Sprint 2- Manual Testing:

The purpose of this document is to detail the manual testing of the UI of the Polybius app. This includes tests for multiple stories located in our current sprints planning document. In addition, any external tests will be mention at the end of this document.

**Test One:** Registration and login take in user information and send them to the main menu.

Procedure:

Steps followed to test the login and register functionality is outlined below:

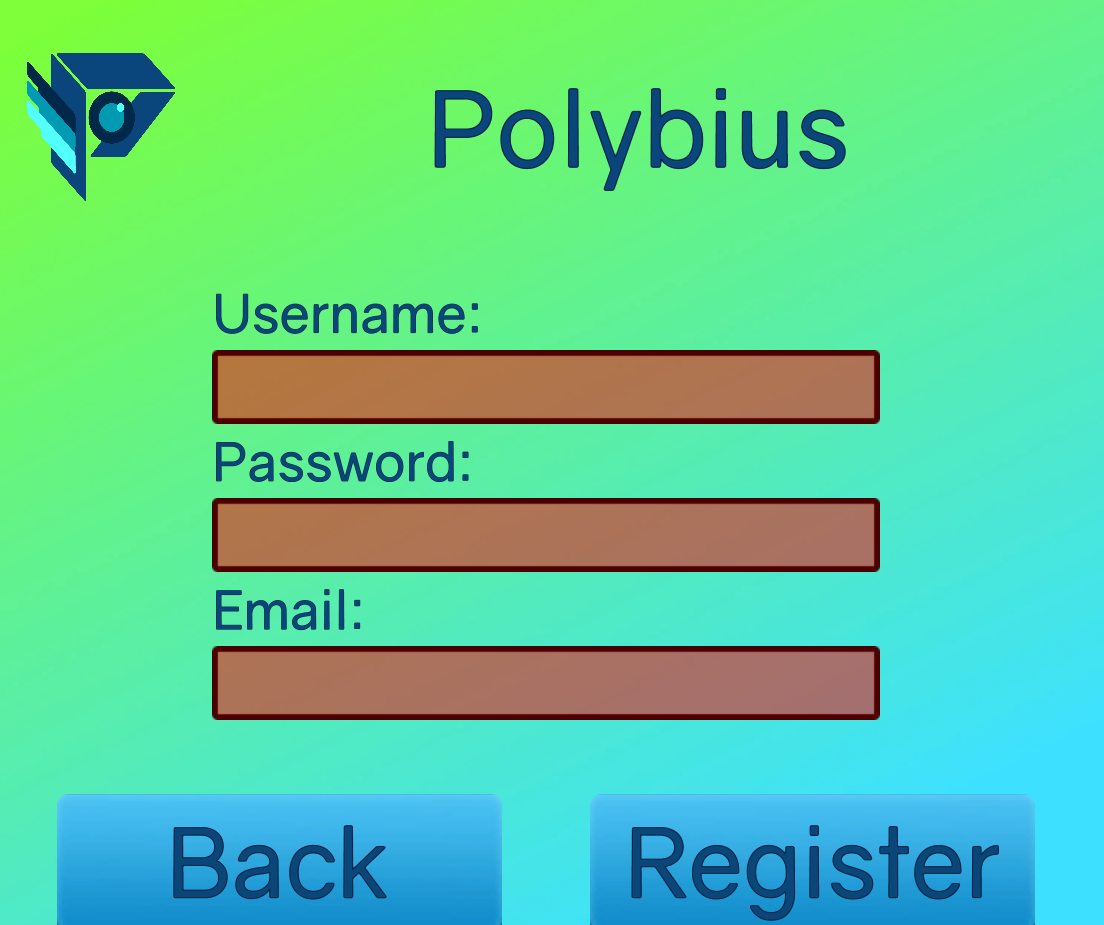
* Open a new instance of Polybius
* Tap or Click on the Register button to open the registration page
* Create new User using a test username, password, and email, (testone, testonepass, [testerone@test.com](mailto:testerone@test.com)) which will then redirect to the main menu
* Click the logout button in the top left to navigate back to the starting screen
* Tap or Click on the Login button to open the login page
* Login to an account using username and password (testone, testonepass) which will then redirect to the main menu.

Results and Follow Up:

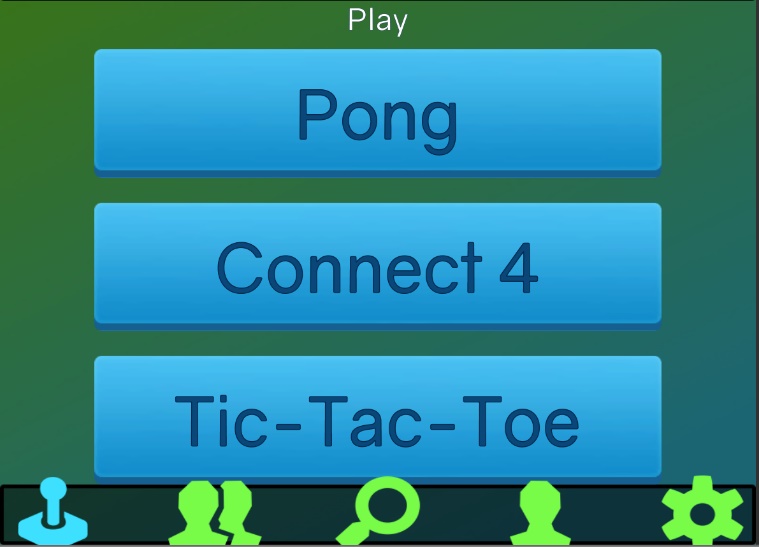
1. Open a new instance of Polybius is successful. Once the app is open it leads to the login and register menu. The image below displays this function:

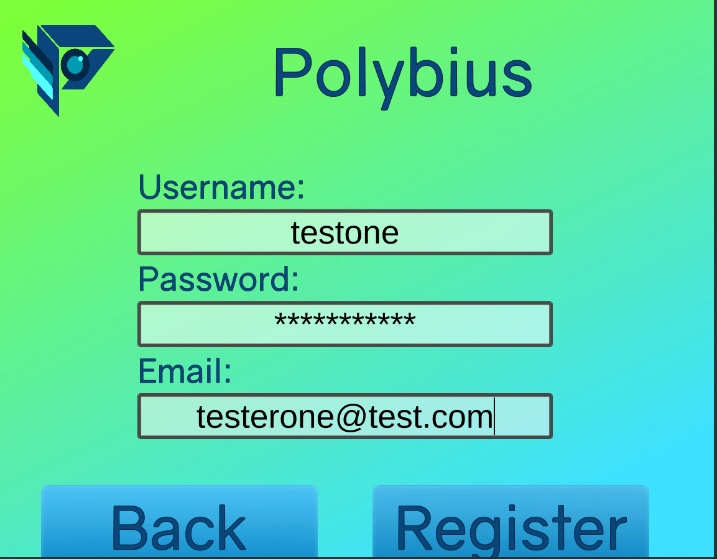


1. Tap or Click on the Register button to open the registration page is successful. Once the register button is pressed the app opens to the registration window. Picture is showen below:

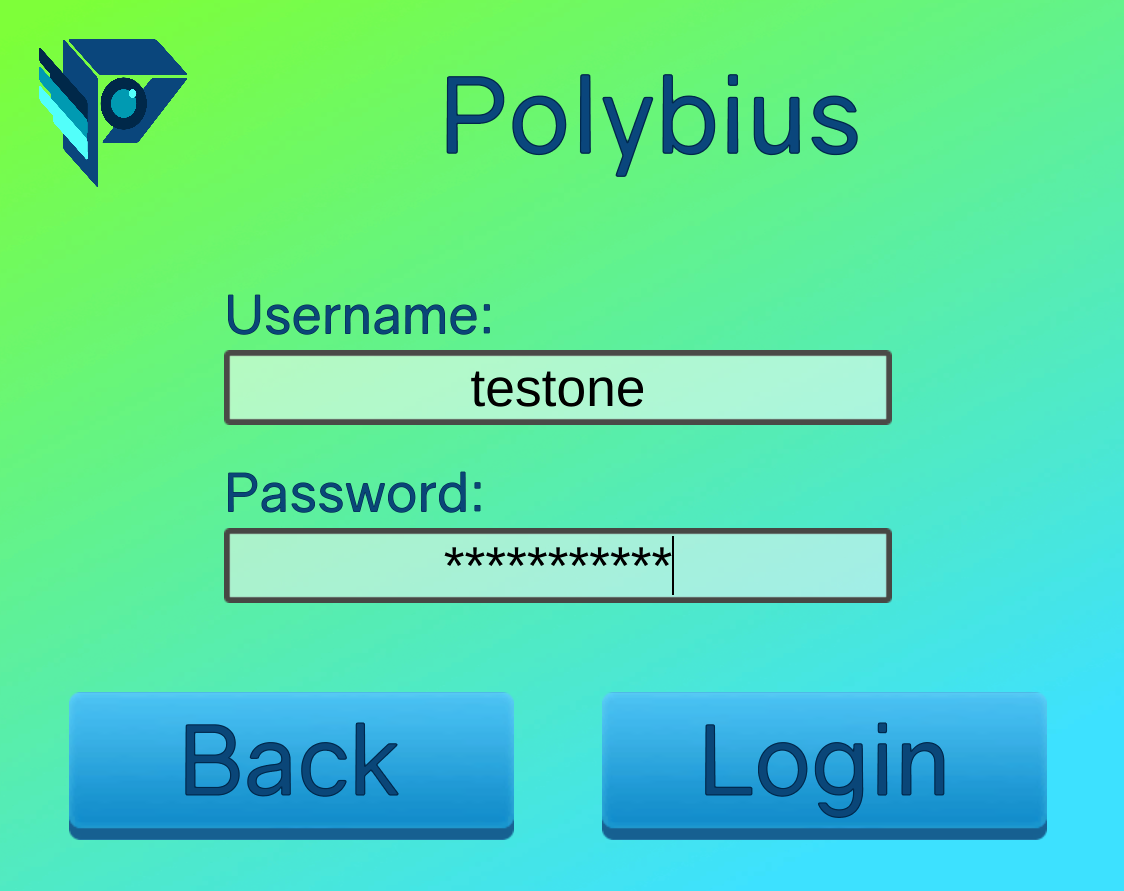


1. Create new User is successful. Once the user enters in their username password they are immediately redirected to the main menu. Pictures indicate the process:





1. Click the logout button in the top left to navigate back to the starting screen is successful. It moves the menu back to the starting screen to login and register. No pictures due to redundancy.
2. Tap or Click on the Login button to open the login page, and login to the test user, successful. There was an issue where login would hang after the logging out, but that has since been fixed by including built-in functions. Picture below shows the login screen, and the main menu has been removed due to redundancy.



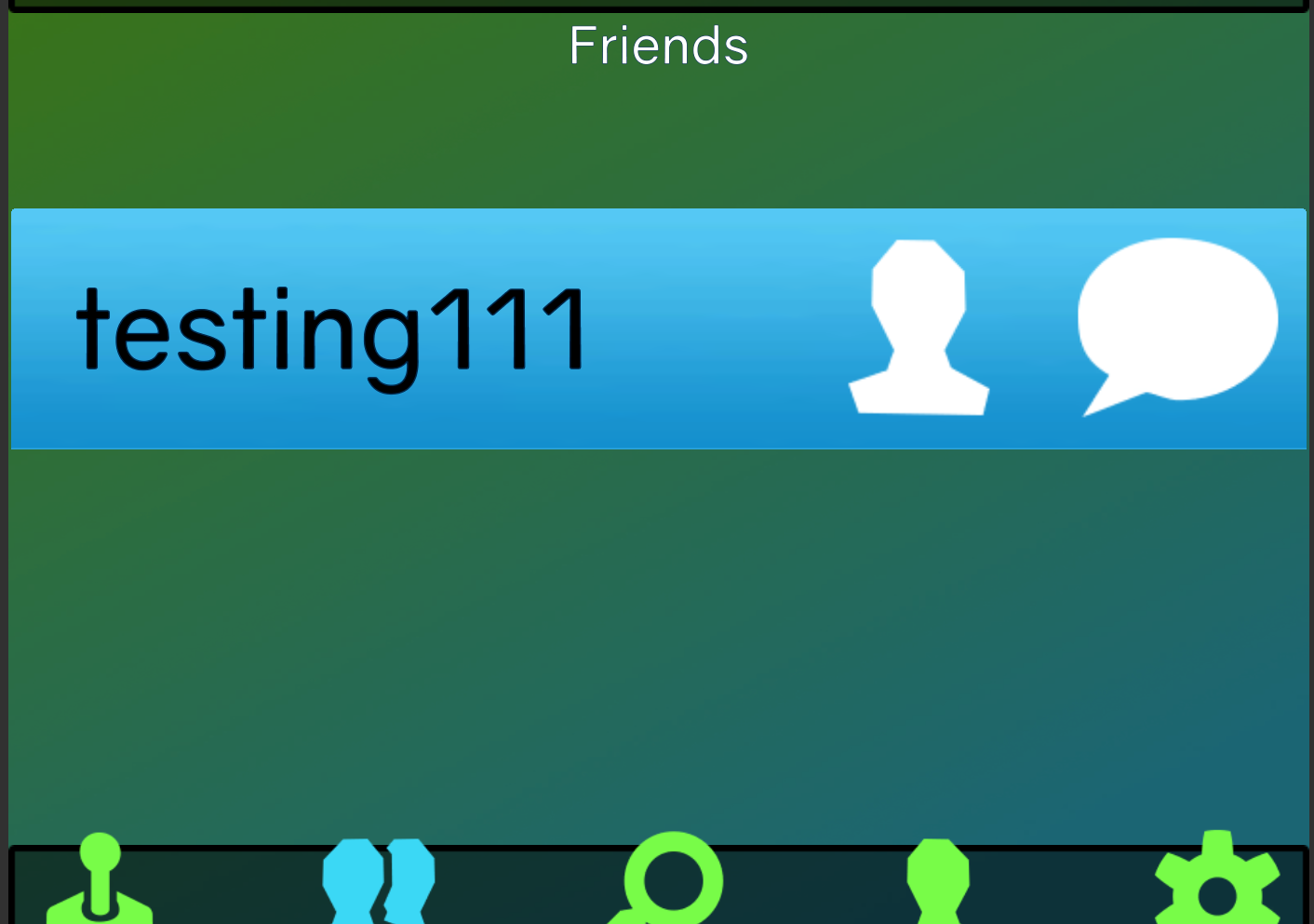
**Test Two:** Validating friends list is functional and able to view current friends.

Procedure:

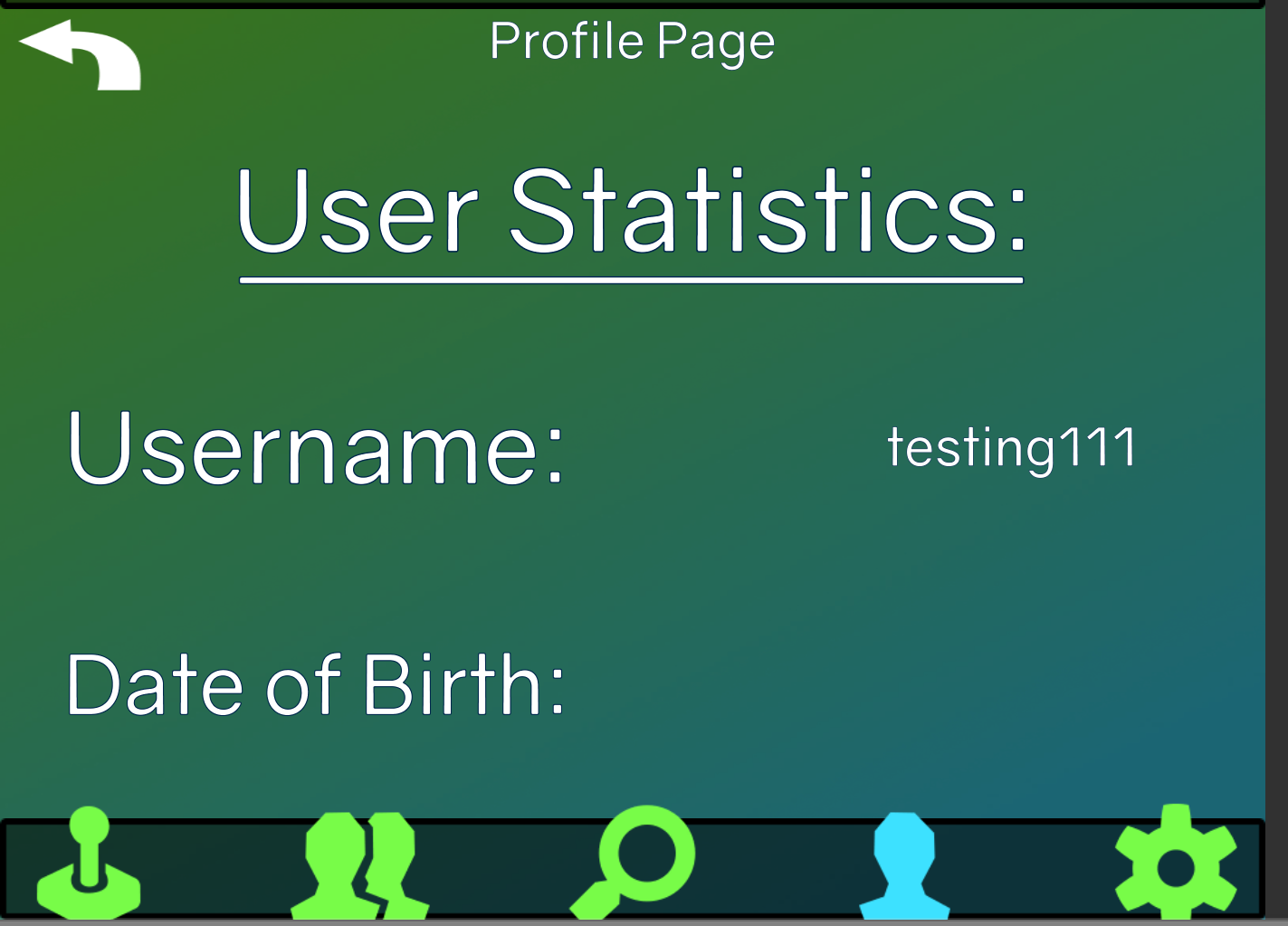
* User has logged into their Polybius account and navigates to the friends tab.
* The user can scroll through their friends list viewing a basic tab of each of their friends
* The User selects a friend and navigates to their profile.
* The Friend profile successfully displays the data on the selected friend.

Results and follow up:

1. User has logged into their Polybius account and navigates to the friends tab: successful. The friends tab successfully displays when the button is displayed.



1. The user can scroll through their friends list viewing a basic tab of each of their friends: successful. The friends tab has a scrolling function that changes depending on how many the friends the user has.(Picture removed for redundancy)
2. The User selects a friend and navigates to their profile: successful. The user can tap on the player icon to be brought to their friend’s profile page. This then displays their information.



1. The Friend profile successfully displays the data on the selected friend: successful. The data shown on the friends page is correct even though the date of birth is not shown this is correct.(Picture Removed for redundancy)

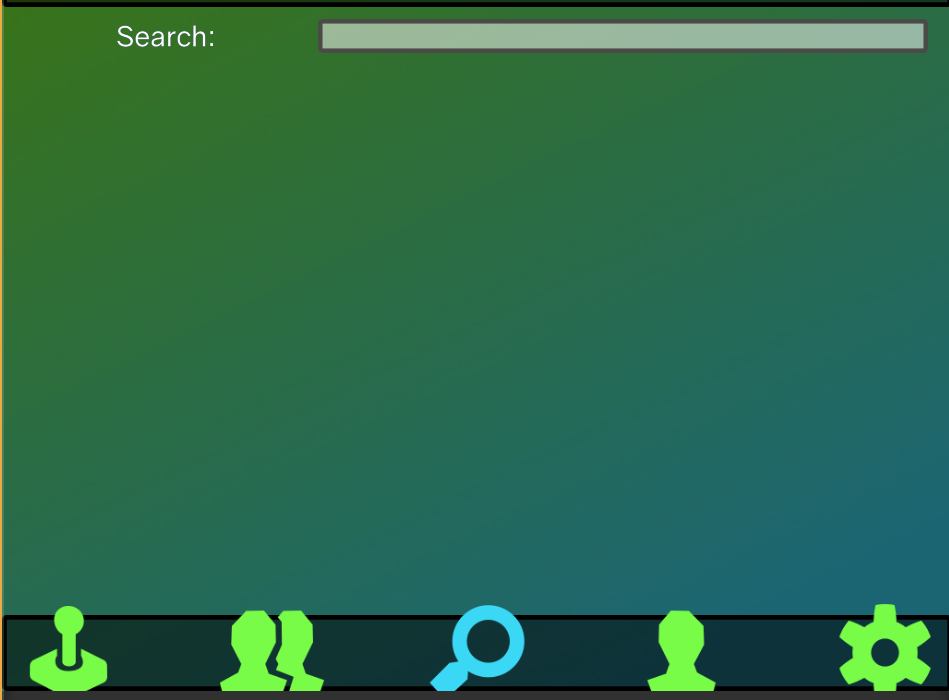
**Testing Three:** Users can successfully add and remove select users.

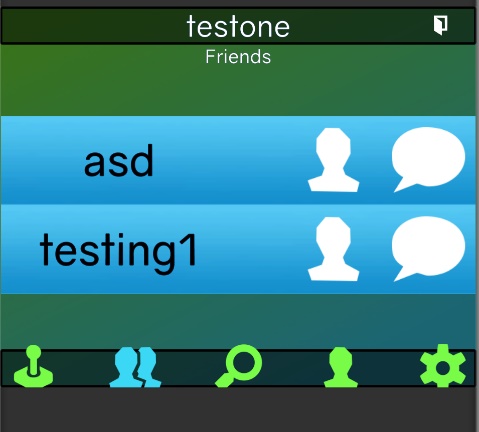
Procedure:

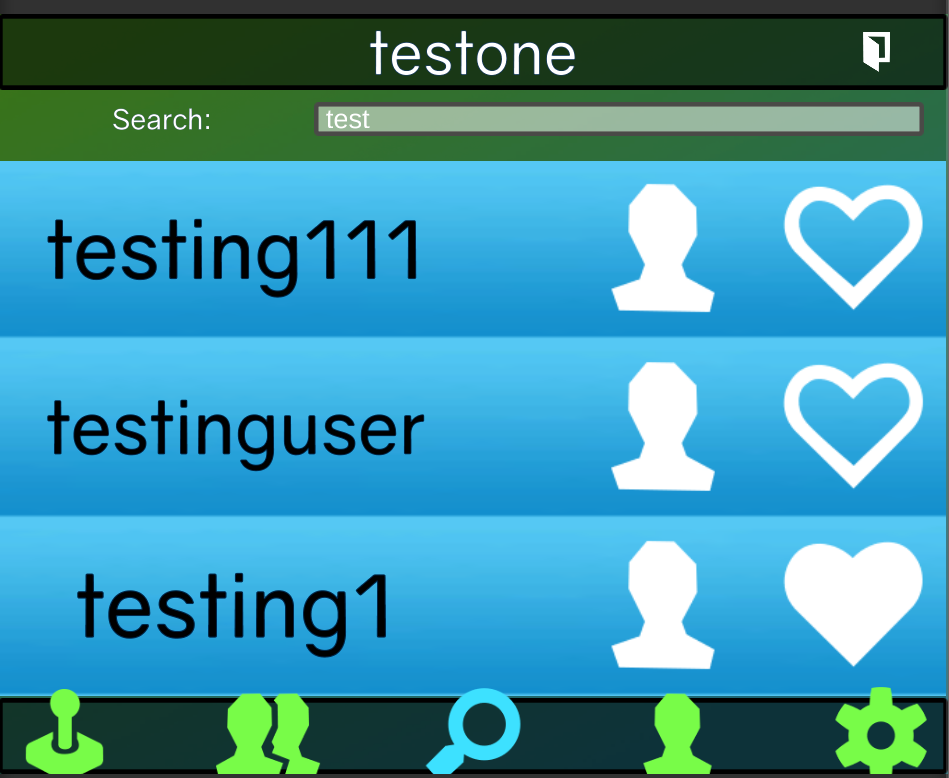
* The user taps on the add friends’ button to navigate to the add friend page.
* The user enters their friends name and sends a request to their friend
* The user taps a friend and taps the removes them from their friends list.

Results and Follow up:

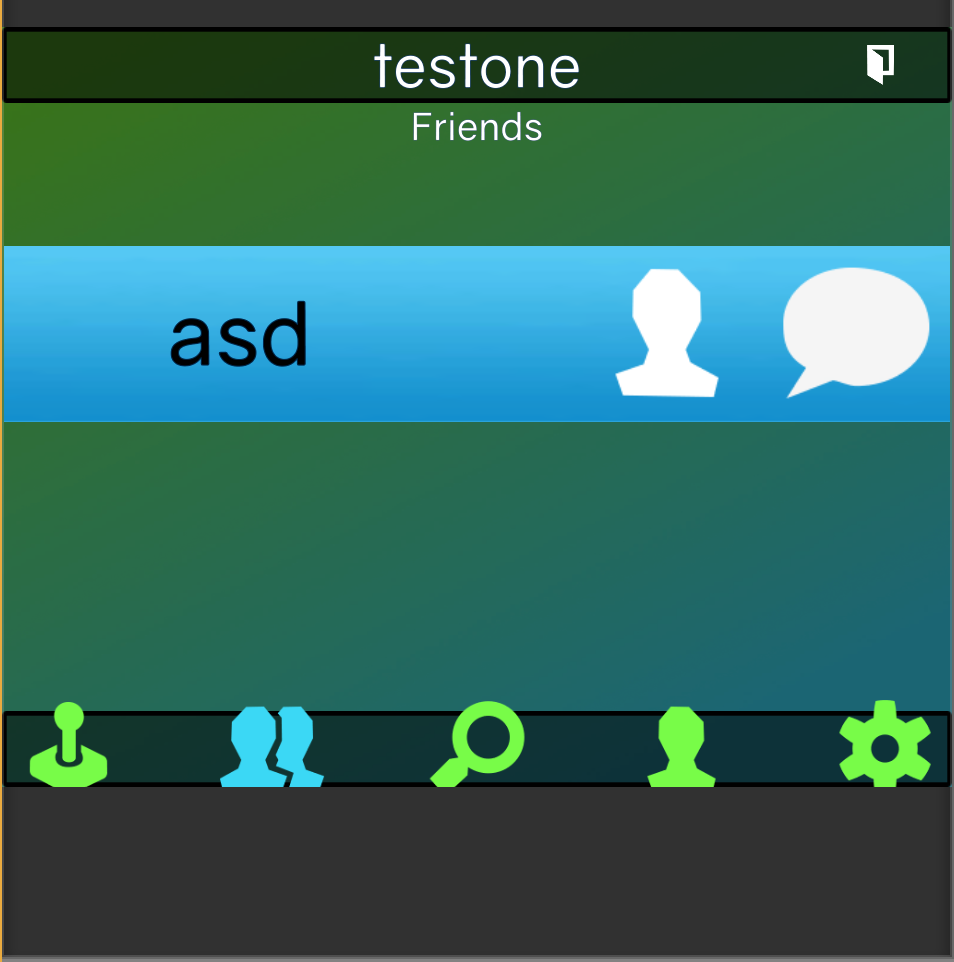
1. The user taps on the add friends’ button to navigate to the add friend page: is successful. The page opens up with an open friends page that allows a search functionality. Picture provided below.



1. The user enters their friends name and sends a request to their friend: is successful. The searched friend displays, and the friends button adds a person to someone friends list. Pictures provided below.



1. The user taps a friend and taps the removes them from their friends list: is successful. Once a person has been removed from the friends list they disappear from the friends tab.



**Testing Four:** Users can send and receive messages.

Procedure:

* Navigate on each instance to the users that they will both be communicating with.
* Type any message then send said message to the other user
* Verify that the message is received

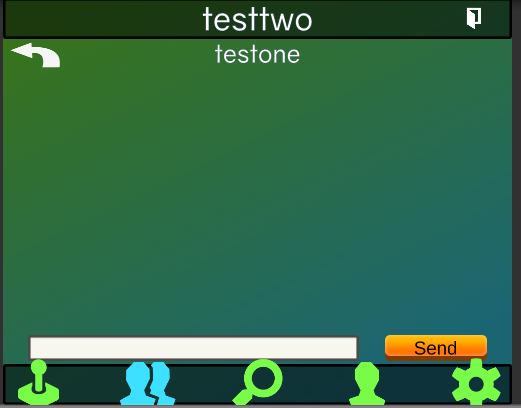
1. Navigate on each instance to the users that they will both be communicating with: is successful. The user can navigate to the messaging menu such that they are able to message the user when they please.



1. Type any message then send said message to the other user: is successful. A user can send a message to their corresponding friend, which gets sent to the server.



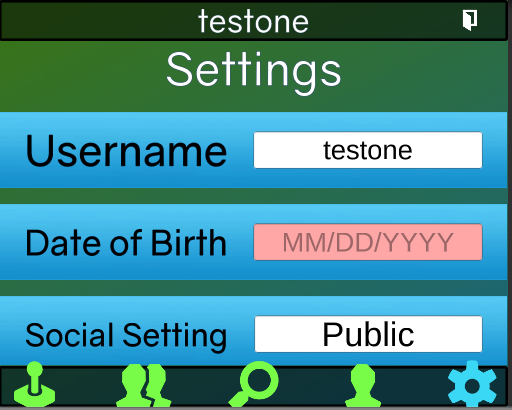
1. Verify that the message is received: failed. This was unable o be accomplished due to multiple issues client side and server side making it possible to send messages, but not receive them.

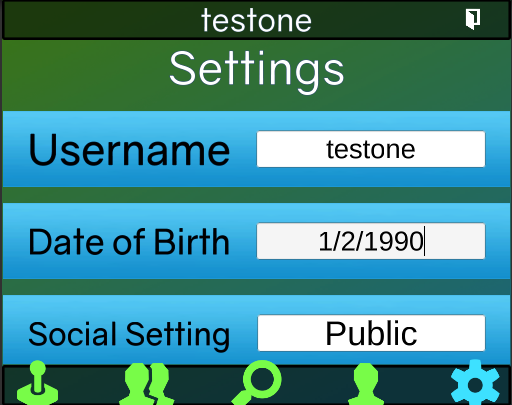


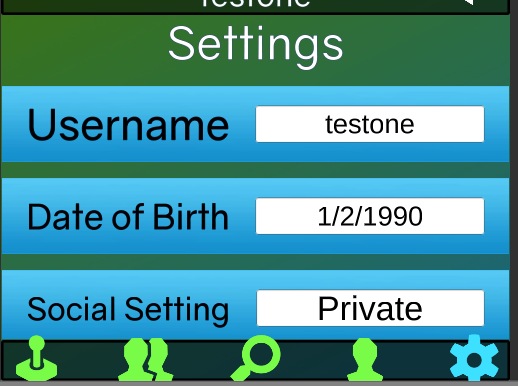
**Testing Five:** Settings in the user profile can be modified and changed as needed.

Procedure:

* The user can see their current profile settings
* The UI changes accordingly with the new value entered
* The user can set their privacy settings
* The users privacy is adjusted to reflect their interests.







**Testing Six:** Users can create and host lobbies.

Procedure:

* Navigate to the list of pong games
* Create a lobby that other players can view
* Once all users leave the lobby is removed

1. Navigate to the list of pong games: is successful. This view is successfully rendered and displays all current opened games. Picture for reference.



1. Create a lobby that other players can view: is successful. The lobby displays on the main menu of the game when it appears. Picture removed for redundancy.
2. Once all users leave the lobby is removed: is successful. The lobbies disappear once every user has disconnected rom them. Picture below for reference.



**Testing Seven:** MapBox markers can successfully be set and viewed

Procedure:

* The user can navigate to the map interface
* The user can move the map around
* The user can view the location of the selected game

Results and Follow-up:

1. The user can navigate to the map interface: is successful. The user views the map icon to the left of the lobbies and is brought to the map. Picture Below for reference.



1. The user can move the map around: is successful. The map is moveable such that the user can get a better view of the location. Picture below for reference.



1. The user can view the location of the selected game: is successful. The user can zoom in and accurately see the location of the game. Picture for reference.



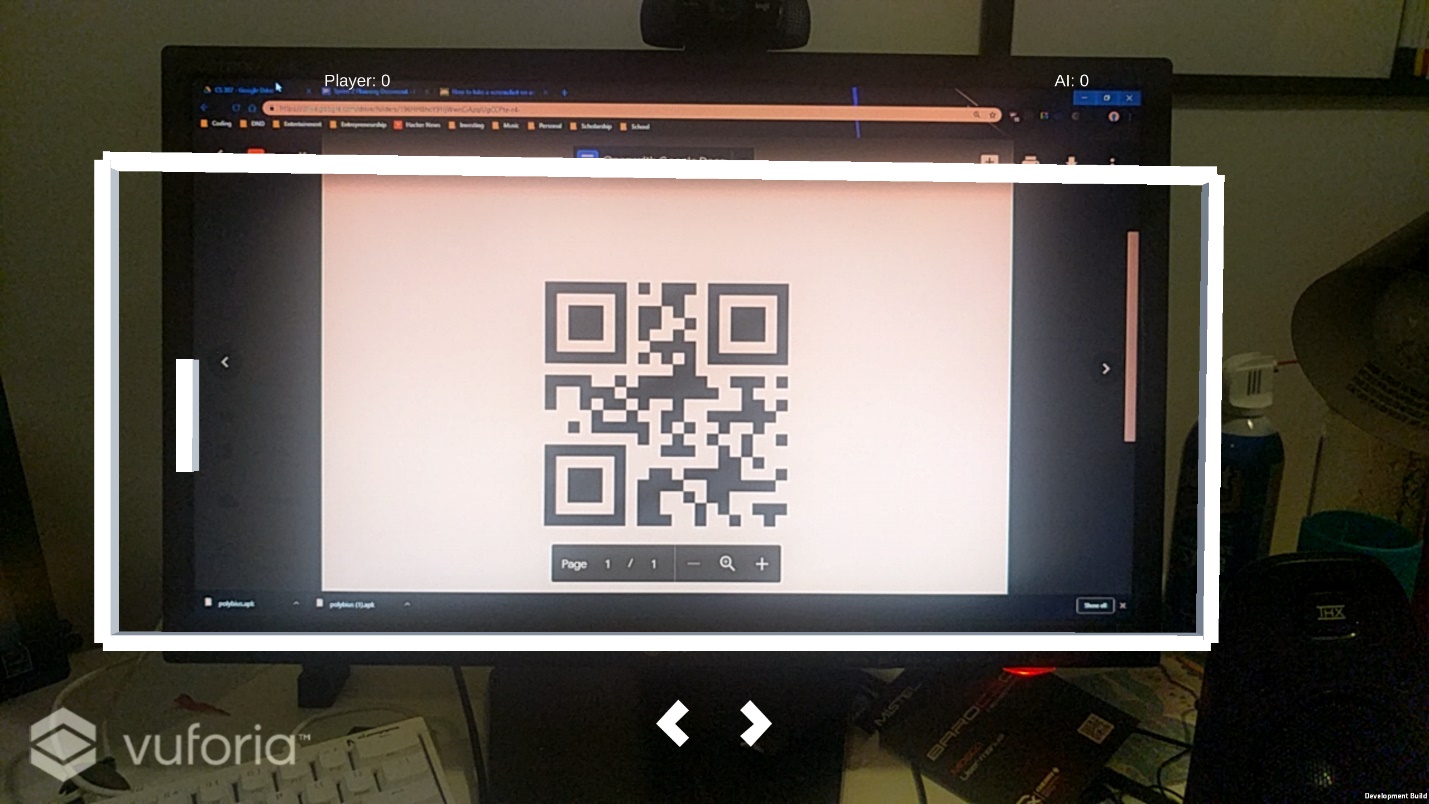
**Testing Eight:** Pong can be played by the user through their device

Procedure:

* The user opens an instance on pong
* Pong is interactive on their device

Results and Follow-up:

1. The user opens an instance on pong: is successful. Pong correctly opens on the users device and displays over the QR Code. Picture below.



1. Pong is interactive on the user device: is successful. The paddle on pong moved based on what buttons are pressed on the bottom of the screen. Picture removed for redundancy.

Final Testing Summary:

We have found that while most of the features work there are still some functionality that needs to be reworked to have a functional application. In addition to our manual testing we have an automated test cases which test the server’s functionality to verify that database scripts are functional and that can verify that the database is being updated.