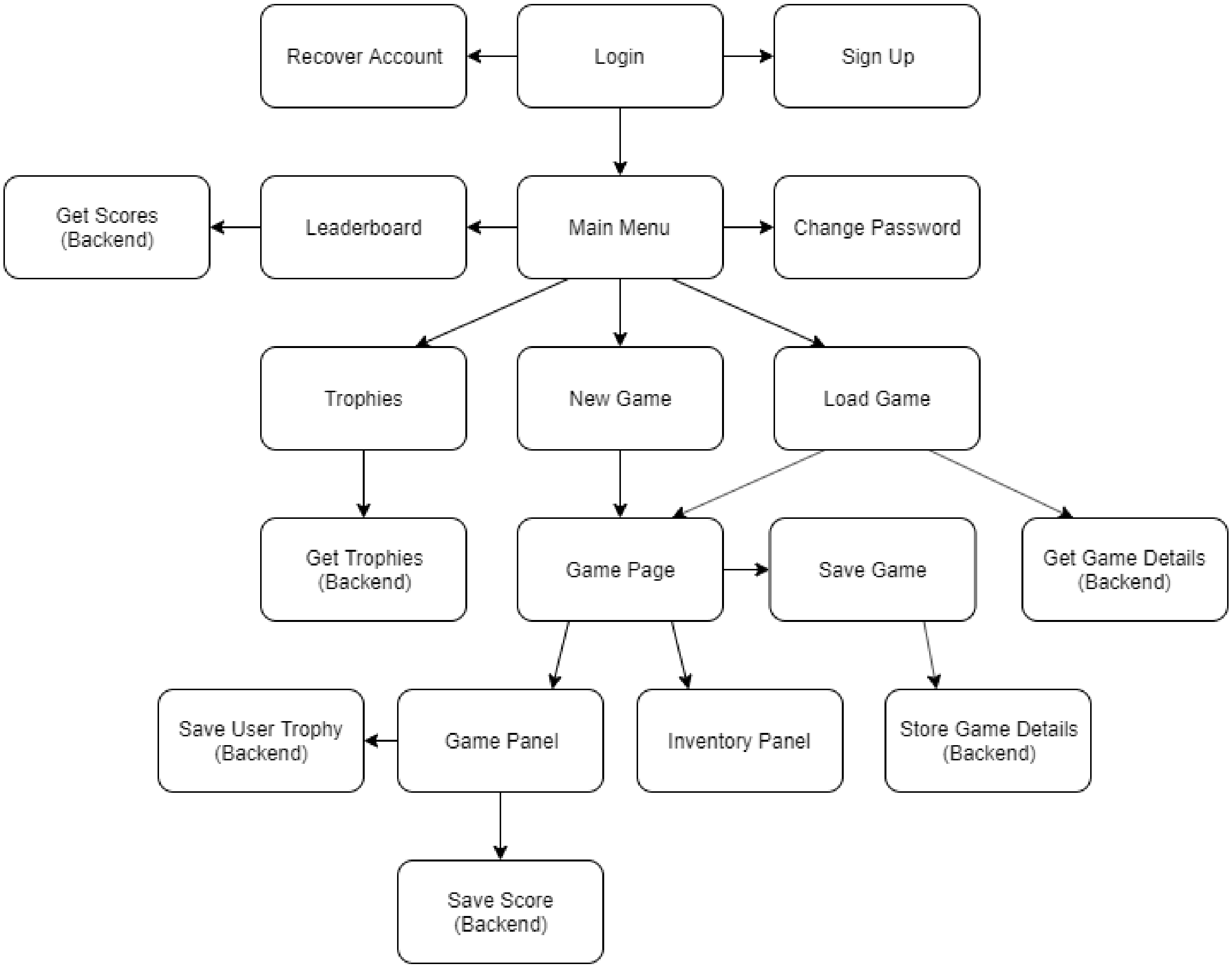
Sprint 1: Incremental and Regression Testing

Julian Ng, Cole Compton, Adrian Taubner, Christian McKain, Jacob Riggs

Component Structure



Modules

Login

- The login module will authenticate users through the backend and allow access to the rest of the site

Sign Up

- The Sign Up module will handle new user creation through the backend and enable users to access the rest of the site through the login module

Recover Account

- The recover account module will send you an email to reset your password given an email

Main Menu

- This module will allow the user to choose other modules to navigate to, including the change password page, the leaderboard, the trophy room, their saved game, and new game.

Leaderboard Module

- The leaderboard will display a certain number of the highest scores so far achieved by any user along with their username and other data about their game.
- Users will be able to see their name highlighted in the list if they appear on the leaderboard
- If a user does not appear on the leaderboard, they will appear below it with their rank out of all runs.

Change Password

- The Change Password module will allow a user to modify their login password.

Trophy Room

- The Trophy Room module will act as a way to show users special achievements that they have earned while playing the game. Trophies will have a name, image and description associated with them which will be stored on the frontend and displayed based on a response from the backend listing which trophy ids have been earned by a given user.

New Game

- The New Game module will provide a prompt for the user to enter information for a new game

Load Game

- The Load Game module will allow users to continue a saved game by retrieving it from the database and displaying the saved inventory and current event right where it had been left off.

Game Module

- The Game Page has two child components, the game panel and the inventory panel.
- Game Panel
  - This is where users can play the game, experience different events, make decisions, and ultimately try to get the highest score before they get to game over.
  - Current score is displayed at the top of the game screen.
  - The backend will save current score if a user chooses to save their game or when they lose
- Inventory Panel
  - This is where the items users have collected will be stored.
  - The backend will be responsible for saving the contents of the inventory whenever a user chooses to save their game.
  - Items may affect player stats or allow certain choices in events.
  - Items may be dropped anytime or acquired at events.

Testing Approach

We decided to use top-down testing for our application because most of our work this sprint is focused on the upper-level modules, such as creating an account, logging in, or seeing trophies and the leaderboard. This way, when we later add modules onto these, we can continue to incrementally test from the top down.

Incremental & Regression Testing

Module: Login

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	When logging in, the message ‘Incorrect login, please try again’ shows up, even if the username and password is correct	Critical	Check how variables are being passed from the forms field to the method call and vue store
2	When logging in with static values page will not change to the main page	Critical	Check how the response from server is being handled.
3	Default page is not the login page	critical	Edit routing table in main.js to add a catch all redirect field when attempting to visit non described routes

Regression Testing

Defect No.	Description	Severity	How to Correct
1	Post request changed into option	critical	Add a CORS ignore to the server as the

	request preventing the server from responding		front and backend do not share a domain
2	Failed logins cause the page to redirect to /login?	low	Set force redirect to false of a failed login to prevent url change

Module: Sign Up

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	When signing up, email account without "@"-sign can be used	Important	Check email input when signing up
2	When signing up, you will stay on the same page	Critical	Redirect to new page when signing up
3	When creating an account, the error message 'Incorrect login. Please try again' would always appear	Critical	Replace incorrectly named variables this.username and this.usernameAgain to this.password and this.confirmPassword.

Regression Testing

Defect No.	Description	Severity	How to Correct
1	Checking to make sure an account can be created with an email, username, and password	Critical	Send a request to the server to create a test user account to test the creation functionality, then another request to delete the test user

Module: Server

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	Checking login information by attempting to encrypt the password sent by client and compare it to password held in database.	Critical	Hash passwords instead and compare a hash of password from client to hash of password in the database.
2	Attempted to encrypt email addresses and retrieve emails addresses.	Workaround	Hash emails instead instead of using encryption.
3	Did not consider CORS for receiving requests.	Critical	Imported the cors library for node.js.
4	Requests not being read correctly as a JSON.	Workaround	Use the body-parser library for node.js.
5	Inserted values out of order when creating a new user.	Critical	Fixed the order of the inputs to match that of the database attributes for the User table.
6	Was ordering scores incorrectly when passing them to the client.	Workaround	Used the mysql ORDER BY to query the information in the correct order.
7	Did not change the username for every table value that was related to a	Important	Also UPDATED the values in the Scores and UserTrophies

	username that was changed e.g. (score, trophies)		tables for usernames.

Regression Testing (Automated using Postman)

Defect No.	Description	Severity	How to Correct
1	Checking that the login request is doing the correct checks to filter the correct user-password combination.	Critical	Test using Postman that makes login requests including good and bad requests.
2	Checking that changing usernames does not affect any interactions with both the front end and database.	Important	Test using Postman that makes a change username request and then analysis of the database and front end afterwards.

Module: Database

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	No relationship between users and their trophies established.	Workaround	Made a table UserTrophies that connects users with the trophies they've earned.
2	Scores in the score table had a primary key of user.	Workaround	This made it so only 1 user score was displayed per user, but changed the table to have the user and score attributes be key attributes.

3	Trophies table was keeping track of image files.	Workaround	Not enough space in the database we were using until we had to pay money so instead we saved references to these images.

Regression Testing (Automated using Postman)

Defect No.	Description	Severity	How to Correct
1	Adding users to the database and duplicates were allowed.	Workaround	Make a check to the database before adding to the database. Request in Postman setup to check for successful adding of user.
2	Show all of the current scores of all users in the scoreboard.	Workaround	Would like to only send the top 5 or so high scores of each user. Postman setup for testing score return results.
3	Show all scores of a specific user.	Workaround	Due to only the top 5 scores of each user being shown usually, instead we need to display all scores of a user. Postman setup for user score return results.
4	Password changes need to be stored hashed rather than as a string.	Important	Postman testing of changing a password and then attempts on login. Also checking database to make

			sure that the new passwords are hashed and kept secure.
5	No functionality to add trophies to test the Trophies table in the database.	Workaround	Created a temporary method used to add trophy entries so we are able to test with user relation to trophies.
6	No functionality to delete entries from the database.	Workaround	When testing with the database, a lot of information is stored that is purely for testing and we need a way to get rid of it. We use the username as the way to delete entries from multiple tables.

Module: Leaderboard

Incremental Testing

Defect No.	Description	Severity	How to Correct
1	When arriving on leaderboard page, no scores are shown	Critical	Check the database to see if there are any users in the scores table and check to see how the server retrieves the scores from the database and sends them to the client
2	Leaderboard	Critical	Check how the

	scores are not sorted		client receives and sorts the scores before displaying them
3	Usernames are not listed	Critical	Check the messages being sent by the server to make sure all the information is there

Regression Testing

Defect No.	Description	Severity	How to Correct
1	Test functionality with test data	Important	If there is no information in the scores table on the database, create test scores and check the functionalities of the leaderboard

Module: Trophy Room (No Defects)