

# 1. Introduction

This test plan describes the approach to be taken in testing the Student Engagement Mobile Application.

## 2. Test Design

### 2.1 Summary of Test Types

Test Type [ID]	Definition
White Box (Unit) Testing [UT]	Covers individual methods, decision branches, and lines of code.
Black Box (Output) Testing [OT]	Validates system output when unit testing is insufficient.
User Acceptance Testing [UAT]	Ensures the software meets functional requirements for end users.

### 2.2 Test Items and Software Risk Issues

This test plan applies White Box (Unit), Black Box (Output), and User Acceptance Testing. Testing needs to be consistently monitored and edited to ensure accuracy to avoid accurate and working software being classified as “broken” and “buggy”. Testing will be created and/or run every occasion the application is updated.

## 3. Features to be tested

Test Case ID	Requirement Key	Test Description	Expected Outcome	Inputs/Actions to test
UT-1	1.1.1 UserID Key	Verify account creation stores the correct Account ID.	The Account ID is stored correctly.	Input Ursinus email to account creation field and verify all UserIDs are discrete
OT-1	1.1.3 Administrative Permissions	Ensure administrative permissions are correctly assigned.	The correct admin permissions are granted.	Mark an account as being an Admin account during creation and verify that the

				field within the User table is marked as true. If an account is not an Admin, ensure they do not have the capability to delete posts and/or poster accounts
OT-2	1.1.5 Post Permissions	Validate posting permissions functionality.	Users with permission from the institution will be able to create posts.	Mark an account as being a poster account during creation and verify that the field within the User table is marked as true. If an account is not
UT-4	1.2.2 Post-Date Key	Verify post creation stores the correct Post-Date Key.	The Post-Date Key is stored correctly.	Send a created post with a Date object within the Post object and verify that the date appears in the correct field within the database (no negative dates, etc.)
UT-5	1.2.3 Likes Key	Verify post likes are stored in the correct Likes Key.	The Likes Key is stored correctly.	Input various integer numbers (eg. 0, -1, 999999) to ensure accurate be
UT-6	1.3 Credentials Interface	Verify a popup appears on the	The interface sends a unique Account Token to	Input various Ursinus emails to ensure that

		screen and prompts for a user email	the database to be stored as a new user or return an existing user	all generate different User Tokens within the database. Then input incorrect usernames, passwords, and incorrect email formats to ensure they do not log into the app.
UT-7	1.4.1 Following List	Verify database correctly stores and transfers the ids of the posters that the current user follows	The following list is accurate and consistently stores and displays the user's	Generate fake User() accounts, set one of the accounts' following list to a select few, and verify the followed accounts appear on following list screen
UT-8	2.2 Institution Selection	Validate institution selection is stored properly.	Institutions are correctly associated with the user.	Login with various institutional emails and ensure that each are recorded as various institutions
UT-9	2.3 Account Type Field	Validate account type field is stored properly.	Account type is stored correctly.	Send accounts with all potential account type fields (poster, admin, both, neither) and ensure correct

				storage and behavior.
UT-10	2.4 Email Field	Validate email field is stored properly.	Email of the user is stored correctly.	Attempt to create an account with no email field create an account with an email field over the char limit (255). Verify legal address are stored correctly in the database.
UT-11	3.1 Admin Account Creation	Ensure an admin account is created properly.	An admin account is created with the appropriate permissions.	Delete Others' posts Delete Others' Accounts
UT-12	4.1.1 Choose Following	Ensure a user can select and unselect other posting accounts to add/remove them from their following list	The user can easily add and remove people from their following list and have the application update during use.	Have a user add some current posters to their following list, ensure they all appear on the following screen, then add all remaining poster accounts, and ensure they all appear on the following list screen upon refresh
UT-13	4.1.2 View Following	Ensure a user can traverse to the following list screen and view who they follow	The user can maintain a following list over multiple instances of the app	Have a user create a following list, close out from the application, and reopen the application to

				ensure proper saving and translation of the following list from the database
UT-14	4.1.3 Delete Following	Ensure a user can remove a poster account from their following list	The user can easily remove an account that they no longer wish to follow	Have a user with an existing following list remove a single poster account, multiple poster accounts, and all accounts, refreshing after each instance to ensure proper behavior in the following posts screen and following list screen
UT-15	4.2.1 Create Post	Ensure a poster account only can access the create post screen, fill out the required title and text text boxes, and post the data to the database.	Only poster accounts can access the create post interface, and then can easily fill out the text boxes to create their post	Have a non-poster account attempt to make a post, and have a poster account fill out the text and title text boxes with no inputs, a very large input, potential SQL injection inputs, and numerical inputs to ensure that the required boxes behave as desired

UT-16	4.2.2 Attach Image	Ensure a poster can attach an image file to their post	Posters have the option of adding an image to their post, but it is not required	Have a poster attempt to add audio, video, image, and text files to the post and ensure the field only accepts image files
UT-17	4.2.3 Attach Link	Ensure a poster can attach a link to an external site to their post	Posters have the option of adding a link to their post, but it is not required	Have a poster attempt to add random strings that are not actual links as well as actual hyperlinks to the post to ensure that only real links are accepted
UT-18	4.2.4 Attach Video	Ensure a poster can attach a video file to their post	Posters have the option of adding a video to their post, but it is not required	Have a poster attempt to add audio, video, image, and text files to the post and ensure the field only accepts video files
UT-19	4.2.5 View Posts	Verify the feed displays all non-deleted posts	All eligible posts are accurately translated from the database and displayed.	Make a test posts and ensure they all appear, then set a select few to deleted to ensure they no longer appear upon app refreshing in both the main page and following page

UT-20	5.2 Following Only Feed	Verify the feed only loads posts from followed accounts.	The user can select a button to filter the feed on the home page to only the accounts which they follow	Create multiple posts from discrete posters and set the current user to follow a subset of posters and ensure that the filtration on the main screen only shows the following account's posts when the button is selected
UT-21	6.1.1 Scrolling	Verify the scroll view scales with list of posts	The user should be able to scroll through all posts uninterrupted unless they refresh the page	Create multiple instances of posts that fill past the view of the scroll view and ensure that the user can scroll to the end (create a very large number of temporary posts that do not need to be saved to the database to ensure that there is no limit to scrolling)
OT-3	6.2.1 Search Button	Verify search button shows all available posting accounts in the view following list screen	The user should be able to search for a posting account by their email to choose to follow them	Create a few posting accounts and ensure that searching based off of the posting account email

				(which is displayed in the posts they have created)
OT-4	6.2.2 Settings Button	Verify settings button directs to the appropriate page	The button loads the correct page	<p>Click the button while logged in with an authorized account</p> <p>Click the button while logged in with an unauthorized account</p> <p>Click the button while logged out</p>
OT-5	6.3.1 Main Page Button	Verify Main Page button directs to the appropriate page	The Button loads the main page	<p>Click the button while logged in with an authorized account</p> <p>Click the button while logged in with an unauthorized account</p> <p>Click the button while logged out</p>
OT-6	6.3.2 Following Page Button	Verify Following page button directs to the Following page	Following Page loads on click while the user is logged in.	<p>Click the button while logged in with an authorized account</p> <p>Click the button while logged in</p>



				<p>with an unauthorized account</p> <p>Click the button while logged out</p>
UT-26	6.3.3 Create a Post Button	Verify Post Creation button directs to the post-creation page	Create a Post Page loads on click while the user is logged into an authorized account.	<p>Click the button while logged in with an authorized account</p> <p>Click the button while logged in with an unauthorized account</p> <p>Click the button while logged out</p>
UT-27	6.6 Post Creation Page	Verify the Post Creation Page has the appropriate fields to create a post	Post Creation page has all fields necessary to create a post	Ensure only authorized accounts have access to post creation page
UT-28	6.6.1 Text Box	Verify the Post Creation Page has the appropriate fields to fill in a text box	Post Creation page has all fields necessary to insert text	Ensure only authorized users can insert text
UT-29	6.6.2 Image Insertion (Aspirant)	Verify the Post Creation Page has the appropriate fields to insert an image	Post creation page has all fields necessary to insert images	Ensure only authorized users can insert images
UT-30	6.6.3 Link Insertion (Aspirant)	Verify the Post Creation Page has the appropriate fields to insert a link	Post creation page has all fields necessary to insert links	Ensure only authorized users can insert links
UT-31	6.6.4 Video Insertion (Aspirant)	Verify the Post Creation Page has the appropriate fields to insert a video	Post creation page has all fields necessary to insert videos	Ensure only authorized users can insert videos

UT-32	6.7 Created Posts Page	Verify the Created Posts Page properly displays users created posts	Created posts are properly displayed	Enter a valid login and upload a valid image. Test if the user creates a post successfully
OT-7	6.7.1 Scroll Through Created Posts	Verify the Created Posts Page can be scrolled through	Created Posts Page's posts can all be viewed by scrolling	Create a list of posts long enough to scroll through, and attempt to scroll and view all posts.
UT-34	6.7.2 Delete Individual Post Button	Verify button flags post as deleted when pressed by the post author.	Button successfully creates a postTag entry with the target post and deleted tag when pressed by the author or an admin.	<p>Create a post and attempt to delete it as the posting account.</p> <p>Attempt to delete post with another posting account.</p> <p>Attempt to delete post as a viewer</p> <p>Attempt to delete a post as an admin.</p>
UT-35	6.8 Following List Page	Verify that following list page only loads when logged in.	Following page is only accessible by registered users.	<p>Attempt to navigate to the Following List Page while logged in.</p> <p>Attempt to navigate to the Following List Page while logged out.</p>

UT-36	6.8.1 List Population	Verify that only eligible accounts populate the list.	List is populated with active accounts the user is following	Follow an account. Delete said account. Navigate to the following list page and verify the deleted account isn't in the list.
UT-37	6.8.2 List Refresh	Verify that list repopulates when refreshed	List repopulates correctly when followers are removed or added.	Have a basic list of followers. Remove a follower. Refresh the list and observe result. Have a basic list of accounts. Follow an Account. Refresh the list and observe result. Make no changes. Refresh the list and observe result
OT-8	6.8.3 Search Bar	Verify search bar input produces results	Argument in the text box produces a list of results.	Enter the name of an account into the search bar, verify that result matches the input.
UT-39	6.10 Settings Page	Verify the settings page is available to the user and accurately turns on and off notifications	The user should be able to go into the settings page and turn off or on the notifications for the app	Create a user, turn on notifications, and return to the home screen and ensure that there is a push notification when a post is

				created, then return to the app, turn off the notifications, and verify there is no push notification when another post is created
OT-9	6.10.1 Account Deletion (Aspirant)	Verify account deletion tag is ticked on the database	The deleted account should have its 'accountDeleted' field set to TRUE in the accounts table.	Press the button to delete the account and monitor the database to see if it has updated.
UT-41	7. iOS Shell	Verify the application runs within an iOS shell app	The user should be able to access the application from an iOS phone	Download the application in the iOS format to an iOS phone and perform the UAT to ensure app functionality
UT-42	7.1 iOS Download	Verify the download of the app to an iOS phone is successful	An iOS user should be able to seamlessly download the application	Have an iOS user download the application on as many iOS versions as possible (due to the potential limits of the number of available phones)
OT-10	7.2 iOS Notifications	Verify app functions are sending push notifications	On an iOS phone, when notifications are turned on and the application is running, the user should receive a notification of the new post	Attempt to create a post on an account and watch to see that a push notification is sent to a user on an iOS device.

UT-44	8. Android Shell	Verify the application runs within an Android shell app	The user should be able to access the application from an Android phone	Download the application in the Android format to an Android phone and perform the UAT to ensure app functionality
UT-45	8.1 Android Download	Verify the download of the app to an Android phone is successful	An Android user should be able to seamlessly download the application	Have an Android user download the application on as many Android versions as possible (due to the potential limits of the number of available phones)
OT-11	8.2 Android Notifications	Verify app functions are sending push notifications	On an Android phone, when notifications are turned on and the application is running, the user should receive a notification of the new post	Attempt to create a post on an account and watch to see that a push notification is sent on an Android device.

## 4. User Acceptance Testing [UAT]

- User loads app and inputs their Ursinus email to create an account
  - User appropriately decides whether they are a poster/viewer account (students are viewers, student clubs are posters)
- User closes and reopens the app and attempts to log into the app with the same Ursinus email

- User scrolls through the main page of posts
- User checks the following page of posts (should be empty as they have not selected accounts to follow)
- User presses the following list button (should not see any accounts yet)
- User searches for accounts they wish to follow and presses the follow button (after exiting the search function the screen should populate with the accounts they chose)
- User returns to the main screen and switches to the following page posts (should see posts from the accounts they follow)
- User should press their account details screen
  - \*Viewing User\* Should see a screen telling them there is no created posts created
  - \*Posting User\* Should see the posts they have created, if they have not created posts, they should see a screen telling them there is no created posts
- User presses the create post button
  - \*Viewing User\* Should see a page telling them they do not have permission to create posts
  - \*Posting User\* Should see multiple text boxes and insert files, can create a post
    - User should try to press the finish button without a title and/or text and not be allowed
    - User should finish their desired post and press the finish button and return to the main screen to view their own post
- User should press the settings button and see the settings screen
  - User should turn on and off notifications
- \*User created Admin Account\*
  - User should select a post from the main post screen to delete and press the delete button to delete the post (Reloading the page should remove the post)

- User should traverse to the following account screen and search for an account and press the delete button to delete the account (Reloading the page should remove the account)