Course is enrolled * Forum is accessible Test Data: No test data needed Test Steps: 1. Access the forum on a desktop browser 2. Access the forum on a mobile browser 3. Access the forum on a tablet 4. Verify the forum is accessible and functional Postconditions: * Forum is accessible on all devices * Forum functionality is consistent across devices Expected Result: Discussion forums are accessible and functional on various devices and browsers. Severity: Major Type of Testing: Compatibility

Test Case Approach: Positive

Data - 55

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and guizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Live Classes and Webinars

Feature description -

The platform facilitates live interactive sessions, allowing instructors to conduct real-time classes and

webinars through seamless video conferencing tools. These live sessions enable dynamic engagement, Q&A

opportunities, and direct interaction between instructors and learners, replicating the immersive experience of

traditional classroom settings.

Userstory title -

As a learner, I want to attend live classes and webinars so that I can interact with instructors in real-time.

Acceptance criteria -

['Live classes and webinars are conducted through video conferencing tools.', 'Learners can ask questions in

real-time.', 'Sessions are recorded and available for later viewing.', 'Instructors can share screen and resources

during sessions.', 'Notifications are sent before sessions start.']

Generated Test Cases:

Here are the 5 test cases for the given input data:

Test Case 1:

Test Case ID: LCW1

Test Case Title: Verify that live classes and webinars are conducted through video conferencing tools

Test Case Description: Ensure that the online learning platform facilitates live interactive sessions through

seamless video conferencing tools, allowing instructors to conduct real-time classes and webinars.

Test Priority: High

Test Suite: Live Classes and Webinars

Preconditions:

? Instructor is logged in

? Learner is registered for the live class or webinar

Test Data: No test data needed

Test Steps:

1. Instructor logs in to the platform

2. Instructor initiates a live class or webinar

3. Learner joins the live session

4. Verify video conferencing tools are utilized

Postconditions:

? Live session is active

Expected Result: The platform facilitates live classes and webinars through video conferencing tools, enabling real-time interaction between instructors and learners.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: LCW2

Test Case Title: Verify that learners can ask questions in real-time

Test Case Description: Ensure that the online learning platform allows learners to ask questions during live

classes and webinars, promoting dynamic engagement and interaction.

Test Suite: Live Classes and Webinars

Test Priority: High
Preconditions:
? Learner is logged in
? Learner is attending a live class or webinar
Test Data: No test data needed
Test Steps:
1. Learner attends a live class or webinar
2. Learner types a question in the chatbox or Q&A section
3. Verify instructor receives the question
4. Verify instructor responds to the question
Postconditions:
? Learner's question is visible to the instructor
Expected Result: The platform enables learners to ask questions in real-time, and instructors can respond to
them during live sessions.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: LCW3
Test Case Title: Verify that sessions are recorded and available for later viewing
Test Case Description: Ensure that the online learning platform records live classes and webinars, making
them accessible for learners to review at a later time.
Test Suite: Live Classes and Webinars
Test Priority: Medium
Preconditions:

? Live class or webinar has concluded ? Learner is logged in Test Data: No test data needed Test Steps: 1. Learner logs in to the platform 2. Learner navigates to the recorded sessions section 3. Verify recorded live class or webinar is available 4. Verify learner can play the recorded session Postconditions: ? Recorded session is accessible Expected Result: The platform records live classes and webinars, making them available for learners to view at a later time. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 4:** Test Case ID: LCW4 Test Case Title: Verify that instructors can share screen and resources during sessions Test Case Description: Ensure that the online learning platform enables instructors to share their screen and resources during live classes and webinars, enhancing the learning experience. Test Suite: Live Classes and Webinars Test Priority: High Preconditions: ? Instructor is logged in ? Instructor is conducting a live class or webinar

Test Data: No test data needed

Test Steps:

1. Instructor shares their screen during the live session

2. Verify learners can view the shared screen

3. Instructor shares a resource (e.g., document, image)

4. Verify learners can access the shared resource

Postconditions:

? Shared screen and resources are visible to learners

Expected Result: The platform allows instructors to share their screen and resources during live classes and webinars, facilitating dynamic engagement and knowledge sharing.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: LCW5

Test Case Title: Verify that notifications are sent before sessions start

Test Case Description: Ensure that the online learning platform sends notifications to learners before live classes and webinars, reminding them of the upcoming session.

Test Suite: Live Classes and Webinars

Test Priority: Medium

Preconditions:

? Learner is registered for a live class or webinar

? Notification settings are enabled

Test Data: No test data needed

Test Steps:

- 1. Learner registers for a live class or webinar
- 2. Verify notification settings are enabled
- 3. Wait for the notification to be sent before the session starts
- 4. Verify learner receives the notification

Postconditions:

? Learner receives a notification before the session starts

Expected Result: The platform sends notifications to learners before live classes and webinars, ensuring they are reminded of the upcoming session.

Severity: Minor

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:

Here are the 5 test cases for the Online Learning Platform's Mobile Accessibility feature:

Test Case 1:

Test Case ID: MobiAcc_01

Test Case Title: Verify that the platform is accessible on smartphones and tablets

Test Case Description: Ensure that learners can access the online learning platform on their mobile devices,

including smartphones and tablets, without any issues.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Learner has a valid account

- Mobile device has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Access the online learning platform using a smartphone

2. Log in to the learner account

3. Access the course dashboard

Postconditions:

- Learner is logged in successfully

Expected Result: The platform is accessible on smartphones and tablets, and learners can log in successfully.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: MobiAcc_02

Test Case Title: Verify that mobile UI is optimized for touch navigation

Test Case Description: Ensure that the mobile UI is designed for touch navigation, making it easy for learners

to navigate through the platform using their mobile devices.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Learner has a valid account

- Mobile device has a stable internet connection

Test Data: No test data needed

Test Steps:

1. Access the online learning platform using a mobile device

- 2. Navigate through the course dashboard using touch gestures
- 3. Access course materials and lectures

Postconditions:

- Navigation is smooth and intuitive

Expected Result: The mobile UI is optimized for touch navigation, and learners can easily access course

materials and lectures.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case 3:

Test Case ID: MobiAcc 03

Test Case Title: Verify that course materials are accessible and readable on mobile devices

Test Case Description: Ensure that course materials, including lectures, tutorials, and quizzes, are accessible

and readable on mobile devices, providing learners with a seamless learning experience.

Test Suite: Mobile Accessibility

Test Priority: High

Preconditions:

- Learner has a valid account

- Mobile device has a stable internet connection

Test Data: Sample course materials (e.g., PDF, video, quiz)

Test Steps:

1. Access a course on the online learning platform using a mobile device

2. Download and access course materials (e.g., PDF, video, quiz)

3. Verify that course materials are readable and functional

Postconditions:

- Course materials are accessible and readable

Expected Result: Course materials are accessible and readable on mobile devices, ensuring a seamless learning

experience.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: MobiAcc_04

Test Case Title: Verify that notifications are sent to mobile devices

Test Case Description: Ensure that learners receive notifications on their mobile devices, keeping them

informed about important updates and announcements from the platform.

Test Suite: Mobile Accessibility

Test Priority: Medium

Preconditions:

- Learner has a valid account

- Mobile device has a stable internet connection

- Notifications are enabled

Test Data: No test data needed

Test Steps:

1. Access the online learning platform using a mobile device

2. Enable notifications

3. Trigger a notification (e.g., course update, announcement)

Postconditions:

- Notification is received on the mobile device

Expected Result: Notifications are sent to mobile devices, keeping learners informed about important updates

and announcements.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: MobiAcc_05

Test Case Title: Verify that the platform supports offline access to downloaded materials

Test Case Description: Ensure that learners can access downloaded course materials offline, allowing them to

continue learning even without an internet connection.

Test Suite: Mobile Accessibility
Test Priority: High
Preconditions:
- Learner has a valid account
- Mobile device has a stable internet connection
- Course materials are downloadable
Test Data: Sample course materials (e.g., PDF, video)
Test Steps:
1. Access the online learning platform using a mobile device
2. Download course materials
3. Disconnect from the internet
4. Access downloaded course materials
Postconditions:
- Course materials are accessible offline
Expected Result: The platform supports offline access to downloaded materials, allowing learners to continue
learning offline.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Data - 57
Project name -
Online Learning Platform
Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Resource Library

Feature description -

A comprehensive repository of supplementary materials such as e-books, articles, and reference guides

enriches the learning experience by providing additional insights and resources. Learners can access diverse

content to deepen their understanding, explore related topics, and enhance their knowledge beyond the

confines of the core curriculum.

Userstory title -

As a learner, I want access to a resource library so that I can explore additional materials and deepen my

understanding.

Acceptance criteria -

['Resource library includes e-books, articles, and reference guides.', 'Learners can search for specific

materials.', 'Resources are categorized and tagged for easy navigation.', 'Learners can download resources for

offline access.', 'Instructors can recommend resources to learners.']

Generated Test Cases:

Here are the 5 test cases for the Resource Library feature:

Test Case 1

Test Case ID: RL001

Test Case Title: Verify that resource library includes e-books, articles, and reference guides.

Test Case Description: Ensure that the resource library contains diverse materials including e-books, articles,

and reference guides for learners to access.

Test Suite: Resource Library

Test Priority: High

Preconditions:
* Learner is logged in
* Resource library is enabled
Test Data: No test data needed
Test Steps:
1. Navigate to the resource library
2. Browse through the library
3. Verify the presence of e-books, articles, and reference guides
Postconditions:
* Resource library is populated with e-books, articles, and reference guides
Expected Result: The resource library includes e-books, articles, and reference guides.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2
Test Case ID: RL002
Test Case Title: Verify that learners can search for specific materials.
Test Case Description: Ensure that learners can search for specific resources in the library using relevant
keywords.
Test Suite: Resource Library
Test Priority: High
Preconditions:
* Learner is logged in
* Resource library is enabled
Test Data: Sample search queries (e.g., "Python programming", "Data Science")
Test Steps:

1. Navigate to the resource library 2. Enter a search query 3. Verify the search results Postconditions: * Search results are displayed Expected Result: Learners can search for specific materials and retrieve relevant results. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3** Test Case ID: RL003 Test Case Title: Verify that resources are categorized and tagged for easy navigation. Test Case Description: Ensure that resources in the library are organized by category and tagged for easy discovery. Test Suite: Resource Library Test Priority: Medium Preconditions: * Learner is logged in * Resource library is enabled Test Data: No test data needed Test Steps: 1. Navigate to the resource library 2. Browse through the categories 3. Verify the tags associated with each resource

Postconditions:

* Resources are categorized and tagged

Expected Result: Resources are organized by category and tagged for easy navigation.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: RL004

Test Case Title: Verify that learners can download resources for offline access.

Test Case Description: Ensure that learners can download resources from the library for offline access.

Test Suite: Resource Library

Test Priority: High

Preconditions:

* Learner is logged in

* Resource library is enabled

Test Data: Sample resource (e.g., e-book, article)

Test Steps:

1. Navigate to the resource library

2. Select a resource to download

3. Verify the download process

Postconditions:

* Resource is downloaded successfully

Expected Result: Learners can download resources for offline access.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: RL005
Test Case Title: Verify that instructors can recommend resources to learners.
Test Case Description: Ensure that instructors can recommend resources from the library to learners.
Test Suite: Resource Library
Test Priority: Medium
Preconditions:
* Instructor is logged in
* Resource library is enabled
Test Data: Sample resource (e.g., e-book, article)
Test Steps:
1. Navigate to the instructor dashboard
2. Select a learner to recommend a resource to
3. Recommend a resource
4. Verify the recommendation notification
Postconditions:
* Resource recommendation is sent to the learner
Expected Result: Instructors can recommend resources to learners.
Severity: Minor
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:

Here are the five test cases to cover the acceptance criteria for the Peer Review feature:

Test Case 1

Test Case ID: pr_001

Test Case Title: Verify that learners can submit assignments for peer review

Test Case Description: Ensure that learners can successfully submit their assignments for peer review.

Test Suite: Peer Review

Test Priority: High

Preconditions:

* Learner is logged in
* Assignment is created and available for submission
Test Data: Assignment details (title, description, file)
Test Steps:
1. Go to the assignment submission page
2. Click on the "Submit for Peer Review" button
3. Upload the assignment file
4. Click on the "Submit" button
Postconditions:
* Assignment is submitted for peer review
* Submission is visible in the learner's dashboard
Expected Result: The system allows learners to submit assignments for peer review successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2
Test Case ID: pr_002
Test Case Title: Verify that peers can provide feedback on submitted assignments

Test Case Description: Ensure that peers can provide constructive feedback on assignments submitted for peer

review.
Test Suite: Peer Review
Test Priority: High
Preconditions:
* Learner has submitted an assignment for peer review
* Peer is assigned to review the assignment
Test Data: Assignment details, peer feedback
Test Steps:
1. Go to the peer review dashboard
2. Select the assignment to review
3. Provide feedback on the assignment
4. Click on the "Submit Feedback" button
Postconditions:
* Feedback is submitted and visible to the assignment owner
* Feedback is anonymous to ensure unbiased feedback
Expected Result: The system allows peers to provide feedback on submitted assignments successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case 3

Test Case ID: pr_003
Test Case Title: Verify that a structured review process guides the feedback
Test Case Description: Ensure that the peer review process provides a structured guidance for reviewers to
provide feedback.
Test Suite: Peer Review
Test Priority: Medium
Preconditions:
* Learner has submitted an assignment for peer review
* Peer is assigned to review the assignment
Test Data: Assignment details, review guidelines
Test Steps:
1. Go to the peer review dashboard
2. Select the assignment to review
3. Review the review guidelines
4. Provide feedback on the assignment based on the guidelines
Postconditions:
* Feedback is submitted and visible to the assignment owner
* Feedback is guided by the structured review process
Expected Result: The system provides a structured review process to guide peers in providing feedback.

Severity: Minor

Type of Testing: Functional
Test Case Approach: Positive
Test Case 4
Test Case ID: pr_004
Test Case Title: Verify that learners can view feedback on their work
Test Case Description: Ensure that learners can view the feedback provided by peers on their assignments.
Test Suite: Peer Review
Test Priority: High
Preconditions:
* Learner has submitted an assignment for peer review
* Peer has provided feedback on the assignment
Test Data: Assignment details, peer feedback
Test Steps:
1. Go to the assignment submission page
2. Click on the "View Feedback" button
3. Review the feedback provided by peers
Postconditions:
* Feedback is visible to the learner
* Feedback is constructive and helpful

Expected Result: The system allows learners to view feedback on their assignments successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5
Test Case ID: pr_005
Test Case Title: Verify that reviewers remain anonymous to ensure unbiased feedback
Test Case Description: Ensure that the peer review process maintains the anonymity of reviewers to ensure
unbiased feedback.
Test Suite: Peer Review
Test Priority: High
Preconditions:
* Learner has submitted an assignment for peer review
* Peer has provided feedback on the assignment
Test Data: Assignment details, peer feedback
Test Steps:
1. Go to the assignment submission page
2. Click on the "View Feedback" button
3. Review the feedback provided by peers
Postconditions:

* Feedback is visible to the learner

* Reviewer's identity is not disclosed to the learner

Expected Result: The system maintains the anonymity of reviewers to ensure unbiased feedback.

Severity: Critical

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:

Here are the 5 detailed test cases for the "Certificates and Badges" feature:

Test Case 1: Verify that learners earn certificates upon course completion

Test Case ID: CEB001

Test Case Title: Verify that learners earn certificates upon course completion

Test Case Description: Ensure that learners receive certificates after completing a course.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

? Learner is enrolled in a course

? Course is fully completed by the learner

Test Data: No test data needed

Test Steps:

1. Log in as a learner

2. Complete a course

3. Verify the certificate is displayed in the learner's profile

Postconditions:

? Certificate is generated and available in the learner's profile

Expected Result: The learner earns a certificate upon completing a course.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that badges are awarded for specific achievements and milestones

Test Case ID: CEB002

Test Case Title: Verify that badges are awarded for specific achievements and milestones

Test Case Description: Ensure that learners receive badges for achieving specific milestones or completing

specific tasks.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:

? Learner is enrolled in a course with specific milestones

? Learner completes a milestone

Test Data: No test data needed

Test Steps:

1. Log in as a learner

2. Complete a milestone in a course

3. Verify the badge is displayed in the learner's profile

Postconditions:

? Badge is generated and available in the learner's profile

Expected Result: The learner earns a badge for completing a specific milestone.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that certificates and badges are accessible through learner profiles

Test Case ID: CEB003

Test Case Title: Verify that certificates and badges are accessible through learner profiles

Test Case Description: Ensure that learners can access their earned certificates and badges from their profiles.

Test Suite: Certificates and Badges

Test Priority: High

Preconditions:

? Learner has earned a certificate and/or badge

Test Data: No test data needed

Test Steps:

- 1. Log in as a learner
- 2. Access the learner profile
- 3. Verify the certificates and badges are displayed

Postconditions:

? Certificates and badges are accessible from the learner profile

Expected Result: Learners can access their earned certificates and badges from their profiles.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that learners can share their certificates and badges on social media

Test Case ID: CEB004

Test Case Title: Verify that learners can share their certificates and badges on social media

Test Case Description: Ensure that learners can share their earned certificates and badges on social media

platforms.

Test Suite: Certificates and Badges

Test Priority: Medium

Preconditions:
? Learner has earned a certificate and/or badge
Test Data: No test data needed
Test Steps:
1. Log in as a learner
2. Access the learner profile
3. Click on the share button for a certificate or badge
4. Verify the certificate or badge is shared on social media
Postconditions:
? Certificate or badge is shared on social media
Expected Result: Learners can share their earned certificates and badges on social media.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case Approach: Positive **Test Case 5: Verify that instructors can issue custom badges for special achievements**
Test Case 5: Verify that instructors can issue custom badges for special achievements
Test Case 5: Verify that instructors can issue custom badges for special achievements Test Case ID: CEB005
Test Case 5: Verify that instructors can issue custom badges for special achievements Test Case ID: CEB005 Test Case Title: Verify that instructors can issue custom badges for special achievements
Test Case 5: Verify that instructors can issue custom badges for special achievements Test Case ID: CEB005 Test Case Title: Verify that instructors can issue custom badges for special achievements Test Case Description: Ensure that instructors can create and issue custom badges for learners who achieve
Test Case 5: Verify that instructors can issue custom badges for special achievements Test Case ID: CEB005 Test Case Title: Verify that instructors can issue custom badges for special achievements Test Case Description: Ensure that instructors can create and issue custom badges for learners who achieve special milestones or complete specific tasks.
Test Case 5: Verify that instructors can issue custom badges for special achievements Test Case ID: CEB005 Test Case Title: Verify that instructors can issue custom badges for special achievements Test Case Description: Ensure that instructors can create and issue custom badges for learners who achieve special milestones or complete specific tasks. Test Suite: Certificates and Badges
Test Case 5: Verify that instructors can issue custom badges for special achievements Test Case ID: CEB005 Test Case Title: Verify that instructors can issue custom badges for special achievements Test Case Description: Ensure that instructors can create and issue custom badges for learners who achieve special milestones or complete specific tasks. Test Suite: Certificates and Badges Test Priority: Medium

Test Data: No test data needed

Test Steps:

1. Log in as an instructor

2. Create a custom badge for a course

3. Assign the custom badge to a learner

4. Verify the custom badge is displayed in the learner's profile

Postconditions:

? Custom badge is generated and available in the learner's profile

Expected Result: Instructors can create and issue custom badges for special achievements.

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:

Here are the 5 test cases to cover all acceptance criteria:

Test Case 1:

Test Case ID: LGPN1

Test Case Title: Verify that learners earn points for completing activities

Test Case Description: Ensure that learners receive points for completing various activities on the platform,

such as watching videos, attending lectures, or submitting assignments.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

* Learner is logged in

* Activity is available for completion

Test Data: No test data needed

Test Steps:

1. Log in as a learner

2. Complete an activity (e.g., watch a video)

3. Verify points are awarded and displayed in the gamification dashboard

Postconditions:

* Points are reflected in the learner's profile

Expected Result: The learner earns points for completing the activity, and the points are correctly displayed in

the gamification dashboard.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: LGPN2

Test Case Title: Verify that levels are achieved based on accumulated points

Test Case Description: Ensure that learners can progress through levels as they accumulate points, with each

level unlocking new features or privileges.

Test Suite: Gamification Elements

Test Priority: High

Preconditions:

* Learner has accumulated points

* Level structure is configured

Test Data: Point thresholds for each level

Test Steps:

1. Log in as a learner

2. Accumulate points to reach the next level

3. Verify level upgrade and corresponding privileges

Postconditions:

* Level is updated in the learner's profile

Expected Result: The learner progresses to the next level as they accumulate points, with corresponding

privileges and features unlocked.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: LGPN3

Test Case Title: Verify that leaderboards display top-performing learners

Test Case Description: Ensure that leaderboards are updated in real-time, displaying top-performing learners

based on their accumulated points and progress.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

* Leaderboard is configured

* Multiple learners have accumulated points

Test Data: No test data needed

Test Steps:

1. Log in as a learner

2. Verify leaderboard displays top-performing learners

3. Perform an action to earn points and verify leaderboard update

Postconditions:

* Leaderboard is updated in real-time

Expected Result: The leaderboard accurately displays top-performing learners, updating in real-time as

learners earn points and progress.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: LGPN4

Test Case Title: Verify that badges are awarded for reaching milestones

Test Case Description: Ensure that badges are awarded to learners when they reach specific milestones, such

as completing a course or achieving a certain level.

Test Suite: Gamification Elements

Test Priority: Medium

Preconditions:

* Milestone structure is configured

* Learner has reached a milestone

Test Data: Badge images and descriptions

Test Steps:

1. Log in as a learner

2. Reach a milestone (e.g., complete a course)

3. Verify badge is awarded and displayed in the gamification dashboard

Postconditions:

* Badge is reflected in the learner's profile

Expected Result: The learner is awarded a badge when reaching a milestone, and the badge is correctly

displayed in the gamification dashboard.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: LGPN5

Test Case Title: Verify that learners can view their progress in a gamification dashboard

Test Case Description: Ensure that learners can access a gamification dashboard to view their progress,

including points, levels, badges, and leaderboards.

Test Suite: Gamification Elements
Test Priority: High
Preconditions:
* Learner is logged in
* Gamification dashboard is configured
Test Data: No test data needed
Test Steps:
1. Log in as a learner
2. Access the gamification dashboard
3. Verify points, levels, badges, and leaderboards are displayed correctly
Postconditions:
* Learner can access their gamification dashboard
Expected Result: The learner can view their progress in the gamification dashboard, including points, levels,
badges, and leaderboards.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Data - 61
Project name -
Online Learning Platform
Project description -
Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and
topics. The platform should provide interactive features like video lectures, downloadable resources,
discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Multilingual Support

Feature description -

The platform offers robust multilingual support, catering to learners from diverse linguistic backgrounds and

facilitating access to educational resources in their preferred language. Through multilingual interfaces, course

materials, and support services, the platform ensures inclusivity, accessibility, and cultural relevance for all

learners.

Userstory title -

As a learner, I want multilingual support so that I can access the platform in my preferred language.

Acceptance criteria -

['The platform interface is available in multiple languages.', 'Course materials are available in different

languages.', 'Learners can select their preferred language at any time.', 'Multilingual support includes translated

help and support resources.', 'Instructors can provide content in multiple languages.']

Generated Test Cases:

Here are the 5 test cases generated based on the provided input data:

Test Case 1: Verify that the platform interface is available in multiple languages

Test Case ID: ML001

Test Case Title: Verify that the platform interface is available in multiple languages

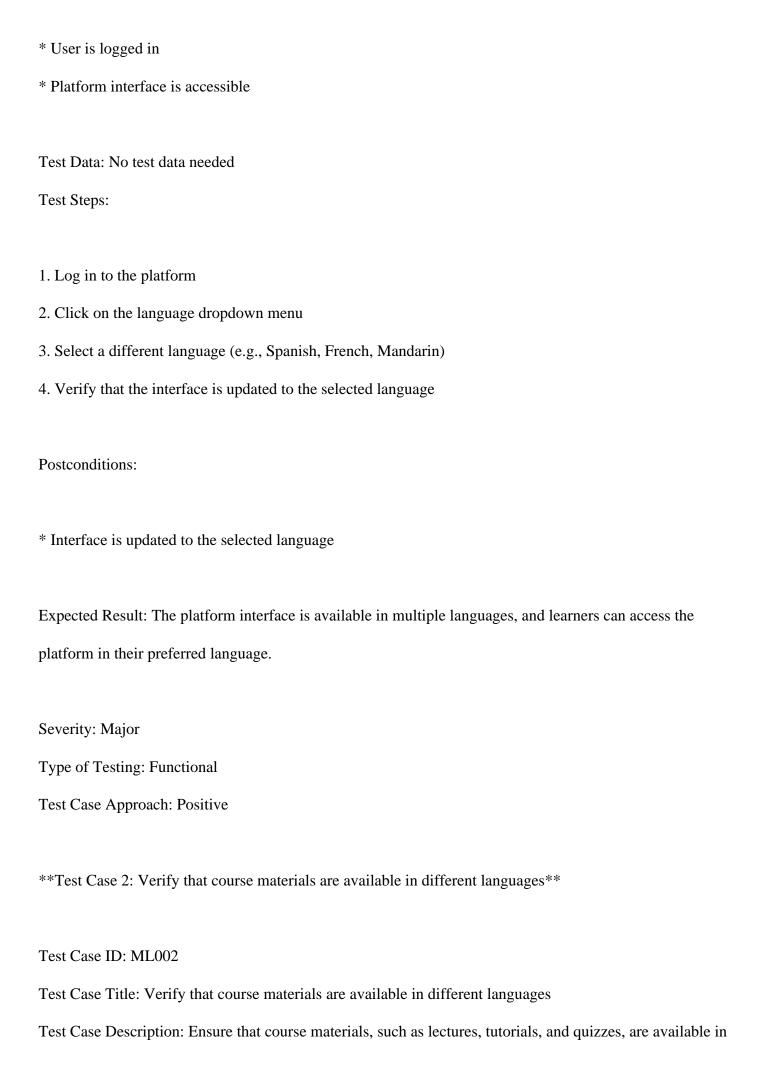
Test Case Description: Ensure that the platform interface supports multiple languages, allowing learners to

access the platform in their preferred language.

Test Suite: Multilingual Support

Test Priority: High

Preconditions:



different languages, catering to learners with diverse linguistic backgrounds.
Test Suite: Multilingual Support
Test Priority: High
Preconditions:
* Course is created with multilingual support
* User is enrolled in the course
Test Data: Course materials in different languages (e.g., English, Arabic, Portuguese)
Test Steps:
1. Enroll in a course with multilingual support
2. Access course materials (e.g., lecture, tutorial, quiz)
3. Verify that course materials are available in different languages
Postconditions:
* Course materials are updated to the selected language
Expected Result: Course materials are available in different languages, allowing learners to access educational
resources in their preferred language.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that learners can select their preferred language at any time

Test Case ID: ML003
Test Case Title: Verify that learners can select their preferred language at any time
Test Case Description: Ensure that learners can change their preferred language at any point, without affecting
their progress or access to course materials.
Test Suite: Multilingual Support
Test Priority: Medium
Preconditions:
* User is logged in
* Platform interface is accessible
Test Data: No test data needed
Test Steps:
1. Log in to the platform
2. Change the preferred language (e.g., from English to Spanish)
3. Verify that the platform interface and course materials are updated to the selected language
Postconditions:
* Preferred language is updated
Expected Result: Learners can select their preferred language at any time, and the platform updates
accordingly.
Severity: Minor

Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that multilingual support includes translated help and support resources
Test Case ID: ML004
Test Case Title: Verify that multilingual support includes translated help and support resources
Test Case Description: Ensure that help and support resources, such as FAQs, tutorials, and customer support,
are translated and available in multiple languages.
Test Suite: Multilingual Support
Test Priority: Medium
Preconditions:
* User is logged in
* Help and support resources are accessible
Test Data: Translated help and support resources in different languages (e.g., English, French, Mandarin)
Test Steps:
1. Access help and support resources (e.g., FAQs, tutorials, customer support)
2. Verify that resources are translated and available in multiple languages
Postconditions:
* Translated resources are available
Expected Result: Multilingual support includes translated help and support resources, providing learners with

access to linguistic support.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that instructors can provide content in multiple languages
Test Case ID: ML005
Test Case Title: Verify that instructors can provide content in multiple languages
Test Case Description: Ensure that instructors can create and upload course materials in multiple languages,
allowing them to cater to learners with diverse linguistic backgrounds.
Test Suite: Multilingual Support
Test Priority: High
Preconditions:
* Instructor is logged in
* Course creation functionality is accessible
Test Data: Course materials in different languages (e.g., English, Arabic, Portuguese)
Test Steps:
1. Log in as an instructor
2. Create a new course with multilingual support
3. Upload course materials in different languages
4. Verify that course materials are saved and accessible in multiple languages

Postconditions:

* Course materials are updated to the selected language

Expected Result: Instructors can create and upload course materials in multiple languages, allowing learners to access educational resources in their preferred language.

Severity: Major

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:

Here are the five detailed test cases for the Online Learning Platform's Accessibility Features:

Test Case 1: Verify that the platform is compatible with screen readers

Test Case ID: OLPTC001

Test Case Title: Verify that the platform is compatible with screen readers

Test Case Description: Ensure that the platform's content and navigation are accessible through screen readers,

facilitating equal access for learners with visual impairments.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

? The platform is installed with a screen reader software.

? A sample course is available with interactive content.

Test Data: No test data needed

Test Steps:

1. Open the platform with a screen reader enabled.

2. Navigate to a sample course with interactive content.

3. Verify that the screen reader correctly reads out the course content, including text, images, and interactive

elements.

Postconditions:

? The platform's content is accessible through the screen reader.

Expected Result: The platform is compatible with screen readers, enabling learners with visual impairments to

access course content.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that learners can adjust font sizes for better readability

Test Case ID: OLPTC002

Test Case Title: Verify that learners can adjust font sizes for better readability

Test Case Description: Ensure that learners can adjust font sizes to improve readability, accommodating

learners with visual impairments or preferences.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

? The platform is accessible with a compatible browser.

? A sample course is available with text-based content.

Test Data: No test data needed

Test Steps:

- 1. Log in to the platform as a learner.
- 2. Navigate to a sample course with text-based content.
- 3. Adjust the font size using the platform's font size adjustment feature.
- 4. Verify that the font size changes accordingly, improving readability.

Postconditions:

? The learner can adjust font sizes to improve readability.

Expected Result: The platform allows learners to adjust font sizes for better readability, promoting inclusivity

Severity: Minor

and comfort.

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that color contrast options are available for improved visibility

Test Case ID: OLPTC003

Test Case Title: Verify that color contrast options are available for improved visibility

Test Case Description: Ensure that the platform offers color contrast options, enabling learners with visual

impairments or sensitivities to customize their learning experience.

Test Suite: Accessibility Features

Test Priority: Medium

Preconditions:

? The platform is accessible with a compatible browser.

? A sample course is available with color-dependent content.

Test Data: No test data needed

Test Steps:

- 1. Log in to the platform as a learner.
- 2. Navigate to a sample course with color-dependent content.
- 3. Access the color contrast options feature.
- 4. Verify that the color contrast options are available and adaptable to individual preferences.

Postconditions:

? The learner can customize color contrast options to improve visibility.

Expected Result: The platform provides color contrast options, enhancing the learning experience for learners with visual impairments or sensitivities.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that accessibility features are easy to activate and customize

Test Case ID: OLPTC004

Test Case Title: Verify that accessibility features are easy to activate and customize

Test Case Description: Ensure that learners can easily activate and customize accessibility features, promoting

a seamless learning experience.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

? The platform is accessible with a compatible browser.

? A sample course is available with interactive content.

Test Data: No test data needed

Test Steps:

- 1. Log in to the platform as a learner.
- 2. Navigate to the accessibility features settings.
- 3. Activate and customize accessibility features (e.g., font size, color contrast, screen reader).
- 4. Verify that the customized settings are saved and applied to the learning experience.

Postconditions:

? The learner can easily activate and customize accessibility features.

Expected Result: The platform provides an intuitive and user-friendly experience for learners to activate and

customize accessibility features.

Severity: Major

Type of Testing: Usability

Test Case Approach: Positive

Test Case 5: Verify that support resources are available to assist learners with disabilities

Test Case ID: OLPTC005

Test Case Title: Verify that support resources are available to assist learners with disabilities

Test Case Description: Ensure that the platform provides accessible support resources, enabling learners with

disabilities to receive timely assistance and support.

Test Suite: Accessibility Features

Test Priority: High

Preconditions:

? The platform is accessible with a compatible browser.

? A sample course is available with interactive content.

Test Data: No test data needed

Test Steps:

1. Log in to the platform as a learner.

2. Navigate to the support resources section.

3. Verify that accessible support resources (e.g., tutorials, FAQs, contact forms) are available and easily

accessible.

4. Verify that the support resources provide clear instructions and appropriate support for learners with

disabilities.

Postconditions:

? Support resources are accessible and available to learners with disabilities.

Expected Result: The platform provides accessible support resources, ensuring that learners with disabilities

receive timely assistance and support.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 63

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Discussion Moderation

Feature description -

Instructors actively moderate discussion forums to maintain a conducive learning environment, ensure respectful discourse, and enforce community guidelines. By facilitating constructive interactions and discouraging disruptive behavior, discussion moderation fosters a positive and collaborative learning atmosphere for all participants.

Userstory title -

As an instructor, I want to moderate discussions so that I can maintain a positive and respectful learning environment.

Acceptance criteria -

['Instructors can edit, delete, or highlight posts in discussions.', 'Instructors can mute or ban disruptive users.', 'Moderation actions are logged and reported.', 'Community guidelines are visible to all participants.', 'Notifications are sent for moderated actions.']

Generated Test Cases:

Here are the 5 detailed test cases for the Discussion Moderation feature:
Test Case 1: Edit Posts
Test Case ID: dscrmn1
Test Case Title: Verify that instructors can edit posts in discussions
Test Case Description: Ensure that instructors can edit posts in discussions to maintain a positive and
respectful learning environment.
Test Suite: Discussion Moderation
Test Priority: High
Preconditions:
- Instructor is logged in
- Discussion forum is created
- A post is made by a participant
Test Data: No test data needed
Test Steps:
1. Login as an instructor
2. Go to the discussion forum
3. Select a post to edit
4. Click on the edit button
5. Make changes to the post
6. Click on save changes
Postconditions:
- Post is updated
Expected Result: The system allows instructors to edit posts in discussions.
Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 2: Mute Disruptive Users

Test Case ID: dscrmn2

Test Case Title: Verify that instructors can mute or ban disruptive users

Test Case Description: Ensure that instructors can mute or ban users who violate community guidelines to

maintain a positive and respectful learning environment.

Test Suite: Discussion Moderation

Test Priority: High

Preconditions:

- Instructor is logged in

- Discussion forum is created

- A user makes a disruptive post

Test Data: No test data needed

Test Steps:

1. Login as an instructor

2. Go to the discussion forum

3. Select a user to mute or ban

4. Click on the mute or ban button

5. Confirm the action

Postconditions:

- User is muted or banned

Expected Result: The system allows instructors to mute or ban disruptive users.

Severity: Critical

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3: Moderation Action Logging

Test Case ID: dscrmn3

Test Case Title: Verify that moderation actions are logged and reported

Test Case Description: Ensure that all moderation actions are logged and reported to maintain transparency

and accountability.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Discussion forum is created
- A moderation action is performed

Test Data: No test data needed

Test Steps:

- 1. Login as an instructor
- 2. Perform a moderation action (e.g., edit, delete, mute)
- 3. Go to the moderation log
- 4. Verify that the action is logged

Postconditions:

- Moderation action is logged

Expected Result: The system logs and reports all moderation actions.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

^{**}Test Case 4: Community Guidelines Visibility**

Test Case ID: dscrmn4

Test Case Title: Verify that community guidelines are visible to all participants

Test Case Description: Ensure that community guidelines are easily accessible and visible to all participants to

promote a positive and respectful learning environment.

Test Suite: Discussion Moderation

Test Priority: Medium

Preconditions:

- Discussion forum is created

Test Data: No test data needed

Test Steps:

1. Go to the discussion forum

2. Verify that community guidelines are displayed

Postconditions:

- Community guidelines are visible

Expected Result: The system displays community guidelines to all participants.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5: Notification for Moderation Actions

Test Case ID: dscrmn5

Test Case Title: Verify that notifications are sent for moderated actions

Test Case Description: Ensure that participants receive notifications when their posts are moderated to

maintain transparency and accountability.

Test Suite: Discussion Moderation

Preconditions:
- Instructor is logged in
- Discussion forum is created
- A participant makes a post
Test Data: No test data needed
Test Steps:
1. Login as an instructor
2. Go to the discussion forum
3. Moderate a post (e.g., edit, delete)
4. Verify that the participant receives a notification
Postconditions:
- Notification is sent
Expected Result: The system sends notifications to participants when their posts are moderated.
Severity: Minor
Type of Testing: Functional
Test Case Behaviour: Positive
Data - 64
Project name -
Online Learning Platform
Project description -
Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and
topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Test Priority: Medium

Feature name -

Learning Analytics

Feature description -

Instructors leverage robust learning analytics dashboards to gain insights into learner engagement,

performance trends, and course effectiveness. By analyzing data on learner interactions, progress, and

outcomes, instructors can make informed decisions, personalize instruction, and optimize the learning

experience to meet learners' needs effectively.

Userstory title -

As an instructor, I want access to learning analytics so that I can understand learner engagement and

performance.

Acceptance criteria -

['Learning analytics dashboards display engagement metrics.', 'Performance trends are visualized over time.',

'Instructors can access detailed reports on learner progress.', 'Analytics data is updated in real-time.',

'Instructors can export analytics reports for further analysis.']

Generated Test Cases:

Here are the 5 detailed test cases for the given input data:

Test Case 1:

Test Case ID: LA_001

Test Case Title: Verify that learning analytics dashboards display engagement metrics

Test Case Description: Ensure that the learning analytics dashboard displays accurate and up-to-date

engagement metrics for instructors to analyze learner engagement.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in
- Learning analytics dashboard is accessible

Test Data: No test data needed

Test Steps:

- 1. Log in as an instructor
- 2. Access the learning analytics dashboard
- 3. Verify that engagement metrics (e.g., course completion rate, time spent on course materials) are displayed

Postconditions:

- Engagement metrics are displayed correctly

Expected Result: The learning analytics dashboard displays accurate and up-to-date engagement metrics for instructors to analyze learner engagement.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: LA_002

Test Case Title: Verify that performance trends are visualized over time

Test Case Description: Ensure that the learning analytics dashboard visualizes performance trends over time,

enabling instructors to identify patterns and trends in learner performance.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

- Instructor is logged in
- Learning analytics dashboard is accessible

Test Data: No test data needed

Test Steps:

1. Log in as an instructor

2. Access the learning analytics dashboard

3. Verify that performance trends (e.g., average score over time, learner progress) are visualized correctly

Postconditions:

- Performance trends are visualized correctly

Expected Result: The learning analytics dashboard visualizes performance trends over time, enabling

instructors to identify patterns and trends in learner performance.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: LA_003

Test Case Title: Verify that instructors can access detailed reports on learner progress

Test Case Description: Ensure that instructors can access detailed reports on learner progress, including

individual learner performance and course completion rates.

Test Suite: Learning Analytics

Test Priority: High

Preconditions:

- Instructor is logged in

- Learning analytics dashboard is accessible

Test Data: No test data needed

Test Steps:

1. Log in as an instructor

2. Access the learning analytics dashboard

3. Verify that detailed reports on learner progress are accessible and accurate Postconditions: - Detailed reports are accessible and accurate Expected Result: Instructors can access detailed reports on learner progress, including individual learner performance and course completion rates. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 4:** Test Case ID: LA_004 Test Case Title: Verify that analytics data is updated in real-time Test Case Description: Ensure that analytics data is updated in real-time, reflecting the latest learner interactions and progress. Test Suite: Learning Analytics Test Priority: High Preconditions: - Instructor is logged in - Learning analytics dashboard is accessible Test Data: No test data needed Test Steps:

- 1. Log in as an instructor
- 2. Access the learning analytics dashboard
- 3. Perform an action that updates learner progress (e.g., complete a quiz)
- 4. Verify that analytics data is updated in real-time

Postconditions:

- Analytics data is updated in real-time

Expected Result: Analytics data is updated in real-time, reflecting the latest learner interactions and progress.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: LA 005

Test Case Title: Verify that instructors can export analytics reports for further analysis

Test Case Description: Ensure that instructors can export analytics reports for further analysis, enabling them

to gain deeper insights into learner performance and progress.

Test Suite: Learning Analytics

Test Priority: Medium

Preconditions:

- Instructor is logged in

- Learning analytics dashboard is accessible

Test Data: No test data needed

Test Steps:

- 1. Log in as an instructor
- 2. Access the learning analytics dashboard
- 3. Verify that analytics reports can be exported in a compatible format (e.g., CSV, PDF)
- 4. Verify that exported reports contain accurate analytics data

Postconditions:

- Analytics reports are exported correctly

Expected Result: Instructors can export analytics reports for further analysis, enabling them to gain deeper insights into learner performance and progress.

Severity: Major

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:

Here are the 5 test cases for the given input data:

Test Case 1: Verify that learners can create and join groups for projects

Test Case ID: OLPSocial1

Test Case Title: Verify that learners can create and join groups for projects

Test Case Description: Ensure that the system allows learners to create and join groups for projects,

facilitating collaboration and social learning.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in

- No existing group with the same name

Test Data: No test data needed

Test Steps:

1. Click on the "Create Group" button

2. Enter group details (name, description, etc.)

3. Click "Create" to create the group

4. Click on the "Join Group" button for an existing group

5. Enter the group code or select from the available groups

Test Postconditions:

- Group is created successfully

- Learner is added to the group

Expected Result: The system allows learners to create and join groups for projects successfully.

Severity: Major

Type of Testing: Functional

Test Case 2: Verify that collaboration tools are available for group communication

Test Case ID: OLPSocial2

Test Case Title: Verify that collaboration tools are available for group communication

Test Case Description: Ensure that the system provides collaboration tools for group communication, such as

messaging, file sharing, and discussion boards.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Learner is logged in

- Learner is part of a group

Test Data: No test data needed

Test Steps:

1. Go to the group dashboard

2. Click on the "Collaboration Tools" button

3. Verify the availability of messaging, file sharing, and discussion boards

4. Send a message to a group member

5. Share a file with the group

Test Postconditions:

- Collaboration tools are available

- Messages and files are shared successfully

Expected Result: The system provides collaboration tools for group communication, enabling effective

collaboration among learners.

Severity: Major

Type of Testing: Functional

Test Case 3: Verify that learners can share resources and feedback within groups

Test Case ID: OLPSocial3

Test Case Title: Verify that learners can share resources and feedback within groups

Test Case Description: Ensure that the system allows learners to share resources and feedback within groups,

promoting knowledge exchange and peer-to-peer support.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner is logged in

- Learner is part of a group

Test Data: No test data needed

Test Steps:

1. Go to the group dashboard

- 2. Click on the "Share Resource" button
- 3. Upload a resource (file, link, etc.)
- 4. Add feedback to a shared resource
- 5. Verify that the resource and feedback are visible to group members

Test Postconditions:

- Resource is shared successfully
- Feedback is visible to group members

Expected Result: The system allows learners to share resources and feedback within groups, facilitating

knowledge exchange and collaboration.

Severity: Major

Type of Testing: Functional

Test Case 4: Verify that instructors can assign group projects and monitor progress

Test Case ID: OLPSocial4

Test Case Title: Verify that instructors can assign group projects and monitor progress

Test Case Description: Ensure that the system allows instructors to assign group projects and monitor learner

progress, enabling effective project management and feedback.

Test Suite: Social Learning Features

Test Priority: High

Preconditions:

- Instructor is logged in

- Course is created with group project assignment

Test Data: No test data needed

Test Steps:

1. Go to the course dashboard

2. Create a new group project assignment

3. Assign learners to the project

4. Monitor learner progress and feedback

5. Verify that instructor can view project submissions and feedback

Test Postconditions:

- Group project assignment is created successfully

- Instructor can monitor learner progress and feedback

Expected Result: The system allows instructors to assign group projects and monitor learner progress,

enabling effective project management and feedback.

Severity: Critical

Type of Testing: Functional

Test Case 5: Verify that social sharing features allow learners to share achievements

Test Case ID: OLPSocial5

Test Case Title: Verify that social sharing features allow learners to share achievements

Test Case Description: Ensure that the system provides social sharing features, enabling learners to share their

achievements on social media platforms.

Test Suite: Social Learning Features

Test Priority: Medium

Preconditions:

- Learner is logged in

- Learner has completed a course or achieved a milestone

Test Data: No test data needed

Test Steps:

1. Go to the learner dashboard

2. Click on the "Share Achievement" button

3. Select a social media platform (e.g., Facebook, Twitter, LinkedIn)

4. Verify that the achievement is shared successfully on the selected platform

5. Verify that the shared post includes relevant details (course name, achievement, etc.)

Test Postconditions:

- Achievement is shared successfully on the selected platform

Expected Result: The system provides social sharing features, allowing learners to share their achievements on

social media platforms.

Severity: Minor

Type of Testing: Functional

Data - 66

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Authoring Tools

Feature description -

Instructors utilize powerful content authoring tools to create and publish course content, including multimedia presentations, interactive modules, and rich media resources. These authoring tools empower instructors to design engaging and interactive learning materials that cater to diverse learning styles and preferences, enhancing the effectiveness and impact of instruction.

Userstory title -

As an instructor, I want powerful content authoring tools so that I can create engaging and interactive course materials.

Acceptance criteria -

['Instructors can create multimedia presentations.', 'Interactive modules can be developed and integrated.', 'Rich media resources, such as videos and animations, are supported.', 'Content authoring tools are user-friendly and intuitive.', 'Instructors can preview and edit content before publishing.']

Generated Test Cases:

Here are the 5 test cases for the provided input data:

Test Case 1:

Test Case ID: OLP_CT_001

Test Case Title: Verify that instructors can create multimedia presentations.

Test Case Description: Ensure that the content authoring tools allow instructors to create multimedia

presentations with various media types, such as images, audio, and video.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

* Instructor is logged in

* Course is created

* Multimedia content is available

Test Data: Sample multimedia content (image, audio, video)

Test Steps:

- 1. Log in as an instructor
- 2. Create a new course
- 3. Select the multimedia presentation module
- 4. Upload sample multimedia content
- 5. Save and preview the presentation

Postconditions:

* Multimedia presentation is created successfully

Expected Result: The system allows instructors to create multimedia presentations with various media types.

Severity: Major

Type of Testing: Functional

Test Case 2:

Test Case ID: OLP_CT_002

Test Case Title: Verify that interactive modules can be developed and integrated.

Test Case Description: Ensure that the content authoring tools allow instructors to develop and integrate

interactive modules, such as quizzes, games, and simulations.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

* Instructor is logged in

* Course is created

* Interactive module templates are available

Test Data: Sample interactive module templates

Test Steps:

1. Log in as an instructor

2. Create a new course

3. Select the interactive module template

4. Customize and develop the interactive module

5. Save and integrate the module into the course

Postconditions:

* Interactive module is developed and integrated successfully

Expected Result: The system allows instructors to develop and integrate interactive modules.

Severity: Major

Type of Testing: Functional

Test Case 3:

Test Case ID: OLP_CT_003

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 the integration of rich media resources,

such as videos and animations, to enhance the learning experience.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

* Instructor is logged in

* Course is created

* Rich media resources are available

Test Data: Sample rich media resources (video, animation)

Test Steps:

- 1. Log in as an instructor
- 2. Create a new course
- 3. Select the rich media resource module
- 4. Upload and integrate the rich media resource
- 5. Save and preview the course content

Postconditions:

* Rich media resource is integrated successfully

Expected Result: The system supports the integration of rich media resources.

Severity: Minor

Type of Testing: Functional

Test Case 4:

Test Case ID: OLP_CT_004

Test Case Title: Verify that content authoring tools are user-friendly and intuitive.

Test Case Description: Ensure that the content authoring tools are user-friendly and intuitive, allowing

instructors to easily create and publish course content.

Test Suite: Content Authoring Tools

Test Priority: Medium

Preconditions:

* Instructor is logged in

* Course is created

* Sample course content is available

Test Data: No test data needed

Test Steps:

1. Log in as an instructor

2. Create a new course

3. Use the content authoring tools to create and publish course content

4. Evaluate the usability and intuitiveness of the tools

Postconditions:

* Content is created and published successfully

Expected Result: The system provides user-friendly and intuitive content authoring tools.

Severity: Minor

Type of Testing: Usability

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLP_CT_005

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 and edit course content before publishing, ensuring accuracy and quality.

Test Suite: Content Authoring Tools

Test Priority: High

Preconditions:

* Instructor is logged in

* Course is created

* Sample course content is available

Test Data: No test data needed

Test Steps:

- 1. Log in as an instructor
- 2. Create a new course
- 3. Create and preview course content
- 4. Edit the course content
- 5. Save and publish the course content

Postconditions:

* Course content is edited and published successfully

Expected Result: The system allows instructors to preview and edit course content before publishing.

Severity: Major

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:

Here are the 5 test cases for the Online Learning Platform project, covering the acceptance criteria for the Learning Paths and Curriculum feature:

Test Case 1: Verify that learning paths include sequential courses and modules

Test Case ID: LP-CR-001

Test Case Title: Verify that learning paths include sequential courses and modules

Test Case Description: This test case ensures that the learning paths in the Online Learning Platform include

sequential courses and modules, allowing learners to follow a structured approach to achieve their learning

objectives.

Test Suite: Learning Paths and Curriculum

Test Priority: High

Preconditions:

* Learner is logged in

* Learning path is created with multiple courses and modules

Test Data: No test data needed

Test Steps:

- 1. Log in as a learner
- 2. Create a new learning path with multiple courses and modules
- 3. Verify that the courses and modules are displayed in a sequential order
- 4. Check that each course and module has a clear description and learning objectives

Postconditions:

* Learning path is created with sequential courses and modules

Expected Result: The system displays a structured learning path with sequential courses and modules,

enabling learners to follow a clear learning trajectory.

Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that curricula are aligned with specific learning objectives
Test Case ID: LP-CR-002
Test Case Title: Verify that curricula are aligned with specific learning objectives
Test Case Description: This test case ensures that the curricula in the Online Learning Platform are aligned
with specific learning objectives, providing learners with a clear direction and focus for their learning.
Test Suite: Learning Paths and Curriculum
Test Priority: High
Preconditions:
* Learner is logged in * Curriculum is created with specific learning objectives
Test Data: Curriculum data with learning objectives
Test Steps:
1. Log in as a learner
2. Create a new curriculum with specific learning objectives
3. Verify that the learning objectives are clearly stated and aligned with the curriculum
4. Check that the curriculum is structured to achieve the specified learning objectives
Postconditions:

* Curriculum is created with aligned learning objectives

Expected Result: The system displays a curriculum that is aligned with specific learning objectives, providing

learners with a clear direction for their learning.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that learners can track their progress within a learning path

Test Case ID: LP-CR-003

Test Case Title: Verify that learners can track their progress within a learning path

Test Case Description: This test case ensures that learners can track their progress within a learning path,

enabling them to monitor their learning journey and stay motivated.

Test Suite: Learning Paths and Curriculum

Test Priority: Medium

Preconditions:

* Learner is logged in

* Learning path is created with multiple courses and modules

Test Data: Learner's progress data

Test Steps:

1. Log in as a learner

2. Enroll in a learning path with multiple courses and modules

3. Complete some courses and modules

4. Verify that the learner's progress is updated and displayed within the learning path

Postconditions:
* Learner's progress is updated and displayed
Expected Result: The system accurately tracks and displays the learner's progress within a learning path,
enabling them to monitor their learning journey.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that instructors can customize learning paths for different needs
Test Case ID: LP-CR-004
Test Case Title: Verify that instructors can customize learning paths for different needs
Test Case Description: This test case ensures that instructors can customize learning paths to cater to different
learner needs, providing personalized learning experiences.
Test Suite: Learning Paths and Curriculum
Test Priority: Medium
Preconditions:
* Instructor is logged in
* Learning path is created with multiple courses and modules
Test Data: Instructor's customization data
Test Steps:
1. Log in as an instructor

2. Create a new learning path with multiple courses and modules 3. Customize the learning path for different learner needs (e.g., language, skill level) 4. Verify that the customized learning path is saved and displayed correctly Postconditions: * Customized learning path is saved and displayed Expected Result: The system allows instructors to customize learning paths to cater to different learner needs, providing personalized learning experiences. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 5: Verify that notifications are sent for upcoming courses or milestones** Test Case ID: LP-CR-005 Test Case Title: Verify that notifications are sent for upcoming courses or milestones Test Case Description: This test case ensures that the Online Learning Platform sends notifications to learners for upcoming courses or milestones, keeping them informed and engaged in their learning journey. Test Suite: Learning Paths and Curriculum Test Priority: Low Preconditions: * Learner is logged in

Test Data: Notification settings

* Learning path is created with upcoming courses or milestones

Test Steps:
1. Log in as a learner
2. Enroll in a learning path with upcoming courses or milestones
3. Verify that notifications are sent to the learner for upcoming courses or milestones
4. Check that the notifications contain relevant information (e.g., course details, deadline)
Postconditions:
* Notifications are sent to the learner
Expected Result: The system sends notifications to learners for upcoming courses or milestones, keeping them
informed and engaged in their learning journey.
Severity: Trivial
Type of Testing: Functional
Test Case Approach: Positive
Data - 68

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Attendance Tracking

Feature description -

Instructors track learner attendance in live classes and webinars to monitor participation, engagement, and

accountability. By recording attendance data and participation metrics, instructors can assess learner

engagement, identify areas for improvement, and provide targeted support and intervention as needed.

Userstory title -

As an instructor, I want to track learner attendance so that I can monitor participation and engagement.

Acceptance criteria -

['Attendance is automatically recorded during live sessions.', 'Instructors can view attendance reports for each

session.', 'Learners can see their attendance records.', 'Attendance data is integrated with participation

metrics.', 'Notifications are sent for missed sessions.']

Generated Test Cases:

Here are the 5 detailed test cases for the Attendance Tracking feature:

Test Case 1: Verify that attendance is automatically recorded during live sessions

Test Case ID: AT01

Test Case Title: Verify that attendance is automatically recorded during live sessions

Test Case Description: Ensure that the system automatically records attendance during live sessions without

requiring manual intervention.

Test Suite: Attendance Tracking

Test Priority: High

Preconditions:

* Instructor has created a live session

* Learners are registered for the live session

Test Data: No test data needed

Test Steps:

1. Instructor starts the live session 2. Learners join the live session 3. System records attendance automatically Postconditions: * Attendance data is updated in the system Expected Result: Attendance is recorded accurately and in real-time during the live session. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 2: Verify that instructors can view attendance reports for each session** Test Case ID: AT02 Test Case Title: Verify that instructors can view attendance reports for each session Test Case Description: Ensure that instructors can access and view attendance reports for each live session, including learner attendance status and participation metrics. Test Suite: Attendance Tracking Test Priority: High Preconditions: * Live session has ended * Attendance data is available Test Data: Sample attendance report Test Steps: 1. Instructor logs in to the platform

3. Instructor reviews the attendance report

2. Instructor accesses the attendance report for a specific live session

Postconditions:

* Attendance report is displayed accurately

Expected Result: Instructors can view attendance reports for each live session, including learner attendance

status and participation metrics.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that learners can see their attendance records

Test Case ID: AT03

Test Case Title: Verify that learners can see their attendance records

Test Case Description: Ensure that learners can access and view their own attendance records, including a

history of their attendance in live sessions.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

* Learner is registered for a live session

* Attendance data is available

Test Data: Sample attendance record

Test Steps:

1. Learner logs in to the platform

2. Learner accesses their attendance record

3. Learner reviews their attendance history

Postconditions:

* Attendance record is displayed accurately

Expected Result: Learners can view their attendance records, including a history of their attendance in live

sessions.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that attendance data is integrated with participation metrics

Test Case ID: AT04

Test Case Title: Verify that attendance data is integrated with participation metrics

Test Case Description: Ensure that attendance data is combined with participation metrics, such as quiz scores,

discussion forum engagement, and assignment submissions, to provide a comprehensive view of learner

engagement.

Test Suite: Attendance Tracking

Test Priority: Medium

Preconditions:

* Attendance data is available

* Participation metrics are available

Test Data: Sample participation metrics

Test Steps:

1. Instructor logs in to the platform

2. Instructor accesses the attendance report with integrated participation metrics

3. Instructor reviews the integrated report

Postconditions:

* Integrated report is displayed accurately

Expected Result: Attendance data is integrated with participation metrics, providing a comprehensive view of

learner engagement.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that notifications are sent for missed sessions

Test Case ID: AT05

Test Case Title: Verify that notifications are sent for missed sessions

Test Case Description: Ensure that learners receive notifications when they miss a live session, including a

reminder to review the recorded session and complete any assignments or quizzes.

Test Suite: Attendance Tracking

Test Priority: Low

Preconditions:

* Learner is registered for a live session

* Learner misses the live session

Test Data: Sample notification template

Test Steps:

1. System detects learner absence from the live session

2. System sends a notification to the learner

3. Learner reviews the notification

Postconditions:

* Notification is sent successfully

Expected Result: Learners receive notifications when they miss a live session, including a reminder to review

the recorded session and complete any assignments or quizzes.

Severity: Minor

Type of Testing: Functional

Data - 69

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Licensing and Rights Management

Feature description -

The platform implements robust content licensing and rights management features to protect intellectual property and ensure compliance with copyright laws. By managing content licenses, usage rights, and permissions, the platform safeguards the interests of content creators and providers, enabling secure and lawful distribution of educational materials.

Userstory title -

As a content creator, I want content licensing and rights management so that my intellectual property is protected and used legally.

Acceptance criteria -

['Content licenses are assigned to each resource.', 'Usage rights and permissions are clearly defined.', 'Content creators can set restrictions on usage and distribution.', 'Instructors and learners are informed of content licensing terms.', 'Notifications are sent for license expiration or renewal.']

Generated Test Cases:

Here are the five test cases for the Content Licensing and Rights Management feature:

Test Case 1: Verify that content licenses are assigned to each resource

Test Case ID: CLRM001

Test Case Title: Verify that content licenses are assigned to each resource

Test Case Description: Ensure that the system assigns a unique license to each educational resource, such as

videos, documents, and quizzes, to protect intellectual property.

Test Suite: Content Licensing and Rights Management

Test Priority: High

Preconditions:

* Content creator is logged in

* Resource is uploaded to the platform

Test Data: No test data needed

Test Steps:

1. Upload a new resource (e.g., video) to the platform

2. Check the resource details page

3. Verify that a unique license is assigned to the resource

Postconditions:

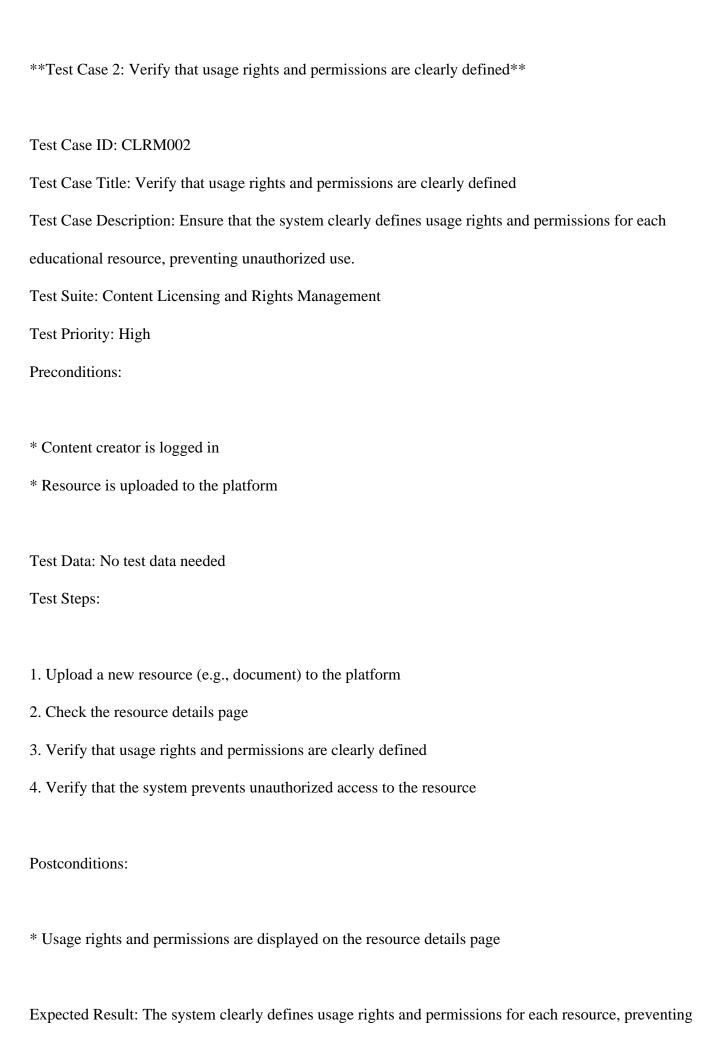
* License is displayed on the resource details page

Expected Result: The system assigns a unique license to each resource, protecting intellectual property.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive



unauthorized use.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that content creators can set restrictions on usage and distribution
Test Case ID: CLRM003
Test Case Title: Verify that content creators can set restrictions on usage and distribution
Test Case Description: Ensure that the system allows content creators to set restrictions on how their resources
can be used and distributed.
Test Suite: Content Licensing and Rights Management
Test Priority: Medium
Preconditions:
* Content creator is logged in
* Resource is uploaded to the platform
Test Data: No test data needed
Test Steps:
1. Upload a new resource (e.g., quiz) to the platform
2. Check the resource details page
3. Set restrictions on usage and distribution (e.g., only for personal use)
4. Verify that the system enforces the restrictions

Postconditions:

* Restrictions are displayed on the resource details page
Expected Result: The system allows content creators to set restrictions on usage and distribution, protecting
their intellectual property.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that instructors and learners are informed of content licensing terms
Test Case ID: CLRM004
Test Case Title: Verify that instructors and learners are informed of content licensing terms
Test Case Description: Ensure that the system informs instructors and learners about the content licensing
terms and conditions before they access the resource.
Test Suite: Content Licensing and Rights Management
Test Priority: Medium
Preconditions:
* Instructor/learner is logged in
* Resource is available on the platform
Test Data: No test data needed
Test Steps:
1. Access a resource on the platform
2. Verify that the system displays the content licensing terms and conditions

3. Verify that the instructor/learner must agree to the terms before accessing the resource
Postconditions:
* Content licensing terms and conditions are displayed on the resource page
Expected Result: The system informs instructors and learners about the content licensing terms and
conditions, ensuring they understand the usage rights and permissions.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that notifications are sent for license expiration or renewal
Test Case ID: CLRM005
Test Case Title: Verify that notifications are sent for license expiration or renewal
Test Case Description: Ensure that the system sends notifications to content creators, instructors, and learners
when a license is about to expire or has been renewed.
Test Suite: Content Licensing and Rights Management
Test Priority: High
Preconditions:
* Content creator/instructor/learner is logged in * Resource is uploaded to the platform with a license
Test Data: License information (e.g., expiration date)

1. Set up a license to expire in a short period (e.g., 1 day) 2. Verify that the system sends a notification to the content creator/instructor/learner 3. Verify that the system updates the license status after renewal Postconditions: * License status is updated on the resource details page Expected Result: The system sends notifications for license expiration or renewal, ensuring that users are aware of changes to the license status. Severity: Critical Type of Testing: Functional Test Case Approach: Positive **Data - 70** Project name -Online Learning Platform **Project description -**Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

Content Versioning

Feature description -

The platform implements content versioning features, allowing instructors to manage and track multiple

versions of course materials. By providing version control, the platform ensures that updates, revisions, and

changes are systematically recorded, enabling instructors to maintain a history of content evolution and easily

revert to previous versions if needed.

Userstory title -

As an instructor, I want content versioning features so that I can manage and track changes to my course

materials.

Acceptance criteria -

['Instructors can create and save multiple versions of content.', 'Version history is maintained and accessible.',

'Instructors can compare different versions of content.', 'Previous versions can be restored if needed.',

'Notifications are sent for significant content updates.']

Generated Test Cases:

Here are the 5 detailed test cases for the Content Versioning feature:

Test Case 1: Verify that instructors can create and save multiple versions of content

Test Case ID: CVF1

Test Case Title: Verify that instructors can create and save multiple versions of content

Test Case Description: Ensure that the system allows instructors to create and save multiple versions of course

materials, enabling version control and tracking changes.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in

- Course material is created

Test Data: No test data needed

Test Steps:

1. Create a new course material

2. Make changes to the material and save as a new version

3. Repeat step 2 multiple times to create multiple versions

Postconditions:

- Multiple versions of the course material are saved

Expected Result: The system saves each version of the course material, allowing instructors to track changes and maintain a version history.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that version history is maintained and accessible

Test Case ID: CVF2

Test Case Title: Verify that version history is maintained and accessible

Test Case Description: Ensure that the system maintains a record of all version changes and allows instructors to access and view version history.

Test Suite: Content Versioning

Test Priority: High

Preconditions:

- Instructor is logged in

- Multiple versions of course material are created

Test Data: No test data needed

Test Steps:

1. Go to the course material page

2. Click on the version history button

3. Verify that all versions are listed with corresponding change dates and descriptions

Postconditions:

- Version history is displayed correctly

Expected Result: The system displays a comprehensive version history, allowing instructors to track changes

and updates to course materials.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that instructors can compare different versions of content

Test Case ID: CVF3

Test Case Title: Verify that instructors can compare different versions of content

Test Case Description: Ensure that the system allows instructors to compare different versions of course

materials, enabling easy identification of changes and updates.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- Instructor is logged in

- Multiple versions of course material are created

Test Data: No test data needed

Test Steps:

1. Select two versions of the course material to compare

2. Click on the compare button

3. Verify that the system highlights changes and updates between the two versions

Postconditions:

- Comparison results are displayed correctly

Expected Result: The system compares the selected versions, highlighting changes and updates, allowing instructors to easily identify differences between versions.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that previous versions can be restored if needed

Test Case ID: CVF4

Test Case Title: Verify that previous versions can be restored if needed

Test Case Description: Ensure that the system allows instructors to restore previous versions of course

materials, enabling easy reversion to a previous state.

Test Suite: Content Versioning

Test Priority: Medium

Preconditions:

- Instructor is logged in

- Multiple versions of course material are created

Test Data: No test data needed

Test Steps:

- 1. Select a previous version of the course material to restore
- 2. Click on the restore button
- 3. Verify that the system reverts to the selected previous version

Postconditions:

- Previous version is restored successfully

Expected Result: The system restores the selected previous version, allowing instructors to easily revert to a

previous state if needed.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that notifications are sent for significant content updates

Test Case ID: CVF5

Test Case Title: Verify that notifications are sent for significant content updates

Test Case Description: Ensure that the system sends notifications to instructors when significant content

updates are made, keeping them informed of changes to their course materials.

Test Suite: Content Versioning

Test Priority: Low

Preconditions:

- Instructor is logged in

- Course material is created with multiple versions

Test Data: No test data needed

Test Steps:

1. Make a significant update to the course material (e.g., add a new chapter)

2. Verify that a notification is sent to the instructor

3. Verify that the notification contains details of the update

Postconditions:

- Notification is sent successfully

Expected Result: The system sends a notification to the instructor when a significant update is made to the

course material, keeping them informed of changes.

Severity: Trivial

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:

Here are the 5 test cases for the API Integration feature of the Online Learning Platform:

Test Case 1: Verify that APIs are available for key platform functionalities

Test Case ID: API-001
Test Case Title: Verify that APIs are available for key platform functionalities
Test Case Description: Ensure that APIs are provided for core platform features, such as course enrollment,
lecture access, and quiz submissions.
Test Suite: API Integration
Test Priority: High
Preconditions:
* Developer account is created
* API documentation is accessible
Test Data: No test data needed
Test Steps:
1. Access the API documentation
2. Verify that APIs are available for course enrollment, lecture access, and quiz submissions
3. Check that API endpoints are properly documented
Postconditions:
* API documentation is updated
Expected Result: APIs are available and properly documented for key platform functionalities.
Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that comprehensive API documentation is provided.
Test Case ID: API-002
Test Case Title: Verify that comprehensive API documentation is provided
Test Case Description: Ensure that API documentation is thorough, up-to-date, and easy to understand,
covering all aspects of API usage.
Test Suite: API Integration
Test Priority: Medium
Preconditions:
* API documentation is accessible
* Developer account is created
Test Data: No test data needed
Test Steps:
1. Review API documentation
2. Verify that documentation covers API endpoints, request/response formats, error handling, and
authentication
3. Check that documentation is easy to understand and navigate
Postconditions:
* API documentation is updated
Expected Result: Comprehensive and easy-to-understand API documentation is provided.
Severity: Major
Type of Testing: Functional

Test Case Approach: Positive
Test Case 3: Verify that secure access protocols are implemented
Test Case ID: API-003
Test Case Title: Verify that secure access protocols are implemented
Test Case Description: Ensure that secure access protocols, such as OAuth or JWT, are implemented to
prevent unauthorized access to the platform's data and features.
Test Suite: API Integration
Test Priority: High
Preconditions:
* Developer account is created
* API access token is obtained
Test Data:
* Valid and invalid API access tokens
Test Steps:
1. Attempt to access a protected API endpoint without a token
2. Verify that access is denied
3. Use a valid token to access the API endpoint
4. Verify that access is granted
Postconditions:

* API access token is updated
Expected Result: Secure access protocols are implemented, preventing unauthorized access to the platform's
data and features.
Severity: Critical
Type of Testing: Security
Test Case Approach: Positive and Negative
Test Case 4: Verify that third-party applications can authenticate and access APIs
Test Case ID: API-004
Test Case Title: Verify that third-party applications can authenticate and access APIs
Test Case Description: Ensure that third-party applications can authenticate and access the platform's data and
features using the provided APIs.
Test Suite: API Integration
Test Priority: High
Preconditions:
* Developer account is created
* Third-party application is registered
* API access token is obtained
Test Data:
* Valid and invalid API access tokens
Test Steps:
1. Configure the third-party application to use the API

2. Verify that the application can authenticate and access the API
3. Test API calls from the third-party application
Postconditions:
* API access token is updated
Expected Result: Third-party applications can authenticate and access the platform's data and features using
the provided APIs.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that rate limiting and monitoring features are in place
Test Case ID: API-005
Test Case Title: Verify that rate limiting and monitoring features are in place
Test Case Description: Ensure that rate limiting and monitoring features are implemented to prevent abuse and
monitor API usage.
Test Suite: API Integration
Test Priority: Medium
Preconditions:
* Developer account is created * API access token is obtained
Test Data:

* Valid and invalid API access tokens
Test Steps:
1. Perform a large number of API calls rapidly
2. Verify that rate limiting is enforced
3. Check the API usage monitoring dashboard
Postconditions:
* API usage monitoring dashboard is updated
Expected Result: Rate limiting and monitoring features are implemented, preventing abuse and monitoring
API usage.
Severity: Major
Type of Testing: Performance
Test Case Approach: Positive and Negative
Data - 72
Project name -
Online Learning Platform
Project description -
Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and
topics. The platform should provide interactive features like video lectures, downloadable resources,
discussion forums, progress tracking, and certifications upon course completion.
Feature name -
Data Analytics and Insights

Feature description -

The platform provides data analytics and insights features, enabling instructors and administrators to analyze

learner performance, engagement, and outcomes. By leveraging data visualization tools and reporting features,

the platform helps stakeholders make informed decisions to enhance the learning experience.

Userstory title -

As an administrator, I want data analytics and insights so that I can analyze learner performance and improve

the platform's effectiveness.

Acceptance criteria -

['Data visualization tools are available for analysis.', 'Reports can be generated for learner performance

metrics.', 'Real-time analytics provide insights into engagement and outcomes.', 'Data can be exported for

further analysis.', 'Notifications are sent for significant data trends or anomalies.']

Generated Test Cases:

Here are the 5 test cases for the Online Learning Platform's Data Analytics and Insights feature:

Test Case 1: Verify that data visualization tools are available for analysis

Test Case ID: OLDP_DA_001

Test Case Title: Verify that data visualization tools are available for analysis

Test Case Description: Ensure that the platform provides data visualization tools for administrators to analyze

learner performance and engagement metrics.

Test Suite: Data Analytics and Insights

Test Priority: High

Preconditions:

* Administrator is logged in

* Course data is available for analysis

Test Data: No test data needed
Test Steps:
1. Log in as an administrator
2. Navigate to the Data Analytics dashboard
3. Verify the availability of data visualization tools (e.g., charts, graphs, tables)
4. Select a course to analyze
5. Verify that the visualization tools display relevant data
Postconditions:
* Data visualization tools are displayed correctly
* Course data is analyzed successfully
Expected Result: The platform provides data visualization tools for administrators to analyze learner
performance and engagement metrics.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that reports can be generated for learner performance metrics
Test Coss ID, OI DD DA 002
Test Case ID: OLDP_DA_002
Test Case Title: Verify that reports can be generated for learner performance metrics

Test Case Description: Ensure that the platform allows administrators to generate reports for learner

performance metrics, such as course completion rates and quiz scores.
Test Suite: Data Analytics and Insights
Test Priority: High
Preconditions:
* Administrator is logged in
* Course data is available for analysis
Course data is available for allarysis
Test Data: No test data needed
Test Steps:
1. Log in as an administrator
2. Navigate to the Data Analytics dashboard
3. Select a course to generate a report for
4. Choose the desired report type (e.g., PDF, CSV)
5. Verify that the report is generated successfully
Postconditions:
* Report is generated successfully
* Report contains relevant learner performance metrics
Expected Result: The platform generates reports for learner performance metrics, providing insights for
administrators.
Severity: Major
Type of Testing: Functional

Test Case Approach: Positive
Test Case 3: Verify that real-time analytics provide insights into engagement and outcomes
Test Case ID: OLDP_DA_003
Test Case Title: Verify that real-time analytics provide insights into engagement and outcomes
Test Case Description: Ensure that the platform provides real-time analytics for administrators to track learner
engagement and outcomes, such as video watch time and quiz submissions.
Test Suite: Data Analytics and Insights
Test Priority: High
Preconditions:
* Administrator is logged in
* Course data is available for analysis
Test Data: No test data needed
Test Steps:
1. Log in as an administrator
2. Navigate to the Data Analytics dashboard
3. Select a course to track real-time analytics for
4. Verify that real-time analytics are displayed correctly
5. Verify that the analytics provide insights into engagement and outcomes
Postconditions:
* Real-time analytics are displayed correctly

* Real-time analytics provide insights into engagement and outcomes
Expected Result: The platform provides real-time analytics, enabling administrators to track learner engagement and outcomes effectively.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that data can be exported for further analysis
Test Case ID: OLDP_DA_004
Test Case Title: Verify that data can be exported for further analysis
Test Case Description: Ensure that the platform allows administrators to export data for further analysis, such
as exporting learner performance metrics to CSV or Excel.
Test Suite: Data Analytics and Insights
Test Priority: Medium
Preconditions:
* Administrator is logged in
* Course data is available for analysis
Test Data: No test data needed
Test Steps:
1. Log in as an administrator
2. Navigate to the Data Analytics dashboard

3. Select a course to export data for
4. Choose the desired export format (e.g., CSV, Excel)
5. Verify that the data is exported successfully
Postconditions:
* Data is exported successfully
* Exported data is in the correct format
Expected Result: The platform allows administrators to export data for further analysis, providing flexibility
and customization.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that notifications are sent for significant data trends or anomalies
Test Case ID: OLDP_DA_005
Test Case Title: Verify that notifications are sent for significant data trends or anomalies
Test Case Description: Ensure that the platform sends notifications to administrators when significant data
trends or anomalies are detected, such as a sudden increase in course dropouts.
Test Suite: Data Analytics and Insights
Test Priority: Medium
Preconditions:
* Administrator is logged in

* Course data is available for analysis
Test Data: No test data needed
Test Steps:
1. Log in as an administrator
2. Navigate to the Data Analytics dashboard
3. Configure notification settings for significant data trends or anomalies
4. Simulate a significant data trend or anomaly (e.g., course dropout rate increases)
5. Verify that a notification is sent to the administrator
Postconditions:
* Notification is sent successfully
* Notification contains relevant information about the data trend or anomaly
Expected Result: The platform sends notifications to administrators when significant data trends or anomalies
are detected, enabling proactive decision-making.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Data - 73
Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and guizzes on various subjects and

topics. The platform should provide interactive features like video lectures, downloadable resources,

discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Roles and Permissions

Feature description -

The platform implements a user roles and permissions system, allowing administrators to define and manage

different levels of access and functionality for various user groups. This feature ensures that users have

appropriate permissions based on their roles, such as instructors, students, and administrators, enhancing

security and user experience.

Userstory title -

As an administrator, I want to manage user roles and permissions so that I can control access to platform

features.

Acceptance criteria -

['Administrators can create and assign user roles.', 'Permissions are customizable for each role.', 'Users are

restricted to functionalities based on their roles.', 'Changes to roles and permissions are logged.', 'Users are

notified of their assigned roles and permissions.']

Generated Test Cases:

Here are the 5 test cases for the given input data:

Test Case 1:

Test Case ID: OLPRole1

Test Case Title: Verify that administrators can create and assign user roles.

Test Case Description: Ensure that administrators can create new user roles and assign them to users,

restricting access to platform features accordingly.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- Administrator is logged in

- User account exists

Test Data: No test data needed

Test Steps:

1. Go to user management dashboard

2. Click on "Create Role"

3. Enter role details (e.g., name, description)

4. Assign the role to a user

5. Verify the user's access to platform features

Postconditions:

- New role is created

- User is assigned the new role

Expected Result: The system allows administrators to create and assign user roles, restricting access to

platform features accordingly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPRole2

Test Case Title: Verify that permissions are customizable for each role.

Test Case Description: Ensure that administrators can customize permissions for each user role, controlling

access to specific platform features.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- Administrator is logged in

- User role exists

Test Data: No test data needed

Test Steps:

1. Go to role management dashboard

2. Select a role

3. Customize permissions (e.g., allow/deny access to specific features)

4. Save changes

5. Verify the updated permissions

Postconditions:

- Permissions are updated for the selected role

Expected Result: The system allows administrators to customize permissions for each user role, controlling access to specific platform features.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPRole3

Test Case Title: Verify that users are restricted to functionalities based on their roles.

Test Case Description: Ensure that users can only access platform features corresponding to their assigned

roles, with no unauthorized access.

Test Suite: User Roles and Permissions

Test Priority: High

Preconditions:

- User is logged in
- User role is assigned

Test Data: No test data needed

Test Steps:

- 1. Log in as a user with a specific role
- 2. Attempt to access a feature not permitted by the role
- 3. Verify the access denial
- 4. Attempt to access a feature permitted by the role
- 5. Verify access is granted

Postconditions:

- User access is restricted to assigned role

Expected Result: The system restricts users to functionalities based on their assigned roles, with no unauthorized access.

Severity: Critical

Type of Testing: Security

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPRole4

Test Case Title: Verify that changes to roles and permissions are logged.

Test Case Description: Ensure that the system maintains an audit trail of changes to user roles and permissions, including who made the changes and when.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- Administrator is logged in

- User role exists

Test Data: No test data needed

Test Steps:

1. Make changes to a user role (e.g., update permissions)

2. Verify the changes are reflected in the system

3. Check the audit log for the changes

Postconditions:

- Changes are logged in the audit trail

Expected Result: The system maintains an audit trail of changes to user roles and permissions, including who made the changes and when.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPRole5

Test Case Title: Verify that users are notified of their assigned roles and permissions.

Test Case Description: Ensure that users receive notifications when their roles or permissions are updated, informing them of the changes.

Test Suite: User Roles and Permissions

Test Priority: Medium

Preconditions:

- Administrator is logged in

- User account exists

Test Data: No test data needed

Test Steps:

1. Assign a new role to a user

- 2. Verify the user receives a notification
- 3. Update the user's permissions
- 4. Verify the user receives another notification

Postconditions:

- User receives notifications of role and permission changes

Expected Result: The system notifies users of their assigned roles and permissions, informing them of changes.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 74

Project name -

Online Learning Platform

Project description -

Develop an online learning platform that offers courses, lectures, tutorials, and quizzes on various subjects and topics. The platform should provide interactive features like video lectures, downloadable resources, discussion forums, progress tracking, and certifications upon course completion.

Feature name -

User Feedback and Rating System

Feature description -

The platform includes a user feedback and rating system, allowing learners to provide feedback on courses, instructors, and learning materials. This feature enables continuous improvement by gathering user insights and ratings, helping instructors refine their content and approach.

Userstory title -

As a learner, I want to provide feedback and ratings so that I can share my experience and help improve the platform.

Acceptance criteria -

['Learners can rate courses and instructors on a scale.', 'Feedback forms are available for detailed comments.', 'Instructors can view and respond to feedback.', 'Aggregated ratings are displayed publicly.', 'Anonymous feedback options are available.']

Generated Test Cases:

Here are the 5 detailed test cases for the User Feedback and Rating System feature:

Test Case 1: Verify that learners can rate courses and instructors on a scale

Test Case ID: LF001

Test Case Title: Verify that learners can rate courses and instructors on a scale

Test Case Description: Ensure that learners can provide ratings for courses and instructors using a rating scale.

Test Suite: User Feedback and Rating System

Test Priority: High

Preconditions:

- * Learner is logged in
- * Course is completed
- * Instructor is assigned to the course

Test Data: No test data needed

Test Steps:

- 1. Log in as a learner
- 2. Go to the course dashboard

3. Click on the "Rate Course" button
4. Select a rating from the scale (e.g., 1-5)
5. Click "Submit"
Postconditions:
* Rating is recorded
Expected Result: The system allows learners to rate courses and instructors on a scale, and the rating is
recorded.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that feedback forms are available for detailed comments
Test Case ID: LF002
Test Case Title: Verify that feedback forms are available for detailed comments
Test Case Description: Ensure that learners can provide detailed feedback comments for courses and
instructors.
Test Suite: User Feedback and Rating System
Test Priority: Medium
Preconditions:
* Learner is logged in
* Course is completed
* Instructor is assigned to the course
Test Data: No test data needed
Test Steps:

1. Log in as a learner
2. Go to the course dashboard
3. Click on the "Provide Feedback" button
4. Fill out the feedback form with comments
5. Click "Submit"
Postconditions:
* Feedback form is submitted
Expected Result: The system provides a feedback form for learners to provide detailed comments, and the
feedback is recorded.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that instructors can view and respond to feedback
Test Case ID: LF003
Test Case Title: Verify that instructors can view and respond to feedback
Test Case Description: Ensure that instructors can view learner feedback and respond to it.
Test Suite: User Feedback and Rating System
Test Priority: High
Preconditions:
* Instructor is logged in
* Learner has provided feedback
Test Data: No test data needed

Test Steps:
1. Log in as an instructor
2. Go to the course dashboard
3. Click on the "View Feedback" button
4. Review learner feedback
5. Respond to feedback
Postconditions:
* Feedback is viewed and responded to
Expected Result: The system allows instructors to view and respond to learner feedback.
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that aggregated ratings are displayed publicly
Test Case ID: LF004
Test Case Title: Verify that aggregated ratings are displayed publicly
Test Case Description: Ensure that aggregated ratings for courses and instructors are displayed publicly.
Test Suite: User Feedback and Rating System
Test Priority: Medium
Preconditions:
* Course is published
* Instructor is assigned to the course
Test Data: No test data needed

Test Steps:
1. Go to the course page
2. Check the aggregated rating display
3. Verify that the rating is accurate
Postconditions:
* Aggregated rating is displayed
Expected Result: The system displays aggregated ratings for courses and instructors publicly.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that anonymous feedback options are available
Test Case ID: LF005
Test Case Title: Verify that anonymous feedback options are available
Test Case Description: Ensure that learners can provide anonymous feedback for courses and instructors.
Test Suite: User Feedback and Rating System
Test Priority: Low
Preconditions:
* Learner is logged in
* Course is completed
* Instructor is assigned to the course
Test Data: No test data needed
Test Steps:

- 1. Log in as a learner
- 2. Go to the course dashboard
- 3. Click on the "Provide Anonymous Feedback" button
- 4. Fill out the feedback form with comments
- 5. Click "Submit"

* Anonymous feedback is submitted

Expected Result: The system provides an option for learners to provide anonymous feedback, and the feedback is recorded.

Severity: Trivial

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:

Here are the 5 test cases based on the provided input data:

Test Case 1:

Test Case ID: OLPSMI1

Test Case Title: Verify that learners can connect their social media accounts to the platform

Test Case Description: Ensure that learners can successfully connect their social media accounts to the online

learning platform, allowing them to share achievements and updates.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

* Learner is logged in to the platform

* Social media account credentials are valid

Test Data: Social media account details (e.g., username, password)

- 1. Go to the platform's settings page
- 2. Click on the "Connect Social Media" button
- 3. Select the social media platform to connect
- 4. Enter social media account credentials
- 5. Click "Connect"

* Social media account is connected to the platform

Expected Result: The learner's social media account is successfully connected to the platform, enabling sharing of achievements and updates.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: OLPSMI2

Test Case Title: Verify that achievements and course completions can be shared on social media

Test Case Description: Ensure that learners can share their achievements and course completions on their connected social media platforms.

Test Suite: Social Media Integration

Test Priority: High

Preconditions:

- * Learner's social media account is connected to the platform
- * Learner has completed a course or achieved a milestone

Test Data: Course completion data, achievement details

- 1. Go to the course completion page
- 2. Click on the "Share" button

- 3. Select the social media platform to share on
- 4. Customize the share message (optional)
- 5. Click "Share"

* Achievement or course completion is shared on social media

Expected Result: The learner's achievement or course completion is successfully shared on their social media platform, with an option to customize the share message.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: OLPSMI3

Test Case Title: Verify that privacy settings control the visibility of shared content

Test Case Description: Ensure that learners can control the visibility of their shared achievements and course completions on social media using privacy settings.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

- * Learner's social media account is connected to the platform
- * Learner has shared an achievement or course completion

Test Data: Privacy settings options (e.g., public, friends, only me)

- 1. Go to the platform's privacy settings page
- 2. Select the desired privacy setting for shared content
- 3. Go to the shared content on social media
- 4. Verify the visibility of the shared content

* Privacy settings are updated

Expected Result: The learner's shared achievements and course completions are only visible to the selected

audience on social media, based on the chosen privacy setting.

Severity: Minor

Type of Testing: Security

Test Case Approach: Positive

Test Case 4:

Test Case ID: OLPSMI4

Test Case Title: Verify that instructors can promote courses and updates through social media

Test Case Description: Ensure that instructors can share their courses and updates on social media, promoting

the platform's offerings and engaging with learners.

Test Suite: Social Media Integration

Test Priority: Medium

Preconditions:

* Instructor's social media account is connected to the platform

* Instructor has created a course or update

Test Data: Course or update details

Test Steps:

1. Go to the course or update page

2. Click on the "Share" button

3. Select the social media platform to share on

4. Customize the share message (optional)

5. Click "Share"

Postconditions:

* Course or update is shared on social media

Expected Result: The instructor's course or update is successfully shared on their social media platform, with an option to customize the share message.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: OLPSMI5

Test Case Title: Verify that analytics track the reach and engagement of shared content

Test Case Description: Ensure that the platform tracks the reach and engagement of shared achievements,

course completions, and instructor-promoted content on social media.

Test Suite: Social Media Integration

Test Priority: Low

Preconditions:

* Social media analytics integration is enabled

* Shared content exists on social media

Test Data: None

Test Steps:

1. Go to the platform's analytics page

2. Select the desired social media platform

3. Verify the analytics data (e.g., reach, engagement, clicks)

Postconditions:

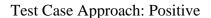
* Analytics data is updated

Expected Result: The platform accurately tracks the reach and engagement of shared content on social media,

providing insights for instructors and learners.

Severity: Trivial

Type of Testing: Functional



Data - 76

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Activity Tracking

Feature description -

The app records various activities such as running, walking, cycling, and swimming, capturing data on distance, duration, pace, and calories burned. This feature enables users to monitor their daily physical activity levels and progress towards fitness goals.

Userstory title -

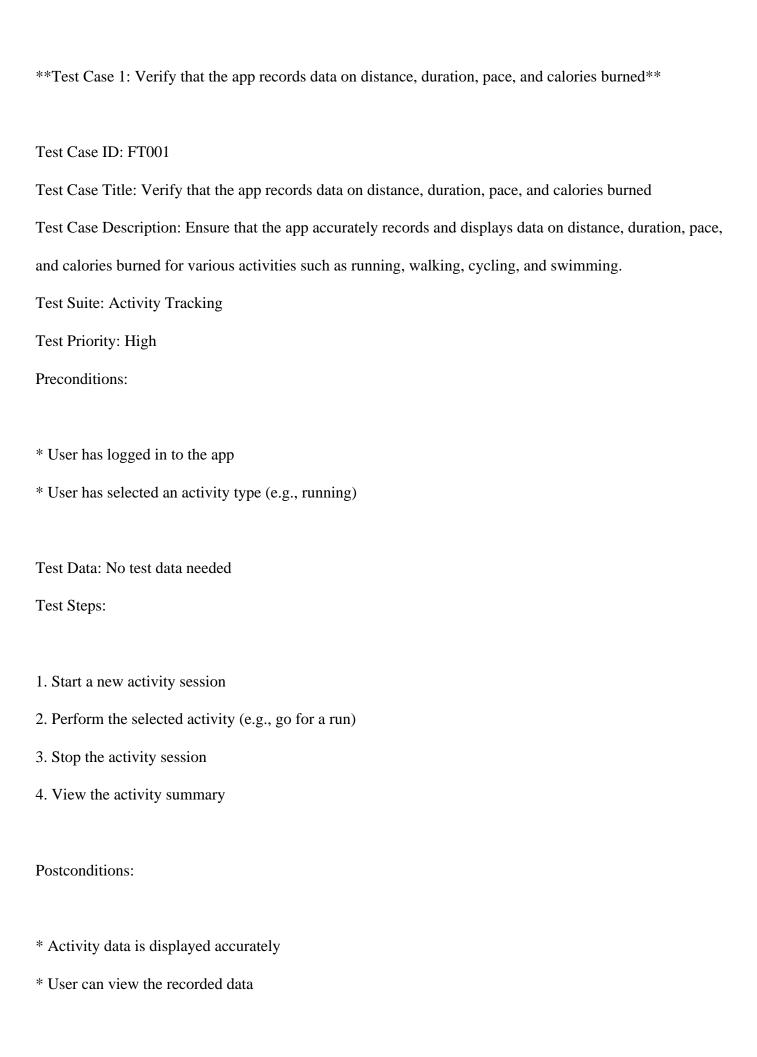
As a user, I want to track my activities so that I can monitor my daily physical activity levels.

Acceptance criteria -

['The app records data on distance, duration, pace, and calories burned.', 'Users can view activity summaries.', 'Data is stored for historical analysis.', 'Activity tracking can be paused and resumed.', 'Users receive notifications for milestones.']

Generated Test Cases:

Here are five detailed test cases for the Activity Tracking feature of the Fitness Tracker App:



Expected Result: The app accurately records and displays data on distance, duration, pace, and calories burned
for the selected activity.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that users can view activity summaries
Test Case ID: FT002
Test Case Title: Verify that users can view activity summaries
Test Case Description: Ensure that users can view a summary of their activities, including total distance, total
calories burned, and other relevant metrics.
Test Suite: Activity Tracking
Test Priority: Medium
Preconditions:
* User has logged in to the app
* User has completed at least one activity session
Test Data: No test data needed
Test Steps:
1. Go to the activity summary page
2. View the activity summary metrics (e.g., total distance, total calories burned)
3. Filter the activity summary by date range or activity type

Postconditions:
* Activity summary is displayed accurately
* User can filter the activity summary by date range or activity type
Expected Result: The app displays an accurate activity summary, and users can filter the summary by date
range or activity type.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that data is stored for historical analysis
Test Case ID: FT003
Test Case Title: Verify that data is stored for historical analysis
Test Case Description: Ensure that the app stores activity data for historical analysis, allowing users to view
their progress over time.
Test Suite: Activity Tracking
Test Priority: Medium
Preconditions:
* User has logged in to the app
* User has completed multiple activity sessions over a period of time
Test Data: No test data needed
Test Steps:

1. Go to the activity history page 2. View activity data for a specific date range or activity type 3. Compare activity data across different date ranges or activity types Postconditions: * Historical activity data is displayed accurately * User can view and compare activity data across different date ranges or activity types Expected Result: The app stores activity data for historical analysis, and users can view and compare their progress over time. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 4: Verify that activity tracking can be paused and resumed** Test Case ID: FT004 Test Case Title: Verify that activity tracking can be paused and resumed Test Case Description: Ensure that users can pause and resume activity tracking, allowing them to take breaks during an activity session. Test Suite: Activity Tracking

Test Priority: Medium

Preconditions:

* User has logged in to the app
* User has started an activity session
Test Data: No test data needed
Test Steps:
1. Start an activity session
2. Pause the activity session
3. Resume the activity session
4. View the activity summary
Postconditions:
1 Osteonarions.
* Activity session is paused and resumed accurately
* User can view the updated activity summary
Expected Result: The app allows users to pause and resume activity tracking, and the activity summary is
updated accurately.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that users receive notifications for milestones
Test Case ID: FT005
Test Case Title: Verify that users receive notifications for milestones

Test Case Description: Ensure that users receive notifications when they reach certain milestones (e.g.,
completing a certain number of workouts or reaching a new personal best).
Test Suite: Activity Tracking
Test Priority: Low
Preconditions:
* User has logged in to the app
* User has set up notification preferences
Test Data: No test data needed
Test Steps:
1. Complete an activity session that triggers a milestone notification
2. View the notification
3. Verify that the notification is accurate and timely
Postconditions:
* User receives a notification for the milestone
* Notification is accurate and timely
Expected Result: The app sends notifications to users when they reach certain milestones, and the notifications
are accurate and timely.
Severity: Trivial
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:

Here are the 5 detailed test cases for the GPS Tracking feature of the Fitness Tracker App:

Test Case ID: GPS-01

Test Case Title: Verify that GPS tracking accurately records route data

Test Case Description: Ensure that the GPS tracking feature accurately records route data, including distance,

speed, and time, during outdoor activities.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- User has enabled GPS tracking on their device
- User is logged in to the app
- Device has a stable GPS connection

Test Data: No test data needed

Test Steps:

- 1. Start a new outdoor activity (e.g., running, cycling) with GPS tracking enabled
- 2. Perform the activity for a set duration (e.g., 30 minutes)
- 3. Stop the activity and review the recorded route data

Postconditions:

- Route data is displayed accurately

Expected Result: The system accurately records and displays route data, including distance, speed, and time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: GPS-02

Test Case Title: Verify that users can view detailed maps and elevation profiles

Test Case Description: Ensure that users can view detailed maps and elevation profiles of their recorded

outdoor activities.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- User has recorded an outdoor activity with GPS tracking

- User is logged in to the app

Test Data: Recorded outdoor activity data

Test Steps:

1. Select a recorded outdoor activity

2. View the activity's map and elevation profile

3. Zoom in and out of the map to verify detail

Postconditions:

- Map and elevation profile are displayed correctly

Expected Result: The system displays detailed maps and elevation profiles of the recorded outdoor activity.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: GPS-03

Test Case Title: Verify that performance data includes location-based metrics

Test Case Description: Ensure that the performance data includes location-based metrics, such as pace,

cadence, and elevation gain.

Test Suite: GPS Tracking

Test Priority: High

Preconditions:

- User has recorded an outdoor activity with GPS tracking
- User is logged in to the app

Test Data: Recorded outdoor activity data

Test Steps:

- 1. Select a recorded outdoor activity
- 2. View the activity's performance data
- 3. Verify that location-based metrics are included

Postconditions:

- Performance data includes location-based metrics

Expected Result: The system includes location-based metrics in the performance data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: GPS-04

Test Case Title: Verify that routes can be saved and revisited

Test Case Description: Ensure that users can save and revisit their recorded outdoor activity routes.

Test Suite: GPS Tracking

Test Priority: Medium

Preconditions:

- User has recorded an outdoor activity with GPS tracking
- User is logged in to the app

Test Data: Recorded outdoor activity data

1. Select a recorded outdoor activity 2. Save the route for later use 3. Revisit the saved route Postconditions: - Route is saved and can be revisited Expected Result: The system allows users to save and revisit their recorded outdoor activity routes. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 5:** Test Case ID: GPS-05 Test Case Title: Verify that notifications are provided for deviations from planned routes Test Case Description: Ensure that users receive notifications when they deviate from their planned outdoor activity routes. Test Suite: GPS Tracking

Preconditions:

Test Priority: High

- User has planned an outdoor activity with a set route
- User is logged in to the app
- Device has a stable GPS connection

Test Data: Planned outdoor activity route data

- 1. Start a planned outdoor activity
- 2. Intentionally deviate from the planned route
- 3. Verify that a notification is received

- Notification is received for deviation from planned route

Expected Result: The system provides a notification when the user deviates from their planned outdoor

activity route.

Severity: Major

Type of Testing: Functional

Test Case Approach: 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:

Based on the provided input data, I will generate exactly 5 detailed test cases to cover all acceptance criteria

for the Heart Rate Monitoring feature of the Fitness Tracker App.

Test Case 1:

Test Case ID: HRM_001

Test Case Title: Verify that heart rate data is recorded in real-time

Test Case Description: Ensure that the app accurately records heart rate data in real-time using built-in sensors

or compatible wearable devices.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

? User has a compatible wearable device connected to the app

? User is logged in and has granted necessary permissions

Test Data: No test data needed

Test Steps:

1. Start a new workout session

2. Begin exercising to increase heart rate

3. Monitor heart rate data on the app

Postconditions:

? Heart rate data is updated in real-time

Expected Result: The app accurately records and displays heart rate data in real-time.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: HRM_002

Test Case Title: Verify that users can set and monitor target heart rate zones

Test Case Description: Ensure that the app allows users to set target heart rate zones and receive notifications

when they enter or exit these zones during a workout.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

? User has a compatible wearable device connected to the app

? User is logged in and has granted necessary permissions

Test Data: Target heart rate zone settings (e.g., 120-140 BPM)

Test Steps:

1. Set a target heart rate zone

2. Start a new workout session

3. Monitor heart rate data and zone notifications

Postconditions:

? Target heart rate zone is set and notifications are received

Expected Result: The app allows users to set and monitor target heart rate zones, providing notifications when

they enter or exit these zones.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: HRM_003

Test Case Title: Verify that historical heart rate data is stored and viewable

Test Case Description: Ensure that the app stores and displays historical heart rate data, allowing users to track

their progress over time.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

? User has a compatible wearable device connected to the app

? User is logged in and has granted necessary permissions

Test Data: Historical heart rate data from previous workouts

Test Steps:

1. Access the workout history section

2. Select a previous workout to view heart rate data

3. Verify that historical heart rate data is displayed

Postconditions:

? Historical heart rate data is displayed

Expected Result: The app stores and displays historical heart rate data, enabling users to track their progress

over time.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: HRM_004

Test Case Title: Verify that alerts are provided for abnormal heart rate readings

Test Case Description: Ensure that the app provides alerts and notifications for abnormal heart rate readings,

such as extremely high or low heart rates.

Test Suite: Heart Rate Monitoring

Test Priority: High

Preconditions:

? User has a compatible wearable device connected to the app

? User is logged in and has granted necessary permissions

Test Data: Abnormal heart rate readings (e.g., 180 BPM or 40 BPM)

Test Steps:

1. Simulate an abnormal heart rate reading

2. Monitor alert notifications

3. Verify that the app provides alert notifications

Postconditions:

? Alert notifications are received for abnormal heart rate readings

Expected Result: The app provides alerts and notifications for abnormal heart rate readings, ensuring user safety and awareness.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Negative

Test Case 5:

Test Case ID: HRM_005

Test Case Title: Verify that heart rate data is compatible with various wearable devices

Test Case Description: Ensure that the app is compatible with different wearable devices, accurately recording and displaying heart rate data from various sources.

Test Suite: Heart Rate Monitoring

Test Priority: Medium

Preconditions:

? Multiple wearable devices are connected to the app

? User is logged in and has granted necessary permissions

Test Data: Heart rate data from different wearable devices

Test Steps:

1. Connect multiple wearable devices to the app

2. Monitor heart rate data from each device

3. Verify that the app accurately records and displays data from each device

Postconditions:

? Heart rate data from multiple devices is accurately recorded and displayed

Expected Result: The app is compatible with various wearable devices, accurately recording and displaying heart rate data from different sources.

Severity: Major

Type of Testing: Compatibility

Test Case Approach: Positive

Data - 79

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Calorie Tracking

Feature description -

Users can log their daily food intake and track calorie consumption against calorie expenditure from physical

activities, helping them maintain a healthy balance between energy intake and expenditure. This feature

promotes mindful eating habits and supports weight management goals.

Userstory title -

As a user, I want to track my calorie intake so that I can maintain a healthy balance between energy intake and

expenditure.

Acceptance criteria -

['Users can log daily food intake and calorie consumption.', 'The app calculates and displays total calories

consumed and burned.', 'Users can set daily calorie goals and track progress.', 'Food items and calorie

information are searchable in the database.', 'Users receive alerts for exceeding calorie goals.']

Generated Test Cases:

Here are the 5 test cases for the Calorie Tracking feature:

Test Case 1: Verify that users can log daily food intake and calorie consumption

Test Case ID: ftCa1

Test Case Title: Verify that users can log daily food intake and calorie consumption

Test Case Description: Ensure that the system allows users to log their daily food intake and calorie

consumption.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

* User is logged in

* User has created a profile

Test Data: Sample food item with calorie information (e.g., Apple, 95 calories)

Test Steps:

1. Go to the food logging page

2. Enter a food item and its calorie information

3. Save the entry

Postconditions:

* Food item is saved to the user's log

Expected Result: The system successfully logs the user's food intake and calorie consumption.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that the app calculates and displays total calories consumed and burned

Test Case ID: ftCa2

Test Case Title: Verify that the app calculates and displays total calories consumed and burned

Test Case Description: Ensure that the system accurately calculates and displays the total calories consumed

and burned based on user input.

Test Suite: Calorie Tracking

Test Priority: High

Preconditions:

* User has logged food intake and physical activity

* User has set daily calorie goals

Test Data: Sample food items with calorie information, sample physical activity with calorie burn information

Test Steps:

1. Log multiple food items with calorie information

2. Log physical activity with calorie burn information

3. View the calorie tracker dashboard

* Total calories consumed and burned are displayed

Expected Result: The system accurately calculates and displays the total calories consumed and burned.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can set daily calorie goals and track progress

Test Case ID: ftCa3

Test Case Title: Verify that users can set daily calorie goals and track progress

Test Case Description: Ensure that the system allows users to set daily calorie goals and track their progress

towards achieving those goals.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

* User is logged in

* User has created a profile

Test Data: Sample daily calorie goal (e.g., 2000 calories)

Test Steps:

- 1. Go to the goal setting page
- 2. Set a daily calorie goal
- 3. Track progress towards the goal

Postconditions:

- * Daily calorie goal is set
- * Progress towards the goal is tracked

Expected Result: The system successfully sets and tracks the user's daily calorie goal.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that food items and calorie information are searchable in the database

Test Case ID: ftCa4

Test Case Title: Verify that food items and calorie information are searchable in the database

Test Case Description: Ensure that the system allows users to search for food items and calorie information in

the database.

Test Suite: Calorie Tracking

Test Priority: Medium

Preconditions:

* User is logged in

* Food items are populated in the database

Test Data: Sample search query (e.g., "Apple")

Test Steps:

- 1. Go to the food search page
- 2. Enter a search query
- 3. View search results

Postconditions:

* Search results are displayed

Expected Result: The system successfully searches and displays food items and calorie information.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: ftCa5
Test Case Title: Verify that users receive alerts for exceeding calorie goals
Test Case Description: Ensure that the system sends alerts to users when they exceed their daily calorie goals
Test Suite: Calorie Tracking
Test Priority: High
Preconditions:
* User has set daily calorie goals
* User has exceeded their daily calorie goals
Test Data: Sample daily calorie goal (e.g., 2000 calories)
Test Steps:
1. Log food intake exceeding the daily calorie goal
2. Check for alerts or notifications
Postconditions:
* Alert is sent to the user
Expected Result: The system successfully sends an alert to the user when they exceed their daily calorie goal
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Data - 80
Project name -
Fitness Tracker App
Project description -

Test Case 5: Verify that users receive alerts for exceeding calorie goals

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

Feature name -

Sleep Tracking

Feature description -

The app monitors sleep duration, quality, and patterns using accelerometer and heart rate sensors, providing

insights into sleep stages, disturbances, and overall sleep quality. Sleep tracking helps users understand their

sleep patterns, identify factors affecting sleep quality, and make lifestyle adjustments for better rest and

recovery.

Userstory title -

As a user, I want to monitor my sleep so that I can understand my sleep patterns and improve my sleep

quality.

Acceptance criteria -

['Sleep data is recorded including duration and quality.', 'Users can view detailed sleep stages and patterns.',

'The app provides insights into sleep disturbances.', 'Users receive recommendations for improving sleep

quality.', 'Sleep data can be viewed in daily, weekly, and monthly summaries.']

Generated Test Cases:

Here are the 5 detailed test cases for the Sleep Tracking feature:

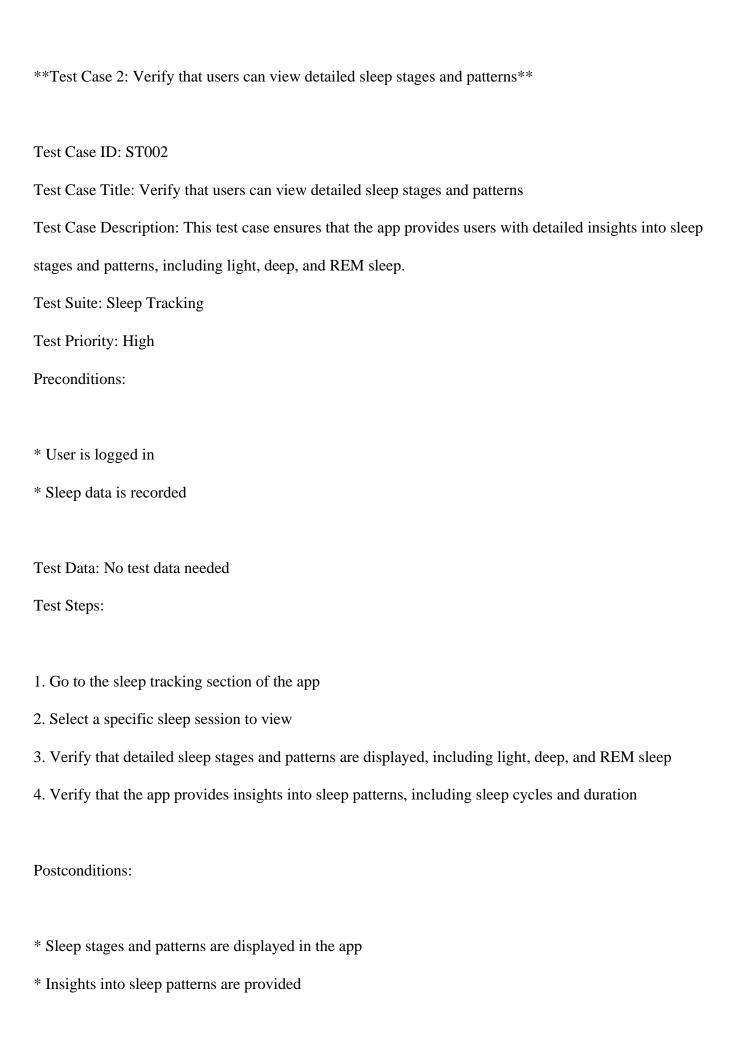
Test Case 1: Verify that sleep data is recorded including duration and quality

Test Case ID: ST001

Test Case Title: Verify that sleep data is recorded including duration and quality

Test Case Description: This test case ensures that the app records sleep data accurately, including duration and

quality, using accelerometer and heart rate sensors.
Test Suite: Sleep Tracking
Test Priority: High
Preconditions:
* User is logged in
* Wearable device or fitness equipment is connected
* Sleep tracking is enabled
Test Data: No test data needed
Test Steps:
1. Wear the wearable device or use fitness equipment
2. Go to sleep and let the app track sleep data
3. Wake up and sync the app with the wearable device or fitness equipment
4. Verify that sleep data is recorded, including duration and quality
Postconditions:
* Sleep data is recorded in the app
* Sleep data includes duration and quality metrics
Expected Result: The app accurately records sleep data, including duration and quality, using accelerometer
and heart rate sensors.
Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive



Expected Result: The app provides users with detailed insights into sleep stages and patterns, including light,
deep, and REM sleep.
Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive
Test Case 3: Verify that the app provides insights into sleep disturbances
Test Case ID: ST003
Test Case Title: Verify that the app provides insights into sleep disturbances
Test Case Description: This test case ensures that the app provides users with insights into sleep disturbances,
including sleep disruptions, awakenings, and movements during sleep.
Test Suite: Sleep Tracking
Test Priority: Medium
Preconditions:
* User is logged in
* Sleep data is recorded
Test Data: No test data needed
Test Steps:
1. Go to the sleep tracking section of the app
2. Select a specific sleep session to view
3. Verify that the app provides insights into sleep disturbances, including sleep disruptions, awakenings, and
movements during sleep

4. Verify that the app highlights periods of disturbed sleep

Postconditions:
* Insights into sleep disturbances are provided * Periods of disturbed sleep are highlighted
Expected Result: The app provides users with insights into sleep disturbances, including sleep disruptions,
awakenings, and movements during sleep.
Severity: Minor
Type of Testing: Functional
Test Case Behaviour: Positive
Test Case 4: Verify that users receive recommendations for improving sleep quality
Test Case ID: ST004
Test Case Title: Verify that users receive recommendations for improving sleep quality
Test Case Description: This test case ensures that the app provides users with personalized recommendations
for improving sleep quality, based on their sleep patterns and habits.
Test Suite: Sleep Tracking
Test Priority: Medium
Preconditions:
* User is logged in
* Sleep data is recorded

- 1. Go to the sleep tracking section of the app
- 2. Select a specific sleep session to view
- 3. Verify that the app provides personalized recommendations for improving sleep quality
- 4. Verify that recommendations are based on sleep patterns and habits

- * Recommendations for improving sleep quality are provided
- * Recommendations are based on sleep patterns and habits

Expected Result: The app provides users with personalized recommendations for improving sleep quality,

based on their sleep patterns and habits.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 5: Verify that sleep data can be viewed in daily, weekly, and monthly summaries

Test Case ID: ST005

Test Case Title: Verify that sleep data can be viewed in daily, weekly, and monthly summaries

Test Case Description: This test case ensures that the app allows users to view sleep data in daily, weekly, and

monthly summaries, providing a comprehensive overview of sleep patterns.

Test Suite: Sleep Tracking

Test Priority: High

Preconditions:

* User is logged in
* Sleep data is recorded
Test Data: No test data needed
Test Steps:
1. Go to the sleep tracking section of the app
2. Select a specific time period to view (daily, weekly, or monthly)
3. Verify that sleep data is displayed in the selected time period
4. Verify that summaries include key sleep metrics, such as sleep duration, quality, and stages
Postconditions:
* Sleep data is displayed in the selected time period
* Summaries include key sleep metrics
Expected Result: The app allows users to view sleep data in daily, weekly, and monthly summaries, providing
a comprehensive overview of sleep patterns.
Severity: Major
Type of Testing: Functional
Test Case Behaviour: Positive
Data - 81
Project name -
Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

Feature name -

Workout Logging

Feature description -

Users can log individual workouts, specifying exercise type, duration, intensity, and any additional notes or

comments. This feature allows users to maintain a detailed workout log, track progress over time, and identify

trends in performance and adherence to training programs.

Userstory title -

As a user, I want to log my workouts so that I can track my progress and improve my performance.

Acceptance criteria -

['Users can log various workout types with details.', 'Workout logs include duration, intensity, and notes.',

'Users can edit and delete workout entries.', 'Data is stored and viewable for tracking progress.', 'Users can

filter logs by date and workout type.']

Generated Test Cases:

Here are five detailed test cases for the Workout Logging feature:

Test Case 1: Verify that users can log various workout types with details

Test Case ID: WL001

Test Case Title: Verify that users can log various workout types with details

Test Case Description: Ensure that the system allows users to log different workout types with details such as

exercise type, duration, intensity, and notes.

Test Suite: Workout Logging
Test Priority: High
Preconditions:
* User is logged in
* Workout logging feature is enabled
Test Data: Various workout types (e.g., running, weightlifting, yoga) with details (e.g., exercise type, duration,
intensity, notes)
Test Steps:
1. Go to the workout logging page
2. Select a workout type
3. Enter workout details (exercise type, duration, intensity, notes)
4. Click 'Log Workout'
Postconditions:
* Workout is successfully logged
* Workout details are accurately stored
Expected Result: The system allows users to log various workout types with details and stores the data
accurately.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive

Test Case 2: Verify that workout logs include duration, intensity, and notes

Test Case ID: WL002
Test Case Title: Verify that workout logs include duration, intensity, and notes
Test Case Description: Ensure that the system includes essential workout log details such as duration,
intensity, and notes.
Test Suite: Workout Logging
Test Priority: High
Preconditions:
* User is logged in
* Workout logging feature is enabled
* A workout is logged
Test Data: Pre-existing workout log with duration, intensity, and notes
Test Steps:
1. Go to the workout logging page
2. View an existing workout log
3. Verify that the log includes duration, intensity, and notes
Postconditions:
* Workout log details are accurately displayed
Expected Result: The system includes essential workout log details such as duration, intensity, and notes.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can edit and delete workout entries
Test Case ID: WL003
Test Case Title: Verify that users can edit and delete workout entries
Test Case Description: Ensure that the system allows users to edit and delete their workout entries.
Test Suite: Workout Logging
Test Priority: Medium
Preconditions:
* User is logged in
* Workout logging feature is enabled
* A workout is logged
Test Data: Pre-existing workout log
Test Steps:
1. Go to the workout logging page
2. Select a workout log
3. Edit the workout log details
4. Save changes
5. Verify that the changes are reflected
6. Delete the workout log
7. Verify that the log is removed
Postconditions:

* Workout log is successfully edited and deleted

Expected Result: The system allows users to edit and delete their workout entries.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that data is stored and viewable for tracking progress
Test Case ID: WL004
Test Case Title: Verify that data is stored and viewable for tracking progress
Test Case Description: Ensure that the system stores workout data and allows users to view their progress over
time.
Test Suite: Workout Logging
Test Priority: High
Preconditions:
* User is logged in
* Workout logging feature is enabled
* Multiple workouts are logged
Test Data: Pre-existing workout logs Test Steps:
1. Go to the workout logging page
2. View workout logs over a period of time
3. Verify that the logs are accurately stored and displayed

Postconditions:

* Workout data is accurately stored and displayed
Expected Result: The system stores workout data and allows users to view their progress over time.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that users can filter logs by date and workout type
Test Case ID: WL005
Test Case Title: Verify that users can filter logs by date and workout type
Test Case Description: Ensure that the system allows users to filter their workout logs by date and workout
type.
Test Suite: Workout Logging
Test Priority: Medium
Preconditions:
* User is logged in
* Workout logging feature is enabled
* Multiple workouts are logged
Test Data: Pre-existing workout logs with varying dates and workout types Test Steps:
 Go to the workout logging page Filter logs by date

- 3. Verify that the filtered logs are accurately displayed
- 4. Filter logs by workout type
- 5. Verify that the filtered logs are accurately displayed

Postconditions:

* Workout logs are accurately filtered by date and workout type

Expected Result: The system allows users to filter their workout logs by date and workout type.

Severity: Minor

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:

Here are the 5 detailed test cases for the Goal Setting feature of the Fitness Tracker App:

Test Case 1: Verify that users can set goals for various fitness metrics

Test Case ID: ftg1

Test Case Title: Verify that users can set goals for various fitness metrics

Test Case Description: This test case ensures that users can set goals for different fitness metrics such as steps

taken, calories burned, distance traveled, and heart rate.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

* User is logged in

* User has created a profile

Test Data: Various fitness metrics (e.g., 10,000 steps, 500 calories, 5 miles, 120 BPM)

Test Steps:

1. Go to the Goal Setting page

2. Select a fitness metric (e.g., steps taken)

3. Set a goal (e.g., 10,000 steps)

4. Save changes
Postconditions:
* Goal is saved
Expected Result: The system allows users to set goals for various fitness metrics and saves the goal
successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that progress tracking is available for each goal
Test Case ID: ftg2
Test Case Title: Verify that progress tracking is available for each goal
Test Case Description: This test case ensures that users can track their progress towards their set goals.
Test Suite: Goal Setting
Test Priority: High
Preconditions:
* User has set a goal
Test Data: No test data needed
Test Steps:
1. Go to the Goal Setting page
2. Select a goal
3. View progress tracking details
Postconditions:
* Progress tracking details are displayed
Expected Result: The system provides progress tracking details for each goal, allowing users to monitor their
progress.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users receive motivational feedback and reminders

Test Case ID: ftg3

Test Case Title: Verify that users receive motivational feedback and reminders

Test Case Description: This test case ensures that users receive motivational feedback and reminders to stay

focused and committed to achieving their goals.

Test Suite: Goal Setting

Test Priority: Medium

Preconditions:

* User has set a goal

Test Data: No test data needed

Test Steps:

- 1. Set a goal
- 2. Wait for a reminder notification
- 3. View motivational feedback

Postconditions:

* Motivational feedback is displayed

Expected Result: The system sends reminder notifications and provides motivational feedback to users to stay

motivated and focused on achieving their goals.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that goals can be adjusted or updated as needed Test Case ID: ftg4 Test Case Title: Verify that goals can be adjusted or updated as needed Test Case Description: This test case ensures that users can adjust or update their goals as needed. Test Suite: Goal Setting Test Priority: Medium Preconditions: * User has set a goal Test Data: Updated goal details (e.g., new target value) Test Steps: 1. Go to the Goal Setting page 2. Select a goal 3. Update goal details 4. Save changes Postconditions: * Goal is updated successfully Expected Result: The system allows users to adjust or update their goals as needed, and saves the changes successfully. Severity: Minor Type of Testing: Functional Test Case Approach: Positive

Test Case 5: Verify that the app provides a summary of goal attainment

Test Case ID: ftg5

Test Case Title: Verify that the app provides a summary of goal attainment

Test Case Description: This test case ensures that the app provides a summary of goal attainment, including progress and achievements.

Test Suite: Goal Setting

Test Priority: High

Preconditions:

* User has set and tracked goals

Test Data: No test data needed

Test Steps:

1. Go to the Goal Setting page

2. View goal attainment summary

Postconditions:

* Goal attainment summary is displayed

Expected Result: The system provides a summary of goal attainment, including progress and achievements, allowing users to track their overall progress.

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:

Here are the 5 test cases for the Workout Plans feature:

Test Case 1: Verify that users can select workout plans based on fitness levels

Test Case ID: WP1

Test Case Title: Verify that users can select workout plans based on fitness levels

Test Case Description: Ensure that the app allows users to choose workout plans based on their fitness levels,

such as beginner, intermediate, or advanced.

Test Suite: Workout Plans

Test Priority: High

Preconditions:

- User is logged in - User has created a profile with fitness level information Test Data: No test data needed Test Steps: 1. Go to the workout plans section 2. Select a fitness level (e.g., beginner) 3. Browse available workout plans 4. Select a plan and review its details Postconditions: - Workout plan is selected and displayed Expected Result: The app displays a list of workout plans filtered by the selected fitness level. Severity: Major Type of Testing: Functional Test Case Behaviour: Positive **Test Case 2: Verify that plans include detailed exercise routines and schedules** Test Case ID: WP2 Test Case Title: Verify that plans include detailed exercise routines and schedules Test Case Description: Ensure that the app provides detailed exercise routines and schedules within each workout plan. Test Suite: Workout Plans Test Priority: High Preconditions: - User has selected a workout plan Test Data: Sample workout plan with exercises and schedules

Test Steps:

- Review the selected workout plan
 Check the exercise routine for each day
- 3. Verify the schedule and duration of each exercise

Postconditions:

- Exercise routine and schedule are displayed

Expected Result: The app displays a detailed exercise routine and schedule for each day of the workout plan.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 3: Verify that users can track adherence and progress within the plan

Test Case ID: WP3

Test Case Title: Verify that users can track adherence and progress within the plan

Test Case Description: Ensure that the app allows users to track their adherence and progress within the selected workout plan.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User has selected a workout plan

Test Data: Sample workout plan with progress tracking

Test Steps:

- 1. Start a workout plan
- 2. Complete a workout and log it
- 3. Review progress and adherence metrics

Postconditions:

- Progress and adherence metrics are updated

Expected Result: The app updates the user's progress and adherence metrics based on their workout logs.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: Positive

Test Case 4: Verify that the app provides notifications and reminders for workouts

Test Case ID: WP4

Test Case Title: Verify that the app provides notifications and reminders for workouts

Test Case Description: Ensure that the app sends notifications and reminders to users for upcoming workouts

in their selected plan.

Test Suite: Workout Plans

Test Priority: Medium

Preconditions:

- User has selected a workout plan

- Notifications are enabled

Test Data: No test data needed

Test Steps:

1. Set up notifications for upcoming workouts

2. Wait for the scheduled workout time

3. Check for notifications and reminders

Postconditions:

- Notifications and reminders are sent

Expected Result: The app sends notifications and reminders to the user for upcoming workouts in their

selected plan.

Severity: Minor

Type of Testing: Functional

Test Case Behaviour: Positive
Test Case 5: Verify that users can adjust plans based on progress and preferences
Test Case ID: WP5
Test Case Title: Verify that users can adjust plans based on progress and preferences
Test Case Description: Ensure that the app allows users to adjust their workout plans based on their progres
and preferences.
Test Suite: Workout Plans
Test Priority: High
Preconditions:
- User has started a workout plan
Test Data: Sample workout plan with adjustable settings
Test Steps:
1. Review progress and adjust the plan
2. Change the exercise routine or schedule
3. Save the updated plan
Postconditions:
- Workout plan is updated

Expected Result: The app allows users to adjust their workout plan based on their progress and preferences.

Severity: Major

Type of Testing: Functional

Test Case Behaviour: 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 -

Community Support

Feature description -

Users can connect with a community of like-minded individuals, share progress updates, participate in

challenges, and offer support and encouragement to fellow users. Community support fosters accountability,

motivation, and a sense of camaraderie among users striving towards similar fitness goals.

Userstory title -

As a user, I want to connect with a community so that I can share my progress and stay motivated.

Acceptance criteria -

['Users can join and participate in community groups.', 'Users can share progress updates and achievements.',

'The app facilitates participation in community challenges.', 'Users can offer and receive support and

encouragement.', 'Community activity and engagement are tracked.']

Generated Test Cases:

Here are the 5 detailed test cases for the Community Support feature of the Fitness Tracker App:

Test Case 1: Verify that users can join and participate in community groups

Test Case ID: fsdT5a

Test Case Title: Verify that users can join and participate in community groups

Test Case Description: Ensure that users can join and participate in community groups to connect with

like-minded individuals.

Test Suite: Community Support

Test Priority: High

Preconditions:

* User is logged in

* Community groups are available

Test Data: No test data needed

Test Steps:

1. Go to the community section

2. Browse available community groups

3. Select a group to join

4. Click on the "Join" button

Postconditions:

* User is added to the community group

Expected Result: The system allows users to join and participate in community groups.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that users can share progress updates and achievements

Test Case ID: hgf4d

Test Case Title: Verify that users can share progress updates and achievements

Test Case Description: Ensure that users can share their progress updates and achievements with the

community.

Test Suite: Community Support

Test Priority: Medium Preconditions: * User is logged in * User has recorded progress updates and achievements Test Data: Sample progress updates and achievements Test Steps: 1. Go to the community section 2. Click on the "Share Update" button 3. Enter progress update and achievement details 4. Click on the "Post" button Postconditions: * Progress update and achievement are posted Expected Result: The system allows users to share their progress updates and achievements with the community. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that the app facilitates participation in community challenges** Test Case ID: jkl2e Test Case Title: Verify that the app facilitates participation in community challenges Test Case Description: Ensure that the app allows users to participate in community challenges to stay

Test Case Description: Ensure that the app allows users to participate in community challenges to stay motivated and engaged.

Test Suite: Community Support

Test Priority: High

Preconditions:

- * User is logged in
- * Community challenges are available

Test Data: Sample community challenge

Test Steps:

- 1. Go to the community section
- 2. Browse available community challenges
- 3. Select a challenge to participate in
- 4. Click on the "Join" button

Postconditions:

* User is added to the community challenge

Expected Result: The system facilitates participation in community challenges.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that users can offer and receive support and encouragement

Test Case ID: mn3b

Test Case Title: Verify that users can offer and receive support and encouragement

Test Case Description: Ensure that users can offer and receive support and encouragement from fellow

community members.

Test Suite: Community Support

Test Priority: Medium

Preconditions:

- * User is logged in
- * User has joined a community group

Test Data: Sample support and encouragement messages

Test Steps: 1. Go to a community group discussion 2. Click on a user's post 3. Enter a support and encouragement message 4. Click on the "Post" button Postconditions: * Support and encouragement message is posted Expected Result: The system allows users to offer and receive support and encouragement from fellow community members. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 5: Verify that community activity and engagement are tracked** Test Case ID: pq1r Test Case Title: Verify that community activity and engagement are tracked Test Case Description: Ensure that the app tracks community activity and engagement to foster accountability and motivation among users. **Test Suite: Community Support** Test Priority: Low Preconditions: * User is logged in * Community groups are available Test Data: No test data needed

Test Steps:

1. Go to the community section

2. Browse available community groups

3. Observe community activity and engagement metrics

Postconditions:

* Community activity and engagement metrics are displayed

Expected Result: The system tracks community activity and engagement to foster accountability and

motivation among users.

Severity: Minor

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:

Here are the 5 test cases for the given input data:

Test Case 1: Verify that the app supports integration with multiple wearable devices

Test Case ID: ftIXg

Test Case Title: Verify that the app supports integration with multiple wearable devices

Test Case Description: Ensure that the app can integrate with various wearable devices, such as fitness

trackers, smartwatches, and heart rate monitors, to sync activity data.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User has multiple wearable devices paired with the app

- Devices are connected to the internet

Test Data: No test data needed

Test Steps:

1. Launch the app and navigate to the wearable device integration page

2. Select multiple wearable devices to integrate with the app

3. Verify that the app successfully integrates with each device

Postconditions:

- Devices are listed as connected in the app

Expected Result: The app supports integration with multiple wearable devices, allowing users to sync activity data from each device.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that users can sync activity data from their wearables

Test Case ID: vX2Hc

Test Case Title: Verify that users can sync activity data from their wearables

Test Case Description: Ensure that users can synchronize their activity data, including steps taken, heart rate,

and calories burned, from their wearable devices to the app.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User has a wearable device paired with the app

- Device has recorded activity data

Test Data: Wearable device with recorded activity data

Test Steps:

- 1. Launch the app and navigate to the wearable device integration page
- 2. Select the wearable device to sync data from
- 3. Verify that the app receives and displays the synced activity data

Postconditions:

- Activity data is updated in the app

Expected Result: Users can successfully sync activity data from their wearables, ensuring accurate tracking

and analysis of their fitness activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that the app provides real-time data updates from wearables

Test Case ID: rT5zG

Test Case Title: Verify that the app provides real-time data updates from wearables

Test Case Description: Ensure that the app receives and displays real-time updates of activity data, including

steps taken, heart rate, and calories burned, from connected wearable devices.

Test Suite: Integration with Wearable Devices

Test Priority: High

Preconditions:

- User has a wearable device paired with the app

- Device is connected to the internet

Test Data: Wearable device with real-time data streaming capabilities

Test Steps:

1. Launch the app and navigate to the wearable device integration page

2. Select the wearable device to receive real-time updates from

3. Perform an activity (e.g., take a few steps) while wearing the device

4. Verify that the app receives and displays the real-time update of activity data

Postconditions:

- Real-time data is updated in the app

Expected Result: The app provides real-time data updates from connected wearable devices, ensuring users

have accurate and timely tracking of their fitness activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that wearable data includes steps, heart rate, and calories burned

Test Case ID: t8NkR

Test Case Title: Verify that wearable data includes steps, heart rate, and calories burned

Test Case Description: Ensure that the app retrieves and displays comprehensive wearable data, including

steps taken, heart rate, and calories burned, to provide users with a complete picture of their fitness activities.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions:

- User has a wearable device paired with the app

- Device has recorded activity data

Test Data: Wearable device with recorded activity data

Test Steps:

1. Launch the app and navigate to the wearable device integration page

2. Select the wearable device to retrieve data from

3. Verify that the app displays steps taken, heart rate, and calories burned data

Postconditions:

- Comprehensive wearable data is displayed in the app

Expected Result: The app retrieves and displays comprehensive wearable data, including steps, heart rate, and

calories burned, providing users with a detailed understanding of their fitness activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that users can troubleshoot connectivity issues with wearables

Test Case ID: wZh9F

Test Case Title: Verify that users can troubleshoot connectivity issues with wearables

Test Case Description: Ensure that users can diagnose and resolve connectivity issues between their wearable

devices and the app, minimizing data loss and ensuring seamless integration.

Test Suite: Integration with Wearable Devices

Test Priority: Medium

Preconditions:

- User has a wearable device paired with the app

- Device is experiencing connectivity issues

Test Data: Wearable device with connectivity issues

Test Steps:

1. Launch the app and navigate to the wearable device integration page

2. Select the wearable device experiencing connectivity issues

3. Verify that the app provides troubleshooting options or guidance to resolve the issue

Postconditions:

- Connectivity issue is resolved

Expected Result: Users can troubleshoot and resolve connectivity issues with their wearables, ensuring

uninterrupted data syncing and accurate tracking of their fitness activities.

Severity: Major

Type of Testing: Functional

Test Case Approach: 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:

Here are the 5 detailed test cases for the Water Intake Tracking feature:

Test Case 1: Verify that users can log their daily water intake

Test Case ID: WIT-001

Test Case Title: Verify that users can log their daily water intake

Test Case Description: Ensure that the system allows users to log their daily water intake correctly.

Test Suite: Water Intake Tracking

Test Priority: High
Preconditions:
? User is logged in
? User has access to the water intake tracking feature
Test Data: No test data needed
Test Steps:
1. Go to the water intake tracking dashboard
2. Click on the "Log Water Intake" button
3. Enter the amount of water consumed
4. Click "Save"
Postconditions:
? Water intake is logged and displayed on the dashboard
Expected Result: The system successfully logs the user's daily water intake and displays it on the dashboard.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that the app calculates total water consumption for the day
Test Case ID: WIT-002
Test Case Title: Verify that the app calculates total water consumption for the day
Test Case Description: Ensure that the system accurately calculates the total water consumption for the day
based on user input.
Test Suite: Water Intake Tracking
Test Priority: Medium
Preconditions:
? User has logged water intake at least twice

? User has access to the water intake tracking feature Test Data: Water intake data (e.g., 2 cups, 3 cups) Test Steps: 1. Log multiple water intake entries for the day 2. Go to the water intake tracking dashboard 3. Check the total water consumption for the day Postconditions: ? Total water consumption is displayed on the dashboard Expected Result: The system correctly calculates and displays the total water consumption for the day based on user input. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that users can set daily hydration goals** Test Case ID: WIT-003 Test Case Title: Verify that users can set daily hydration goals Test Case Description: Ensure that the system allows users to set daily hydration goals and save them successfully. Test Suite: Water Intake Tracking Test Priority: High Preconditions: ? User is logged in

? User has access to the water intake tracking feature

Test Data: Hydration goal data (e.g., 8 cups, 10 cups)

Test Steps:

1. Go to the water intake tracking dashboard
2. Click on the "Set Hydration Goal" button
3. Enter the desired hydration goal
4. Click "Save"
Postconditions:
? Hydration goal is set and displayed on the dashboard
Expected Result: The system successfully sets and saves the user's daily hydration goal.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that reminders are provided to encourage regular water intake
Test Case 1. Verify that reminders are provided to encourage regular water intake
Test Case ID: WIT-004
Test Case Title: Verify that reminders are provided to encourage regular water intake
Test Case Description: Ensure that the system sends reminders to users to drink water at regular intervals.
Test Suite: Water Intake Tracking
Test Priority: Medium
Preconditions:
? User has set a hydration goal
? User has enabled reminders
Test Data: No test data needed
Test Steps:
1. Set a hydration goal
2. Enable reminders
3. Wait for the reminder time interval (e.g., 2 hours)
Postconditions:

? Reminder notification is sent to the user

Expected Result: The system sends reminders to the user to drink water at regular intervals, encouraging

regular water intake.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that users can view hydration progress and trends

Test Case ID: WIT-005

Test Case Title: Verify that users can view hydration progress and trends

Test Case Description: Ensure that the system displays hydration progress and trends to users, allowing them

to track their hydration levels over time.

Test Suite: Water Intake Tracking

Test Priority: High

Preconditions:

? User has logged water intake data

? User has access to the water intake tracking feature

Test Data: Water intake data (e.g., 1 week, 1 month)

Test Steps:

1. Go to the water intake tracking dashboard

2. Click on the "Hydration Progress" tab

3. Check the hydration progress and trends

Postconditions:

? Hydration progress and trends are displayed on the dashboard

Expected Result: The system correctly displays hydration progress and trends to users, enabling them to track

their hydration levels over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 87

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Nutrition Tracking

Feature description -

The app includes a food diary and nutrition tracker, allowing users to log meals, track macronutrient intake, and monitor nutritional balance. Nutrient tracking provides insight into dietary habits, supports informed food choices, and helps users align their nutrition with their fitness goals.

Userstory title -

As a user, I want to track my nutrition so that I can monitor my diet and make informed food choices.

Acceptance criteria -

['Users can log their daily meals and snacks.', 'The app calculates macronutrient intake (carbs, fats, proteins).', 'Users can set nutritional goals and track progress.', 'The app provides nutritional information and suggestions.', 'Users can view their nutritional history and trends.']

Generated Test Cases:

Here are the 5 test cases for the Nutrition Tracking feature of the Fitness Tracker App:

Test Case 1: Verify that users can log their daily meals and snacks

Test Case ID: NT1

Test Case Title: Verify that users can log their daily meals and snacks

Test Case Description: Ensure that the app allows users to log their daily meals and snacks, enabling them to

track their nutrition.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

? User is logged in

? Nutrition tracking feature is enabled

Test Data: Sample meal and snack data (e.g., breakfast, lunch, dinner, snack)

Test Steps:

1. Open the nutrition tracking feature

2. Click on the "Log Meal" button

3. Enter meal details (e.g., food, quantity, time)

4. Click "Save"

Postconditions:

? Meal is successfully logged

Expected Result: The app successfully logs the meal and snack, displaying it in the user's nutrition diary.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that the app calculates macronutrient intake (carbs, fats, proteins)

Test Case ID: NT2

Test Case Title: Verify that the app calculates macronutrient intake (carbs, fats, proteins)

Test Case Description: Ensure that the app accurately calculates the macronutrient intake based on the logged

meals and snacks.

Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

? User has logged meals and snacks

? Nutrition tracking feature is enabled

Test Data: Sample meal data with known macronutrient values

Test Steps:

1. Open the nutrition tracking feature

2. View the macronutrient intake calculation for a logged meal

3. Compare the calculated values with expected values

Postconditions:

? Macronutrient intake is accurately calculated

Expected Result: The app accurately calculates the macronutrient intake (carbs, fats, proteins) based on the

logged meals and snacks.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can set nutritional goals and track progress

Test Case ID: NT3

Test Case Title: Verify that users can set nutritional goals and track progress

Test Case Description: Ensure that the app allows users to set nutritional goals and track their progress towards achieving those goals. Test Suite: Nutrition Tracking

Test Priority: High

Preconditions:

? User is logged in

? Nutrition tracking feature is enabled

Test Data: Sample nutritional goals (e.g., daily calorie intake, macronutrient ratios)

Test Steps:

- 1. Open the nutrition tracking feature
- 2. Click on the "Set Goals" button
- 3. Enter nutritional goal details
- 4. Track progress towards the goal

Postconditions:

? Nutritional goal is successfully set

Expected Result: The app allows users to set nutritional goals and tracks their progress towards achieving those goals.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case ID: NT4

Test Case Title: Verify that the app provides nutritional information and suggestions

Test Case 4: Verify that the app provides nutritional information and suggestions

Test Case Description: Ensure that the app provides users with relevant nutritional information and suggestions to help them make informed food choices.

Test Suite: Nutrition Tracking Test Priority: Medium Preconditions: ? User has logged meals and snacks ? Nutrition tracking feature is enabled Test Data: Sample meal data Test Steps: 1. Open the nutrition tracking feature 2. View the nutritional information and suggestions for a logged meal 3. Check if the suggestions are relevant and helpful Postconditions: ? Nutritional information and suggestions are provided Expected Result: The app provides users with relevant nutritional information and suggestions to help them make informed food choices. Severity: Minor Type of Testing: Functional Test Case Approach: Positive **Test Case 5: Verify that users can view their nutritional history and trends** Test Case ID: NT5 Test Case Title: Verify that users can view their nutritional history and trends Test Case Description: Ensure that the app allows users to view their nutritional history and trends, enabling them to track their progress over time.

Test Suite: Nutrition Tracking

Test Priority: Medium

Preconditions:

? User has logged meals and snacks

? Nutrition tracking feature is enabled

Test Data: No test data needed

Test Steps:

1. Open the nutrition tracking feature

2. View the nutritional history and trends

3. Check if the data is accurately displayed

Postconditions:

? Nutritional history and trends are displayed

Expected Result: The app allows users to view their nutritional history and trends, enabling them to track their progress 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:

Here are the 5 detailed test cases for the Progress Reports feature of the Fitness Tracker App:

Test Case 1

Test Case ID: PR001

Test Case Title: Verify that users receive weekly progress reports

Test Case Description: Ensure that the system sends weekly progress reports to users, summarizing their

activity levels, workout performance, and goal attainment.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

* User is logged in

* User has completed at least one week of activity tracking

Test Data: No test data needed

Test Steps:

1. Log in to the app

2. Navigate to the Progress Reports section

3. Verify that a weekly progress report is generated and sent to the user

4. Review the report to ensure it includes activity levels, workout performance, and goal attainment

Postconditions:

* Weekly progress report is sent to the user

Expected Result: The system sends a weekly progress report to the user, summarizing their activity levels,

workout performance, and goal attainment.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2

Test Case ID: PR002

Test Case Title: Verify that reports include activity levels, workout performance, and goal attainment

Test Case Description: Ensure that the progress reports include detailed information about the user's activity

levels, workout performance, and goal attainment.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

* User is logged in

* User has completed at least one week of activity tracking

Test Data: No test data needed

Test Steps:

1. Log in to the app

2. Navigate to the Progress Reports section

3. Review the weekly progress report to ensure it includes activity levels, workout performance, and goal

attainment

4. Verify that the report provides a detailed breakdown of achievements and trends

Postconditions:

* Report includes activity levels, workout performance, and goal attainment

Expected Result: The system includes detailed information about the user's activity levels, workout

performance, and goal attainment in the progress report.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3

Test Case ID: PR003

Test Case Title: Verify that users can view detailed breakdowns of achievements and trends

Test Case Description: Ensure that the system provides users with a detailed breakdown of their achievements

and trends in the progress report.

Test Suite: Progress Reports

Test Priority: High

Preconditions:

* User is logged in

* User has completed at least one week of activity tracking

Test Data: No test data needed

Test Steps:

- 1. Log in to the app
- 2. Navigate to the Progress Reports section
- 3. Review the weekly progress report to ensure it provides a detailed breakdown of achievements and trends
- 4. Verify that the breakdown is accurate and relevant to the user's activity data

Postconditions:

* Detailed breakdown of achievements and trends is provided

Expected Result: The system provides a detailed breakdown of the user's achievements and trends in the

progress report.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4

Test Case ID: PR004

Test Case Title: Verify that the app provides recommendations based on report findings

Test Case Description: Ensure that the system provides personalized recommendations to users based on their

progress report findings.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* User is logged in

* User has completed at least one week of activity tracking

Test Data: No test data needed

Test Steps:

- 1. Log in to the app
- 2. Navigate to the Progress Reports section
- 3. Review the weekly progress report to ensure it provides personalized recommendations
- 4. Verify that the recommendations are relevant and actionable

Postconditions:

* Personalized recommendations are provided

Expected Result: The system provides personalized recommendations to the user based on their progress

report findings.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5

Test Case ID: PR005

Test Case Title: Verify that users can customize the frequency and content of reports

Test Case Description: Ensure that the system allows users to customize the frequency and content of their

progress reports.

Test Suite: Progress Reports

Test Priority: Medium

Preconditions:

* User is logged in

* User has completed at least one week of activity tracking

Test Data: No test data needed

Test Steps:

1. Log in to the app

2. Navigate to the Progress Reports section

3. Verify that the user can customize the frequency of reports (e.g., weekly, monthly)

4. Verify that the user can customize the content of reports (e.g., activity levels, workout performance, goal

attainment)

Postconditions:

* User can customize report frequency and content

Expected Result: The system allows users to customize the frequency and content of their progress 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:

Here are five detailed test cases for the Fitness Tracker App's In-App Challenges feature, ensuring thorough coverage of the acceptance criteria:



Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2: Verify that challenges have clear objectives and rules
Test Case ID: FT-002
Test Case Title: Verify that challenges have clear objectives and rules
Test Case Description: Ensure that each challenge provides clear objectives, rules, and requirements, allowing
users to understand what is expected to achieve the challenge.
Test Suite: In-App Challenges
Test Priority: Medium
Preconditions:
* User is logged in
* A challenge is selected
Test Data: Challenge details
Test Steps:
1. View challenge details
2. Verify the challenge objective
3. Review challenge rules and requirements
4. Check for any specific instructions or restrictions
Postconditions:

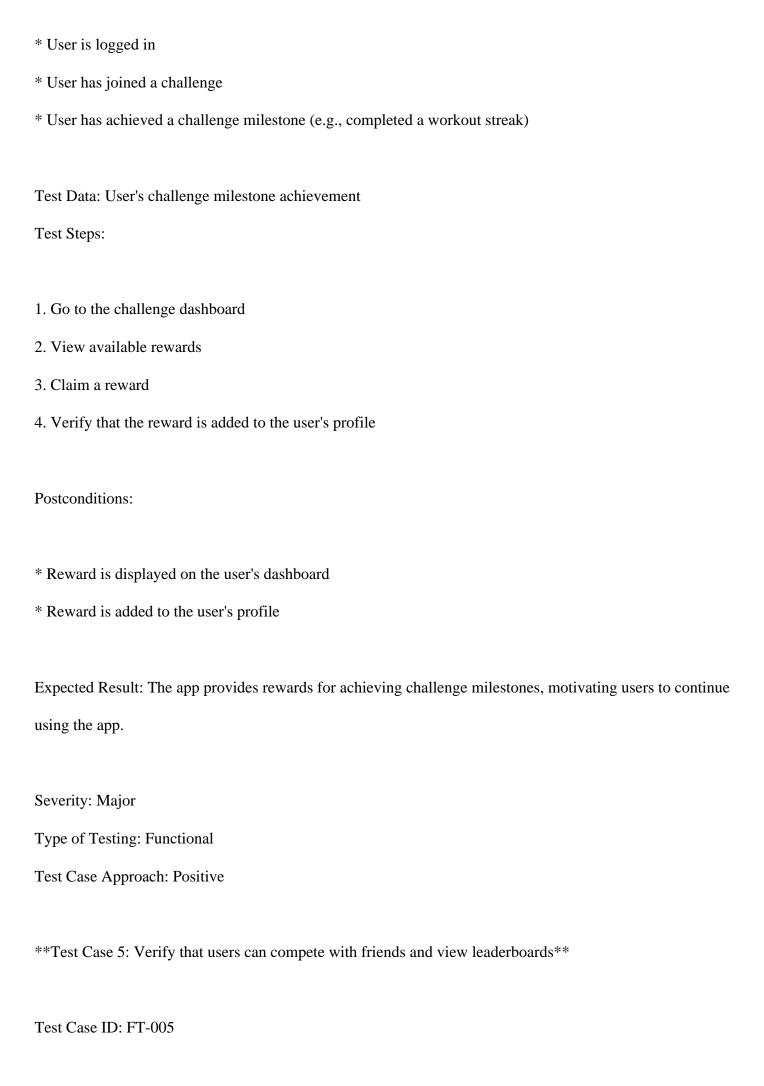
* Challenge objective is clear and concise * Challenge rules and requirements are well-defined Expected Result: The app provides clear objectives and rules for each challenge, enabling users to understand what is required to achieve the challenge. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that users can track their progress within the challenge** Test Case ID: FT-003 Test Case Title: Verify that users can track their progress within the challenge Test Case Description: Ensure that the app allows users to track their progress within the challenge, providing a sense of accomplishment and motivation. Test Suite: In-App Challenges Test Priority: High Preconditions: * User is logged in * User has joined a challenge

Test Data: User's challenge progress

Test Steps:

1. Go to the challenge dashboard 2. View current progress (e.g., steps taken, calories burned, distance traveled) 3. Perform an action that contributes to the challenge (e.g., complete a workout) 4. Verify that the progress is updated Postconditions: * Progress is updated in real-time * Progress is reflected on the challenge leaderboard Expected Result: The app allows users to track their progress within the challenge, providing an engaging experience. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 4: Verify that rewards are provided for achieving challenge milestones** Test Case ID: FT-004 Test Case Title: Verify that rewards are provided for achieving challenge milestones Test Case Description: Ensure that the app rewards users for achieving challenge milestones, providing an incentive to stay motivated and continue using the app. Test Suite: In-App Challenges Test Priority: Medium

Preconditions:



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, adding a social and competitive element to the fitness tracking experience. Test Suite: In-App Challenges Test Priority: High Preconditions: * User is logged in * User has joined a challenge * Friends are connected on the app Test Data: User's friends' progress in the challenge Test Steps: 1. Go to the challenge dashboard 2. View the leaderboard 3. Compare progress with friends 4. Engage in a friendly competition (e.g., send a motivational message) Postconditions: * Leaderboard displays user and friends' progress * Users can compare progress and engage in a friendly competition Expected Result: The app enables users to compete with friends and view leaderboards, enhancing the overall fitness tracking experience.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 90

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Customizable Dashboards

Feature description -

Users can customize their app dashboard to display relevant fitness metrics, goals, and progress trackers based on their preferences and priorities. Customizable dashboards provide a personalized user experience, allowing users to focus on the metrics that matter most to them.

Userstory title -

As a user, I want to customize my dashboard so that I can focus on the fitness metrics that are most important to me.

Acceptance criteria -

['Users can select which metrics to display on their dashboard.', 'The app allows customization of dashboard layout and theme.', 'Users can save and update their dashboard settings.', 'Dashboard displays are dynamic and update in real-time.', 'Users can reset the dashboard to default settings.']

Generated Test Cases:

Here are the	five test cases	for the Fitness	Tracker App's	Customizable	Dashboards feature:

Test Case 1: Verify that users can select which metrics to display on their dashboard

Test Case ID: FTC-001

Test Case Title: Verify that users can select which metrics to display on their dashboard

Test Case Description: Ensure that users can choose the relevant fitness metrics to display on their dashboard

based on their preferences.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in
- Dashboard is displayed

Test Data: No test data needed

Test Steps:

- 1. Log in to the app
- 2. Navigate to the dashboard
- 3. Click on the "Customize" button
- 4. Select desired metrics to display (e.g., steps taken, calories burned, distance traveled)
- 5. Save changes

Postconditions:

- Dashboard updated with selected metrics

Expected Result: The app allows users to select which metrics to display on their dashboard.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that the app allows customization of dashboard layout and theme

Test Case ID: FTC-002

Test Case Title: Verify that the app allows customization of dashboard layout and theme

Test Case Description: Ensure that users can customize the layout and theme of their dashboard to suit their

preferences.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in

- Dashboard is displayed

Test Data: No test data needed

Test Steps:

1. Log in to the app

2. Navigate to the dashboard

3. Click on the "Customize" button

4. Select a desired layout (e.g., grid, list, chart)

5. Choose a theme (e.g., dark, light, custom)

6. Save changes

Postconditions:

- Dashboard updated with selected layout and theme

Expected Result: The app allows customization of dashboard layout and theme.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users can save and update their dashboard settings

Test Case ID: FTC-003

Test Case Title: Verify that users can save and update their dashboard settings

Test Case Description: Ensure that users can save their customized dashboard settings and update them later as

needed.

Test Suite: Customizable Dashboards

Test Priority: Medium

Preconditions:

- User is logged in

- Dashboard is customized

Test Data: No test data needed

Test Steps:

1. Log in to the app

- 2. Navigate to the customized dashboard
- 3. Make changes to the dashboard settings (e.g., add/remove metrics, change layout)
- 4. Save changes
- 5. Log out and log back in
- 6. Verify that dashboard settings are retained

Postconditions:

- Dashboard settings are saved and updated successfully

Expected Result: The app saves and updates user dashboard settings correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that dashboard displays are dynamic and update in real-time

Test Case ID: FTC-004

Test Case Title: Verify that dashboard displays are dynamic and update in real-time

Test Case Description: Ensure that the dashboard displays update dynamically in real-time as the user's fitness

data changes.

Test Suite: Customizable Dashboards

Test Priority: High

Preconditions:

- User is logged in

- Dashboard is customized

Test Data: No test data needed

Test Steps:

1. Log in to the app

2. Navigate to the customized dashboard

3. Perform an activity that updates fitness data (e.g., take a few steps, burn calories)

4. Verify that dashboard metrics update in real-time

Postconditions:

- Dashboard metrics update dynamically and in real-time

Expected Result: The app updates dashboard displays dynamically and in real-time.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that users can reset the dashboard to default settings

Test Case ID: FTC-005

Test Case Title: Verify that users can reset the dashboard to default settings
Test Case Description: Ensure that users can reset their customized dashboard to the default settings if needed.
Test Suite: Customizable Dashboards
Test Priority: Medium
Preconditions:
- User is logged in
- Dashboard is customized
Test Data: No test data needed
Test Steps:
1. Log in to the app
2. Navigate to the customized dashboard
3. Click on the "Reset to Default" button
4. Verify that dashboard settings are reset to default
Postconditions:
- Dashboard settings are reset to default
Expected Result: The app allows users to reset the dashboard to default settings.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Data - 91
Project name -
Fitness Tracker App
Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

Feature name -

Social Sharing

Feature description -

Users can share workout summaries, achievements, and progress updates on social media platforms, inspiring

others and fostering accountability. Social sharing enables users to celebrate their successes, seek support from

their social networks, and stay motivated on their fitness journey.

Userstory title -

As a user, I want to share my fitness achievements on social media so that I can inspire others and stay

motivated.

Acceptance criteria -

['Users can share workout summaries and achievements.', 'The app supports sharing on multiple social media

platforms.', 'Users can customize the content and format of shared posts.', 'Sharing features are integrated

seamlessly with social media.', 'Users can track the engagement of their shared posts.']

Generated Test Cases:

Here are the 5 test cases for the Social Sharing feature of the Fitness Tracker App:

Test Case 1: Verify that users can share workout summaries and achievements on social media

Test Case ID: FSA1

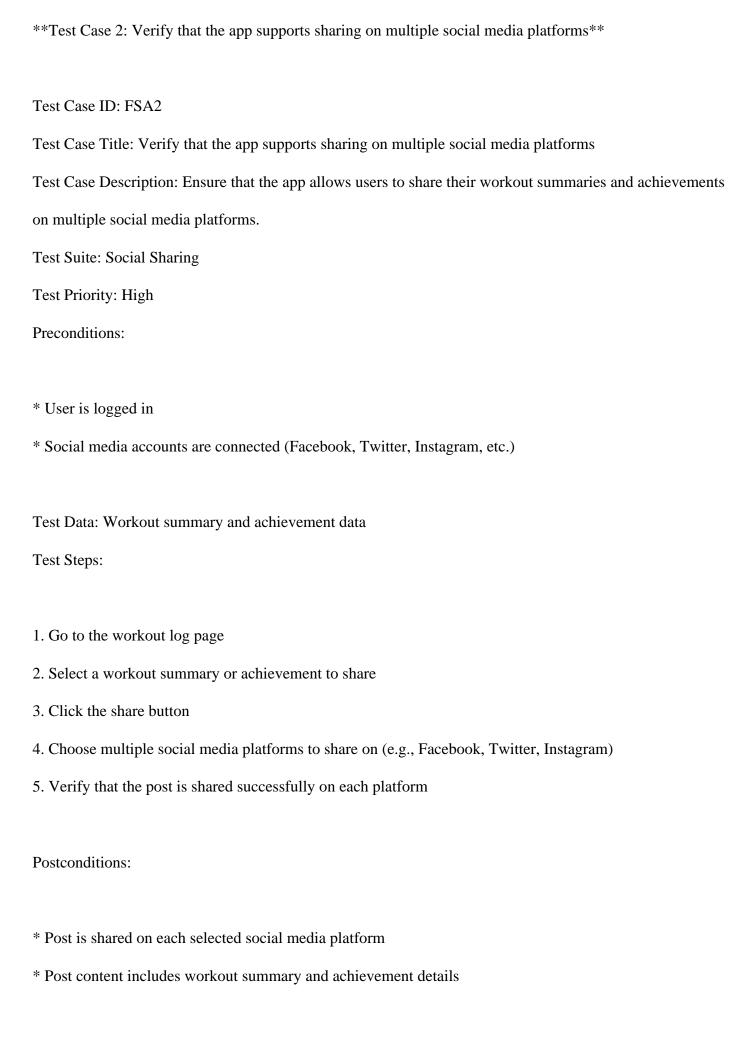
Test Case Title: Verify that users can share workout summaries and achievements on social media

Test Case Description: Ensure that the app allows users to share their workout summaries and achievements

on social media platforms.

Test Suite: Social Sharing

Test Priority: High
Preconditions:
* User is logged in
* Workout data is available
* Social media account is connected
Test Data: Workout summary and achievement data
Test Steps:
1. Go to the workout log page
2. Select a workout summary or achievement to share
3. Click the share button
4. Choose a social media platform to share on
5. Verify that the post is shared successfully
Postconditions:
* Post is shared on the selected social media platform
* Post content includes workout summary and achievement details
Expected Result: The app allows users to share their workout summaries and achievements on social media
platforms.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive



Expected Result: The app supports sharing on multiple social media platforms.
Severity: Critical
Type of Testing: Integration
Test Case Approach: Positive
Test Case 3: Verify that users can customize the content and format of shared posts
Test Case ID: FSA3
Test Case Title: Verify that users can customize the content and format of shared posts
Test Case Description: Ensure that the app allows users to customize the content and format of their shared
workout summaries and achievements.
Test Suite: Social Sharing
Test Priority: Medium
Preconditions:
* User is logged in
* Workout data is available
* Social media account is connected
Test Data: Workout summary and achievement data
Test Steps:
1. Go to the workout log page
2. Select a workout summary or achievement to share
3. Click the share button
4. Customize the post content and format (e.g., add a photo, change the text)

5. Verify that the customized post is shared successfully

Postconditions:
* Customized post is shared on the selected social media platform
* Post content includes the customized details
Expected Result: The app allows users to customize the content and format of their shared workout summaries
and achievements.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that sharing features are integrated seamlessly with social media
Test Case ID: FSA4
Test Case Title: Verify that sharing features are integrated seamlessly with social media
Test Case Description: Ensure that the app's sharing features are integrated seamlessly with social media
platforms, allowing users to share their workout summaries and achievements without issues.
Test Suite: Social Sharing
Test Priority: High
Preconditions:
* User is logged in
* Social media account is connected
Test Data: Workout summary and achievement data
Test Steps:

- 1. Go to the workout log page
- 2. Select a workout summary or achievement to share
- 3. Click the share button
- 4. Verify that the app redirects to the social media platform's sharing page
- 5. Verify that the post is shared successfully

Postconditions:

- * Post is shared on the selected social media platform
- * No errors or issues occur during the sharing process

Expected Result: The app's sharing features are integrated seamlessly with social media platforms.

Severity: Blocker

Type of Testing: Integration

Test Case Approach: Positive

Test Case 5: Verify that users can track the engagement of their shared posts

Test Case ID: FSA5

Test Case Title: Verify that users can track the engagement of their shared posts

Test Case Description: Ensure that the app allows users to track the engagement of their shared workout

summaries and achievements, including likes, comments, and shares.

Test Suite: Social Sharing

Test Priority: Medium

Preconditions:

* User is logged in
* Social media account is connected
* Post is shared on a social media platform
Test Data: Shared post data
Test Steps:
1. Go to the shared post on the social media platform
2. Verify that the app displays the engagement metrics (likes, comments, shares)
3. Verify that the engagement metrics are updated in real-time
Postconditions:
* Engagement metrics are displayed accurately
* Engagement metrics are updated in real-time
Expected Result: The app allows users to track the engagement of their shared workout summaries and
achievements.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Data - 92
Project name -
Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

Feature name -

Rest and Recovery Tracking

Feature description -

The app includes features for tracking rest days, recovery activities, and recovery metrics such as resting heart

rate and sleep quality. Rest and recovery tracking helps users prioritize recovery, prevent overtraining, and

optimize their overall health and performance.

Userstory title -

As a user, I want to track my rest and recovery so that I can optimize my fitness performance and prevent

overtraining.

Acceptance criteria -

['Users can log rest days and recovery activities.', 'The app tracks metrics such as resting heart rate and sleep

quality.', 'Users receive recommendations for recovery based on their activities.', 'The app provides alerts for

insufficient recovery time.', 'Users can view trends and history of their recovery data.']

Generated Test Cases:

Here are the 5 test cases for the Rest and Recovery Tracking feature of the Fitness Tracker App:

Test Case 1: Verify that users can log rest days and recovery activities

Test Case ID: RT001

Test Case Title: Verify that users can log rest days and recovery activities

Test Case Description: Ensure that users can log rest days and recovery activities in the app, including

specifying the type and duration of the activity.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User is logged in

- User has a registered wearable device or fitness equipment

Test Data: Rest day and recovery activity details (e.g., type, duration)

Test Steps:

1. Go to the app's dashboard

2. Click on the "Log Activity" button

3. Select "Rest Day" or "Recovery Activity" as the activity type

4. Enter the activity details (e.g., type, duration)

5. Save the activity

Postconditions:

- The activity is logged in the user's profile

Expected Result: The system allows users to log rest days and recovery activities successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that the app tracks metrics such as resting heart rate and sleep quality

Test Case ID: RT002

Test Case Title: Verify that the app tracks metrics such as resting heart rate and sleep quality

Test Case Description: Ensure that the app tracks and displays resting heart rate and sleep quality metrics for

users who have connected wearable devices or fitness equipment.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium Preconditions: - User has a connected wearable device or fitness equipment - User has enabled heart rate and sleep tracking Test Data: No test data needed Test Steps: 1. Go to the app's dashboard 2. Click on the "Recovery" tab 3. Check the resting heart rate and sleep quality metrics Postconditions: - The metrics are displayed in the user's profile Expected Result: The system successfully tracks and displays resting heart rate and sleep quality metrics. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3: Verify that users receive recommendations for recovery based on their activities** Test Case ID: RT003 Test Case Title: Verify that users receive recommendations for recovery based on their activities

Test Case Description: Ensure that the app provides personalized recovery recommendations to users based on their logged activities and progress.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- User has logged activities and progress
- User has enabled recovery recommendations

Test Data: No test data needed

Test Steps:

1. Go to the app's dashboard

2. Click on the "Recovery" tab

3. Check the recommended recovery activities and tips

Postconditions:

- The recommendations are displayed in the user's profile

Expected Result: The system provides personalized recovery recommendations to users based on their activities and progress.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that the app provides alerts for insufficient recovery time

Test Case ID: RT004

Test Case Title: Verify that the app provides alerts for insufficient recovery time

Test Case Description: Ensure that the app sends alerts to users when they have insufficient recovery time, warning them of potential overtraining risks.

Test Suite: Rest and Recovery Tracking

Test Priority: High

Preconditions:

- User has set a recovery goal

- User has insufficient recovery time

Test Data: No test data needed

Test Steps:

1. Go to the app's dashboard

2. Check for alerts and notifications

Postconditions:

- The alert is displayed in the user's profile

Expected Result: The system sends alerts to users when they have insufficient recovery time.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that users can view trends and history of their recovery data

Test Case ID: RT005

Test Case Title: Verify that users can view trends and history of their recovery data

Test Case Description: Ensure that the app allows users to view their recovery data trends and history,

enabling them to track their progress and make informed decisions.

Test Suite: Rest and Recovery Tracking

Test Priority: Medium

Preconditions:

- User has logged recovery data
- User has enabled data tracking

Test Data: No test data needed

Test Steps:

- 1. Go to the app's dashboard
- 2. Click on the "Recovery" tab
- 3. Check the recovery data trends and history

Postconditions:

- The data is displayed in the user's profile

Expected Result: The system allows users to view their recovery data trends and history successfully.

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.']

Here are the five test cases for the Audio Coaching feature in the Fitness Tracker App:
Test Case 1:
Test Case ID: AC1
Test Case Title: Verify that users can access a variety of audio-guided workouts
Test Case Description: Ensure that the app provides access to a diverse range of audio-guided workouts, such
as cardio, strength training, yoga, and others.
Test Suite: Audio Coaching
Test Priority: High
Preconditions:
* User is logged in
* Audio coaching feature is enabled
Test Data: No test data needed
Test Steps:
1. Open the app and navigate to the workouts section
2. Select the audio-guided workouts option
3. Browse through the available workout types
4. Select a workout and start the audio coaching session
Postconditions:
* Audio coaching session starts successfully

Generated Test Cases:

Expected Result: The app provides access to a variety of audio-guided workouts for users to choose from.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2:
Test Case ID: AC2
Test Case Title: Verify that audio coaching includes motivational messages and coaching cues
Test Case Description: Ensure that the audio coaching feature provides motivational messages and coaching
cues during the workout to enhance user engagement and adherence.
Test Suite: Audio Coaching
Test Priority: High
Preconditions:
* User is logged in
* Audio coaching feature is enabled
Test Data: No test data needed
Test Steps:
1. Start an audio-guided workout session
2. Listen to the audio coaching cues and motivational messages
3. Monitor the workout progress and adjust paceaccording to the coaching cues
Postconditions:
* Motivational messages and coaching cues are provided throughout the workout

Expected Result: The audio coaching feature includes motivational messages and coaching cues to enhance
user engagement and adherence.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: AC3
Test Case Title: Verify that users can choose coaching based on workout type and intensity
Test Case Description: Ensure that users can select the type of coaching and intensity level that suits their
workout needs and preferences.
Test Suite: Audio Coaching
Test Priority: Medium
Preconditions:
* User is logged in
* Audio coaching feature is enabled
Test Data: No test data needed
Test Steps:
1. Navigate to the workouts section
2. Select the workout type (e.g., cardio, strength training)
3. Choose the intensity level (e.g., beginner, intermediate, advanced)
4. Start the audio coaching session

Postconditions:

* Audio coaching session starts with the selected workout type and intensity level
Expected Result: The app allows users to choose coaching based on workout type and intensity level.
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4:
Test Case ID: AC4
Test Case Title: Verify that audio coaching integrates seamlessly with other app features
Test Case Description: Ensure that the audio coaching feature integrates smoothly with other app features,
such as workout logging, goal setting, and progress tracking.
Test Suite: Audio Coaching
Test Priority: High
Preconditions:
* User is logged in
* Audio coaching feature is enabled
Test Data: No test data needed
Test Steps:
1. Start an audio-guided workout session
2. Monitor the workout progress and sync with the app's logging feature
3. Review the workout summary and adjust goals accordingly
Postconditions:

* Audio coaching session integrates seamlessly with other app features

Expected Result: The audio coaching feature integrates smoothly with other app features to provide a

comprehensive fitness tracking experience.

Severity: Critical

Type of Testing: Integration

Test Case Approach: Positive

Test Case 5:

Test Case ID: AC5

Test Case Title: Verify that users can control audio settings and preferences

Test Case Description: Ensure that users can customize audio settings and preferences to their liking, such as

volume, playback speed, and audio language.

Test Suite: Audio Coaching

Test Priority: Medium

Preconditions:

* User is logged in

* Audio coaching feature is enabled

Test Data: No test data needed

Test Steps:

1. Navigate to the audio coaching settings

2. Adjust the audio volume, playback speed, and language preferences

3. Save the changes and start an audio-guided workout session

Postconditions:

* Audio coaching session reflects the updated settings and preferences

Expected Result: The app allows users to control audio settings and preferences to their liking.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 94

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Integration with Health Apps

Feature description -

The app integrates with health and wellness apps such as calorie counters, meditation apps, and mental health trackers, enabling users to consolidate their health data and gain a comprehensive view of their overall well-being. Integration with health apps promotes holistic health management and synergy between different aspects of wellness.

Userstory title -

As a user, I want to integrate the fitness tracker app with other health apps so that I can have a comprehensive

view of my well-being.

Acceptance criteria -

['The app integrates with popular health and wellness apps.', 'Users can sync data from connected health apps.',

'The app provides a consolidated view of health metrics.', 'Users receive insights based on combined data from

multiple apps.', 'Integration settings are easy to manage and configure.']

Generated Test Cases:

Here are the 5 test cases for the given input data:

**Test Case 1: Verify that the app integrates with popular health and wellness apps. **

Test Case ID: FT001

Test Case Title: Verify that the app integrates with popular health and wellness apps.

Test Case Description: This test case ensures that the fitness tracker app integrates with popular health and

wellness apps, enabling users to consolidate their health data.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User is logged in to the app

- Popular health and wellness apps are installed on the device

Test Data: No test data needed

Test Steps:

1. Go to the integration settings in the app

2. Select a popular health and wellness app (e.g., MyFitnessPal)

3. Authenticate with the selected app

4. Authorize data sharing between the apps

Postconditions:

- Integration is successful

Expected Result: The app integrates with the selected popular health and wellness app, enabling data sharing.

Severity: Critical

Type of Testing: Integration Testing

Test Case Approach: Positive

Test Case 2: Verify that users can sync data from connected health apps.

Test Case ID: FT002

Test Case Title: Verify that users can sync data from connected health apps.

Test Case Description: This test case ensures that users can sync data from connected health apps, allowing

them to access consolidated health data.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User is logged in to the app

- Health app integration is successful

Test Data: Sample health data from connected app

Test Steps:

- 1. Go to the data sync settings in the app
- 2. Select the connected health app
- 3. Initiate data sync
- 4. Verify that data is synced successfully

Postconditions:

- Data is synced successfully

Expected Result: The app syncs data from the connected health app, providing users with a comprehensive view of their health data.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 3: Verify that the app provides a consolidated view of health metrics.

Test Case ID: FT003

Test Case Title: Verify that the app provides a consolidated view of health metrics.

Test Case Description: This test case ensures that the app provides a consolidated view of health metrics,

enabling users to track their overall well-being.

Test Suite: Integration with Health Apps

Test Priority: High

Preconditions:

- User is logged in to the app
- Data is synced from connected health app

Test Data: No test data needed

Test Steps:

- 1. Go to the health metrics dashboard
- 2. Verify that health metrics from connected apps are displayed
- 3. Verify that metrics are accurate and up-to-date

Postconditions:

- Consolidated view of health metrics is displayed

Expected Result: The app provides a consolidated view of health metrics, enabling users to track their overall

well-being.

Severity: Major

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 4: Verify that users receive insights based on combined data from multiple apps.

Test Case ID: FT004

Test Case Title: Verify that users receive insights based on combined data from multiple apps.

Test Case Description: This test case ensures that users receive insights based on combined data from multiple

apps, helping them make informed decisions about their health.

Test Suite: Integration with Health Apps

Test Priority: Medium

Preconditions:

- User is logged in to the app

- Data is synced from multiple connected health apps

Test Data: Sample health data from multiple connected apps

Test Steps:

1. Go to the insights section in the app

2. Verify that insights are generated based on combined data

3. Verify that insights are accurate and relevant

Postconditions:

- Insights are generated based on combined data

Expected Result: The app provides users with insights based on combined data from multiple apps, helping

them make informed decisions about their health.

Severity: Minor

Type of Testing: Functional Testing

Test Case Approach: Positive

Test Case 5: Verify that integration settings are easy to manage and configure.

Test Case ID: FT005

Test Case Title: Verify that integration settings are easy to manage and configure.

Test Case Description: This test case ensures that integration settings are easy to manage and configure,

providing a seamless user experience.

Test Suite: Integration with Health Apps

Test Priority: Low

Preconditions:

- User is logged in to the app

Test Data: No test data needed

Test Steps:

- 1. Go to the integration settings in the app
- 2. Verify that settings are easily accessible and understandable
- 3. Verify that changes can be made to integration settings

Postconditions:

- Integration settings are easily managed and configured

Expected Result: The app provides an intuitive and user-friendly interface for managing and configuring integration settings.

Severity: Trivial

Type of Testing: Usability Testing

Test Case Approach: Positive

Data - 95

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health

metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features

like workout logging, goal setting, progress tracking, personalized recommendations, and integration with

wearable devices or fitness equipment.

Feature name -

Body Composition Tracking

Feature description -

Users can track changes in body composition metrics such as weight, body fat percentage, muscle mass, and

BMI over time. Body composition tracking provides insights into changes in physique, fitness progress, and

the effectiveness of exercise and nutrition interventions.

Userstory title -

As a user, I want to track my body composition so that I can monitor my fitness progress and make informed

decisions about my training and nutrition.

Acceptance criteria -

['Users can log body composition metrics such as weight, body fat percentage, and muscle mass.', 'The app

calculates and displays BMI based on logged data.', 'Users can view trends and history of body composition

changes.', 'The app provides insights and recommendations based on body composition data.', 'Users can set

goals for body composition metrics.']

Generated Test Cases:

Here are the 5 test cases for the Body Composition Tracking feature of the Fitness Tracker App:

Test Case 1: Verify that users can log body composition metrics

Test Case ID: BT001

Test Case Title: Verify that users can log body composition metrics

Test Case Description: Ensure that users can log body composition metrics such as weight, body fat

percentage, and muscle mass.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:

- User is logged in

- User has access to the body composition tracking feature

Test Data: Sample body composition metrics data (e.g., weight: 70kg, body fat percentage: 20%, muscle mass:

40kg)

Test Steps:

1. Go to the body composition tracking feature

2. Enter the body composition metrics data

3. Save the data

Postconditions:

- Data is saved successfully

Expected Result: The system allows users to log body composition metrics successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that the app calculates and displays BMI based on logged data

Test Case ID: BT002

Test Case Title: Verify that the app calculates and displays BMI based on logged data

Test Case Description: Ensure that the app calculates and displays BMI based on the logged body composition

metrics data.

Test Suite: Body Composition Tracking

Test Priority: High

Preconditions:
- User has logged body composition metrics data
Test Data: Sample body composition metrics data (e.g., weight: 70kg, height: 175cm)
Test Steps:
1. Go to the body composition tracking feature
2. View the calculated BMI
3. Verify the calculations
Postconditions:
- BMI is calculated and displayed correctly
Expected Result: The system calculates and displays BMI correctly based on the logged data.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3: Verify that users can view trends and history of body composition changes
Test Case ID: BT003

Test Case Title: Verify that users can view trends and history of body composition changes

Test Case Description: Ensure that users can view trends and history of body composition changes over time.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User has logged multiple body composition metrics data points

Test Data: Sample body composition metrics data with multiple data points

Test Steps:

- 1. Go to the body composition tracking feature
- 2. View the trend and history of body composition changes

3. Verify the data is displayed correctly

Postconditions:

- Trend and history of body composition changes are displayed correctly

Expected Result: The system displays the trend and history of body composition changes correctly.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that the app provides insights and recommendations based on body composition data

Test Case ID: BT004

Test Case Title: Verify that the app provides insights and recommendations based on body composition data

Test Case Description: Ensure that the app provides insights and recommendations based on the logged body

composition metrics data.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User has logged body composition metrics data

Test Data: Sample body composition metrics data

Test Steps:

1. Go to the body composition tracking feature

2. View the insights and recommendations

3. Verify the relevance of the insights and recommendations

Postconditions:

- Insights and recommendations are displayed correctly

Expected Result: The system provides relevant insights and recommendations based on the logged data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that users can set goals for body composition metrics

Test Case ID: BT005

. D1002

Test Case Title: Verify that users can set goals for body composition metrics

Test Case Description: Ensure that users can set goals for body composition metrics such as weight, body fat

percentage, and muscle mass.

Test Suite: Body Composition Tracking

Test Priority: Medium

Preconditions:

- User is logged in

- User has access to the body composition tracking feature

Test Data: Sample goal data (e.g., weight goal: 65kg, body fat percentage goal: 18%)

Test Steps:

1. Go to the body composition tracking feature

2. Set a goal for a body composition metric

3. Save the goal

Postconditions:

- Goal is saved successfully

Expected Result: The system allows users to set goals for body composition metrics successfully.

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:

Here are the generated test cases for the provided input data:

Test Case 1:

Test Case ID: WkAn1

Test Case Title: Verify that users can view detailed performance metrics for each workout.

Test Case Description: Ensure that the app provides a detailed view of performance metrics for each workout,

including exercise intensity, duration, and calorie burn.

Test Suite: Workout Analysis

Test Priority: High

Preconditions:

? User is logged in

? Workout data is available

Test Data: Sample workout data with various performance metrics

Test Steps:

1. Go to the workout analysis page

2. Select a workout to analyze

3. Verify that performance metrics are displayed correctly

Postconditions:

? Performance metrics are updated correctly

Expected Result: The app displays detailed performance metrics for each workout, including exercise

intensity, duration, and calorie burn.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: WkAn2

Test Case Title: Verify that the app provides trends and insights into workout intensity and duration.

Test Case Description: Ensure that the app analyzes workout data to provide trends and insights into exercise intensity and duration. Test Suite: Workout Analysis Test Priority: High Preconditions: ? User is logged in ? Workout data is available Test Data: Sample workout data with various intensity and duration levels Test Steps: 1. Go to the workout analysis page 2. Select a workout to analyze 3. Verify that trends and insights into workout intensity and duration are displayed correctly Postconditions: ? Trends and insights are updated correctly Expected Result: The app provides accurate trends and insights into workout intensity and duration. Severity: Major Type of Testing: Functional Test Case Approach: Positive **Test Case 3:** Test Case ID: WkAn3 Test Case Title: Verify that users can track calorie burn and other health metrics. Test Case Description: Ensure that the app allows users to track calorie burn and other health metrics, such as heart rate and distance traveled. Test Suite: Workout Analysis

Test Priority: High

Preconditions:
? User is logged in
? Workout data is available
Test Data: Sample workout data with calorie burn and health metrics
Test Steps:
1. Go to the workout analysis page
2. Select a workout to analyze
3. Verify that calorie burn and health metrics are displayed correctly
Postconditions:
? Calorie burn and health metrics are updated correctly
Expected Result: The app accurately tracks calorie burn and other health metrics.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4:
Test Case ID: WkAn4
Test Case Title: Verify that the app offers recommendations based on workout analysis.
Test Case Description: Ensure that the app provides personalized recommendations for users based on their
workout analysis.
Test Suite: Workout Analysis
Test Priority: Medium
Preconditions:
? User is logged in
? Workout data is available
Test Data: Sample workout data with various performance metrics

Test Steps:
1. Go to the workout analysis page
2. Select a workout to analyze
3. Verify that personalized recommendations are displayed correctly
Postconditions:
? Recommendations are updated correctly
Expected Result: The app provides accurate and relevant recommendations based on workout analysis.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5:
Test Case ID: WkAn5
Test Case Title: Verify that users can compare current workouts with past performance.
Test Case Description: Ensure that the app allows users to compare their current workouts with past
performance metrics.
Test Suite: Workout Analysis
Test Priority: Medium
Preconditions:
? User is logged in
? Historical workout data is available
Test Data: Sample workout data with historical performance metrics
Test Steps:
1. Go to the workout analysis page
2. Select a workout to analyze
3. Verify that users can compare current workouts with past performance metrics

Postconditions:

? Comparison data is updated correctly

Expected Result: The app allows users to compare current workouts with past performance metrics accurately.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Data - 97

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Weather Integration

Feature description -

Integrated weather forecasts and conditions allow users to plan outdoor workouts and activities based on current weather conditions and forecasts. Weather integration helps users make informed decisions, stay safe, and optimize their outdoor exercise routines according to weather conditions.

Userstory title -

As a user, I want to view weather forecasts in the app so that I can plan my outdoor workouts accordingly.

Acceptance criteria -

[The app provides current weather conditions and forecasts.', 'Users can view weather details for different

locations.', 'Weather data is updated in real-time.', 'The app offers recommendations based on weather conditions.', 'Users can receive weather alerts for extreme conditions.'] **Generated Test Cases:** Here are the 5 detailed test cases for the Weather Integration feature of the Fitness Tracker App: **Test Case 1:** Test Case ID: WE_001 Test Case Title: Verify that the app provides current weather conditions Test Case Description: Ensure that the app displays the current weather conditions, including temperature, humidity, and wind speed, for the user's location. Test Suite: Weather Integration Test Priority: High Preconditions: * User is logged in * Location services are enabled Test Data: No test data needed Test Steps: 1. Launch the app 2. Navigate to the weather section 3. Verify that the current weather conditions are displayed 4. Check that the temperature, humidity, and wind speed are accurate

Postconditions:

* Weather data is updated in real-time
Expected Result: The app displays the current weather conditions accurately
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 2:
Test Case ID: WE_002
Test Case Title: Verify that users can view weather details for different locations
Test Case Description: Ensure that the app allows users to select different locations and view their weather
details.
Test Suite: Weather Integration
Test Priority: Medium
Preconditions:
* User is logged in
* Multiple locations are available in the app
Test Data: List of different locations (e.g., New York, London, Paris)
Test Steps:
1. Launch the app
2. Navigate to the weather section
3. Select a different location
4. Verify that the weather details for the selected location are displayed
5. Repeat steps 3-4 for multiple locations
Postconditions:

* Weather data is updated for the selected location
Expected Result: The app displays weather details for different locations accurately
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 3:
Test Case ID: WE_003
Test Case Title: Verify that weather data is updated in real-time
Test Case Description: Ensure that the app updates weather data in real-time, reflecting changes in weather
conditions.
Test Suite: Weather Integration
Test Priority: High
Preconditions:
* User is logged in
* Weather data is available
Test Data: No test data needed
Test Steps:
1. Launch the app
2. Navigate to the weather section
3. Observe the weather data for a specific location
4. Wait for a short period (e.g., 10 minutes)
5. Verify that the weather data has updated to reflect any changes in weather conditions

Postconditions:
* Weather data is updated in real-time
Expected Result: The app updates weather data in real-time, reflecting changes in weather conditions
Severity: Critical
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4:
Test Case ID: WE_004
Test Case Title: Verify that the app offers recommendations based on weather conditions
Test Case Description: Ensure that the app provides workout recommendations based on the current weather
conditions.
Test Suite: Weather Integration
Test Priority: Medium
Preconditions:
* User is logged in
* Weather data is available
Test Data: No test data needed
Test Steps:
1. Launch the app
2. Navigate to the workout section
3. Verify that the app provides workout recommendations based on the current weather conditions
4. Check that the recommendations are relevant and accurate

Postconditions:
* Workout recommendations are undated based on weather conditions
* Workout recommendations are updated based on weather conditions
Expected Result: The app provides workout recommendations based on the current weather conditions
Severity: Minor
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5:
Test Case ID: WE_005
Test Case Title: Verify that users can receive weather alerts for extreme conditions
Test Case Description: Ensure that the app sends weather alerts to users for extreme weather conditions (e.g.,
heavy rain, heatwave, strong winds).
Test Suite: Weather Integration
Test Priority: High
Preconditions:
* User is logged in
* Weather alerts are enabled
Test Data: List of extreme weather conditions (e.g., heavy rain, heatwave, strong winds)
Test Steps:
1. Launch the app
2. Navigate to the weather section
3. Set up weather alerts for extreme conditions
4. Trigger an extreme weather condition (e.g., simulate heavy rain)
5. Verify that the app sends a weather alert to the user

Postconditions:

* Weather alert is sent to the user

Expected Result: The app sends weather alerts to users for extreme weather conditions

Severity: Blocker

Type of Testing: Functional

Test Case Approach: Positive

Data - 98

Project name -

Fitness Tracker App

Project description -

Create a fitness tracker app that allows users to track their physical activity, exercise routines, and health metrics such as steps taken, calories burned, distance traveled, and heart rate. The app should include features like workout logging, goal setting, progress tracking, personalized recommendations, and integration with wearable devices or fitness equipment.

Feature name -

Mood Tracking

Feature description -

Users can track their mood, energy levels, and motivation before and after workouts, providing context for their exercise experiences and insights into the impact of physical activity on mental well-being. Mood tracking promotes self-awareness, mindfulness, and emotional regulation in the context of fitness and exercise.

Userstory title -

As a user, I want to track my mood and energy levels before and after workouts so that I can understand the

impact of exercise on my mental well-being.

Acceptance criteria -

['Users can log their mood, energy levels, and motivation.', 'The app provides insights into mood and energy

trends over time.', 'Users receive recommendations based on their mood and energy data.', 'The app supports

journaling for personal reflections on mood and workouts.', 'Users can set goals for improving mood and

energy levels.']

Generated Test Cases:

Here are the 5 detailed test cases for the Mood Tracking feature of the Fitness Tracker App:

Test Case 1: Verify that users can log their mood, energy levels, and motivation

Test Case ID: MT001

Test Case Title: Verify that users can log their mood, energy levels, and motivation

Test Case Description: Ensure that the app allows users to log their mood, energy levels, and motivation

before and after workouts.

Test Suite: Mood Tracking

Test Priority: High

Preconditions:

- User is logged in

- Workout is scheduled or completed

Test Data: No test data needed

Test Steps:

1. Go to the mood tracking section

2. Select the current mood, energy level, and motivation

3. Save the entry

Postconditions:

- Mood, energy level, and motivation are saved

Expected Result: The app successfully logs the user's mood, energy levels, and motivation.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that the app provides insights into mood and energy trends over time

Test Case ID: MT002

Test Case Title: Verify that the app provides insights into mood and energy trends over time

Test Case Description: Ensure that the app displays user mood and energy trends over time, providing

valuable insights into the impact of exercise on mental well-being.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User has logged mood and energy data for at least 7 days

- User is logged in

Test Data: No test data needed

Test Steps:

1. Go to the mood tracking section

2. View the mood and energy trend graph

3. Analyze the displayed trends

Postconditions:

- Mood and energy trends are displayed accurately

Expected Result: The app provides accurate insights into the user's mood and energy trends over time.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that users receive recommendations based on their mood and energy data

Test Case ID: MT003

Test Case Title: Verify that users receive recommendations based on their mood and energy data

Test Case Description: Ensure that the app provides personalized recommendations to users based on their

mood and energy levels, helping them enhance their mental well-being.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User has logged mood and energy data for at least 3 days

- User is logged in

Test Data: No test data needed

Test Steps:

1. Go to the mood tracking section

2. View the recommended workouts and exercises

3. Analyze the relevance of the recommendations

Postconditions:

- Relevant recommendations are displayed

Expected Result: The app provides personalized recommendations to the user based on their mood and energy

data.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4: Verify that the app supports journaling for personal reflections on mood and workouts

Test Case ID: MT004

Test Case Title: Verify that the app supports journaling for personal reflections on mood and workouts

Test Case Description: Ensure that the app allows users to journal their thoughts and feelings about their mood

and workouts, promoting self-awareness and mindfulness.

Test Suite: Mood Tracking

Test Priority: Medium

Preconditions:

- User is logged in

- User has completed a workout

Test Data: No test data needed

Test Steps:

1. Go to the journaling section

2. Write a journal entry about the workout and mood

3. Save the entry

Postconditions:

- Journal entry is saved successfully

Expected Result: The app supports journaling for personal reflections on mood and workouts, enabling users

to gain deeper insights.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5: Verify that users can set goals for improving mood and energy levels

Test Case ID: MT005

Test Case Title: Verify that users can set goals for improving mood and energy levels

Data - 99

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload

their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Creation

Feature description -

Users can create and share their own recipes on the platform, including ingredients, cooking instructions, and

optional photos. This feature allows users to showcase their culinary creations and contribute to the platform's

recipe database.

Userstory title -

As a user, I want to create and share my own recipes so that I can showcase my culinary creations.

Acceptance criteria -

['Users can input and save ingredients and cooking instructions.', 'Users can upload photos for their recipes.',

"Recipes are saved in the user's profile.", 'Recipes are accessible to other users on the platform.', 'Users receive

a confirmation upon successful recipe submission.']

Generated Test Cases:

Here are the 5 detailed test cases for the Recipe Creation feature:

Test Case 1: Verify that users can input and save ingredients and cooking instructions

Test Case ID: REC001

Test Case Title: Verify that users can input and save ingredients and cooking instructions

Test Case Description: Ensure that users can successfully input and save ingredients and cooking instructions

for their recipes.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

? User is logged in

? Recipe creation form is accessible

Test Data: Sample recipe with ingredients and cooking instructions

Test Steps:

1. Log in to the platform

2. Access the recipe creation form

3. Enter the recipe title, ingredients, and cooking instructions

4. Click "Save" to save the recipe

Postconditions:

? Recipe is saved in the user's profile

Expected Result: The system saves the recipe with ingredients and cooking instructions successfully.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2: Verify that users can upload photos for their recipes

Test Case ID: REC002

Test Case Title: Verify that users can upload photos for their recipes

Test Case Description: Ensure that users can successfully upload photos for their recipes.

Test Suite: Recipe Creation

Test Priority: Medium

Preconditions:

? User is logged in

? Recipe creation form is accessible

Test Data: Sample recipe with a photo

Test Steps:

1. Log in to the platform

- 2. Access the recipe creation form
- 3. Enter the recipe title, ingredients, and cooking instructions
- 4. Upload a photo for the recipe
- 5. Click "Save" to save the recipe

Postconditions:

? Photo is uploaded and associated with the recipe

Expected Result: The system uploads the photo successfully and associates it with the recipe.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3: Verify that recipes are saved in the user's profile

Test Case ID: REC003

Test Case Title: Verify that recipes are saved in the user's profile

Test Case Description: Ensure that recipes are saved in the user's profile after creation.

Test Suite: Recipe Creation

Test Priority: High

Preconditions:

? User is logged in

? Recipe creation form is accessible

Test Data: Sample recipe

Test Steps:

- 1. Log in to the platform
- 2. Access the recipe creation form
- 3. Enter the recipe title, ingredients, and cooking instructions
- 4. Click "Save" to save the recipe

5. Navigate to the user's profile
Postconditions:
? Recipe is listed in the user's profile
Expected Result: The system saves the recipe in the user's profile successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 4: Verify that recipes are accessible to other users on the platform
Test Case ID: REC004
Test Case Title: Verify that recipes are accessible to other users on the platform
Test Case Description: Ensure that recipes created by one user are accessible to other users on the platform.
Test Suite: Recipe Creation
Test Priority: High
Preconditions:
? User is logged in
? Recipe creation form is accessible
? Another user is logged in
Test Data: Sample recipe
Test Steps:
1. Log in to the platform as User A
2. Access the recipe creation form
3. Enter the recipe title, ingredients, and cooking instructions
4. Click "Save" to save the recipe
5. Log in to the platform as User B
6. Search for the recipe

Postconditions:
? Recipe is listed in the search results
Expected Result: The system makes the recipe accessible to other users on the platform successfully.
Severity: Major
Type of Testing: Functional
Test Case Approach: Positive
Test Case 5: Verify that users receive a confirmation upon successful recipe submission
Test Case ID: REC005
Test Case Title: Verify that users receive a confirmation upon successful recipe submission
Test Case Description: Ensure that users receive a confirmation message after successfully submitting a
recipe.
Test Suite: Recipe Creation
Test Priority: Medium
Preconditions:
? User is logged in
? Recipe creation form is accessible
Test Data: Sample recipe
Test Steps:
1. Log in to the platform
2. Access the recipe creation form
3. Enter the recipe title, ingredients, and cooking instructions
4. Click "Save" to save the recipe
5. Verify the confirmation message
Postconditions:
? Confirmation message is displayed

Expected Result: The system displays a confirmation message to the user after successfully submitting a recipe.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Data - 100

Project name -

Recipe Sharing Platform

Project description -

Build a recipe sharing platform where users can discover, save, and share recipes, cooking tips, and culinary inspiration. The platform should include features like recipe search and browse, user-generated content, ratings and reviews, meal planning, grocery list generation, and social sharing. Users should be able to upload their own recipes, photos, and videos, and engage with other users' content through comments and likes.

Feature name -

Recipe Discovery

Feature description -

The platform offers a robust search functionality that allows users to discover new recipes based on keywords, ingredients, cuisine types, dietary preferences, and more. Users can explore a wide variety of recipes to find inspiration for their next meal.

Userstory title -

As a user, I want to search for recipes using keywords and ingredients so that I can find new recipes to try.

Acceptance criteria -

['Users can search for recipes using keywords and ingredients.', 'Search results are relevant and accurately match the search criteria.', 'Users can filter search results by cuisine type, dietary preferences, and other

criteria.', 'Recipes in the search results are accessible and detailed.', 'Users can save search results for future reference.']

Generated Test Cases:

Here are the five detailed test cases for the Recipe Discovery feature of the Recipe Sharing Platform:

Test Case 1:

Test Case ID: RDP_001

Test Case Title: Verify that users can search for recipes using keywords and ingredients

Test Case Description: Ensure that the search functionality allows users to find recipes based on keywords and ingredients.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User is logged in
- Search bar is visible on the homepage

Test Data: Keywords and ingredients for searching (e.g., "chicken", "pasta", "italian")

Test Steps:

- 1. Enter a keyword or ingredient in the search bar
- 2. Click the search icon
- 3. Verify that the search results display relevant recipes

Postconditions:

- Search results are displayed

Expected Result: The system displays a list of recipes that match the search criteria.

Severity: Critical

Type of Testing: Functional

Test Case Approach: Positive

Test Case 2:

Test Case ID: RDP_002

Test Case Title: Verify that search results are relevant and accurately match the search criteria

Test Case Description: Ensure that the search results are accurate and relevant to the user's search query.

Test Suite: Recipe Discovery

Test Priority: High

Preconditions:

- User is logged in
- Search bar is visible on the homepage

Test Data: Keywords and ingredients for searching (e.g., "vegan", "gluten-free", "desserts")

Test Steps:

- 1. Enter a keyword or ingredient in the search bar
- 2. Click the search icon
- 3. Verify that the search results only display recipes that match the search criteria

Postconditions:

- Search results are displayed

Expected Result: The system only displays recipes that match the search criteria.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 3:

Test Case ID: RDP_003

Test Case Title: Verify that users can filter search results by cuisine type, dietary preferences, and other

criteria

Test Case Description: Ensure that the search results can be filtered based on various criteria to provide a more

customized search experience.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

- User is logged in

- Search bar is visible on the homepage

- Search results are displayed

Test Data: Filter options (e.g., cuisine type, dietary preferences, cooking time)

Test Steps:

1. Enter a keyword or ingredient in the search bar

2. Click the search icon

3. Apply filter options to the search results

4. Verify that the filtered search results only display recipes that match the selected criteria

Postconditions:

- Filtered search results are displayed

Expected Result: The system allows users to filter search results based on various criteria.

Severity: Major

Type of Testing: Functional

Test Case Approach: Positive

Test Case 4:

Test Case ID: RDP_004

Test Case Title: Verify that recipes in the search results are accessible and detailed

Test Case Description: Ensure that users can access and view detailed recipe information from the search

results.

Test Suite: Recipe Discovery

Test Priority: Medium

Preconditions:

- User is logged in
- Search bar is visible on the homepage
- Search results are displayed

Test Data: Recipe details (e.g., ingredients, instructions, cooking time)

Test Steps:

- 1. Enter a keyword or ingredient in the search bar
- 2. Click the search icon
- 3. Select a recipe from the search results
- 4. Verify that the recipe details are displayed

Postconditions:

- Recipe details are displayed

Expected Result: The system displays detailed recipe information for each recipe in the search results.

Severity: Minor

Type of Testing: Functional

Test Case Approach: Positive

Test Case 5:

Test Case ID: RDP 005

Test Case Title: Verify that users can save search results for future reference

Test Case Description: Ensure that users can save their search results for later use.

Test Suite: Recipe Discovery

Test Priority: Low

Preconditions:

- User is logged in
- Search bar is visible on the homepage
- Search results are displayed

Test Data: No test data needed

Test Steps:

1. Enter a keyword or ingredient in the search bar

2. Click the search icon

3. Verify that a "Save" or "Favorite" button is available for each recipe

4. Click the "Save" or "Favorite" button for a recipe

5. Verify that the recipe is added to the user's saved recipes

Postconditions:

- Recipe is saved

Expected Result: The system allows users to save search results for future reference.

Severity: Trivial

Type of Testing: Functional

Test Case Approach: Positive