Collaborative University Android App

WAYNE STATE
UNIVERSITY

Test Case Specifications Document | Version 1.2
April 2, 2020

Team: Adeel Asghar, Tyler Gross, Palak Patel, and Hala Ali
Clients: Thomas Anter and Jahnu Best
GTA: Son Dang

VERSION HISTORY

Version	Author	Date	Description
1.0	Hala Ali	3/20/20	Initial Design Definition draft
1.1	Hala Ali, Adeel Asghar, Palak Patel, Tyler Gross	3/31/20	Rough Draft of Testing Plan Document
1.2	Hala Ali, Palak Patel	4/1/20	Formatting Testing Plan, and Adding System Test Cases

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to blueprint the test plan and provide the required information to execute the test cases as defined below. The document contains functional test cases, nonfunctional test cases, integration testing, and system testing. This will ensure that the document will match the specifications as stated in the Software Requirements Specification document.

1.2 References

The following test cases in this document originate from the use cases in the Collaborative University Android App Design Specifications which were created from the requirements gathered in the Software Specifications Document. Each use case has at least one or more test cases assigned to it. All use cases that were referenced for the test cases used the standard protocol of having actors, description, triggers, preconditions, postconditions, normal flow, alternative flows, frequency of use, and assumptions.

2. TEST CASES

2.1 Approach

We will approach this with a fully working application that is populated with data and other users. All dependencies and libraries will be installed.

2.2 Pass/Fail Criteria

For a test case to pass, it must fulfill all the requirements of the expected results in each test case. A fail would be if any of the expected results do not occur; this will fail the test case.

2.3 Entry/ Exit Criteria

The criteria for entry are the completion of Prototype 3, and the exit criteria is that all test cases have passed.

2.4 Suspension/Resumption Criteria

Suspension of testing will occur if any of the test cases pass and will only resume after the issue or error has been resolved and that specific test case will result in a pass.

2.5 Risks/Issues Criteria

Some test cases may result in the app crashing if it they haven’t been properly addressed or some data might be added to the database without all the parameters and requirements fulfilled for that specific model. The specific phone running the application must be somewhat up to date and be running Android 5.0 or above.

2.6 Functional Test Cases

2.6.1 User Registration

Test Case ID	TC-1
Test Case Name	Register for an Account Screen
Priority	High
Preconditions	1. Download and run app. 2. Login screen appears on app being opened.
Postconditions	1. User is added to Firebase Authentication. 2. User is added to Firebase User table.
Test Steps	1. Open application. 2. Click the “Register for an account” text on Login Screen.
Expected Results	1. User is redirected to Login screen.

Test Case ID	TC-2
Test Case Name	Register for an account - Empty Field
Priority	High
Preconditions	1. User is on the registration page.
Postconditions	N/A
Test Steps	1. Leave either the “Password”, “Email”, or “Name” fields empty. 2. Click on “Register” button.
Expected Results	1. User receives the following toast message: “Please enter your username, email, and password.”

Test Case ID	TC-3
Test Case Name	Register for an account - Password Too Short
Priority	Low
Preconditions	1. User is on the registration page.
Postconditions	N/A
Test Steps	1. Open Registration screen. 2. Enter in Name field: “Unit”. 3. Enter in Email field: “unit.school.team.help@gmail.com”. 4. Enter in Password field: “cat”. 5. Click on “Register” button.

Expected Results	1. User receives the following toast message: “Password must be at least 6 characters long.”

Test Case ID	**TC-4**
Test Case Name	Register for an Account - Email is Already Associated with an Account
Priority	Low
Preconditions	1. User is on the registration page.
Postconditions	N/A
Test Steps	1. Enter in Name field: “Son”. 2. Enter in Email Field: “son@gmail.com”. 3. Enter in Password field: “password”. 4. Click “Register” button.
Expected Results	1. User receives the following toast message: “This user is already registered to an account”.

Test Case ID	**TC-5**
Test Case Name	Register for an Account - Click "Already have an account?"
Priority	High
Preconditions	1. User is on the registration page.
Postconditions	N/A
Test Steps	1. Click “Already have an account?” text on Register Screen.
Expected Results	1. User is redirected to Login screen.

Test Case ID	**TC-6**
Test Case Name	Account Registered Successfully
Priority	High
Preconditions	1. User is on the registration page.
Postconditions	1. User account is stored in Firebase database. 2. User account is stored in Firebase Authentication. 3. Email verification is sent to user’s email address.
Test Steps	1. Enter in Name field: “John Doe”. 2. Enter in Email field: “unit.school.team.help@gmail.com”. 3. Enter in Password field: “password”.
Expected Results	1. User is redirected to Login Screen.

2.6.2 User Login

Test Case ID	TC-7
Test Case Name	Login - Successful
Priority	High
Preconditions	1. User has registered for an account. 2. Email Address was verified by user.
Postconditions	N/A
Test Steps	1. Enter in Email Field: “unit.school.team.help@gmail.com” 2. Enter in Password field: “password” 3. Click on “LogIn” button.
Expected Results	1. User is redirected to Home Screen in Main Forum of app. 2. User receives the following toast message: “Successfully logged in!”

Test Case ID	TC-8
Test Case Name	Login - Empty Email and/or Password Fields
Priority	High
Preconditions	1. User is on the login page.
Postconditions	N/A
Test Steps	1. Leave “Password” or “Email” fields empty. 2. Click on “Login” button.
Expected Results	1. User receives the following toast message: “Please enter both your email and password.”

Test Case ID	TC-9
Test Case Name	Login - Incorrect Password
Priority	Low
Preconditions	1. User is on the login page.
Postconditions	N/A
Test Steps	1. Enter in Email Field: “unit.school.team.help@gmail.com” 2. Enter in Password field: “unitschoolteamhelp” 3. Click on “Login” button
Expected Results	1. User receives the following toast message: “The password is invalid, or the user does not have a password”

Test Case ID	**TC-10**
Test Case Name	Login - Incorrect Email
Priority	Low
Preconditions	1. User is on the login page.
Postconditions	N/A
Test Steps	1. Enter in Email Field: “unithelp@gmail.com” 2. Enter in Password field: “password” 3. Click on “LogIn” button.
Expected Results	1. User receives the following toast message: “There is no user record corresponding to this identifier. The user may have been deleted.”

Test Case ID	**TC-11**
Test Case Name	Forgot Password Text Click
Priority	High
Preconditions	1. User is on the login page.
Postconditions	N/A
Test Steps	1. Click on “Forgot Password?” text on Login Screen.
Expected Results	1. User is redirected to Reset Password Screen.

Test Case ID	**TC-12**
Test Case Name	Login Button Redirects to Home Screen
Priority	High
Preconditions	1. User is on the login page. 2. Email Address was verified by user.
Postconditions	N/A
Test Steps	1. Enter in Email Field: “unit.school.team.help@gmail.com” 2. Enter in Password field: “password” 3. Click on “LogIn” button.
Expected Results	1. User is redirected to Home Screen in Main Forum of app. 2. User receives the following toast message: “Successfully logged in!”

Test Case ID	**TC-13**
Test Case Name	Email Address was Not Verified

Priority	High
Preconditions	1. User is on the login page. 2. User received an email from noreply@university-social-media.firebaseapp.com with directions on how to verify their email. 3. User was redirected to Login screen. 4. User did not follow email's directions and did not verify their email address.
Postconditions	N/A
Test Steps	1. Enter in Email Field: "unit.school.team.help@gmail.com". 2. Enter in Password field: "password". 3. Click on "Log In" button.
Expected Results	1. User receives the following toast message: "Please verify your email".

2.6.3 Reset Password

Test Case ID	TC-14
Test Case Name	Send Reset Password Link to User's Email
Priority	High
Preconditions	1. User is on the Reset Password page.
Postconditions	N/A
Test Steps	1. Enter in Email field: "unit.school.team.help@gmail.com." 2. Click button "Reset Password."
Expected Results	1. User will receive an email from Firebase with directions on how to reset their password.

Test Case ID	TC-15
Test Case Name	Resetting Password
Priority	High
Preconditions	1. User has entered the email address "unit.school.team.help@gmail.com" and clicked "Reset password" button on Reset Password page.
Postconditions	1. User's password is changed in Firebase Authentication.
Test Steps	1. Go to gmail.com. 2. Enter for email: unit.school.team.help@gmail.com. 3. Enter for password: adeelhalapalaktyler. 4. Open the email link sent to by **noreply@university-social-media.firebaseapp.com**. 5. Click on the underlined blue link to reset your password. 6. Enter new password "unitschoolpassword" in field generated by link and click "Done".
Expected Results	1. The user can now login with their new password.

Test Case ID	TC-16
Test Case Name	Incorrect Email Parameter
Priority	High
Preconditions	1. User is on the Reset Password page.
Postconditions	N/A
Test Steps	1. Enter in Email field: "unithelp@gmail.com". 2. Click button "Reset Password".

Expected Results	1. User receives the following toast message, “There is no user record corresponding to this identifier. The user may have been deleted.”

Test Case ID	**TC-17**
Test Case Name	Logging in with New Password
Priority	High
Preconditions	1. User is on the Login page. 2. User has successfully reset their password.
Postconditions	N/A
Test Steps	1. Open the app and Login screen will appear. 2. Enter in Email field: “unit.school.team.help@gmail.com”. 3. Enter in password field: “unitschoolpassword”. 4. Click “Login” button.
Expected Results	1. User is redirected to Home Screen in Main Forum of app. 2. User receives the following toast message: “Successfully logged in!”

2.6.4 Subscribe to Community Forum

Test Case ID	TC-18
Test Case Name	Subscribed Button UI
Priority	Low
Preconditions	1. Successful Login
Postconditions	1. Subscription is added to the user in the database.
Test Steps	1. Navigate to the subscriptions screen. 2. Select a subscription button.
Expected Results	1. The subscription button changes color.

Test Case ID	TC-19
Test Case Name	User Sees Subscribed Communities' Posts
Priority	Low
Preconditions	1. Successful Login
Postconditions	N/A
Test Steps	1. Log in to the application.
Expected Results	1. The user's home screen will load and show two posts from every class that they are subscribed to.

2.6.5 Unsubscribe from Community Forum

Test Case ID	TC-20
Test Case Name	User Unsubscribes from a Community
Priority	Medium
Preconditions	1. Users is subscribed to a certain community.
Postconditions	2. The subscription is removed from the user in database.
Test Steps	1. Navigate to search functionality by clicking search icon on the bottom navigation. 2. Search for the class they wish to unsubscribe to. 3. Select the "unsubscribe button next to the class found from the search results.
Expected Results	1. The unsubscribe button text changes to subscribe. 2. When the user navigates to the subscriptions page the class searched for is missing from the list. 3. When the user navigates to the home page all posts from the class they unsubscribed from no longer appear.

2.6.6 Post to Subscribed Forum

Test Case ID	TC-21
Test Case Name	Text Post to Subscribed forum
Priority	High
Preconditions	User should have an account with at least 1 class subscription.
Postconditions	1. Post is added to Community Table in Database. 2. Post is added to User Table in Database.
Test Steps	1. Navigate to the new post functionality. 2. Use the class spinner to select a class to post to, this should be a class that the you are subscribed to. 3. Enter the title and text of the post they would like to create. 4. Select submit.
Expected Results	1. The post the user created shows up in the forum.

Test Case ID	TC-22
Test Case Name	Image Post to Subscribed forum
Priority	High
Preconditions	User should have an account with at least 1 class subscription.
Postconditions	1. Post is added to Community Table in Database. 2. Post is added to User Table in Database.
Test Steps	1. Navigate to the new post functionality. 2. Use the class spinner to decide what class they would like to post to. This should be a class that they are subscribed to already. 3. Select the image post button on the new post page. 4. Pick an image from downloaded images in the phone's storage. 5. Enter title and text to go along with the image. 6. Select submit.
Expected Results	1. The image post that the user created shows up in the forum.

Test Case ID	TC-23
Test Case Name	Post to Forum Not Subscribed to
Priority	High
Preconditions	User needs to be a member of the app.
Postconditions	N/A
Test Steps	1. Navigate to new text post.

	2. Use the class spinner to pick a class that you are not subscribed to. 3. Enter the title and text for the post and clicks new post.
Expected Results	1. The user receives a toast message notifying them they cannot post to a forum they aren’t subscribed to.

Test Case ID	**TC-24**
Test Case Name	Post to a Forum Without a Title
Priority	Low
Preconditions	1. User should have an account with at least 1 class subscription.
Postconditions	N/A
Test Steps	1. User navigates to new post functionality. 2. User uses class spinner to pick a class they are subscribed to. 3. User enters the text for the post but leaves the title blank.
Expected Results	1. User receives a toast message notifying them the title is blank. 2. The post will not show up in the class page or in the user profile.

Test Case ID	**TC-25**
Test Case Name	Post to a Forum Without a Text
Priority	Low
Preconditions	1. User should have an account with at least 1 class subscription.
Postconditions	N/A
Test Steps	1. User navigates to new post functionality. 2. User uses class spinner to pick a class they are subscribed to. 3. User enters the text for the post but leaves the text blank.
Expected Results	1. User receives a toast message notifying them the title is blank. 2. The post will not show up in the class page or in the user profile.

2.6.7 Edit Posts

Test Case ID	TC-26
Test Case Name	Edit Existing Post
Priority	High
Preconditions	1. User is subscribed to a class forum. 2. User has created a post in that class forum.
Postconditions	1. Post is edited in Firebase Database. 2. Post is edited in LiveData mutable list.
Test Steps	1. Select Profile from Hamburger menu. 2. Navigate to Post fragment. 3. Choose a Post to edit and swipe left on it. 4. Click the edit option that appears. 5. Change the title and/or text of the post. 6. Click done.
Expected Results	1. The post title and/or text is changed in the list of posts in User Profile. 2. The post title and/or text is changed throughout the application.

Test Case ID	TC-27
Test Case Name	Edit Existing Post – Empty Text Field
Priority	High
Preconditions	1. User is subscribed to a class forum. 2. User has created a post in that class forum.
Postconditions	N/A
Test Steps	1. Select Profile from Hamburger menu. 2. Navigate to Posts fragment. 3. Choose a comment to edit and swipe left on it. 4. Click the edit option that appears. 5. Change title and/or text to empty. 6. Click done.
Expected Results	1. Toast message appears warning that the text box is empty 2. App will stay on edit comment page.

Test Case ID	TC-28
Test Case Name	Edit Existing Post – No Internet Connection

Priority	Medium
Preconditions	1. User is subscribed to a class forum. 2. User has created a post in that class forum. 3. Internet must be disconnected.
Postconditions	N/A
Test Steps	1. Select Profile from Hamburger menu. 2. Navigate to Comments fragment. 3. Choose a comment to edit and swipe left on it. 4. Click the edit option that appears.
Expected Results	1. Toast message appears warning that there is no internet connection and will prevent user from editing the comment.

2.6.8 Delete Post

Test Case ID	TC-29
Test Case Name	Delete Post
Priority	Medium
Preconditions	1. User is successfully logged in (TC-7). 2. User is subscribed to a class forum. 3. User has created a post in that class forum.
Postconditions	1. Post removed from Firebase Database.
Test Steps	1. Navigate to profile from navigation drawer. 2. Click Post fragment. 3. Navigate to post to be deleted. 4. Swipe left on post to display delete options. 5. Click delete. 6. Confirm you want to delete post by clicking yes.
Expected Results	1. Post deleted from post. 2. Post deleted from list of user comments.

Test Case ID	TC-30
Test Case Name	Delete Post – No Internet Connection
Priority	Medium
Preconditions	1. User is successfully logged in (TC-7). 2. User is subscribed to a class forum. 3. User has created a post in that class forum.
Postconditions	N/A
Test Steps	1. Navigate to profile from navigation drawer. 2. Click Post fragment. 3. Navigate to post to be deleted. 4. Swipe left on post to display delete options. 5. Click delete. 6. Confirm you want to delete post by clicking yes.
Expected Results	1. Toast message appearing that states that the post cannot be deleted because there is no internet connection. 2. The post will remain and not be deleted.

2.6.9 Comment to Posts in Subscribed Forum

Test Case ID	TC-31
Test Case Name	Comment on Post
Priority	High
Preconditions	1. User is successfully logged in (TC-7). 2. User must be subscribed to whatever community the post is in.
Postconditions	1. Comment added to Firebase Database. 2. Comment added to LiveData.
Test Steps	1. User navigates to post they wish to comment on. 2. Click the comment textbox and the bottom of the page. 3. Type a comment. 4. Click “Comment.”
Expected Results	1. Comment shows up under post. 2. Comment is added to current user’s profile page.

Test Case ID	TC-32
Test Case Name	Comment on Post – Empty Text Field
Priority	High
Preconditions	1. User is successfully logged in (TC-7). 2. User must be subscribed to whatever community the post is in.
Postconditions	N/A
Test Steps	1. User navigates to post they wish to comment on. 2. Click the comment textbox and the bottom of the page. 3. Leave text field empty. 4. Click “Comment.”
Expected Results	1. Toast message that warns user that textbox is empty and must not be blank.

Test Case ID	TC-33
Test Case Name	Comment on Post – Not Subscribed to Specified Community
Priority	High
Preconditions	1. User is successfully logged in (TC-7). 2. User must not be subscribed to whatever community the post is in.
Postconditions	N/A
Test Steps	1. User navigates to post they wish to comment on.

	2. Click the comment textbox and the bottom of the page. 3. Type a comment. 4. Click “Comment.”
Expected Results	1. Toast message that warns user that they must be subscribed to that community to post comments.

2.6.10 Edit Comments

Test Case ID	TC-34
Test Case Name	Edit Existing Comment
Priority	High
Preconditions	1. User is subscribed to a class forum. 2. User is successfully logged in (TC-7).User has created a comment to a post in that class forum.
Postconditions	1. Comment is edited in Firebase Database. 2. Comment is edited in LiveData.
Test Steps	1. Select Profile from Hamburger menu. 2. Navigate to Comments fragment. 3. Choose a comment to edit and swipe left on it. 4. Click the edit option that appears. 5. Change the text of the comment. 1. Click done.
Expected Results	1. The comment text is changed in the list of comments in User Profile. 2. The comment is changed on whatever post it is for.

Test Case ID	TC-35
Test Case Name	Edit Existing Comment – Empty Text
Priority	High
Preconditions	1. User is successfully logged in (TC-7). 1. User is subscribed to a class forum. 2. User has created a comment to a post in that class forum.
Postconditions	N/A
Test Steps	1. Select Profile from Hamburger menu. 2. Navigate to Comments fragment. 3. Choose a comment to edit and swipe left on it. 4. Click the edit option that appears. 5. Change the text of the comment to empty. 6. Click done.
Expected Results	1. Toast message appears warning that the text box is empty 2. App will stay on edit comment page.

Test Case ID	TC-36

Test Case Name	Edit Existing Comment – No Internet Connection
Priority	Medium
Preconditions	1. User is successfully logged in (TC-7). 1. User is subscribed to a class forum. 2. User has created a comment to a post in that class forum. 3. Internet must be disconnected.
Postconditions	N/A
Test Steps	1. Select Profile from Hamburger menu. 2. Navigate to Comments fragment. 3. Choose a comment to edit and swipe left on it. 4. Click the edit option that appears.
Expected Results	1. Toast message appears warning that there is no internet connection and will prevent user from editing the comment.

2.6.11 Delete Comment

Test Case ID	**TC-37**
Test Case Name	Delete Comment
Priority	Medium
Preconditions	1. User is successfully logged in (TC-7). 1. User is subscribed to a class forum. 2. User has created a comment to a post in that class forum.
Postconditions	1. Comment removed from Firebase Database.
Test Steps	1. Navigate to profile from navigation drawer. 2. Click comment fragment. 3. Navigate to comment to be deleted. 4. Swipe left on comment to display delete options. 5. Click delete. 6. Confirm you want to delete comment by clicking yes.
Expected Results	1. Comment deleted from post. 2. Comment deleted from list of user comments.

Test Case ID	**TC-38**
Test Case Name	Delete Comment – No Internet Connection
Priority	Medium
Preconditions	1. No internet connection. 1. User is subscribed to a class forum. 2. User has created a comment to a post in that class forum. 3. Internet must be disconnected.
Postconditions	N/A
Test Steps	1. Navigate to profile from navigation drawer. 2. Click comment fragment. 3. Navigate to comment to be deleted. 4. Swipe left on comment to display delete options. 5. Click delete. 6. Confirm you want to delete comment by clicking yes.
Expected Results	1. Toast message appearing that states that the comment cannot be deleted because there is no internet connection. 2. The comment will remain and not be deleted.

2.6.12 Search for Forum

Test Case ID	**TC-39**
Test Case Name	Search for a Full Class Name
Priority	Medium
Preconditions	1. User has an account and is looking for a certain section.
Postconditions	N/A
Test Steps	1. Navigate to the search functionality by using the bottom navigation. 2. Select the search bar and enters a Forum name.
Expected Results	1. The searched class name should be the only one that is still on the page.

Test Case ID	TC-40
Test Case Name	Search for a Subject Prefix
Priority	Medium
Preconditions	1. User has an account and is looking for a certain class subject.
Postconditions	N/A
Test Steps	1. Navigate to the search functionality. 2. Select the search bar and enters the subject pretext into the search bar.
Expected Results	1. All the classes that appear on the page should have that prefix.

Test Case ID	TC-41
Test Case Name	Search for all classes with the same number
Priority	Medium
Preconditions	1. User has an account and is looking for a certain class level.
Postconditions	N/A
Test Steps	1. Navigate to the search functionality. 2. Select the search bar and enters a class number.
Expected Results	1. The list on the page should only show classes with the same class number.

2.6.13 Hamburger Menu

Test Case ID	TC-42
Test Case Name	Opening Hamburger Menu
Priority	High
Preconditions	1. User has logged in and is on a page where the Menu Icon is available.
Postconditions	N/A
Test Steps	1. Select the menu button.
Expected Results	1. The menu slides open.

Test Case ID	TC-43
Test Case Name	Open Profile Screen from The Hamburger Menu
Priority	High
Preconditions	1. User has logged in and is on a page where the Menu Icon is available.
Postconditions	N/A
Test Steps	1. Select the menu button.
Expected Results	1. The user can view their current username, bio, and profile image. 2. The user can view all post and comments that they have made as well as.

Test Case ID	TC-44
Test Case Name	Close Hamburger Menu
Priority	High
Preconditions	1. User has logged in and has opened the Hamburger Menu.
Postconditions	N/A
Test Steps	1. Click anywhere outside of the Hamburger Menu.
Expected Results	1. The Hamburger menu closes.

Test Case ID	TC-45
Test Case Name	Open Settings Screen from The Hamburger Menu
Priority	High
Preconditions	1. User has logged in and has opened the Hamburger Menu.

Postconditions	N/A
Test Steps	1. Select the Settings option.
Expected Results	1. The settings page opens.

Test Case ID	**TC-46**
Test Case Name	Logout from Hamburger Menu
Priority	High
Preconditions	1. User has logged in and has opened the Hamburger Menu.
Postconditions	N/A
Test Steps	1. Select log out option.
Expected Results	1. The user is logged out.

2.6.14 User Profile

Test Case ID	TC-47
Test Case Name	Edit Profile Picture
Priority	High
Preconditions	1. User has logged in and navigated to the edit profile screen.
Postconditions	1. The profile image URL is updated in the database.
Test Steps	1. Select profile image button. 2. Select a photo from your device storage. 3. Select done.
Expected Results	1. The profile image is replaced with the one that the user chose in every place that it can be viewed.

Test Case ID	TC-48
Test Case Name	Edit Bio
Priority	High
Preconditions	1. The user has logged in and navigated to the edit profile screen.
Postconditions	1. The new user bio is saved to the database.
Test Steps	1. Select the “please add a new bio” hint. 2. Type in a new bio. 3. Select done.
Expected Results	1. The user bio text in the profile screen is replaced with the user’s new bio.

Test Case ID	TC-49
Test Case Name	Edit Username
Priority	High
Preconditions	1. The user has logged in and navigated to the edit profile screen.
Postconditions	1. The new username is saved to the database and the username is changed in the database for ever post and comment that it appears in.
Test Steps	1. Select the username text line. 2. Enter a new username. 3. Select done.
Expected Results	1. The current user’s username is changed in the profile page as well as all posts and comments.

Test Case ID	**TC-50**
Test Case Name	Opening A User’s Profile from a Post
Priority	High
Preconditions	1. The user has navigated to clicked post screen or any class list screen.
Postconditions	N/A
Test Steps	1. Select a user’s username.
Expected Results	1. The user is navigated to the profile screen of the user that they clicked on.

Test Case ID	**TC-51**
Test Case Name	Opening A User’s Profile from a Comment
Priority	High
Preconditions	1. The user has navigated to a clicked post screen that has comments.
Postconditions	N/A
Test Steps	1. Select a user’s username.
Expected Results	1. The user is navigated to the profile screen of the user that they clicked on.

2.6.15 Settings

Test Case ID	**TC-52**
Test Case Name	Night Mode
Priority	High
Preconditions	1. The user has logged in and navigated to settings page.
Postconditions	N/A
Test Steps	1. Select night mode.
Expected Results	1. The color scheme of the application becomes better suited for a dark room.

2.6.16 Messaging

Test Case ID	TC-53
Test Case Name	Select Latest Messages Fragment
Priority	High
Preconditions	1. User is successfully logged in (TC-7). 2. User has started chatting with another user.
Postconditions	N/A
Test Steps	1. Click the Messages Fragment.
Expected Results	1. Messages fragment loads. 2. List of latest messages is displayed showing the user profile image, username, and most recent message.

Test Case ID	TC-54
Test Case Name	Select Latest Messages Fragment – No Messages Sent Yet
Priority	High
Preconditions	1. User is successfully logged in (TC-7).
Postconditions	N/A
Test Steps	1. Click the Messages Fragment
Expected Results	1. Display message fragment with no list of recent messages. 2. Display message prompting user to select New Message button to start a conversation.

Test Case ID	TC-55
Test Case Name	Select Latest Messages Fragment – No Connection
Priority	High
Preconditions	1. User has started chatting with another user. 2. Internet must be disconnected.
Postconditions	N/A
Test Steps	1. Click the Messages Fragment.
Expected Results	1. Display cached list of messages which has been cached but is not updated. 2. Display message listing that there is no connection.

Test Case ID	TC-56
Test Case Name	Select New Message Button
Priority	High
Preconditions	1. Navigate to Latest Messages Fragment.
Postconditions	N/A
Test Steps	1. Click New Message button on title bar.
Expected Results	1. New screen loads which shows list of all users. 2. Their username and profile images will be available.

Test Case ID	TC-57
Test Case Name	Select New Message Button – No Network Connection
Priority	High
Preconditions	1. Navigate to Latest Messages Fragment. 2. Internet must be disconnected.
Postconditions	N/A
Test Steps	1. Click New Message button on title bar.
Expected Results	1. List of users will be list of users that have been cached but is not updated. 2. Display message listing there is no connection.

Test Case ID	TC-58
Test Case Name	Open Chatlog from New Message Button
Priority	High
Preconditions	1. Navigate to Latest Messages Fragment.
Postconditions	1. User Chatlog with selected is loaded into LiveData.
Test Steps	1. Click any user from list of Users in New Message Activity.
Expected Results	1. Show chatlog of current user and selected user. 2. Display username of selected user in title bar.

Test Case ID	TC-59
Test Case Name	Open Chatlog from Recent Messages Fragment
Priority	High
Preconditions	1. Navigate to Latest Messages Fragment.

Postconditions	1. User Chatlog with selected user is loaded into LiveData.
Test Steps	1. Click any user from list of Users in New Message Activity.
Expected Results	1. Show chatlog of current user and selected user. 2. Display username of selected user in title bar.

Test Case ID	**TC-60**
Test Case Name	Send Message to User - Successful
Priority	High
Preconditions	1. Follow steps of TC-56 – Open Chatlog from Recent Messages Fragment.
Postconditions	1. User message is added to Chatlog LiveData 2. User message is added to Recent Messages LiveData
Test Steps	1. Type a message into input box. 2. Click send button.
Expected Results	1. Message will appear in chatlog of sending user. 2. Message will appear in chatlog of receiving user.

Test Case ID	**TC-61**
Test Case Name	Send Message to User – No Network Connection
Priority	High
Preconditions	1. Follow steps of TC-56 – Open Chatlog from Recent Messages Fragment. 2. Internet must be disconnected.
Postconditions	1. User messages are cached and added to Firebase when connection is restored. 2. Stored in LiveData to display in chatlog.
Test Steps	1. Type a message into input box. 2. Click send button.
Expected Results	1. Message will appear in chatlog of sending user. 2. Message will appear in chatlog of receiving user.

Test Case ID	**TC-62**
Test Case Name	Click Send Button with no Empty Message
Priority	High

Preconditions	1. Follow steps of TC-56 – Open Chatlog from Recent Messages Fragment.
Postconditions	N/A
Test Steps	1. Leave message input box empty. 2. Click send button.
Expected Results	1. Toast message will display stating that input box is empty.

Test Case ID	**TC-63**
Test Case Name	Receive Message from User
Priority	High
Preconditions	N/A
Postconditions	1. Received message is added to Chatlog LiveData. 2. Received message is added to Recent Messages LiveData.
Test Steps	N/A
Expected Results	1. Message will appear in chatlog of sending user. 2. Message will appear in chatlog of receiving user.

Test Case ID	**TC-64**
Test Case Name	Chatlog Saves Messages
Priority	High
Preconditions	1. Follow steps of TC-57 - Send Message to User – Successful.
Postconditions	1. Message saved to Firebase Database. 2. Message saved to LiveData.
Test Steps	1. Send message to user. 2. Exit application. 3. Open application. 4. Navigate back to chat with user.
Expected Results	1. Chat should contain all messages sent to and from user to another user.

2.6.17 Report User to Admin

Test Case ID	TC-65
Test Case Name	Report Another User’s Post
Priority	High
Preconditions	1. User logged into app with email: “son@gmail.com” and password: “password” 2. Follow test steps of TC-17 Subscribe to Community Forum BIO 1030 3. User clicks on BIO 1030 and is redirected to Community Posts screen where all posts pertaining to BIO 1030 are available.
Postconditions	4. User blocks all of John Smith’s posts and comments across all community forums. User is notified that Admin received their report.
Test Steps	1. User swipes on John Smith’s post with title: “Dd” and text: “Dd”. 2. User clicks on Report User. Pop-up dialog appears 3. Of the three options ("This is spam", "This is abusive or harassing", "Other issues"), the user selects “This is spam” and clicks “Submit”
Expected Results	1. Email is automatically triggered to admin account unit.school.team.help@gmail.com under subject “Report Issue”, the message states in the email, “Hello Admin! The accuser, son@gmail.com, is reporting the user with the uid of "CqkILDrFkiUoPYFYwr3pXN08nEJ2" of inappropriate content. Here is the reported message: Dd” 2. Admin can copy-paste the uid inside Firebase Authentication search and disable user’s account based on whether admin decides if that post was inappropriate. 3. User receives toast message with the following message: “We’ve received your report.”

Test Case ID	TC-66
Test Case Name	Report Another User’s Comments
Priority	High
Preconditions	1. User logged into app with email: “son@gmail.com” and password: “password” 2. Follow test steps of TC-17 Subscribe to Community Forum BIO 1030 3. User clicks on BIO 1030 and is redirected to Community Posts screen where all posts pertaining to BIO 1030 are available.

	4. User clicks on John Smith’s post with title: “Hey what’s up” and text: “This is a picture of me”
Postconditions	1. User blocks all of Hala Ali’s posts and comments across all community forums User is notified that Admin received their report.
Test Steps	1. User swipes on Hala Ali’s comment with text: “You look really ugly.” 2. User clicks on Report Comment. Pop-up dialog appears 3. Of the three options ("This is spam", "This is abusive or harassing", "Other issues"), the user selects “This is abusive or harassing” and clicks “Submit.”
Expected Results	1. Email is automatically triggered to admin account unit.school.team.help@gmail.com under subject “Report Issue”, the message states in the email, “Hello Admin! The accuser, son@gmail.com, is reporting the user with the uid of "UsOnbrDFBCanYpjZyu0GKgJIWEt1" of inappropriate content. Here is the reported message: You look really ugly.” 2. Admin can copy-paste the uid inside Firebase Authentication search and disable user’s account based on whether admin decides if that post was inappropriate. 3. User receives toast message with the following message: “We’ve received your report.”

Test Case ID	**TC-67**
Test Case Name	User cannot Report their Own Posts or Comments
Priority	High
Preconditions	1. User logged into app with email: “son@gmail.com” and password: “password” 2. Follow test steps of TC-17 Subscribe to Community Forum BIO 1030 3. User clicks on BIO 1030 and is redirected to Community Posts screen where all posts pertaining to BIO 1030 are available.
Postconditions	Swipe is disabled, so no options appear to report
Test Steps	1. User swipes on their own post title: “Syllabus” text: “GTA hours are 4:30-5:30 on Thursdays.” 2. User locates their comment that says, “Cool photo!” in John Smith’s post with title: “Hey what’s up” and text: “This is a picture of me.” 3. User swipes on their comment.
Expected Results	Nothing happens when user swipes in both instances, so reporting feature can’t be accessed on their own post or comment.

2.6.18 Block User

Test Case ID	TC-68
Test Case Name	User cannot Block their Own Posts or Comments
Priority	High
Preconditions	1. User logged into app with email: "son@gmail.com" and password: "password" 2. Follow test steps of TC-17 Subscribe to Community Forum BIO 1030 3. User clicks on BIO 1030 and is redirected to Community Posts screen where all posts pertaining to BIO 1030 are available.
Postconditions	Swipe is disabled, so no options appear to report
Test Steps	1. User swipes on their own post title: "Syllabus" text: "GTA hours are 4:30-5:30 on Thursdays." 2. User locates their comment that says, "Cool photo!" in John Smith's post with title: "Hey what's up" and text: "This is a picture of me." 3. User swipes on their comment.
Expected Results	Nothing happens when user swipes in both instances, so reporting feature can't be accessed on their own post or comment.

Test Case ID	TC-69
Test Case Name	User can Block Another User through their Posts
Priority	High
Preconditions	1. User logged into app with email: "son@gmail.com" and password: "password" 2. Follow test steps of TC-17 Subscribe to Community Forum BIO 1030 3. User clicks on BIO 1030 and is redirected to Community Posts screen where all posts pertaining to BIO 1030 are available.
Postconditions	1. John Smith's posts and comments are no longer visible to user. 2. User is notified that John Smith has been blocked.
Test Steps	1. User swipes on John Smith's post title: "Ff" text: "Fd" 1. User selects "Block User," Pop up appears with message that says, "Are you sure? You won't be able to see posts or comments from this user." 2. User selects "Block" button.
Expected Results	1. All of John Smith's posts and comments across all community forums are blocked, meaning that the user will no longer be able to

	see content from this user (content is replaced with “[blocked]” placeholder). 2. User receives toast message with the following message: “This user has been blocked.”

Test Case ID	**TC-70**
Test Case Name	User can Block Another User through their Comments
Priority	High
Preconditions	1. User logged into app with email: “son@gmail.com” and password: “password” 2. Follow test steps of TC-17 Subscribe to Community Forum BIO 1030 3. User clicks on BIO 1030 and is redirected to Community Posts screen where all posts pertaining to BIO 1030 are available.
Postconditions	1. John Smith’s posts and comments are no longer visible to user. 2. User is notified that John Smith has been blocked.
Test Steps	1. User clicks on Tyler’s post title: “Help with Learning Bones” text: “Does anyone have any advice for memorizing the bones names?” 2. User swipes on John Smith’s comment, “What’s a bone? Haha” 3. User selects “Block User,” Pop up appears with message that says, “Are you sure? You won’t be able to see posts or comments from this user.” 4. User selects “Block” button.
Expected Results	1. All of John Smith’s posts and comments across all community forums are blocked, meaning that the user will no longer be able to see content from this user (content is replaced with “[blocked]” placeholder). 2. User receives toast message with the following message: “This user has been blocked.”

2.6 Non-Functional Test Cases

2.2.1 Performance Testing

Test Case ID	TC-71
Test Case Name	Toast Messages to User
Priority	High
Preconditions	Follow steps of TC-1
Postconditions	Message is displayed to user.
Test Steps	Follow steps of all TC cases with expected results of user receiving toast message. (i.e TC-2, TC-3, TC-4, TC-7, TC-10 etc.)
Expected Results	Toast message should appear to user within 200-500 milliseconds.

Test Case ID	TC-72
Test Case Name	Redirection of Screens Triggered by User Activity
Priority	Medium
Preconditions	User is on the registration page.
Postconditions	Screen is redirected.
Test Steps	Follow steps of all TC cases with the expected results of user redirecting to another screen. (I.E. TC)
Expected Results	User is redirected to another screen within 500 milliseconds.

Test Case ID	TC-73
Test Case Name	Create Posts
Priority	High
Preconditions	1. User is logged in. 2. User is subscribed to CSC 1500 forum.
Postconditions	User's post is added to CSC 1500 forum.
Test Steps	1. User clicks on the pencil icon and changes spinner to CSC 1500. 2. User creates a text post with title: "Hello" and text: "When's our first quiz?" 3. User clicks "Post" button
Expected Results	User's post is uploaded to class forum in under 1 second.

Test Case ID	TC-74
Test Case Name	Create Comments
Priority	High
Preconditions	1. User is logged in. 2. User is subscribed to CSC 1500 forum.
Postconditions	User’s comment is added to Adeel Asghar’s post in CSC 1500 forum.
Test Steps	1. User clicks Adeel’s image post with title: “Here’s some cool code” and text: “Check out this login feature.” 2. User enters “Thanks for sharing!” in text field next to “Comment” button. 3. User clicks “Comment” button
Expected Results	User’s comment is uploaded to class forum in under 1 second.

Test Case ID	TC-75
Test Case Name	Subscribing to Class Forum
Priority	Medium
Preconditions	1. User is logged in. 2. User is on list fragment screen with all the classes.
Postconditions	User is subscribed to BIO 1030
Test Steps	User clicks the “Subscribe” button next to BIO 1030
Expected Results	User is subscribed to BIO 1030 in about 200-500 milliseconds.

Test Case ID	TC-76
Test Case Name	Unsubscribing to Class Forum
Priority	Medium
Preconditions	1. Follow the preconditions of TC-74: Subscribe to Class Forum 2. User is subscribed to BIO 1030
Postconditions	User is unsubscribed from BIO 1030
Test Steps	User clicks the “Unsubscribe” button next to BIO 1030
Expected Results	User is unsubscribed from BIO 1030 in about 200-500 milliseconds.

2.2.2 Security

Test Case ID	**TC-77**
Test Case Name	Password Encryption on Firebase Real-time Database
Priority	Low
Preconditions	User has successfully registered an account on the app.
Postconditions	Password is hashed and stored using auto-generated token in Firebase Real-time Database and Firebase Authentication.
Test Steps	1. Go to https://console.firebase.google.com/project/university-social-media/database/university-social-media/data. 2. Open up some of the child nodes under “users” node and observe that the uid is an auto-generated token designed to protect the password of all the users who have accounts in this app.
Expected Results	Password protection security measures in the form of password hashing is present.

3. INTEGRATION TESTING

3.1 Result of Attempts with Espresso Automated Testing

Integration testing for this app was attempted through creating an automated UI test with Espresso. Unfortunately, several runs using this automated testing technique proved to be very slow and ineffective. Also, the tests behaved erratically, meaning that the same tests would somehow fail after proving successfully passing in the previous runs. Stretch goals in include more emphasis on unit testing cases rather than integration testing as they are more structurally sound and effective.

4. SYSTEM TESTING

4.1 Firebase Realtime Database Test Cases

4.1.1 User Registration

Test Case ID	TC-78
Test Case Name	Register for an account
Priority	High
Preconditions	1. User has downloaded and opened app. 2. User is on the Registration Page.
Postconditions	1. User account information is stored in Firebase Real-time Database and in Firebase Authentication
Test Steps	1. Enter in Email field: “unit.school.team.help@gmail.com” 2. Enter in password field: “password” 3. Enter in name field: “John Doe” 4. Click the “Register” button

Expected Results	1. The account associated with email, “unit.school.team.help@gmail.com” is now located in Firebase Authentication and the password (i.e. the uid) is protected with hashing and shows an auto-generated value. Inside the database schema:
	Users{
	uid{
	Username: “John Doe”
	Email: “unit.school.team.help@gmail.com”
	Password: “[auto-generated uid value]”
	profileImageUrl: null
	UserBio: “noBio”
	}
	}

4.1.2 User Login

Test Case ID	**TC-79**
Test Case Name	Find Firebase Authentication token to Login User
Priority	High
Preconditions	1. Download and run app. 2. Login screen appears on app being opened. 3. User account information is stored in Firebase Real-time Database and in Firebase Authentication
Postconditions	1. User is logged in
Test Steps	1. Enter in Email field: “unit.school.team.help@gmail.com” 2. Enter in password field: “password” 3. Click the “Login” button
Expected Results	1. Firebase Authentication’s built-in signInWithEmailAndPassword(email, password) function searches through Firebase Authentication to find email and matches the real password hashed into the uid token in the system to the password inputted by user. If a match is found with email and password, Firebase authenticates this user and grants them access to the app. 2. User is logged in to the app and is directed to the home fragment screen

4.1.3 Reset Password

Test Case ID	TC-80
Test Case Name	Incorrect Email
Priority	High
Preconditions	1. Download and run app. 2. Login screen appears on app being opened. 3. User account information is stored in Firebase Real-time Database and in Firebase Authentication 4. User clicks “Forgot Password?” text and is redirected to Reset Password screen.
Postconditions	1. Error is logged 2. User cannot reset password
Test Steps	1. Enter in Email field: “unithelp @gmail.com” 2. Click the “Reset Password” button
Expected Results	1. Firebase Authentication’s built-in sendPasswordResetEmail(email) functionality checks inside Firebase Authentication to see if email user inputted is stored. For this case, Firebase Authentication will generate an error message that says “There is no user record corresponding to this identifier. The user may have been deleted.”

Test Case ID	TC-81
Test Case Name	Email found in Firebase Authentication
Priority	High
Preconditions	1. Download and run app. 2. Login screen appears on app being opened. 3. User account information is stored in Firebase Real-time Database and in Firebase Authentication 4. User clicks “Forgot Password?” text and is redirected to Reset Password screen.
Postconditions	1. Error is logged 2. User cannot reset password
Test Steps	1. Enter in Email field: unit.school.team.help@gmail.com. 2. Click the “Reset Password” button.
Expected Results	1. Firebase Authentication’s built-in sendPasswordResetEmail(email) functionality checks inside Firebase Authentication to see if email user inputted is stored. For

	this case, Firebase Authentication will find a match and automatically trigger an email to user’s email address.

4.1.4 Subscribe to Community Forum

Test Case ID	**TC-82**
Test Case Name	Subscribing to a Community forum
Priority	High
Preconditions	1. User is logged in. 2. User has navigated to the all classes screen. 3. The user is not already subscribed to BIO 1030.
Postconditions	1. User is subscribed to BIO 1030.
Test Steps	1. User clicks the “Subscribe” button next to BIO 1030.
Expected Results	1. The path UID/subscriptions under the current user is updated with BIO 1030 in the Firebase Database.

4.1.5 Unsubscribe from Community Forum

Test Case ID	**TC-83**
Test Case Name	Unsubscribe from a Community Forum
Priority	High
Preconditions	1. User is subscribed to BIO 1030
Postconditions	1. The user is unsubscribed from BIO 1030
Test Steps	1. User clicks the “Unsubscribe” button next to BIO 1030
Expected Results	1. BIO 1030 is removed from the list of subscribed classes in the path UID/subscriptions of the firebase database.

4.1.6 Post to Subscribed Forum

Test Case ID	**TC-84**
Test Case Name	Text Post to a Subscribed Forum
Priority	High
Preconditions	1. User is logged in. 2. User is subscribed to CSC 1500 forum.
Postconditions	1. The post appears on the CSC 1500 screen 2. The post appears on the current user's user profile
Test Steps	1. User clicks on the pencil icon and changes spinner to CSC 1500. 2. User creates a text post with title: "Hello" and text: "When's our first quiz?" 3. User clicks "Post" button
Expected Results	1. A post key is added to the Firebase database under both Subjects/CSC1500/Posts as well as under UID/posts. That post includes a title of "Hello", the text of "When's out first quiz?", a class key, the time of creation, the image post label false, a subject and an author.

4.1.7 Edit Post

Test Case ID	**TC-85**
Test Case Name	Editing a Post
Priority	High
Preconditions	1. The user has completed TC-83. 2. The user has navigated to their profile page.
Postconditions	1. The edited post text and title appear in the user profile and in the CSC 1500 post list.
Test Steps	1. User Swipes left on the text post with the title “Hello”. 2. User Selects the edit button. 3. User Enters “new post title” in the title section. 4. User Enters “new post text” in text section. 5. User Selects done.
Expected Results	1. In the Firebase database under both Subjects/CSC1500/Posts as well as under UID/posts. The post data elements are changed to include a title of “new post title”, the text of “new post text”, the same class key, an updated time, the image post label false, a subject and an author.

4.1.8 Delete Post

Test Case ID	TC-86
Test Case Name	Email found in Firebase Authentication
Priority	High
Preconditions	1. The user has completed TC-84 2. The user has navigated to their profile page.
Postconditions	1. The post is no long visible from anywhere in the application.
Test Steps	1. User Swipes left on the text post with the title “new post”. 2. User Selects the delete icon. 3. User Agrees to delete post in the pop-up window.
Expected Results	1. The post is deleted from the Subjects/CSC1500/Posts and UID/posts paths of the Firebase Database.

4.1.9 Comment to Posts in Subscribed Forum

Test Case ID	TC-87
Test Case Name	Email found in Firebase Authentication
Priority	High
Preconditions	1. User is logged in. 2. User is subscribed to CSC 1500 forum. 3. User has navigated to the CSC forum screen
Postconditions	1. User's comment is added to Adeel Asghar's post in CSC 1500 forum
Test Steps	1. User clicks Adeel's image post with title: "Here's some cool code" and text: "Check out this login feature." 2. User enters "Thanks for sharing!" in text field next to "Comment" button. 3. User clicks "Comment" button
Expected Results	1. Comment key is added Subjects/CSC1500/Posts/[the post key associated with Adeel's post]/Comments. Path in database as well as UID/Comments. The comment includes the text of "Thanks for sharing", the time the comment was made, a user key, a post key, a profile com key, the posters UID, a class key and the CRN of the class.

4.1.10 Edit Comment

Test Case ID	TC-88
Test Case Name	Edit Comment
Priority	High
Preconditions	1. User is logged in. 2. User is subscribed to CSC 1500 forum. 3. User has navigated to their user profile 4. User has completed TC-86
Postconditions	1. The comment is no long visible from anywhere in the application
Test Steps	1. User Selects the comments tab 2. User Swipes left on the comment titled “Thanks for sharing” 3. User Selects the edit icon 4. The user types “new comment” in the text section
Expected Results	1. In Subjects/CSC1500/Posts/[the post key associated with Adeel’s post]/Comments. Path in database as well as UID/Comments. The comment elements changes to the text of “new comment”, the time the comment was edited, a user key, a post key, a profile com key, the posters UID, a class key and the CRN of the class.

4.1.11 Delete Comment

Test Case ID	TC-89
Test Case Name	Delete Comment
Priority	High
Preconditions	1. User is logged in. 2. User has navigated to their user profile. 3. User has completed TC-86.
Postconditions	1. The comment is no longer visible from anywhere in the application
Test Steps	1. User Selects the comments tab 2. User Swipes left on the comment titled “Thanks for sharing” 3. User Selects the edit icon 4. The user types “new comment” in the text section
Expected Results	1. The comment will be removed from Firebase Realtime Database in 2 locations with queries: the user node path and the subject node path. The queries will find the child stored under an auto-generated key for the comment the user would like to delete. 2. Use the Firebase path, “users/[auto-generated uid]/Comments/ProfileComKey” to find comment from user path to remove the comment. 3. Use the Firebase path, “Subjects/CSC1500/Posts/[auto-generated post key]/Comments/Classkey” to find comment from subject path to remove the comment.

4.1.12 User Profile

Test Case ID	TC-90
Test Case Name	Edit Profile Image
Priority	High
Preconditions	1. User has logged in and navigated to the edit profile screen.
Postconditions	1. The profile image URL is updated in the database.
Test Steps	1. Select profile image button. 2. Select a photo from device storage photos or gallery. 3. Select done.
Expected Results	1. The image will be uploaded onto Firebase Storage. 2. The image will have a file location URI that will be stored as a string inside the key "profileImageUrl" with following Firebase path, "users/[auto-generated uid for user]/profileImageUrl."

Test Case ID	TC-91
Test Case Name	Edit Bio
Priority	Medium
Preconditions	1. The user has logged in and navigated to the edit profile screen.
Postconditions	1. The new user bio is saved to the database.
Test Steps	1. Select the "please add a new bio" hint. 2. Type in a new bio. 3. Select done.
Expected Results	1. The user input will be stored as a string inside the key "UserBio" with following Firebase path, "users/[auto-generated uid for user]/profileImageUrl."

Test Case ID	TC-92
Test Case Name	Edit Username
Priority	Low
Preconditions	1. The user has logged in and navigated to the edit profile screen.
Postconditions	1. The new username is saved to the database and the username is changed in the database for ever post and comment that it appears in.
Test Steps	1. Select the username text line. 2. Enter a new username.

	3. Select done.
Expected Results	1. The user input will be stored as a string inside the key “Username” with following Firebase path, “users/[auto-generated uid for user]/Username.”

4.1.13 Messaging

Test Case ID	TC-93
Test Case Name	Send Message to User - Successful
Priority	High
Preconditions	1. Follow steps of TC-56 – Open Chatlog from Recent Messages Fragment.
Postconditions	1. Message is stored in two locations of database under node user-messages.
Test Steps	1. Type “Hello text” in the input box. 2. Click send button.

Expected Results	1. Inside the user-messages node Firebase Realtime database, when a user (Sender) sends a message to another user (Receiver), the auto-generated uid tokens for both sender and receiver is added as children of user-messages node. The database schema of the test case looks like this:
	user-messages{
	[Sender uid]{
	[Receiver uid]{
	[auto-generated key for message child]{
	FromID: [Sender uid]
	Id: [auto-generated key for message child]
	Text: “Hello test”
	Timestamp: 1585806166
	Told: [Receiver uid]
	}
	}
	}
	[Receiver uid]{
	[Sender uid]{
	[auto-generated key for message child]{
	FromID: [Sender uid]
	Id: [auto-generated key for message child]
	Text: “Hello test”
	Timestamp: 1585806166
	Told: [Receiver uid]
	}
	}
	}

	}

Test Case ID	**TC-94**
Test Case Name	Receive Message from User - Successful
Priority	High
Preconditions	1. Follow steps of TC-56 – Open Chatlog from Recent Messages Fragment.
Postconditions	1. Message is stored in two locations of database under node user-messages.
Test Steps	N/A

Expected Results	1. Inside the user-messages node Firebase Realtime database, when a user (Receiver) receives a message from another user (Sender), the auto-generated uid tokens for both sender and receiver is added as children of user-messages node. The database schema of the test case looks like this:
	
	user-messages{
	[Sender uid]{
	[Receiver uid]{
	[auto-generated key for message child]{
	FromID: [Sender uid]
	Id: [auto-generated key for message child]
	Text: “Hello test”
	Timestamp: 1585806166
	Told: [Receiver uid]
	}
	}
	}
	
	[Receiver uid]{
	[Sender uid]{
	[auto-generated key for message child]{
	FromID: [Sender uid]
	Id: [auto-generated key for message child]
	Text: “Hello test”
	Timestamp: 1585806166
	Told: [Receiver uid]
	}
	}

	} }