

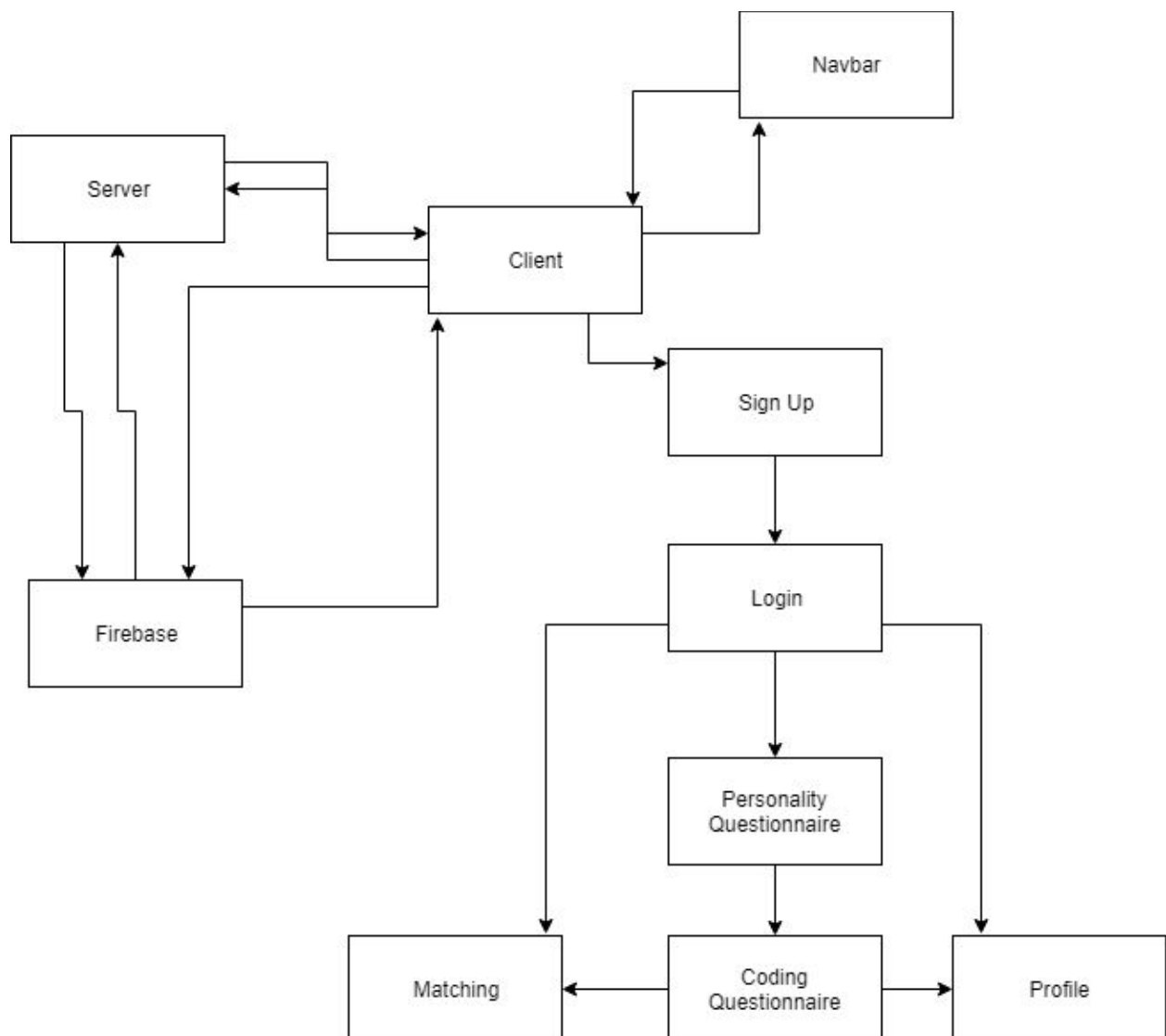
# Incremental and Regression Testing

Team 8

Team Members: David Carr, Shivan Desai, Reid Kippenbrock, Laura Kristie, Paul Krivacka

## 1. Classification of Components

Define Components



**Firebase:**

*Function:* Stores and retrieves the data for our website.

*Interaction(s):*

Server: The server requests data from and stores data in firebase.

Navbar: Uses firebase's authentication to track whether or not the user is logged in, and their account if they are. It also lets users log in or sign up, both of which send and get data from firebase.

Personality Questionnaire: Receives the user's answers for the personality questionnaire and stores them

Coding Questionnaire: Receives the user's answers for the coding questionnaire and stores them

Matching Page: Returns possible matches for the user to the matching page.

*Input:* Incoming request with JSON-formatted information in the body

*Output:* A "success" or the requested information

*Dependent Components:* Server, Sign Up, Login, Navbar, Personality Questionnaire, Coding Questionnaire, Matching Page

*Dependent On:* None

**Server:**

*Function:* Host the website, store data in the database, retrieve data from the database, and the matching algorithm.

*Interaction(s):*

Firebase: The server requests data from and stores data in firebase.

Personality Questionnaire: Sorts through a user's personality questionnaire answers to determine who they will match to.

Coding Questionnaire: Sorts through a user's coding questionnaire answers to determine who they will match to.

Matching Page: Returns possible matches for the user to the matching page.

*Input:* A user client connecting with the server

*Output:* The information the client requested/a response back

*Dependent Components:* Client as a whole, including Sign Up, Login, Navbar, Personality Questionnaire, Coding Questionnaire, Matching Page

*Dependent On:* Firebase

**Nav Bar:**

*Function:* Acts as a navigator for the site. A link to most of the main pages is in the nav bar. Users can use the nav bar to create or log in to their accounts.

*Interaction(s):*

Firebase: Gets the user authentication from firebase to see whether or not a user is logged in. It also sends the login and sign up requests to firebase and receives a response.

Matching Page: Appears on the matching page.

*Input:* User Click/Typing

*Output:* Redirecting the user to a new page or sending something to firebase

*Dependent Components:* Matching Page

*Dependent On:* Client, Server, Firebase, Navbar

### **Personality Questionnaire:**

*Function:* The function of the personality questionnaire is to get information about the user and their preferences so it can be used in matching them to other people.

*Interaction(s):*

Firebase: Stores the answers to its questions in firebase.

Matching Page: (Indirectly) Provides the data with which the matching page and algorithm depend on.

Coding Questionnaire: Once the personality questionnaire is completed, it redirects to the coding questionnaire.

*Input:* User Click/Typing

*Output:* Redirecting the user to the coding questionnaire and storing their choices in the database

*Dependent Components:* Matching Page

*Dependent On:* Client, Server, Firebase, Navbar

### **Coding Questionnaire:**

*Function:* The function of the coding questionnaire is to get information about the user and their coding preferences so it can be used in matching them to other people.

*Interaction(s):*

Firebase: Stores the answers to its questions in firebase.

Matching Page: (Indirectly) Provides the data with which the matching page and algorithm depend on.

*Input:* User Click/Typing

*Output:* Redirecting the user to the coding questionnaire and storing their choices in the database

*Dependent Components:* Matching Page

*Dependent On:* Client, Server, Firebase, Navbar, Personality Questionnaire

### **Profile Page:**

*Function:* The function of the profile page is to display the profile information to the user, and allow them to edit information or change preferences about their profile.

*Interaction(s):*

Firebase: Fetches the required user information from firebase, and updates firebase database according to edited information.

Questionnaires: Allows users to edit their answers on the questionnaires.

*Input:* User Click/Typing

*Output:* Redirecting the user to a new page or sending something to firebase

*Dependent Components:* Coding Questionnaire, Personality Questionnaire.

*Dependent On:* Client, Server, Firebase

### **Matching:**

*Function:* The function of matching is to pair together users based on the answers to their personality and coding questionnaires and their preferences.

*Interaction(s):*

Firebase: Fetches information needed from Firebase and stores the current matches and previously matched users to Firebase.

Questionnaires: (Indirectly) The questionnaires provide the data needed to match users together.

*Input*: User Click/Typing

*Output*: Redirecting the user to the potential matches page along with the list of matched users.

*Dependent Components*: Potential Matches Page

*Dependent On*: Client, Server, Firebase, Navbar, Personality Questionnaire

## Form of Incremental Testing

We used bottom-up testing to test our project. Each module was first tested individually with various inputs and test cases. For example, input boxes would be checked locally on the client during development, and we would use drivers to test the algorithms. After the individual module testing was done, we tested the system as a whole.

## 2. Incremental and Regression Testing

Module	Component A: Creating an Account
--------	----------------------------------

### Incremental Testing

Defect #	Description	Severity	How to Correct
1	If the inputted username is not alphanumeric, is less than 6 characters, or greater than 20 characters, an error should display.	2	Create the proper regex for the username and a function that checks whether or not it's valid.
2	If the inputted password is below 8 characters or above 256 characters, an error should display.	2	Create the proper regex for the password and a function that checks whether or not it's valid.
3	The email needs to display an error if it's not of the form: xxx@emailprovider.com	2	Create the regex to determine whether or not the email is valid
4	If an input box is cleared/empty,	2	Add the onBlur html react tag

	the error should no longer show until it is typed into.		and a change event to check whether or not the input box is empty.
5	If the “Sign Up” button is clicked, and at least one input field is not filled in, an error will show for the fields that aren’t filled in.	2	Add a function to check for empty input

#### Regression Testing

Defect #	Description	Severity	How to Correct
1	Ensure that disabling the sign up button until the information is valid doesn’t break the button	1	Add slightly redundant event listeners to each input field to ensure that they’re how they should be.

<b>Module</b>	<b>Component B: Login</b>
---------------	---------------------------

#### Incremental Testing

Defect #	Description	Severity	How to Correct
1	If login request fails, it doesn’t retry	1	Add a .error() to the request and make it retry/show an error
2	There is no client response for incorrect/missing login credentials	3	Added text to appear next to login prompt whenever there is a login failure

#### Regression Testing

Defect #	Description	Severity	How to Correct
1	Retrying a login may duplicate something in the database depending on the functioned invoked and the data in question	1	Make sure that the login request failed and error check the data retrieved from the database

<b>Module</b>	<b>Component C: Potential Matches / Matching Algorithm</b>
---------------	--

#### Incremental Testing

<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How to Correct</b>
1	The 'get more matches' button is shown even though there are potential matches on the page	2	Checked the number of matches on the page, if there is less than 1, show the 'get more matches' button
2	Users should not be matched unless both users have the correct sexual preference.	2	Check the gender and interested in fields for both users and make sure they match up.
3	Users should not be matched if the personality and coding questionnaire have not been filled out.	2	When computing score check that each of the answers is not blank.
4	Current matches should be stored for both users.	2	Made sure to add current matches to both users for each other by pushing to the database.

#### Regression Testing

<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How to Correct</b>
1	Fixing current matches being stored caused correct matches to be left out.	2	Accessing quiz data needed to be split into the quiz data and the userid which meant that the original way matching was done needed to also be split.

<b>Module</b>	<b>Component D: Personality Questionnaire</b>
---------------	---

#### Incremental Testing

Defect #	Description	Severity	How to Correct
1	Once the personality questionnaire is successfully submitted the coding questionnaire should be brought up after the personality questionnaire information is submitted into the database.	2	Add a next button to link to the Coding questionnaire. Since adding a link to the skipped other adding information to the database.
2	Ensure data is added correctly to the database when connect with account system.	1	Store uid as an entry in the collection to identify user's personality questionnaires.
3	Ensure user submits information before clicking the next button and continuing to the coding questionnaire.	3	Add function check Submit to occur before page is linked to the Coding questionnaire.
4	Two Alert windows appear after a questionnaire is submitted.	3	Moved the redirect button to outside the form area.

#### Regression Testing

Defect #	Description	Severity	How to Correct
1	Fixing Code to add information to the database with one collection for both personality and coding questionnaire.	1	Adjusted how the data was stored in database by setting each collection with an user id.

<b>Module</b>	<b>Component E: Coding Questionnaire</b>
---------------	--

#### Incremental Testing

Defect #	Description	Severity	How to Correct
1	Ensure user submits coding questionnaire into the shared query for both questionnaires.	1	Change the database to update the collection referenced by user id.
2	Ensure data is added correctly to	1	Store uid as an entry in the

	the database when connect with account system.		collection to identify user's personality questionnaires.
3	Ensure the user can't click done until the coding questionnaire is complete.	1	Add if statement to enable the button after check is made the questionnaire is complete.
4	Two alert windows appear after a questionnaire is submitted.	3	Moved the redirect button to outside the form area.

### Regression Testing

Defect #	Description	Severity	How to Correct
1	Fixing Code to add information to the database with one collection for both personality and coding questionnaire.	1	Adjusted how the data was stored in database by setting each collection with an user id.

<b>Module</b>	<b>Component F: Profile Page</b>
---------------	----------------------------------

### Incremental Testing

Defect #	Description	Severity	How to Correct
1	When the user clicks on the 'edit' button for either questionnaire on the profile page, the questionnaires should appear with the fields pre-filled with the data that the user entered during registration	1	Added code to retrieve user data when user clicks on 'profile'. Also added code to send the state from profile page as props for both questionnaires.
2	When the user navigates to either of the questionnaires from the profile page, the submit button should correctly update data on the database.	3	Added code to update specific fields on the database only if there is a change in any of the answers.
3	The app doesn't show an alert when the profile has been	1	Added javascript alert code to execute after the information



	successfully edited		has been updated on the database.
4	The app accepts alphabetic character strings as input to the age field	1	Added checks to make sure the information entered for each field is in the appropriate format.
5	After successful account creation, the app doesn't automatically redirect to the quizzes, mandating the use of profile page to fill the quizzes.	2	Added router-redirects to navigate directly to quizzes after a user successfully creates an account.
6	The app saves data to the database even if multiple fields are left blank while editing profile information	2	Added code to check if all fields are filled with the necessary requirements before updating the database.

#### Regression Testing

Defect #	Description	Severity	How to Correct
1	Fixing the alert display issue after editing profile caused the app to not redirect to the profile page	2	Understood and re-implemented router-redirection in react. The correct local address had to be pushed to redirect history.
2	Changing code to pre-fill the questionnaires while editing answers caused the app to break down when the questionnaires appeared after account creation.	2	Added code to conditionally use variables passed as props from the parent component.

<b>Module</b>	<b>Component G: Nav Bar</b>
---------------	-----------------------------

#### Incremental Testing

Defect #	Description	Severity	How to Correct
----------	-------------	----------	----------------

1	The navigation component does not show when a user is logged in	1	Added a check for the current user using firebase api, if there is a user logged in, show the navigation component
2	The logout button does not redirect the user to the correct page	2	Use window.location.href to correctly set the location the user should be redirected to after logging out

### Regression Testing

Defect #	Description	Severity	How to Correct
1	Some links were hardcoded, so when the site is hosted the user may be directed to an unreachable page	1	Create a check for what the current window location is, and redirect based on whether it's localhost or not.

## 3. Update Product Backlog

### Functional Requirements

Backlog Id	Functional Requirements	Hours	Status
1	As a user I would like to create a H&LO WORLD account.	4 hours	<Complete>
2	As a user I would like to login and logout of my H&LO WORLD account.	4 hours	<Complete>
3	As a user, I would like to fill out a personality questionnaire.	10 hours	Completed in Sprint 1
4	As a user, I would like to fill out a fun coding questionnaire.	8 hours	Completed in Sprint 1
5	As a user, I would like to be able to edit my questionnaire answers.	10 hours	Planned for Sprint 2
6	As a user, I would like to be able to upload pictures of myself and my interests to show on my profile.	8 hours	Planned for Sprint 2

7	As a user, I would like to be able to set/change my name, profile description, and preferences on my account.	6 hours	Completed in Sprint 1
8	As a user, I would like to be matched with people who have similar personalities to my own.	10 hours	Completed in Sprint 1
9	As a user, I would like to view my potential matches and our match compatibility score.	10 hours	Completed in Sprint 1
10	As a user, I would like to “like” other accounts to display my interest in getting to know them.	8 hours	Planned for Sprint 2
11	As a user, I would like to be able to chat with users I’ve been matched with.	12 hours	Planned for Sprint 2
12	As a user, I would like to view a list of others who have been matched with me.	8 hours	Planned for Sprint 2
13	As a user, I would like to receive notifications for when I receive a new chat message.	8 hours	Planned for Sprint 2
14	As a user, I would like to be able to “unlike” someone if they are not right for me.	6 hours	Planned for Sprint 2
15	As a user, I would like to be able to report other users.	4 hours	Planned for Sprint 2
16	As a developer, I would like to be able to verify if a user is logged in on all required pages.	8 hours	<Completed>
17	As a user, I would like to be able to easily navigate between web pages.	5 hours	<Completed>
18	As a user, I would like a secure login page (HTTPS).	8 hours	<Completed>
19	As a user, I would like a Matching webpage where I can view potential matches.	6 hours	<In Progress: Moved to Sprint 2>
20	As a user, I would like a Matches webpage where I can view people I have been matched with.	6 hours	Planned for Sprint 2
21	As a user, I would like to have homepage that explains what HE&gt;LO WORLD is.	5 hours	<Completed>
22	As a user, I would like to be able to see the online status of my matched users.	8 hours	Planned for Sprint 2

23	As a user, I would like to select a picture as my profile picture to display on the matching page. .	6 hours	Planned for Sprint 2
----	--	---------	----------------------