

Sprint 2 Manual Testing Documentation

This document will outline the procedures and results of manually testing some of the user stories as specified in our sprint planning document. Any user story tests not mentioned in this document were tested with unit tests.

User Story 2 Test:

User Story: "I would like to be able to change my display name so that it gives my account more character/personality."

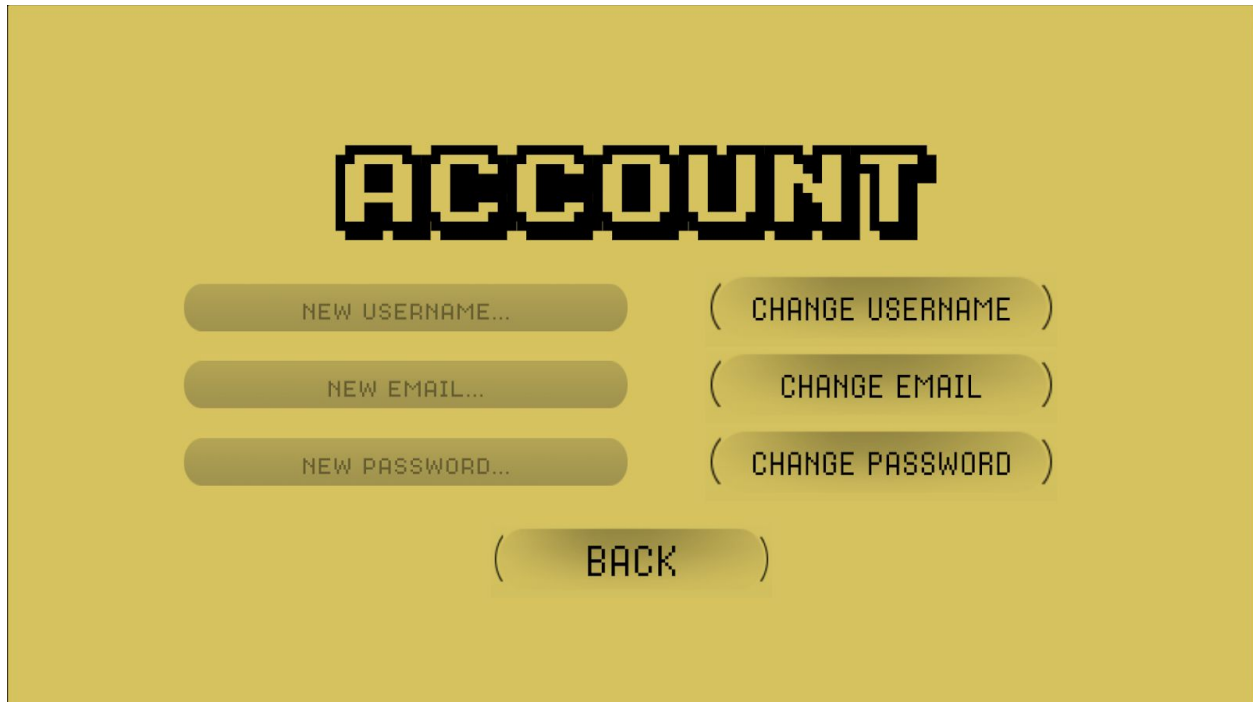
Test: Test display name possibilities and make sure we catch edge cases

Testing Procedure:

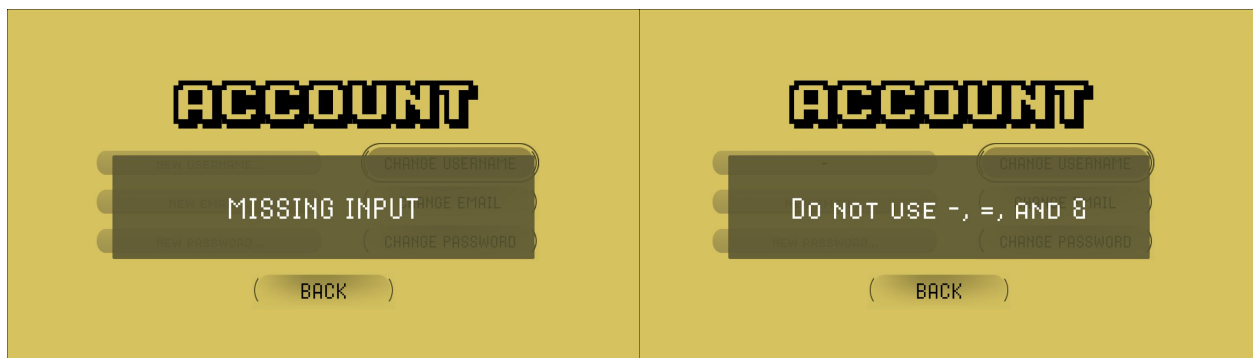
1. Navigate to account screen
2. Test edge cases for bad inputs regardless of the database information
 - No input
 - Inputs longer than 50 characters
 - Inputs containing "-", "=", or "&"
3. Compare results to expected results
 - An error message should pop up stating "MISSING INPUT"
 - An error message should pop up stating "..."
 - An error message should pop up stating "DO NOT USE -, =, &"
4. Test edge cases for bad inputs dependent of the database information
 - Username already exists in the database
 - Email already exists in the database
5. Compare results to expected results
 - An error message should pop up stating "USERNAME ALREADY TAKEN"
 - An error message should pop up stating "EMAIL ALREADY TAKEN"
6. Use successful input
7. Compare results to expected (should state "SUCCESSFULLY CHANGED")

Results:

1. Account screen: Login → Profile → Account - successful. The app can navigate to the account page without any issues.

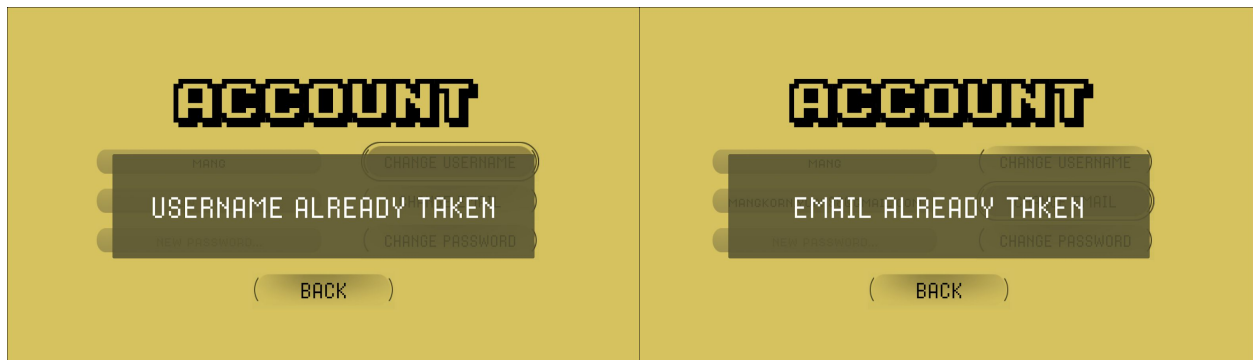


2. Edge cases regardless of the database information - successful. If there is no input or an error in the input then the respective error message will appear for each case.

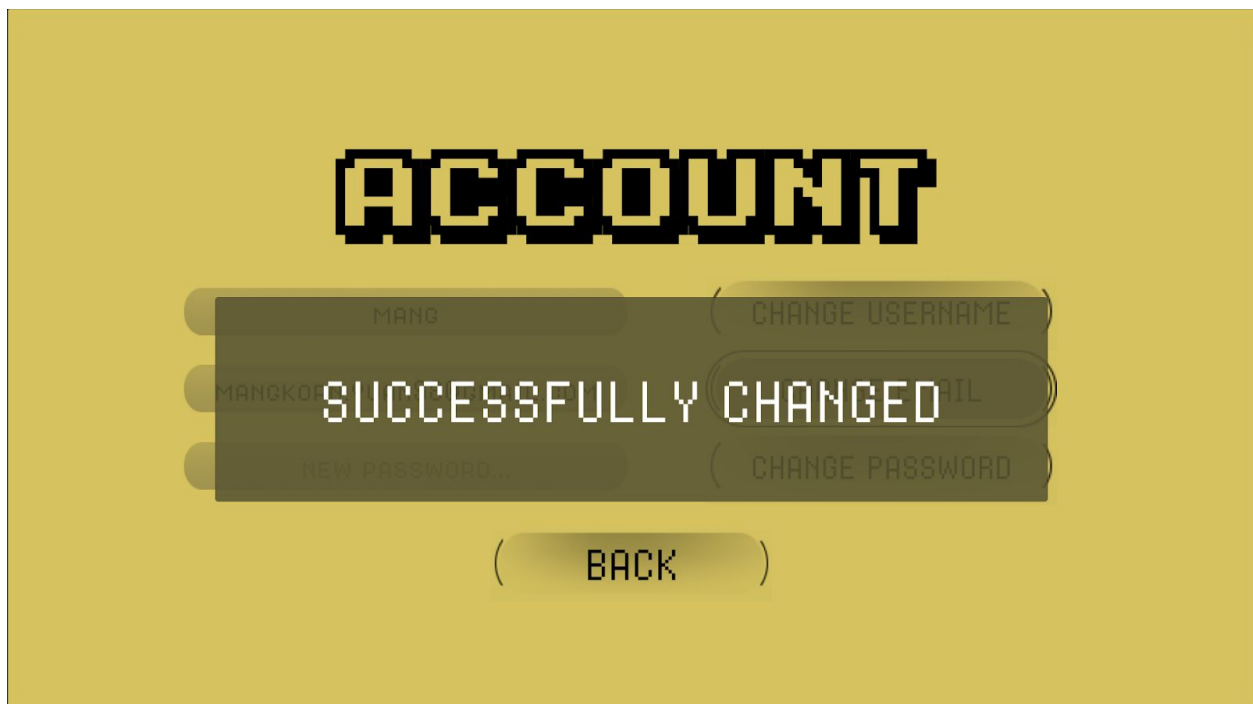


{INSERT 50 CHARS MAX SCREEN}

3. Edge cases dependent of the database information - successful. If the username is already taken or the email is already taken then their respective error message will appear.



4. Successful input - successful. A successful change in username, email, or password will show the following screen.



Testing Summary: The manual testing for User Story 2 was successful. Inserting an empty input, one containing over 50 characters, or one containing "-", "=", "&" resulted in an unsuccessful change in username, email, or password. Entering an username or email that already exists in the database also resulted in a failed change. And entering an input that does not contain any of the edge cases was successfully changed. Each of these cases resulted in a message that allowed users to know whether or not the change was successful.

User Story 6 Test:

User Story: "I would like to be able to search for specific locations inside of the app in case I really want to go see a specific point of interest."

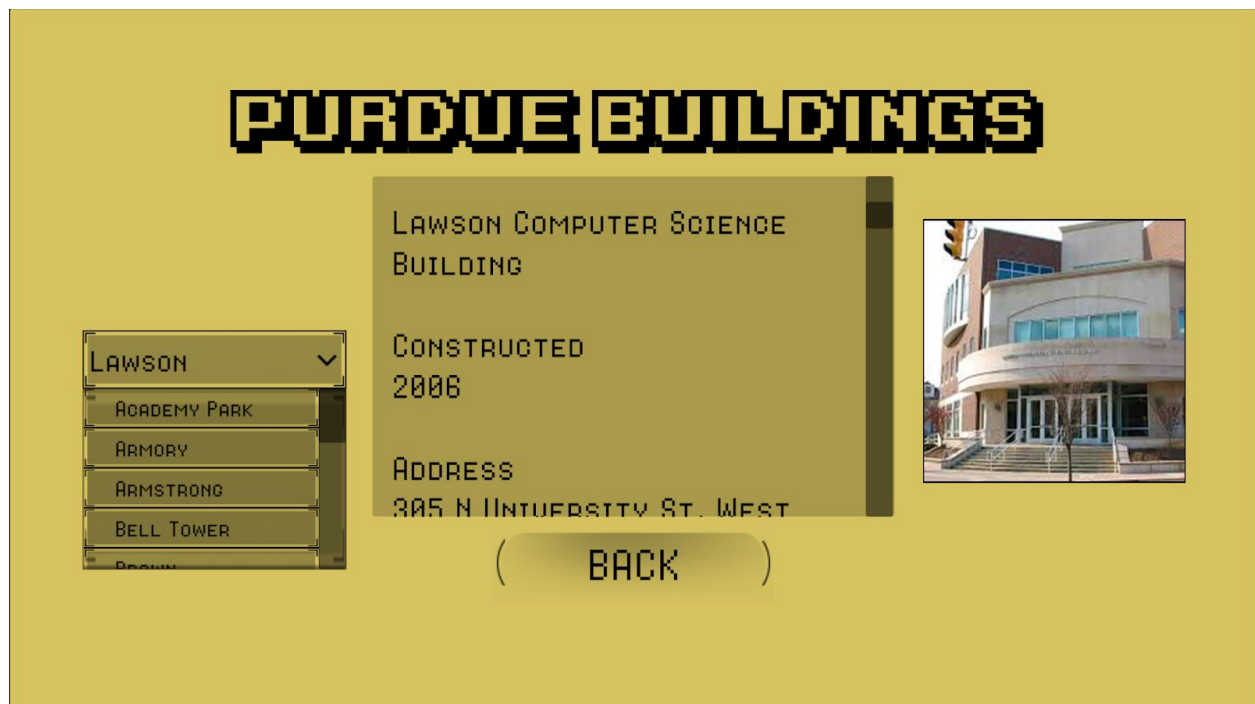
Test: Selecting a location provides information visible to the user

Testing Procedure:

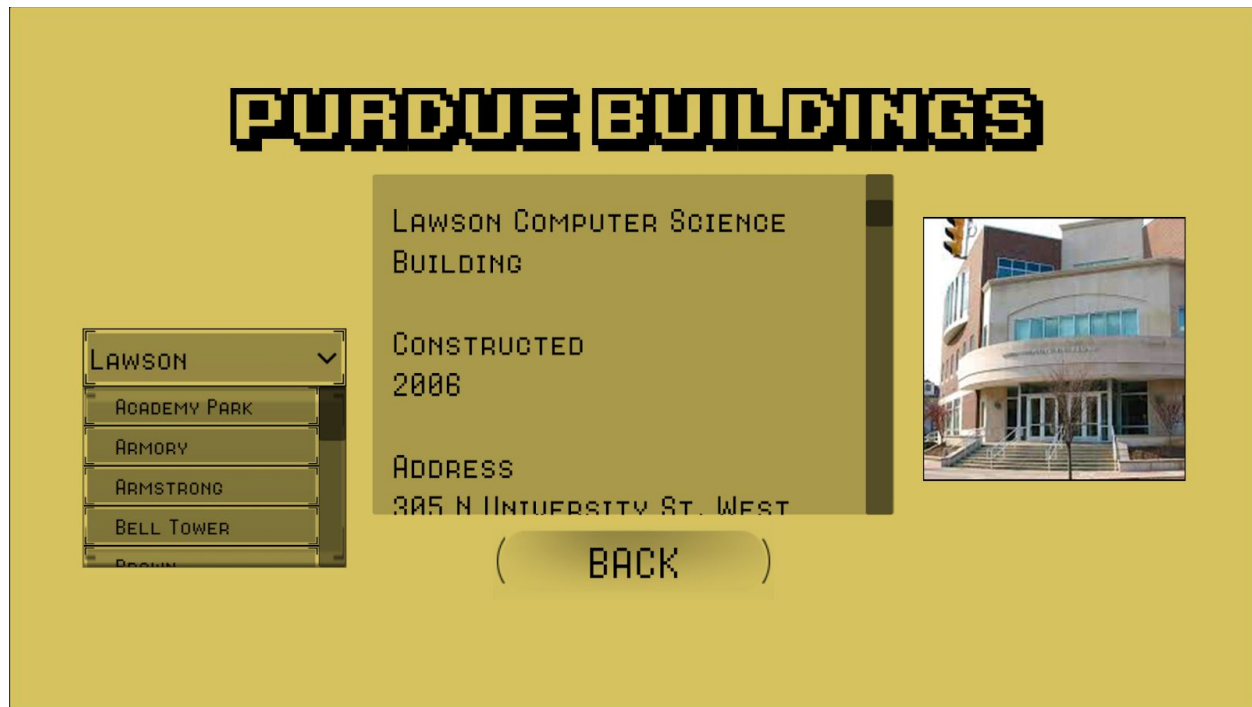
1. Navigate to the Purdue buildings search screen
2. Select a location from the drop down menu to learn more about (Ex: Lawson)
3. Compare the results to expected results (information and an image of Lawson should appear)
4. Repeat steps 2-3 to ensure that the search is functional for all locations

Results:

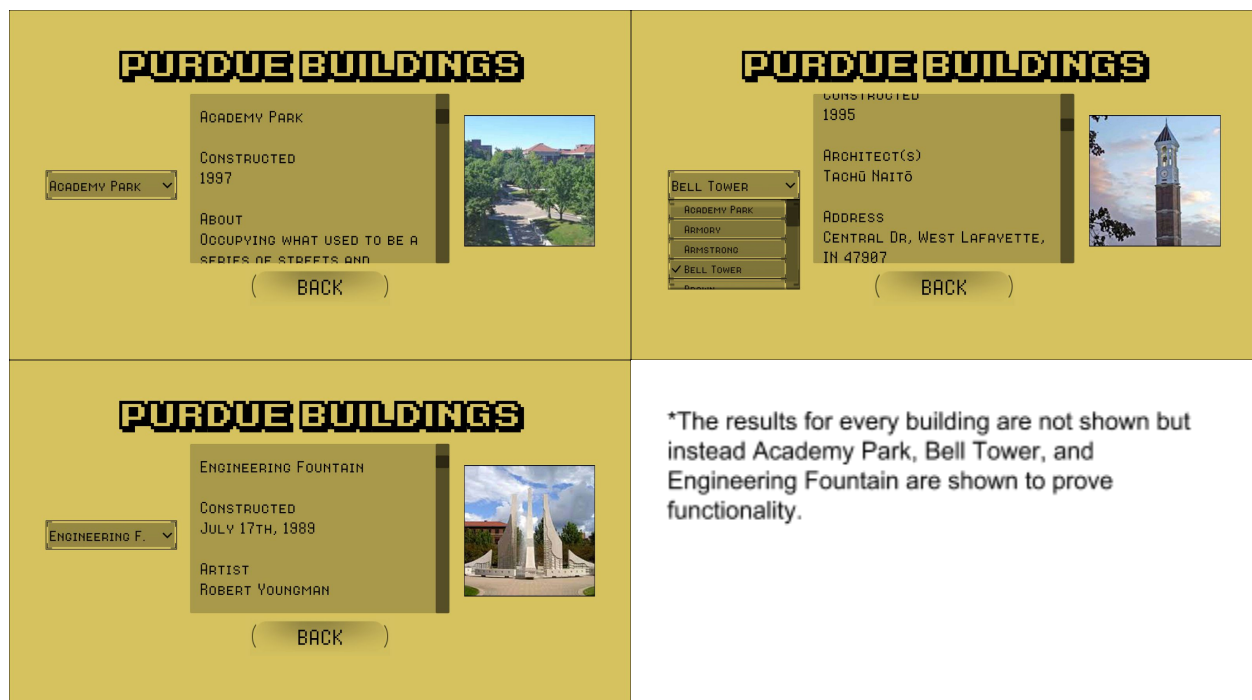
1. Purdue building information search screen Login → Profile → Purdue B - successful.



2. Selecting Lawson from the drop down menu shows information and an image of Lawson - successful. As shown below, Lawson was selected and information about Lawson is shown in the middle and an image on the right.



3. Selecting (any Purdue building or location) from the menu shows information and an image of (the building or location) - successful. Repeating step 2 for different locations works just as well.



*The results for every building are not shown but instead Academy Park, Bell Tower, and Engineering Fountain are shown to prove functionality.

Testing Summary: The manual testing for User Story 6 was successful. By using the drop down menu I could select any of the buildings or locations to learn about. Information about each of the selected building or location was given along with an image of it. I could scroll in the middle portion to read and learn more about the chosen building or location. This applied for all of the options on the drop down menu.

User Story 7 Test:

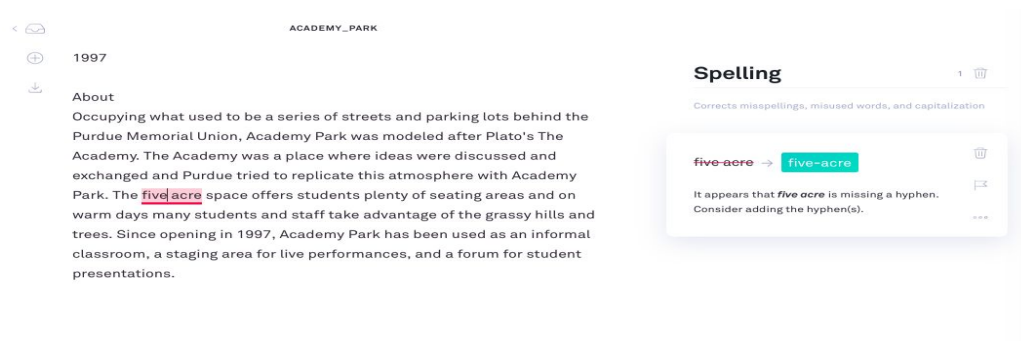
User Story: "As a user, I would like to be able to view the history of buildings on campus."

Test: Written text passes spelling and grammar check in third party applications.

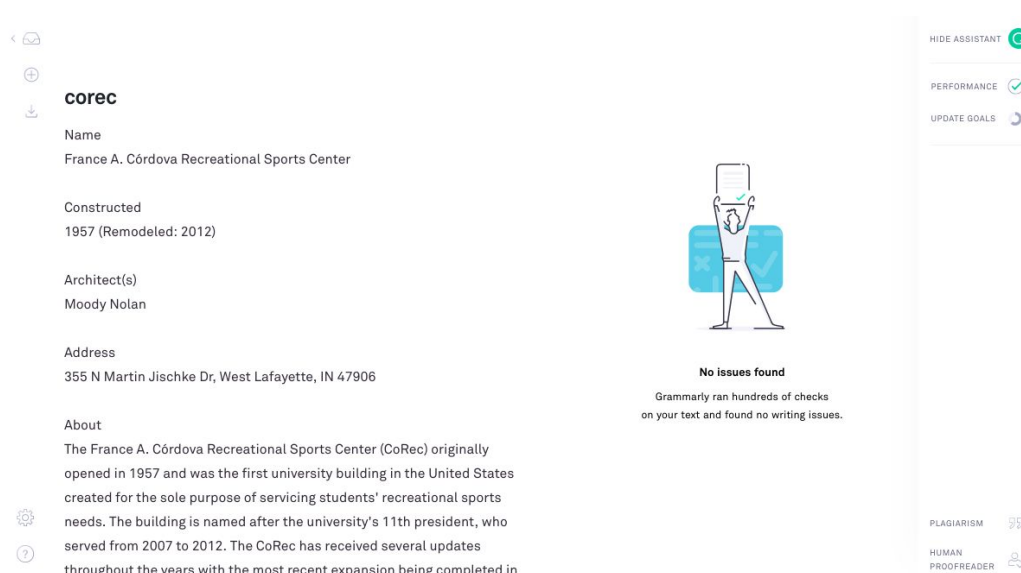
Testing Procedure:

1. Copy written text to the Grammarly Spelling and Grammar check software
2. Fix spelling and grammar errors
3. Move text to final file and add to GitHub

Example Spell Check Procedure 1:



Example Spell Check Procedure 2:



Example Spell Check Procedure 3:

The screenshot displays the Grammarly web interface. On the left, a document titled "hovde" is open, showing fields for Name, Address, and About. The "About" section contains a paragraph about the Hovde Hall of Administration, with the phrase "longest serving" highlighted in red. A spelling correction popup is visible, showing "longest-serving" as the correct form. The right sidebar shows the "Spelling" section with a red indicator for one error. The top right corner shows the "HIDE ASSISTANT" button and a green "G" icon.

Document Content:

hovde

Name
Hovde Hall of Administration

Constructed
1937

Address
610 Purdue Mall, West Lafayette, IN 47907

About
The Hovde Hall of Administration was originally called the Purdue Executive Building before being renamed in honor of former Purdue president, Frederick L. Hovde. Hovde was president during Purdue's greatest growth in enrollment and budget. During Hovde's 25 years as president, he quadrupled the number of students enrolled and increased the university's budget tenfold. Additionally, Hovde also established the schools of industrial engineering, materials engineering, and technology and veterinary medicine. Hovde retired in 1971 as Purdue's longest serving president.

Spelling Correction:

Spelling 1

Checks spelling, confused words, capitalization

longest-serving → **longest-serving**

It appears that **longest serving** is missing a hyphen. Consider adding the hyphen(s).

Right Sidebar:

- HIDE ASSISTANT
- PERFORMANCE 91
- UPDATE GOALS
- SPELLING 1**
- PREMIUM ALERTS
- PLAGIARISM
- HUMAN PROOFREADER

Results:

The Grammarly did a great job identify grammar and spelling mistakes.

Testing Summary:

The tests were a success and all building information files have been created, spell checked, and uploaded to the GitHub repository.

User Story 8 Test:

User Story: "I would like to have preset guide routes I can follow in case I don't really know good locations to visit."

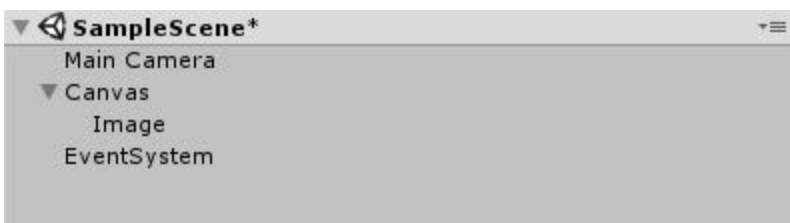
Test: Test that images are displayable in Unity

Testing Procedure:

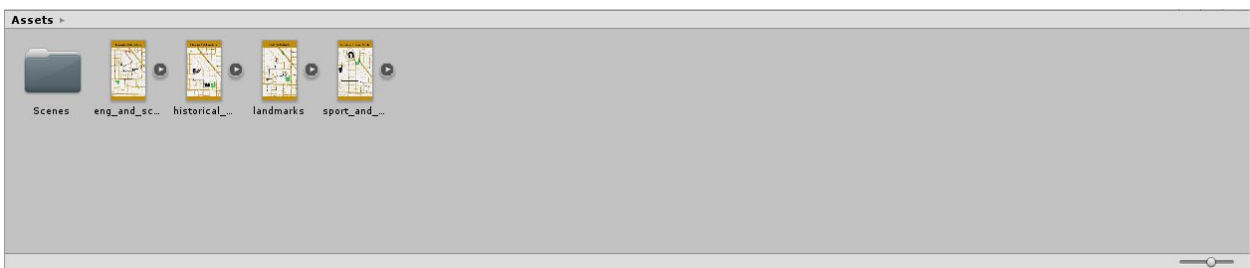
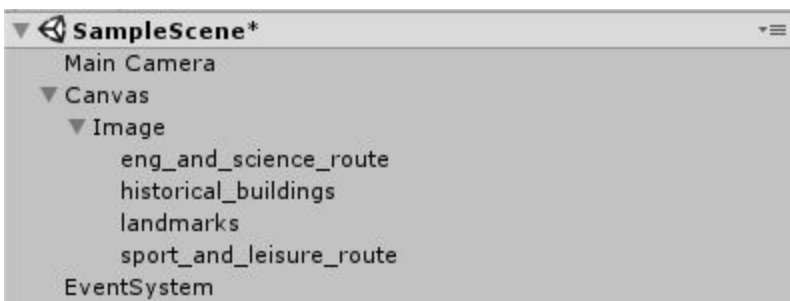
1. Create UI Object
2. Add images to object
3. View images in scene

Results:

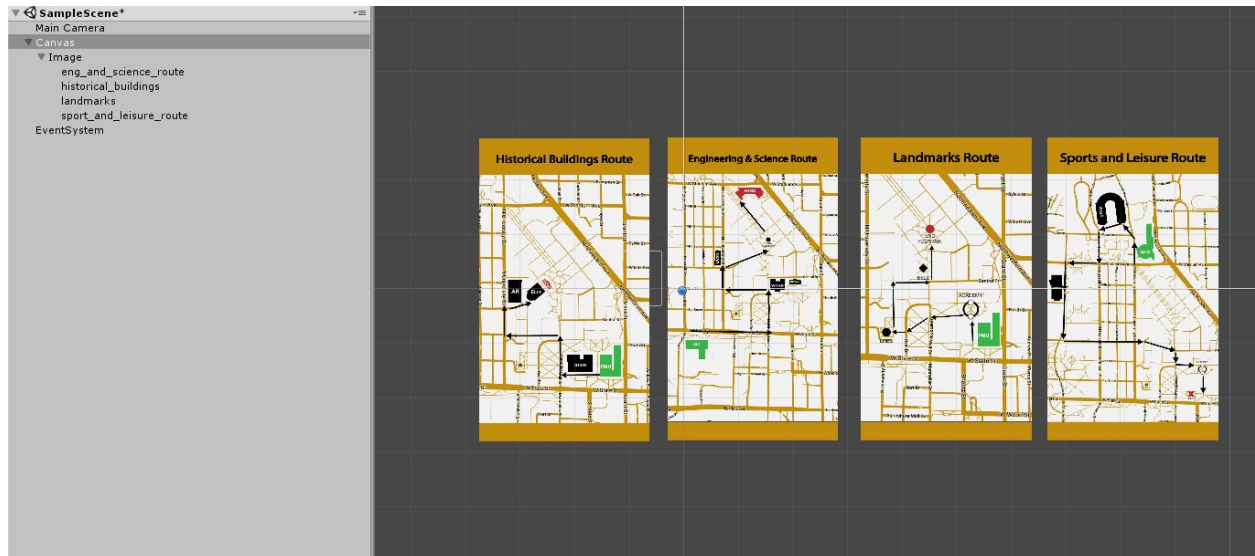
1. Create UI image object



2. Add images to object



3. View images in scene



The results of this test were a success.

Testing Summary:

After performing the test we can see that the images successfully load into Unity and are able to be displayed in a scene. This means that when the routes are ready to be added to the final application, we will be able to add and manipulate the images in any way that we see fit.

User Story 10 Test:

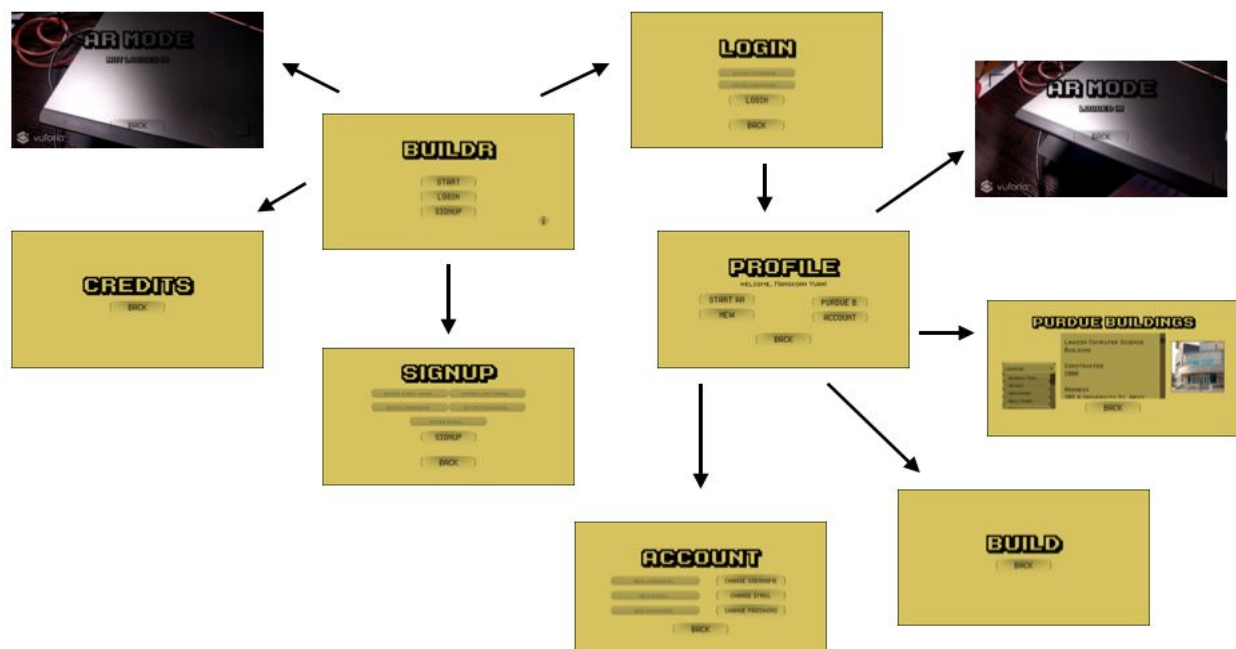
User Story: "I would like to have a polished UI that can be easily navigated."

Test: Test navigations of new UI

Testing Procedure:

1. Navigate the app. The following will test if the navigation to each screen works properly:
 - Open app → main menu
 - Start, login, or sign up, credits
 - Credits
 - Team credits
 - Signup
 - Enter username, email, and password
 - Start → AR mode (Not logged in)
 - Login → Profile
 - Start AR, Account, Build, Purdue B, or Back
 - Start AR → AR mode (logged in)
 - Account
 - Change username, change email, or change password
 - Build → Building objects mode (*not yet implemented in app)
 - Purdue B → Drop down menu selection

Results: App navigation - successful. The following is a chart of the navigation of the app.



Testing Summary: The manual testing of User Story 10 was successful. I could easily navigate the app and select what I wanted to do without difficulty. From the main menu I could begin using the AR, signup, or login. After logging in I could select to use the AR, lookup information about purdue locations, change my account information, or build structures. And there is always the option to go back to the previous screen to make the navigation simpler.

**Note that the building structures and chat room portion of the app has not been implemented into the app yet*

User Story 11 Test:

User Story: "As a user, I would like to be able to login to my Buildr account so that I can interact with the app and other users."

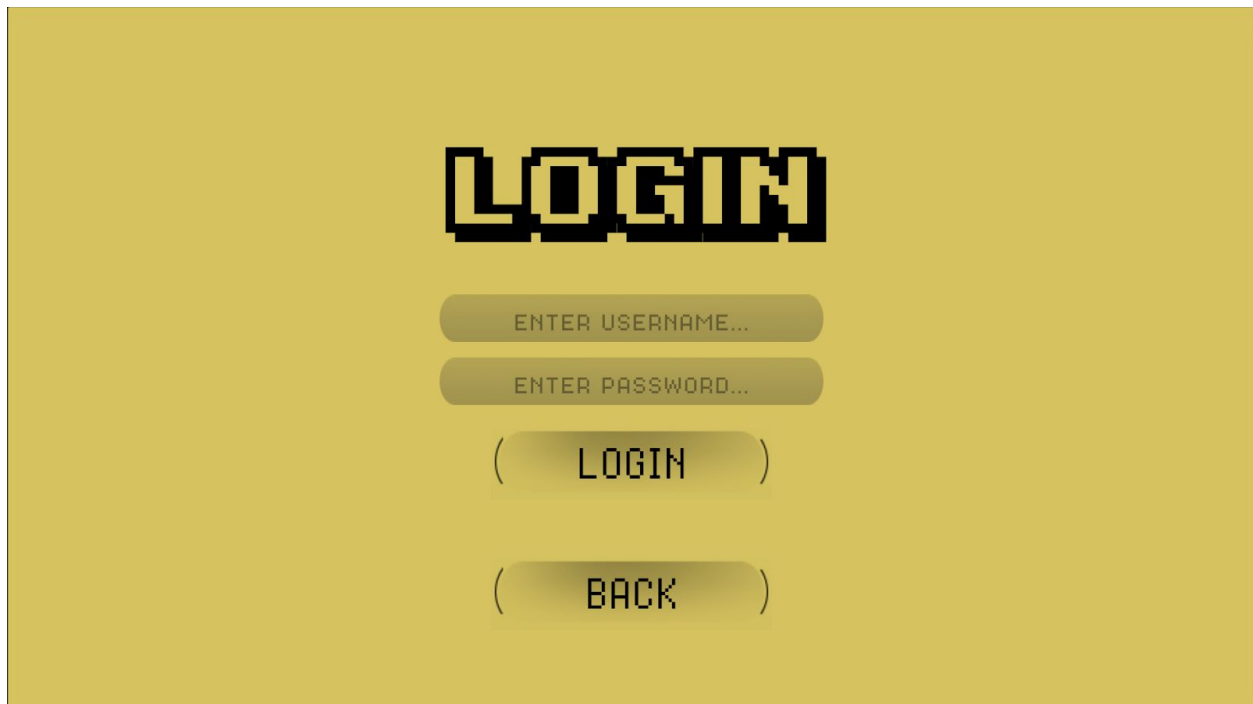
Test: Test proper operation of login request

Testing Procedure:

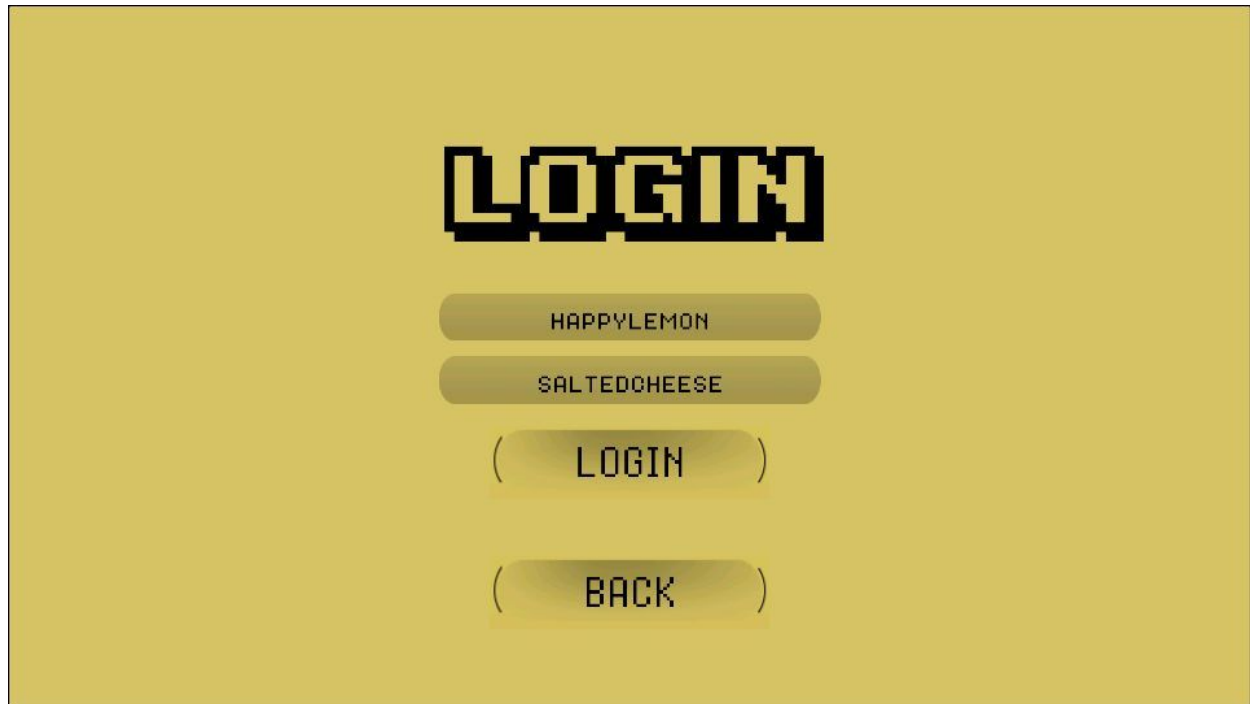
1. Navigate to the login screen
2. Input a username and password that exist in the database
3. Compare the results to the expected results (login should be successful)
4. Go back to the login screen and input a username and password that does not exist in the database
5. Compare the results to the expected results (login should fail)

Results:

1. Login screen - successful. Navigating to the login page was successful where the user can enter their username and password.



2. Input user information that exist in the database - successful. The user can enter their user information in the space given.



LOGIN

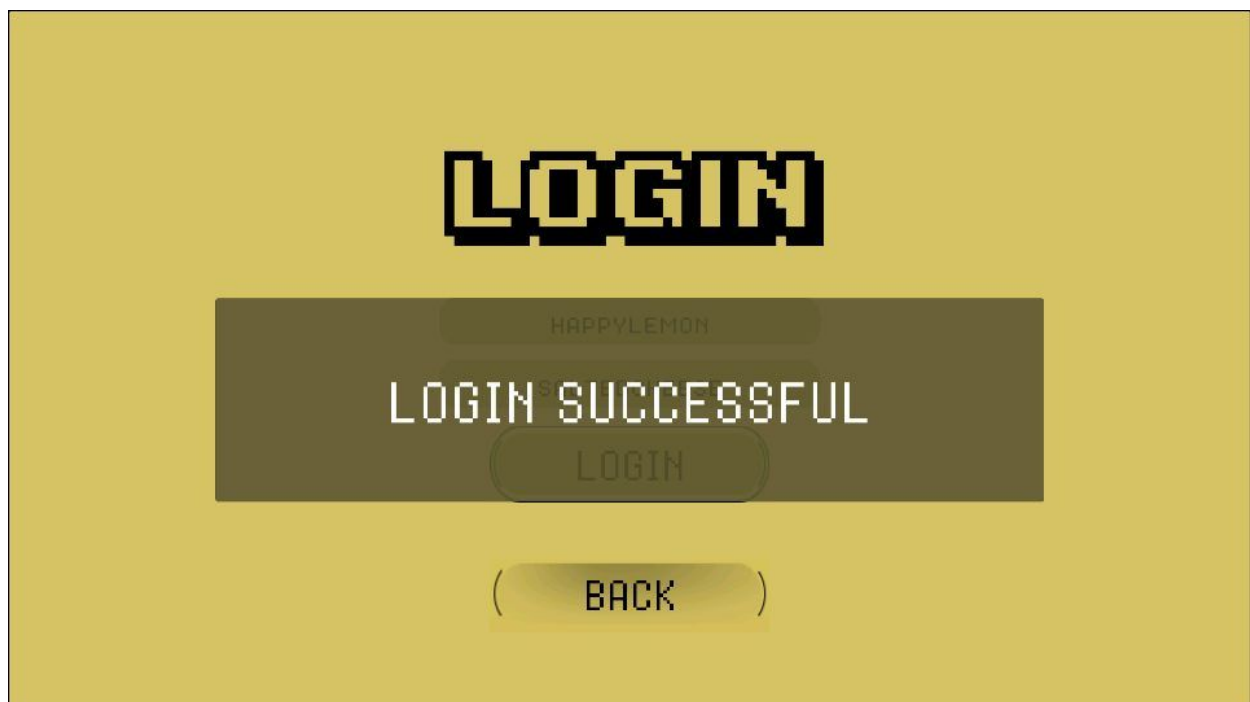
HAPPYLEMON

SALTEDCHEESE

(LOGIN)

(BACK)

3. Login successful - successful. As expected, since the user already exist in the database, the user will be logged into their account successfully.



LOGIN

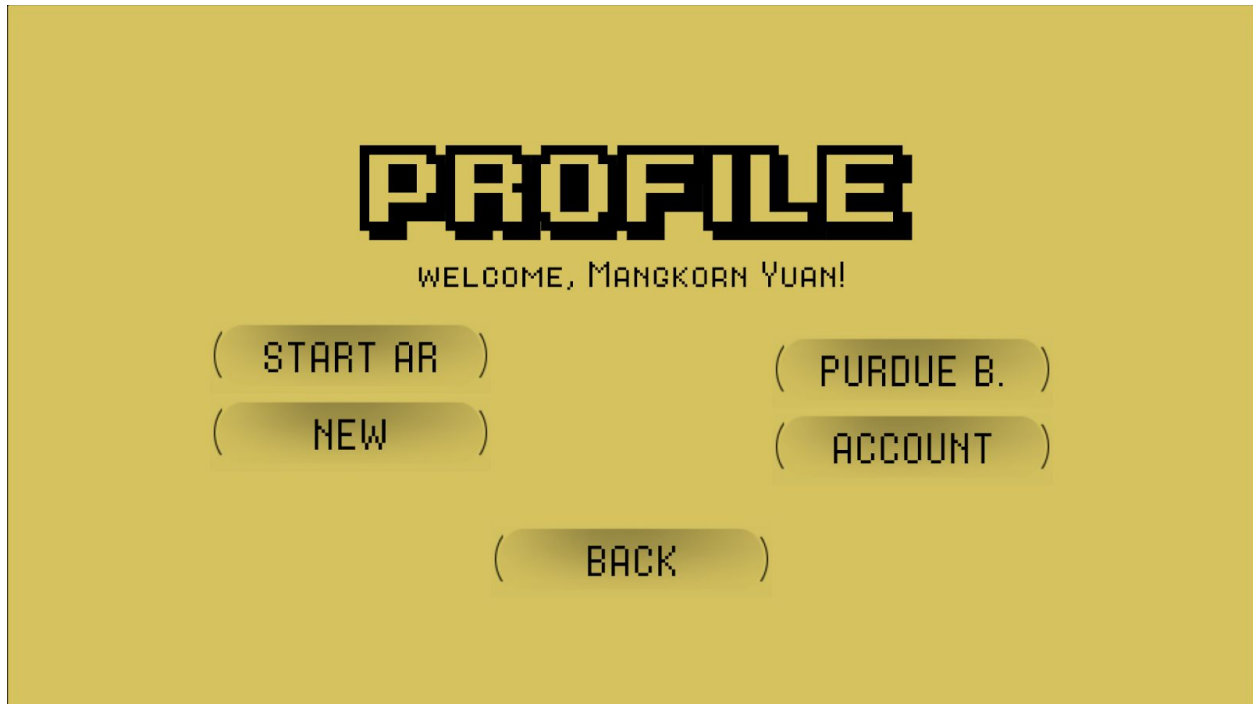
HAPPYLEMON

LOGIN SUCCESSFUL

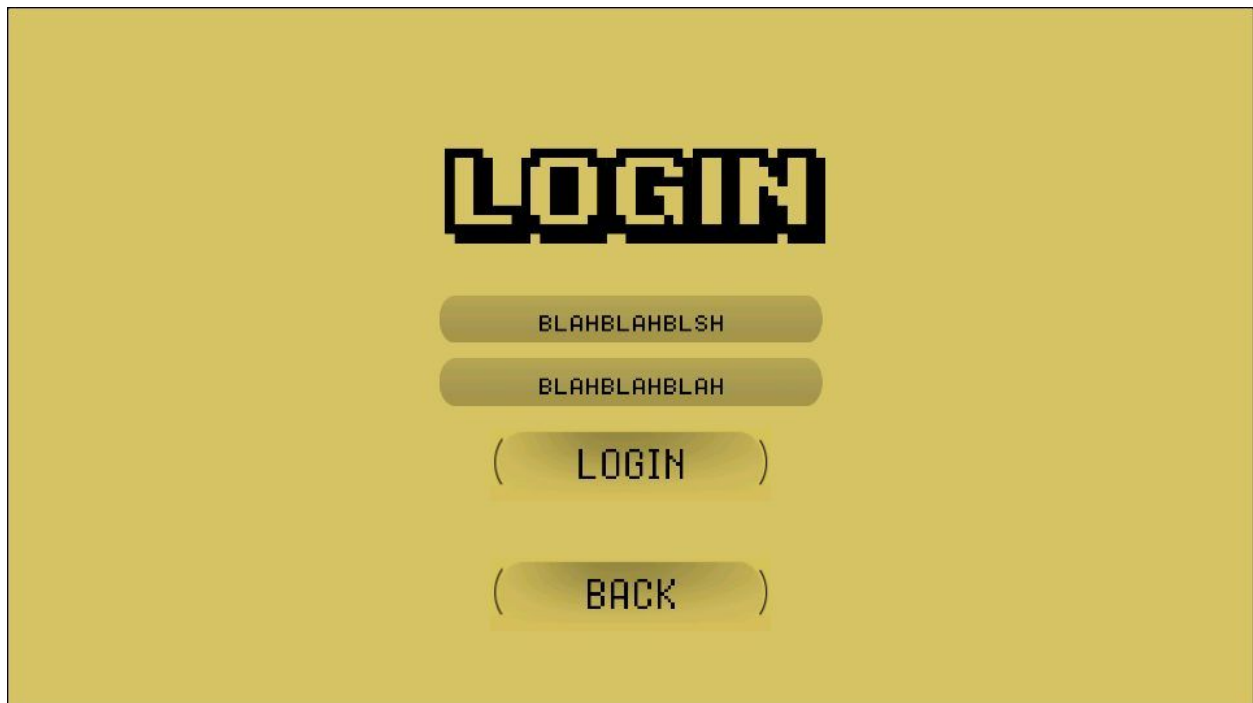
LOGIN

(BACK)

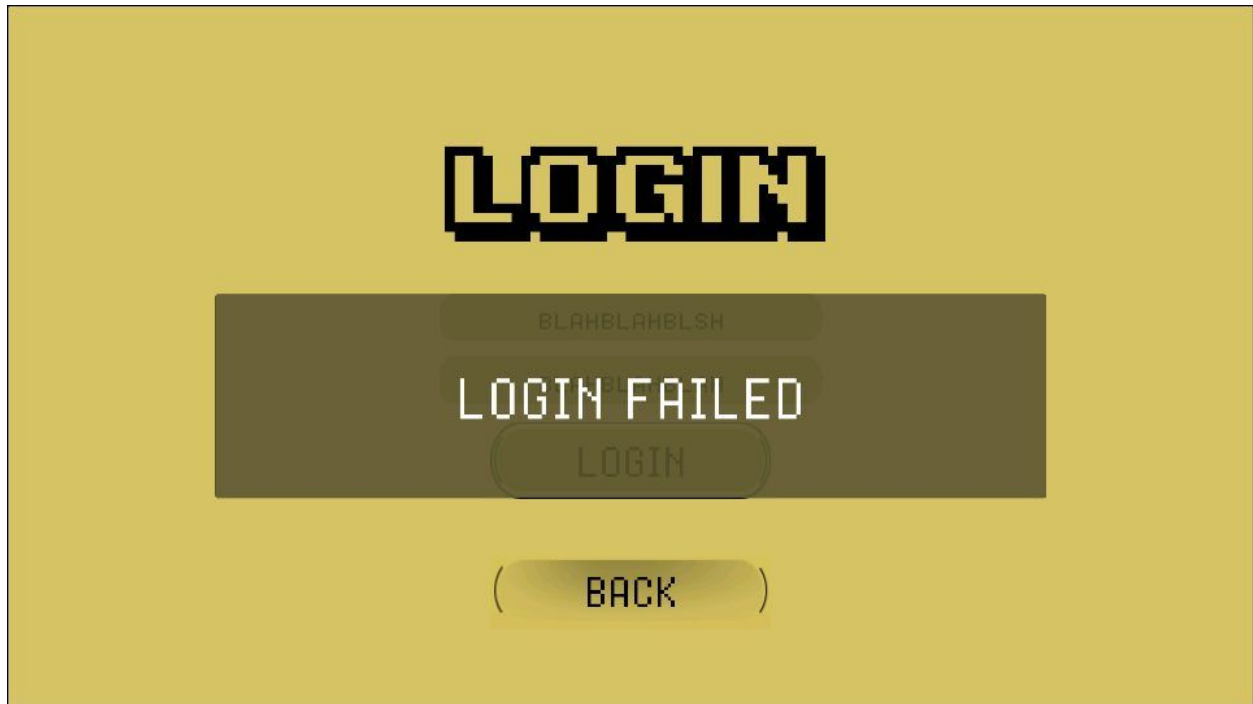
4. Takes user to their buildr home page - successful. Successfully logging in re-logs the user into their account.



5. Insert user information that does not exist in the database - successful. User can enter their information below even if it does not exist.



6. Login Unsuccessful - successful. The attempt to login in with user information that is not existent in the database fails and remains on the login screen.



Testing Summary: The manual testing of User Story 11 was successful as all inputs resulted in outputs that were expected. If a user that exists in the database tried to re-login into their account, the user could do so. And if a user that does not exist in the database attempted to login, the login would fail and remain on the login screen.

User Story 13 Test:

User Story: "As a user, I would like to be able to communicate with other users on the application."

Test: Message is added properly and updated in the chat

Testing Procedure:

1. Navigate to group chat
2. ...

Results:

- 1.

Testing Summary: