National College of Ireland

Software Development

Bachelor's (Honours) in Computing

Vitalie Brescanu

x19241011 (BSHCSD4)

x19241011@student.ncirl.ie

Integration Friend Ireland App

Technical Report

Contents

Executive Summary

I chose to do an Integration Friend Ireland app as integration is crucial to a functioning society, and as it’s a two-way give and take process, it’s also vital that we include established Irish - whether ethnic or born elsewhere. This app works like a dating/friendship app, but without the pressure to find a soulmate or a friend for life, this is about connecting and learning from each other, like a buddy system. One-way integration where newcomers try to make space, find out responsibilities and rights and learning the language and the culture is not enough. Established Irish - ethnic, naturalised and EU citizens, must also want to learn about the newcomers. A social life is as important as food, shelter and warmth. The contact between new and established Irish is the most important aspect of integration and this app is one step closer to making the connection easier. Humans can be altruistic and selfish, but if the opportunity to help is offered, most would find the time to do something. A majority of established Irish are already familiar with online dating, friendship apps and senior companionships app. Ireland needs an integration friend app too. There are a lot of organisations helping newcomers in Ireland, from civil society to businesses and the public sector, but many individual established Irish want to be in charge when deciding to do a good deed without the hassle of too many steps before connected to a newcomer/established Irish. With this app, people can decide for themselves and start chatting within seconds of being matched.

1.0 Introduction

1.1. Background

Ireland is no stranger to welcoming newcomers and taking in refugees, but have forgotten that integration must include the individual. If established Irish who are worried about how society and their part in it will change with the influx of new Irish could connect with a newcomer and learn about a new culture and also teach about their own, they would discover that people are basically the same, only our stories differ. If new and established Irish got to know each other, we could continue to be welcoming.

1.2. Aims

1. Create an app that connects people without the pressure of it having to be anything else than sharing stories and helping a newcomer settle in
2. Help the integration process and eliminate racism. Our difference is mainly cultural and to live in peace we must listen to each other stories
3. Make the app visible and enticing
4. Ultimately, wake up people to the importance of integration as a two-way process

1.3. Technology

- **React/React Native:** It is a framework for developing mobile apps for android and iOS by using one common language JavaScript, with react native the app will have a smooth, responsive user interface. React native will be used for UI (user interface) [1] [2]
- **JavaScript:** For adding distinct variables to the element content, for introducing some functionalities and manipulations in the pages [3]
- **CSS(Tailwind):** Tailwind CSS is an open source framework. Will be used for staling of the content [4]
- **Firebase:** Is a hosting sever supported be the Google. Firebase will be used for user authentication, registration/login, password reset and email verification. User information will be kept and updated in Database storage, deployed and use of Cloud functions. [5] [6]
- **VSCode:** editor (Visual studio code, a source-code editor created by Microsoft) for writing the code [7]
- **Android studio software:** It is a framework for designing application. It will be used to set the necessary tools for building the React Native application for android. [8]
- **Hosting application:** When the application is accomplished can be hosted on the Google store or App Store

1.4. Structure

The structure of this mobile application is defined in three tiers:

Requirements: This section is showing in detail the two requirements functional and non-functional of the systems including the use of case diagram too.

Design and Architecture: This section contains system flow information and various system architecture diagrams.

GUI: Represents the design of how the application was built.

2.0 System

2.1. Requirements

Requirements needed for the application are user's data. This data has a title, description, and a use case diagram that will present different possible actions a user can take at different stages while navigating through the mobile application.

2.1.1. Functional Requirements

The system functional requirements that will be needed to complete the project are:

1) The system collects the data and save it to the database
2) The system checks the data and authorises the data submitted

3) The user can login with the data provided
4) The user can view/update and delete the profile
5) The user can check deferent user/s profile
6) The user can be matched with a different user
7) The user can chat with the other matched/connected user/s
8) The user can check on the map the location of other users

2.1.1.1. Use Case Diagram

My Use Case Diagram outlines 8 functional requirements that the application needs to be able to do in order to work.

2.1.1.1.1 User using the App – Use Case Diagram

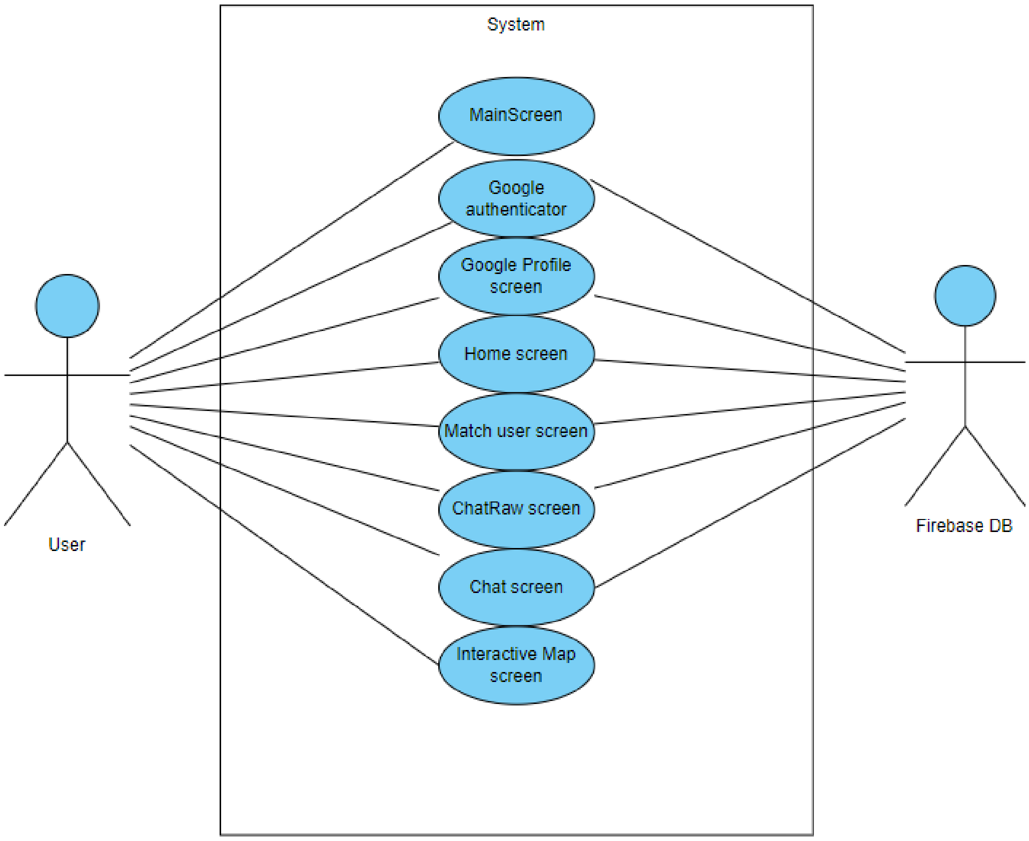


Figure 1 User using the App Use Case Diagram

2.1.1.2. Requirement 1: Main page

This requirement allows the user to see the main page and the information related to the application

2.1.1.3. Description & Priority

The user can read the information on the main page, about the Integration Friend Ireland application

2.1.1.4. Use Case

2.1.1.4.1. User using the App – Use Case Diagram

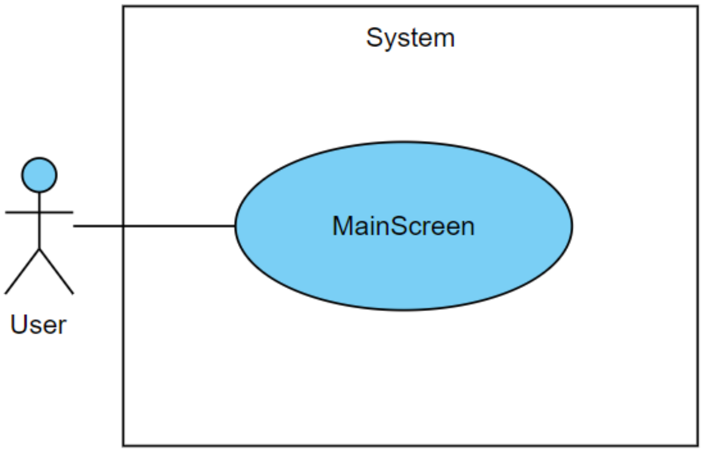


Figure 2 User using the App – Use Case Diagram | MainScreen

Scope

Displaying some Information to the user and how is the buddy matches work

Description

When the application is run the first time the main page will display some Information to the user and also how is the buddy matches work

Main flow

1) The <User> clicks on the login Icon (button)
2) The <User> to register or login

2.1.1.5. Requirement 2: User registration (Google account)

This requirement allows the user to create an account, and access the applications features. The user can not have access to any service if they do not have a registered account.

2.1.1.6. Description & Priority

Establishing the correct data and storing it in the database.

2.1.1.7. Use Case

Scope

The scope of this use case is how a user registers an account.

Description

This use case defines how a user will record a new account, including the data they must provide to the system.

2.1.1.7.1. User using the App – Use Case Diagram

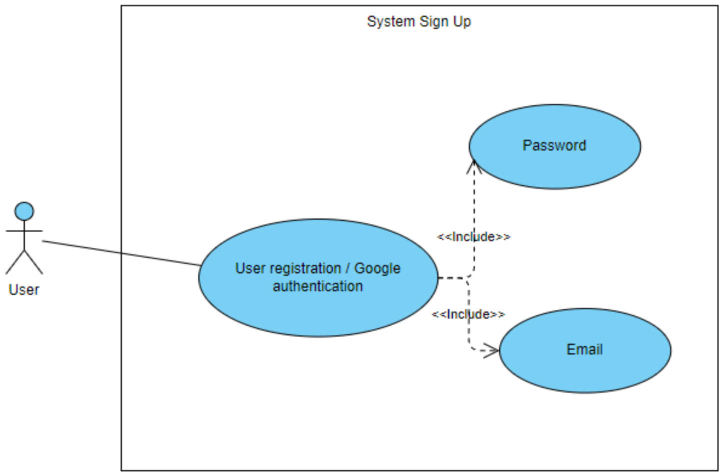


Figure 3 User using the App – Use Case Diagram | User registration

Flow Description

Precondition

There is no data saved in the database.

Activation

Activation will start when the User will register their new account.

Main flow

3) The <User> is typing relevant data
4) The <User> submits registration
5) Firebase query pushing data to database
6) The data is saved to the database

Post condition

A new user registration row is created in the database.

2.1.1.8. Requirement 3: User Login

Use case for user login

2.1.1.9. Description & Priority

The user has to login into the account that they created. The system will only allow accessed with a successful login into a verified account

2.1.1.10. Use Case

Scope

The scope of this use case is to examine the steps needed to log in to the system.

Description

This use case defines the conditions to successfully log in to the system to access the features and account data for a user.

2.1.1.10.1. User using the App – Use Case Diagram

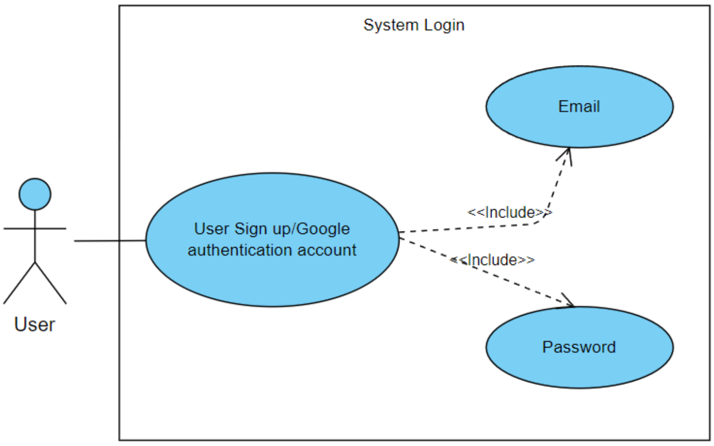


Figure 4 User using the App – Use Case Diagram | User Login

Flow Description

Precondition

To use the login in the system section, the user must have already created an account.

Activation

This use case begins when the user decides to log in to the system by specifying the login option on the start screen.

Main flow

1. The system will ask the user if they want to log in
2. The user selects the login option
3. The system will display the login screen page
4. The user selects their valid information
5. The system logs the user into the system

Alternate flow

1. The system matches the credentials supplied by the user
2. The system believes the credentials are not authentic
3. The error notification is displayed to the user
4. The user is prompted to re-enter the credentials again

Termination

The system displays the main functionality of the application once a successful login attempt happens.

Post condition

The user only needs to login once, the system will remember the user.

When the user logs out of the account, this use case will run again.

2.1.1.11 Requirement 4: Home screen page

Use case for the Home screen page.

2.1.1.12. Description & Priority

The home screen page displays a deck of cards with the user’s information. The user has the ability to choose to swipe the card left if the user doesn't want to be a friend or to swipe right to become a friend

Also, on login, if the user details are not in the database the user profile form will be displayed and the user details needs to be filled in

2.1.1.13. Use Case

Scope

The scope of this use case is to determine if there is a connection between the new Irish and established Irish.

Description

This use case describes the process of the matching algorithm for connecting the user on his selection

2.1.1.13.1. User using the App – Use Case Diagram

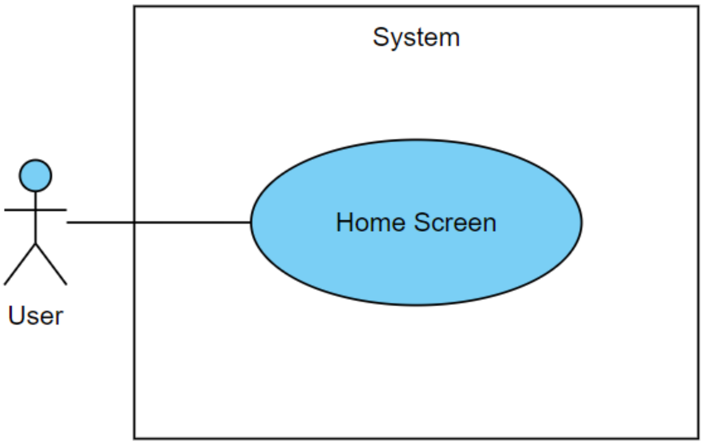


Figure 5 User using the App – Use Case Diagram | Home Screen

Flow Description

Precondition

The system is in a loaded mode

Activation

This use case starts when an <User> selecting the card by swiping the card left or right

Main flow

1. The system displays all the registered users
2. The <User> swipes the card left or right

Alternate flow

A1: <User can check the chat screen>

1. The <User> the user can verify the chat row screen
2. The <User> can check the navigation menu screen be clicking on the user image icon
3. The application is waiting for user action.

Termination

The application stays on the same screen page

Post condition

The application is waiting for the user actions.

2.1.1.14. Requirement 5: Match screen page

Use case for the Match user screen page.

2.1.1.15. Description & Priority

When the matching algorithm will get a match, the Match screen page will appear and display the two users as connected

2.1.1.16. Use Case

Scope

The scope of this use case is to show the connection of two users

Description

This use case shows the matching process connection of the user when his selects another user

2.1.1.16.1. User using the App – Use Case Diagram

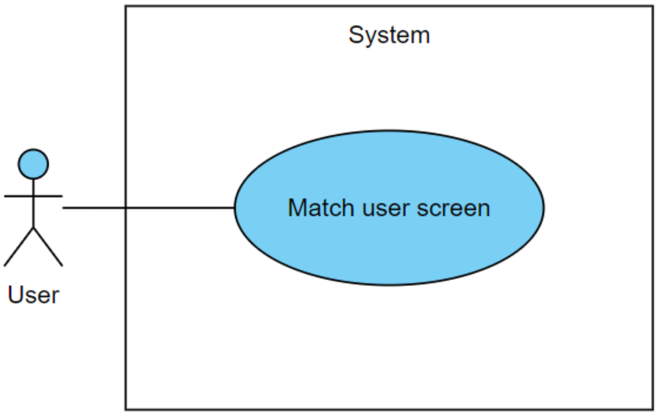


Figure 6 User using the App – Use Case Diagram |Match user screen

Flow Description

Precondition

The system is in a loaded mode

Activation

This use case starts when an <User> selecting the card by swiping, the card left or right

Main flow

1. The system displays the match user screen
2. The <User> can click on the sent a message button

Termination

The application is redirected to the chat row screen page, where all the connected users are displayed

Post condition

The system enters a waiting state

2.1.1.17. Requirement 6: ChatRow screen page

Use case for the ChatRow screen page.

2.1.1.18. Description & Priority

This use case is to view the user list connections, click on the selected user to connection and start chatting

2.1.1.19. Use Case

Scope

The scope of this use case is to allow the user to see his connection for chatting

Description

This use case shows the matching process connection of the user when his selects another user

2.1.1.19.1. User using the App – Use Case Diagram

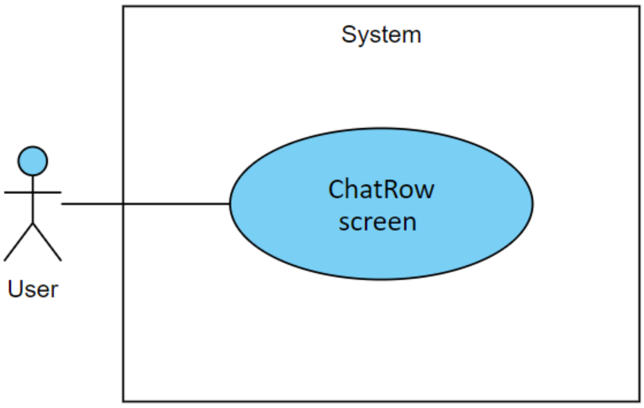


Figure 7 User using the App – Use Case Diagram |ChatRow screen

Flow Description

Precondition

The system is in a loaded mode

Activation

This use case starts when an <User> selecting the card by swiping the card left or right

Main flow

1. The system displays the match user screen
2. The <User> can click on any connection from the list

Termination

The application is redirected to the chat screen page, where both users can chat with each other

Post condition

The system enters a waiting mode

2.1.1.20. Requirement 7: Chat screen page

Use case for the Chat screen page.

2.1.1.21. Description & Priority

This use case will allow the users to chat with each other

2.1.1.22. Use Case

Scope

The scope of this use case is to allow users to exchange messages in real-time

Description

This use case is used for communication and conversation purposes, enabling users to send and receive text messages. Users can engage in one-on-one conversations chats, make plans, and stay connected in real-time.

2.1.1.22.1. User using the App – Use Case Diagram

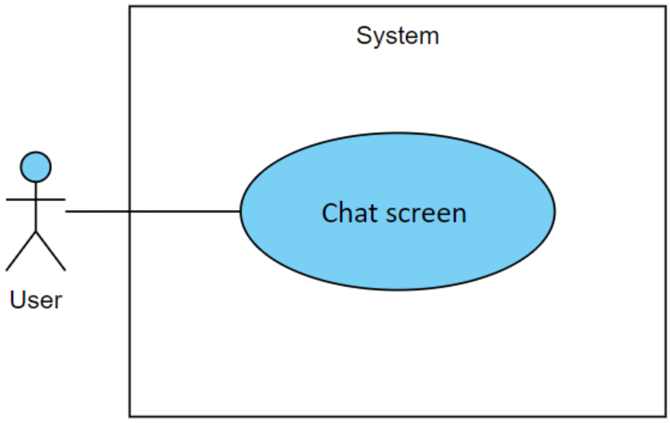


Figure 8 User using the App – Use Case Diagram | Chat screen

Flow Description

Precondition

The system is in a loaded mode

Activation

This use case starts when an <User> starts to send messages

Main flow

1. The system displays the Chat screen page

2. The <User> clicks on the send button

Termination

The application in not redirected anywhere until the user click the top left arrow icon button to go to the previous page

Post condition

The system enters a waiting state

2.1.1.23. Requirement 8: Interactive Map screen page

Use case for the Interactive Map screen page.

2.1.1.24. Description & Priority

This use case: the purpose of an interaction map is to provide a visual representation of how different objects or components within a system communicate or interact with each other

2.1.1.25. Use Case

Scope

The scope of this use case is to create an Interactive map, display a Google map, in this case, Ireland, and display all the users connected to this application in real-time

Description

When a user opens the application, they can navigate to the interactive map feature. The map will display the current location of the users, and they can zoom in or out to see different areas. The map will display markers or icons on the locations of other users.

2.1.1.25.1. User using the App – Use Case Diagram

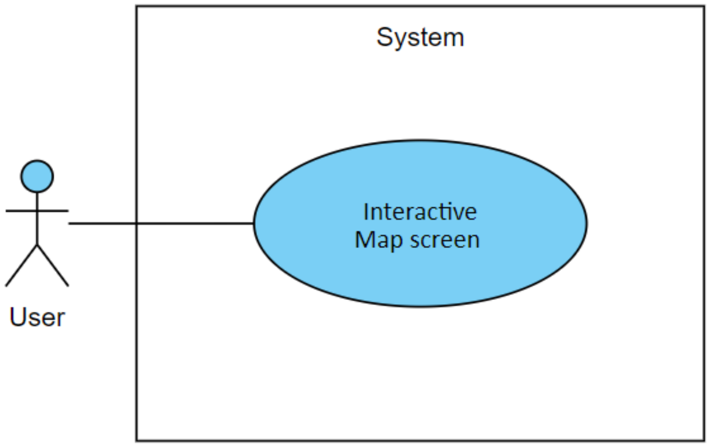


Figure 9 User using the App – Use Case Diagram | Interactive Map screen

Flow Description

Precondition

The system is in a loaded mode

Activation

This use case starts when an <User> click on the Interaction Map button in the nav menu screen which can be found on the top left bar by clicking on the user image/icon

Main flow

1. The system displays Interaction Map
2. The User location can be seen on the map

Termination

The application in not redirected anywhere until the user click the top arrow left icon button

Post condition

The system enters a waiting state

2.1.2. Data Requirements

I am using the Firebase database for storing the data. For the user account the database will store all the registered user's information like email (google account), language, gender, age, photo, if it's a new Irish or established Irish, and for the conversation the database will store message conversation. Firebase uses a real-time database to store and sync data in JSON (JavaScript Object Notation) format.

2.1.3. User Requirements

Requirement 1: The user should be able to create an account and log in through google account authentication portal

Requirement 2: The user should be able to update their information in a limited format

Requirement 3: The user should see other user's information related to this application

Requirement 4: The user should be able to go through a list of users, and swipe the card left or right depending on the user selection

Requirement 5: The user should be able to check the chat messages from within the connected user list.

2.1.4. Environmental Requirements

The application is an android mobile application and must be downloaded and installed on the device from the google store, user must be connected to the internet to access this application.

2.1.5. Usability Requirements

This application is only for Android mobile operating systems. The Integration Friend Ireland application has a pleasant and easy user interface, easy to navigate through the application screens, the information is clearly displayed

2.2. Design & Architecture

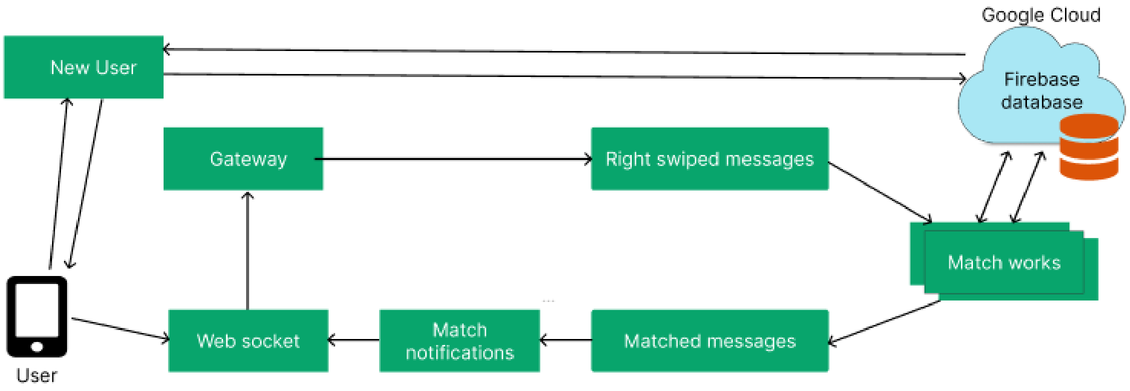


Figure 10 Design ana Architecture diagram, user matchmaking service

The Integration Friend Ireland application is built on a React Native framework.
The architecture diagram above represents the Integration Friend Ireland application

Connecting with a user:

When a new user signs in to the Integration Friend Ireland application using Google oAuth, their profile details are sent to the service using HTTP/WebSocket.

Let's imagine there are two user profiles, User-A and User-B, on a connecting platform

There are three potential scenarios:

1. User-A and User-B both swipe right on each other
2. User-A swipes right on User-B, but User-B does not reciprocate
3. User-B swipes right on User-A, but User-A has not yet swiped right in return

In the image above, when a user performs a right swipe on a matchmaking service, a message is sent through a web socket.

Database

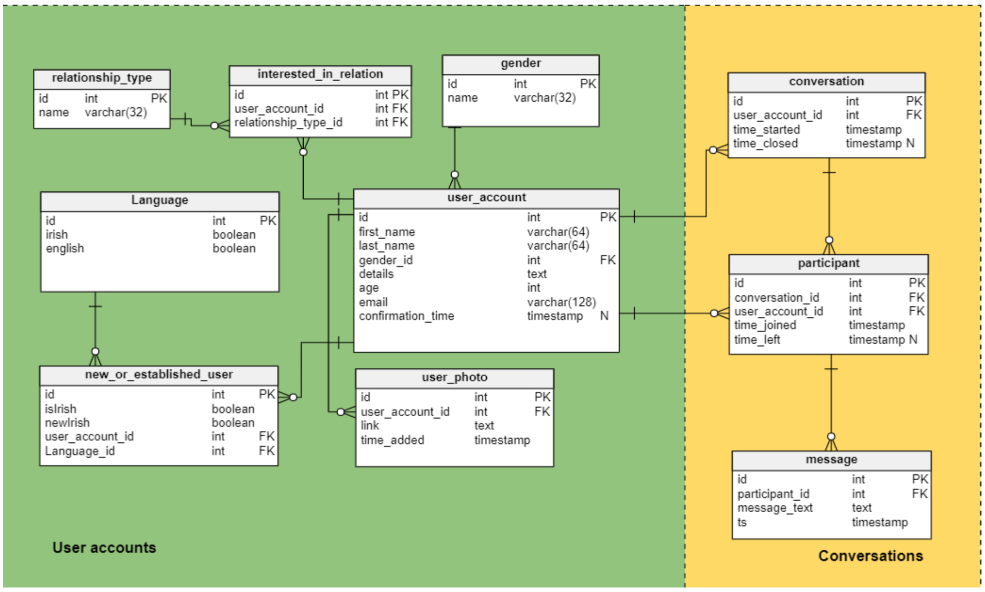


Figure 11 Database diagram | Integration Friend Ireland App

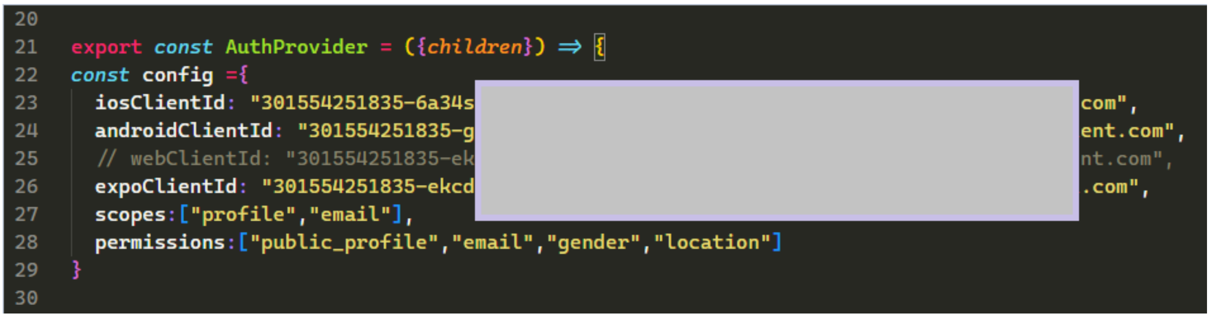
2.3. Implementation

My app is designed to connect users with each other based on swiping left or right. The app's main algorithms revolve around user registration and login, as well as matching and connecting users, and enabling them to chat with each other.

2.3.1. User registration and Login

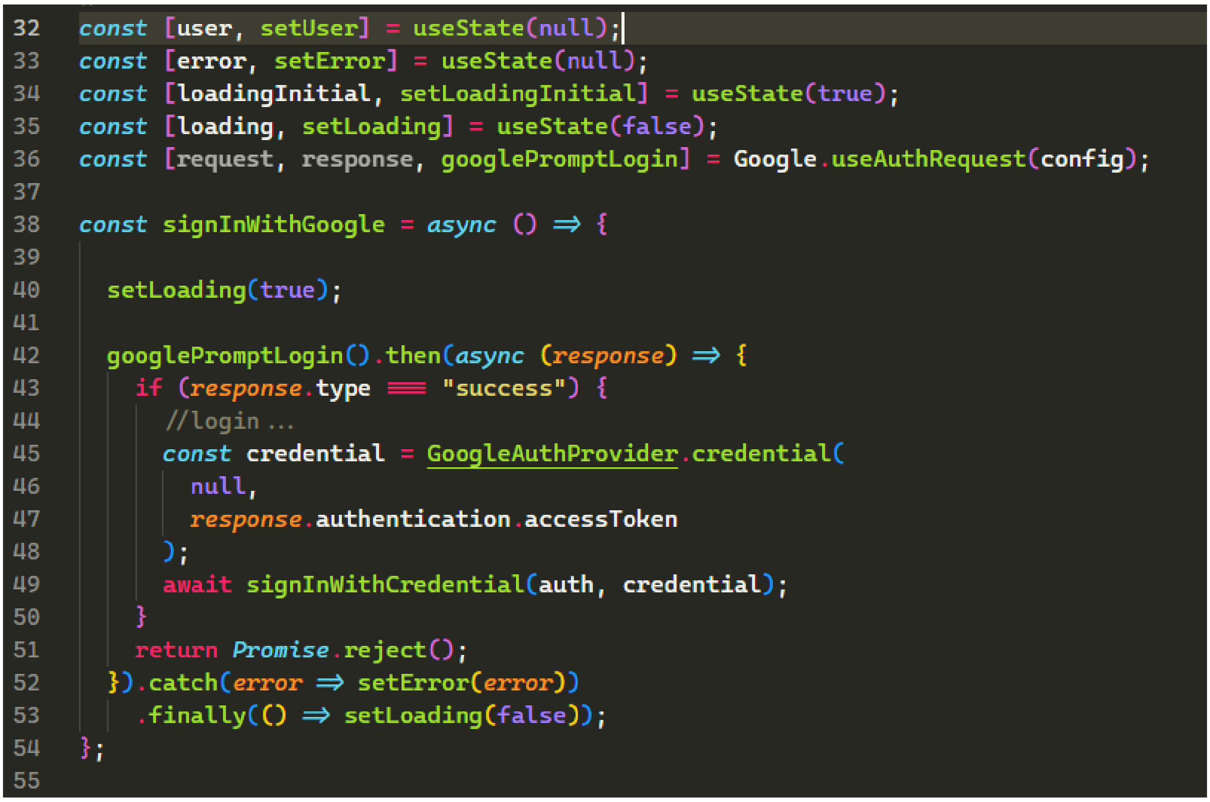
One of the key features of Integration Friend Ireland application is that users can register and log in using their Google accounts, which ensures a quick and secure authentication process. Once they've logged in the first time, users will get a pop-up profile form and asked to provide some basic information, such as, if is a local or established user, gender, language, photo, age, and something about the user. This application is connected to the Cloud Firestore database, and is useful for storing and retrieving user data in the cloud.

hooks/**useAuth.js** file



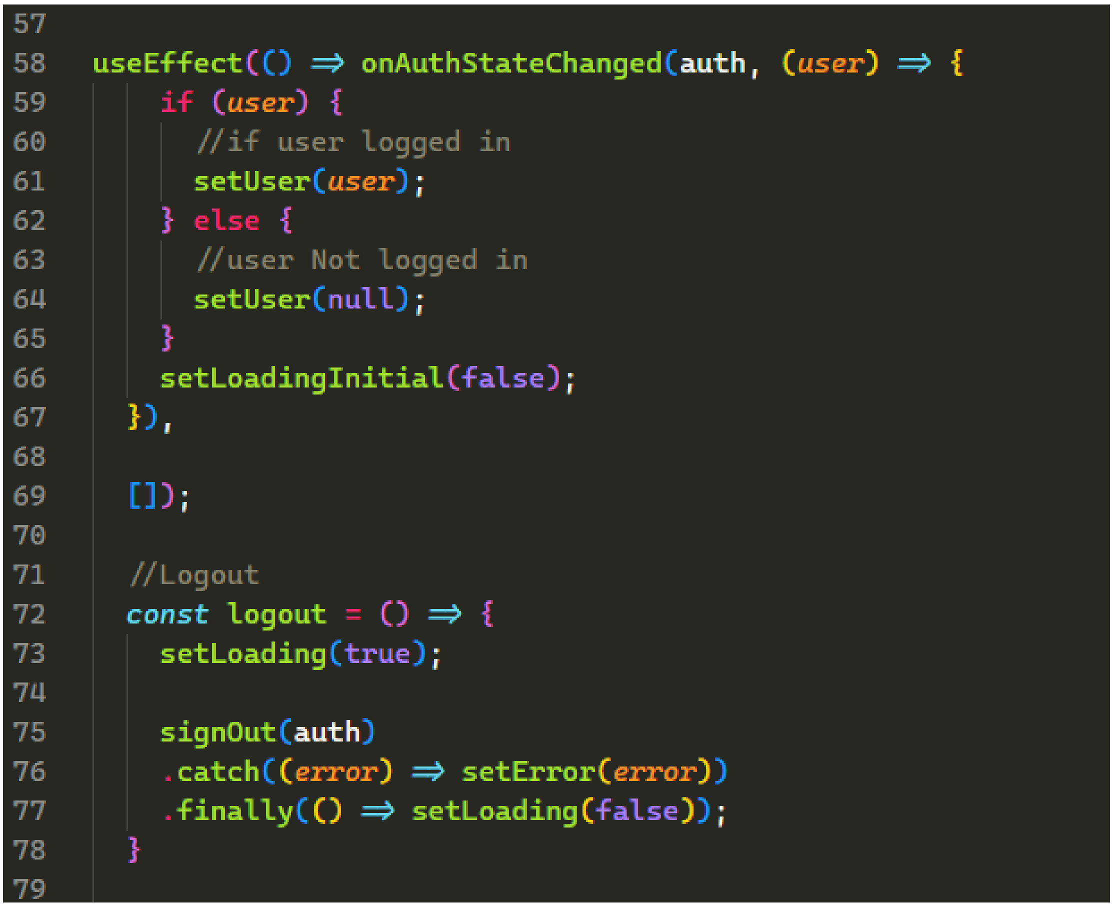
```
export const AuthProvider = ({children}) => {
const config ={
  iosClientId: "301554251835-6a34s[redacted]com",
  androidClientId: "301554251835-gent.com",
  // webClientId: "301554251835-eknt.com",
  expoClientId: "301554251835-ekcd.com",
  scopes:["profile","email"],
  permissions:["public_profile","email","gender","location"]
}
```

Figure 12 Code for registration and Login



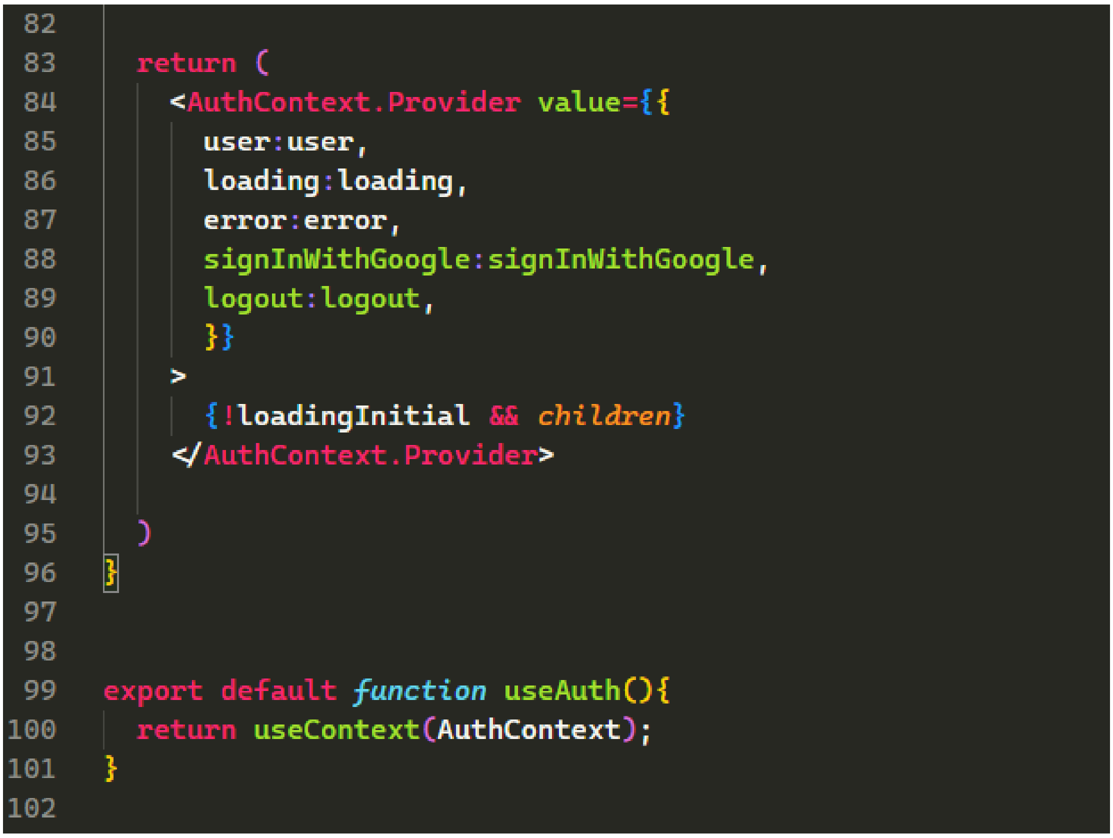
```
const [user, setUser] = useState(null);
const [error, setError] = useState(null);
const [loadingInitial, setLoadingInitial] = useState(true);
const [loading, setLoading] = useState(false);
const [request, response, googlePromptLogin] = Google.useAuthRequest(config);

const signInWithGoogle = async () => {

  setLoading(true);

  googlePromptLogin().then(async (response) => {
    if (response.type === "success") {
      //login ...
      const credential = GoogleAuthProvider.credential(
        null,
        response.authentication.accessToken
      );
      await signInWithCredential(auth, credential);
    }
    return Promise.reject();
  }).catch(error => setError(error))
    .finally(() => setLoading(false));
};
```

Figure 13 continuation of image 12 |registration and Login



```
useEffect(() => onAuthStateChanged(auth, (user) => {
    if (user) {
      //if user logged in
      setUser(user);
    } else {
      //user Not logged in
      setUser(null);
    }
    setLoadingInitial(false);
  }),

  []);

  //Logout
  const logout = () => {
    setLoading(true);

    signOut(auth)
    .catch((error) => setError(error))
    .finally(() => setLoading(false));
  }
```

Figure 14 continuation of image 13 | registration and Login



```

  return (
    <AuthContext.Provider value={{
      user:user,
      loading:loading,
      error:error,
      signInWithGoogle:signInWithGoogle,
      logout:logout,
      }}
    >
      {!loadingInitial && children}
    </AuthContext.Provider>

  )
}


export default function useAuth(){
  return useContext(AuthContext);
}

```

Figure 15 continuation of image 14 | registration and Login

The code in the above imagies is user registration/Login that uses Google authentication. The **AuthProvider** component sets up authentication with Google by defining config settings with Firebase for the **iosClientId**, **androidClientId**, **expoClientId**, scopes, and permissions.

The state variables are defined using the **useState** hook for user, error, **loadingInitial**, and loading. The **useEffect** hook is used to listen for changes to the authentication state with **onAuthStateChanged**, and sets the user state variable based on whether the user is logged in or not.

The code also defines a **signInWithGoogle** function that triggers the Google authentication flow using the **googlePromptLogin** method. If successful, it obtains a credential and logs the user in with **signInWithCredential**. A logout function is also defined to log the user out.

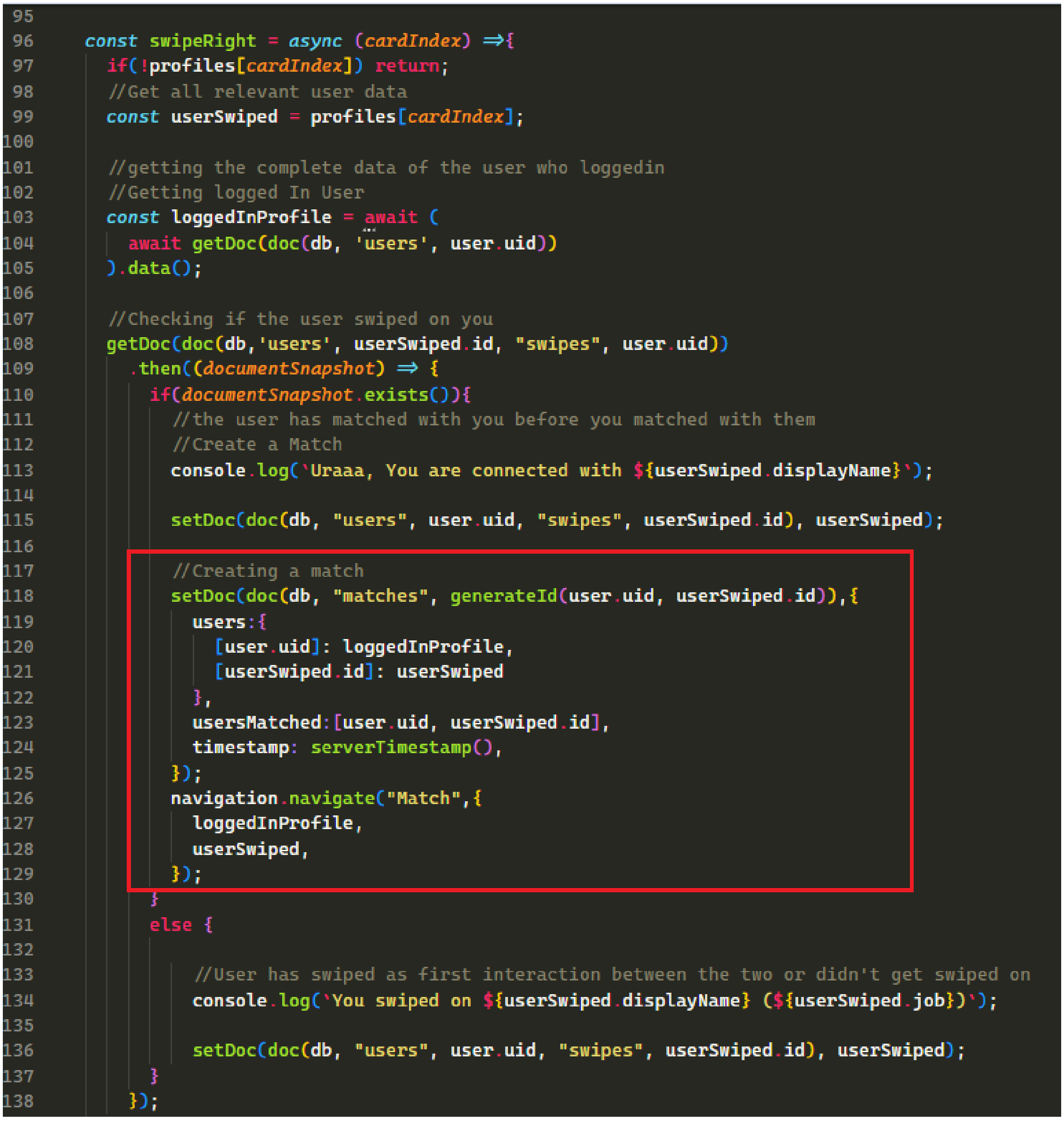
Finally, the **AuthProvider** component returns a Provider that wraps the children components with an **AuthContext.Provider** that provides authentication-related state and functions to its descendants. The **useAuth** hook is also defined for accessing the **AuthContext** values in other components.

2.3.2. Matching algorithm or connecting the users

The matching algorithm takes into account a wide range of factors to ensure that users are connected with the most compatible people possible. Once a match is made, users can initiate a chat session with each other to get to know each other better and start building a connection.

The **swipeRight** function is called when the user swipes right on a profile. It first checks if the profile exists and then retrieves the complete data of the user who is currently logged in. It then checks if the swiped user has already swiped on the logged-in user. If there is a match, it logs a message indicating the match and creates a new match document. Otherwise, it logs the swiped profile details and creates a swipe document for the swiped profile.

Screens/HomeScreen.js (swipeRight function)

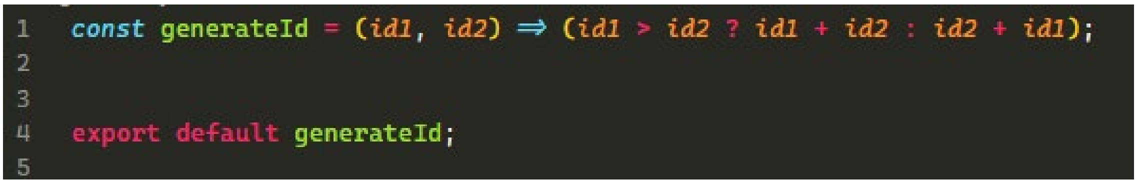


```

const swipeRight = async (cardIndex) =>{
  if(!profiles[cardIndex]) return;
  //Get all relevant user data
  const userSwiped = profiles[cardIndex];

  //getting the complete data of the user who loggedin
  //Getting logged In User
  const loggedInProfile = await (
    await getDoc(doc(db, 'users', user.uid))
  ).data();

  //Checking if the user swiped on you
  getDoc(doc(db,'users', userSwiped.id, "swipes", user.uid))
    .then((documentSnapshot) => {
      if(documentSnapshot.exists()){
        //the user has matched with you before you matched with them
        //Create a Match
        console.log(`Uraaa, You are connected with ${userSwiped.displayName}`);

        setDoc(doc(db, "users", user.uid, "swipes", userSwiped.id), userSwiped);

        //Creating a match
        setDoc(doc(db, "matches", generateId(user.uid, userSwiped.id)),{
          users:{
            [user.uid]: loggedInProfile,
            [userSwiped.id]: userSwiped
          },
          usersMatched:[user.uid, userSwiped.id],
          timestamp: serverTimestamp(),
        });
        navigation.navigate("Match",{
          loggedInProfile,
          userSwiped,
        });
      }
      else {

          //User has swiped as first interaction between the two or didn't get swiped on
          console.log(`You swiped on ${userSwiped.displayName} (${userSwiped.job})`);

          setDoc(doc(db, "users", user.uid, "swipes", userSwiped.id), userSwiped);
      }
    });
```

Figure 16 Matching algorithm code

This function in the image below will always guarantee that one user ID goes before the other user ID

Lib/generateId.js



```
const generateId = (id1, id2) => (id1 > id2 ? id1 + id2 : id2 + id1);


export default generateId;

```

Figure 17 this code is part of the code on the image 16

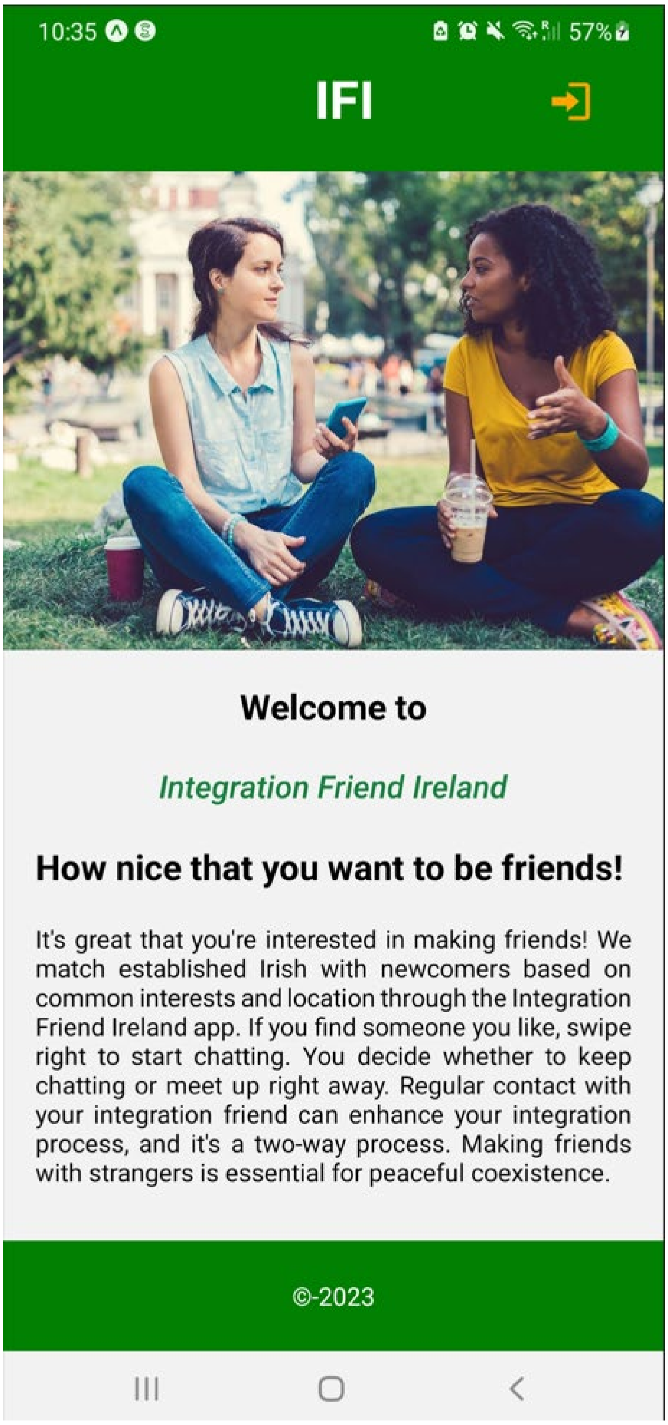
2.3.3. Chat room

The chat algorithm on this application is designed to be easy to use. The chat room will be an online platform where people can communicate with each other in real-time through text-based messages as well as create chat rooms with multiple users. The primary purpose of this chat room is to provide a space where the connected users can engage in discussions and share information with each other.

2.4. Graphical User Interface (GUI)

2.4.1 Welcome Screen and Login Screen

When the application starts the user can view the welcome page and the content, with some information about the application. On the top right corner, there is a login icon and by clicking on it, it will bring user to a login screen to register or sign in with Google account authentication



Welcome to

Integration Friend Ireland

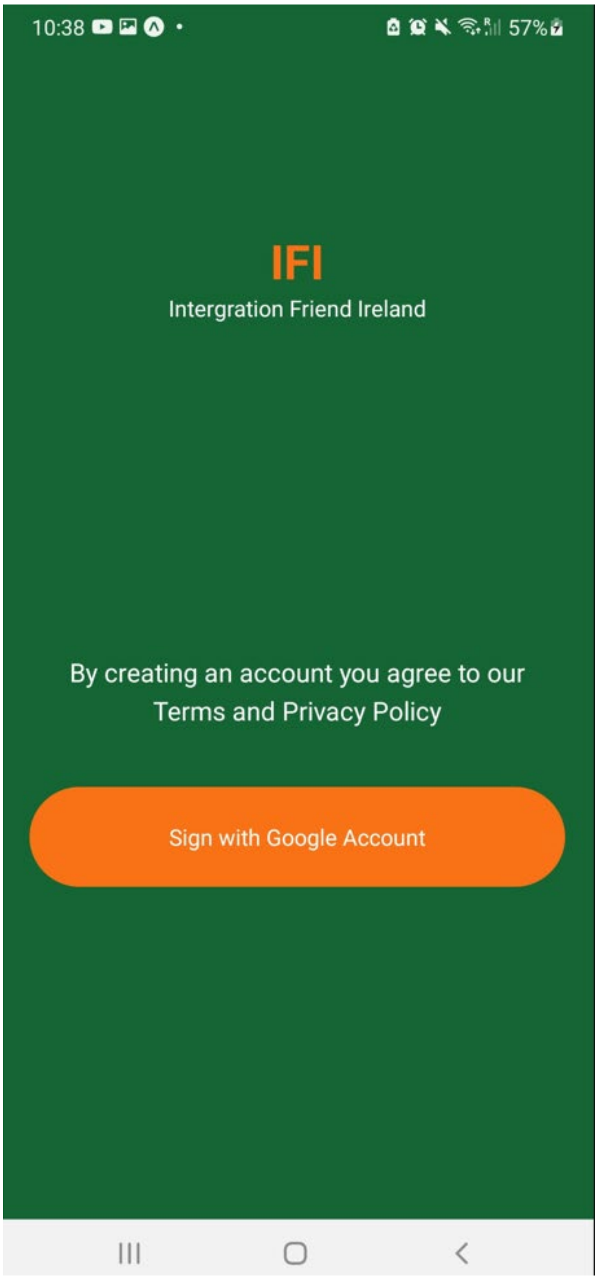
How nice that you want to be friends!

It's great that you're interested in making friends! We match established Irish with newcomers based on common interests and location through the Integration Friend Ireland app. If you find someone you like, swipe right to start chatting. You decide whether to keep chatting or meet up right away. Regular contact with your integration friend can enhance your integration process, and it's a two-way process. Making friends with strangers is essential for peaceful coexistence.

©-2023

Figure 18 Welcome Screen UI

Login Screen



10:38

57%

IFI

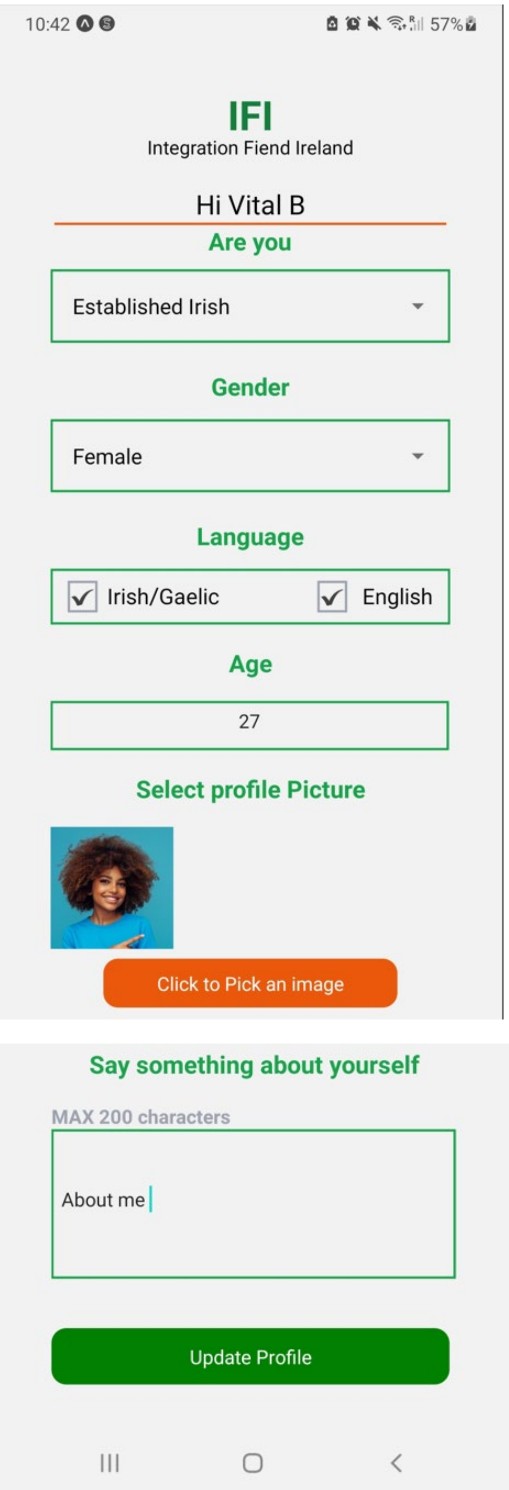
Intergration Friend Ireland

By creating an account you agree to our Terms and Privacy Policy

Figure 19 Login screen where the user can log in with a Google account to sign in to the application UI

2.4.2 User Profile form

The user profile form will appear when, the user is logged in the first time, and the user must provide some information in the form. Also, this is a from where the user can update the user profile information



10:42

57%

IFI

Integration Fiend Ireland

Hi Vital B

Are you

Established Irish

Gender

Female

Language

☑ Irish/Gaelic ☑ English

Age

27

Select profile Picture

Click to Pick an image

Say something about yourself

MAX 200 characters

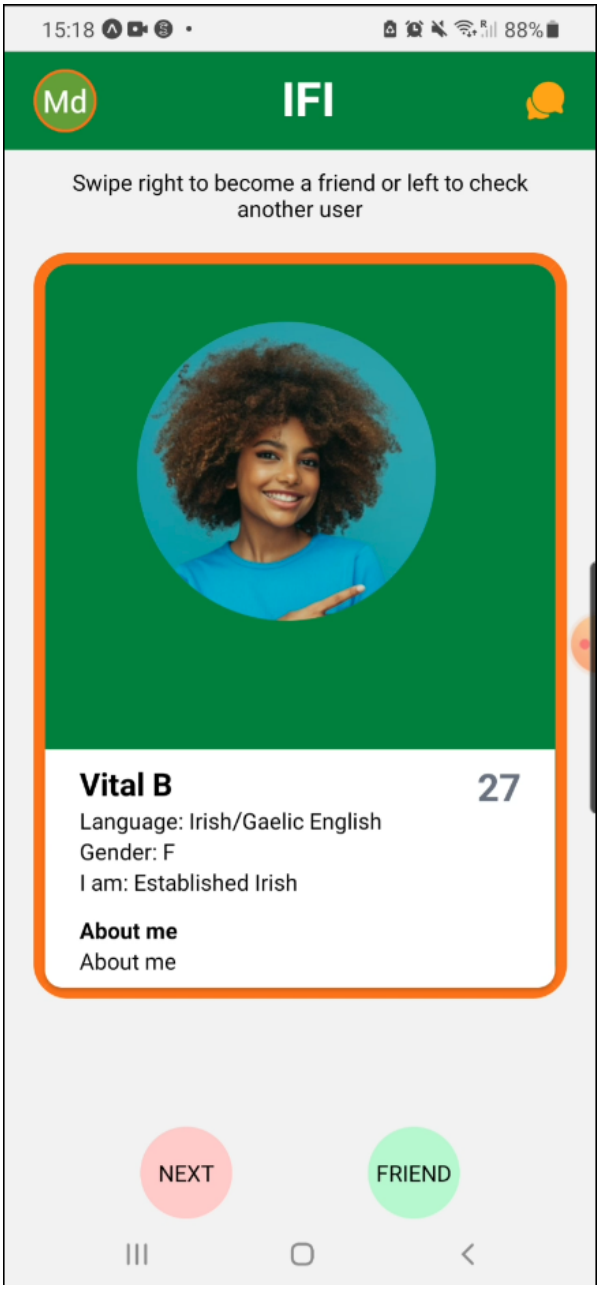
About me

Update Profile

Figure 20 User Profile form UI

2.4.3 Home screen page-Users card info.

The Home screen page will display a deck of cards representing the users and each card will show the user image and info. The card can be swiped left or right. Also, the user can click on the NEXT button to continue to the next user or click on the FRIEND button to become friends. User card Info



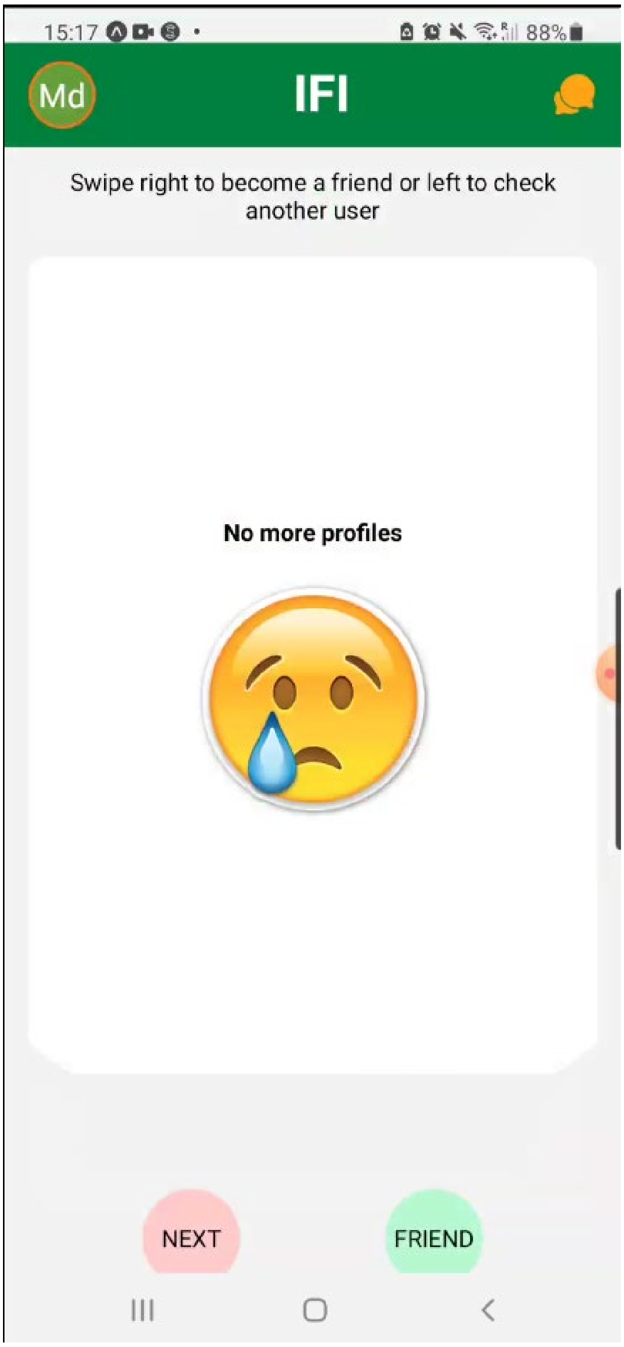
Swipe right to become a friend or left to check another user

Vital B 27
Language: Irish/Gaelic English
Gender: F
I am: Established Irish

About me
About me

Figure 21 Home screen page | User card with the user info UI

No more users screen



Swipe right to become a friend or left to check another user

No more profiles

2.4.4 User Connected/Matched screen

On this screen the user can view that they were connected with a matching user and they can click on the Send a message button to redirect to the ChatRow screen.

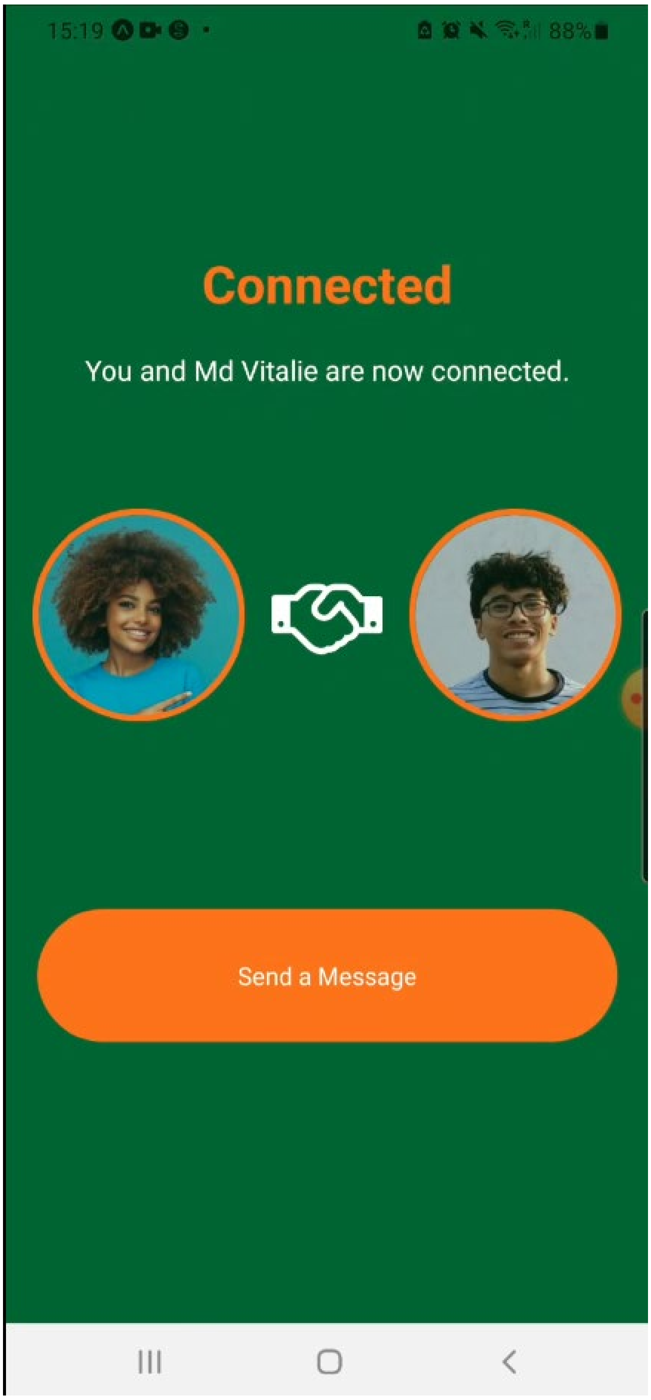
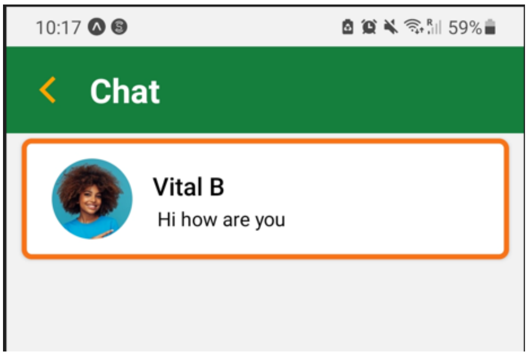


Figure 22 User Connected screen UI

2.4.5 ChatRow screen

The ChatRow screen is displaying a list of connected users, by clicking on one of the users it will bring up the Chat screen

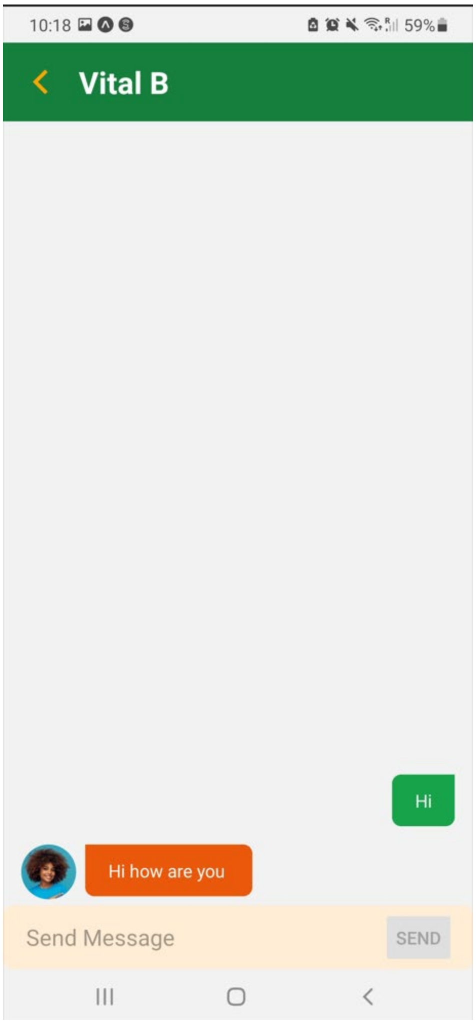


Vital B
Hi how are you

Figure 23 connected users list UI

2.4.6 Chat screen

On the chat screen, the application will allow users to engage in real-time text-based conversations with each other.



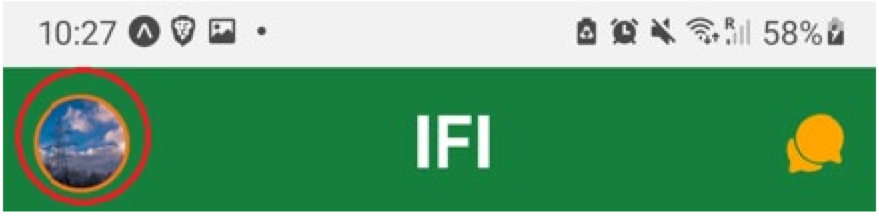
Hi

Hi how are you

Figure 24Users chat room UI

2.4.7 Navigation Screen

To get to the navigation screen, the user needs to click on the user icon image on the top left corner



The Navigation screen will display a list of selective button links with Edit Profile, Interaction Map, delete account and log out.

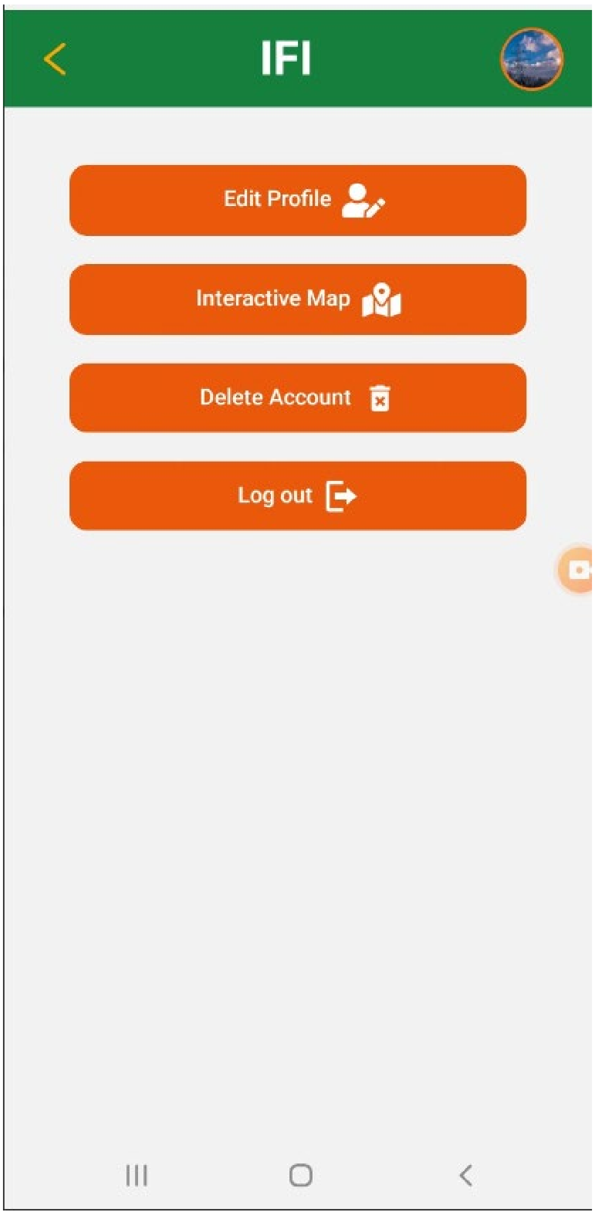


Figure 25The applications main menu UI

2.4.8 Users interactive map.

The interaction Map will show a google map with all the users related to this Integration Friend Ireland application on it

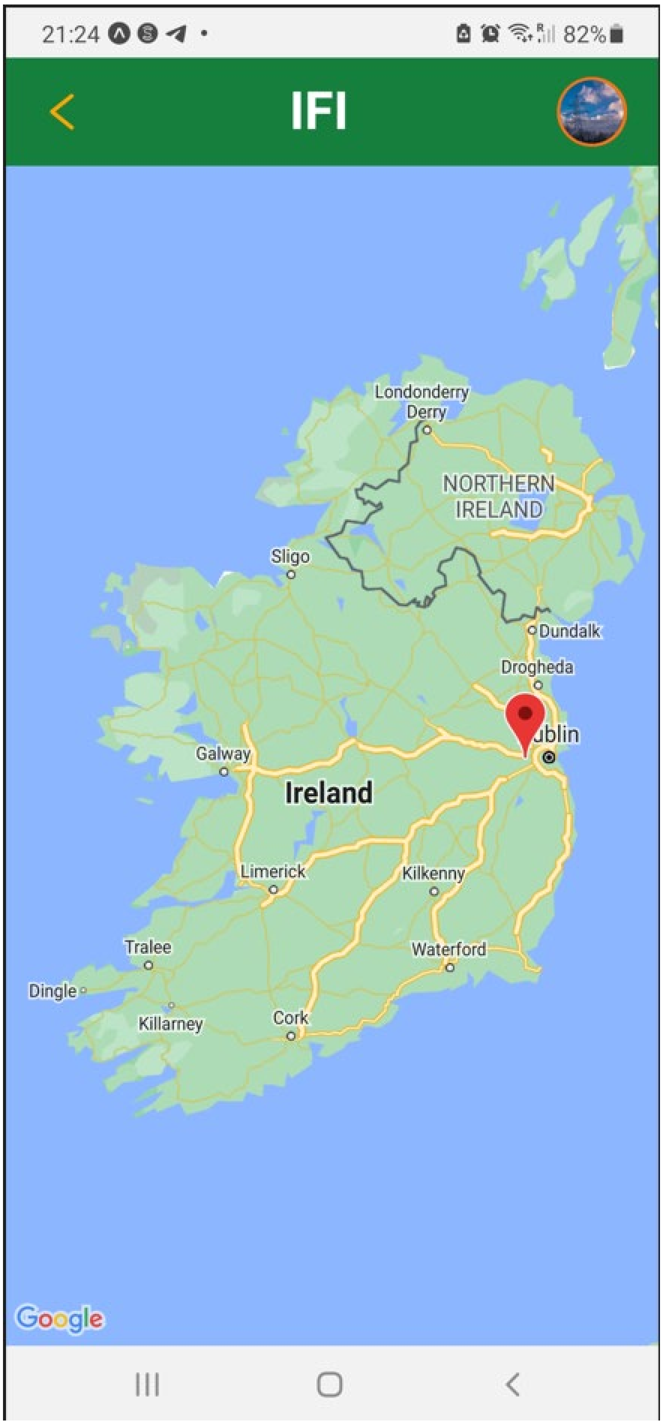


Figure 26 Google interaction Map UI

2.5. Testing

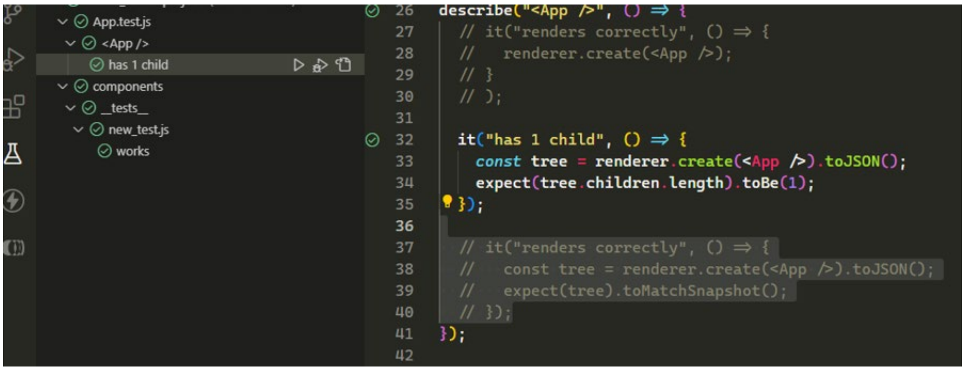
The testing that I use to perform on the Application was the Whitebox testing for checking all application functionality. It was performed on the sign-in with the Google account, matching algorithm, saving the user data to the Firebase database under a specific userID and on the user chat room messages.

After each function was created into the application, I did a manual test every time to make sure that the function/s works on different devices to see if that function was working properly or not.

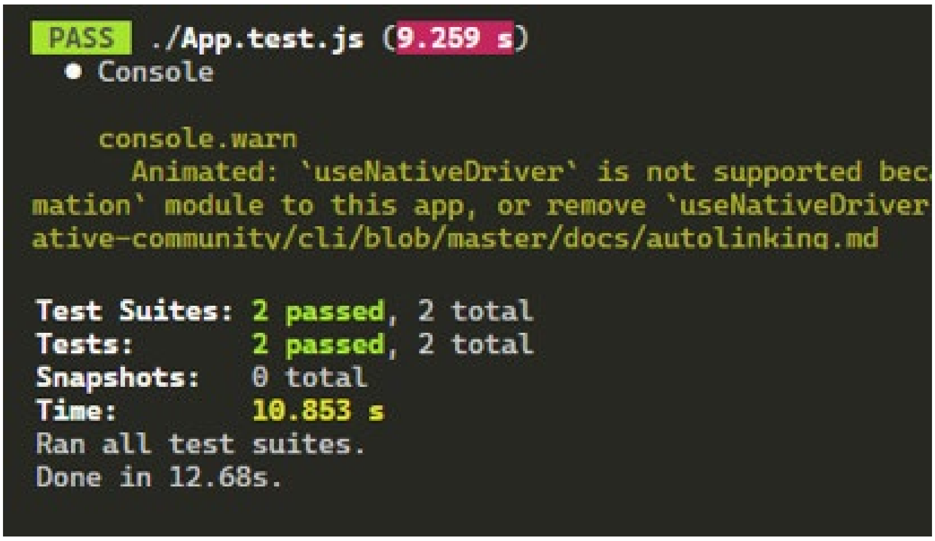
I also did a unit test with a jest-expo package one of the most used JavaScript unit testing frameworks.

- **Unit test** for checking all the functionality of the application
- **Snapshot test** for making sure that the UI (user interface) stays consistent

Unit Test with a jest-expo



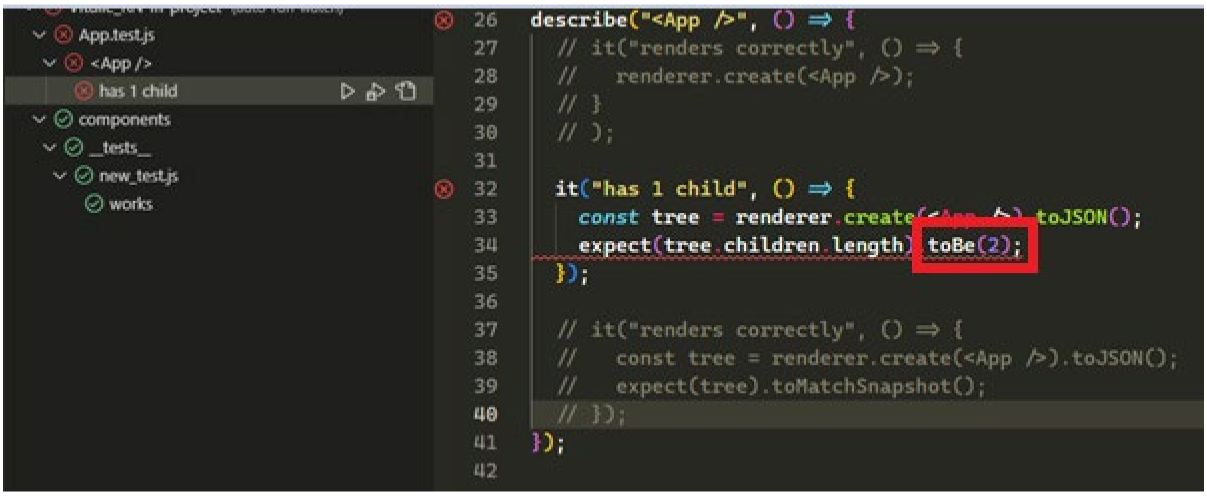
```
describe("<App />", () => {
  // it("renders correctly", () => {
  //   renderer.create(<App />);
  // }
  // );

  it("has 1 child", () => {
    const tree = renderer.create(<App />).toJSON();
    expect(tree.children.length).toBe(1);
  });

  // it("renders correctly", () => {
  //   const tree = renderer.create(<App />).toJSON();
  //   expect(tree).toMatchSnapshot();
  // });
});
```



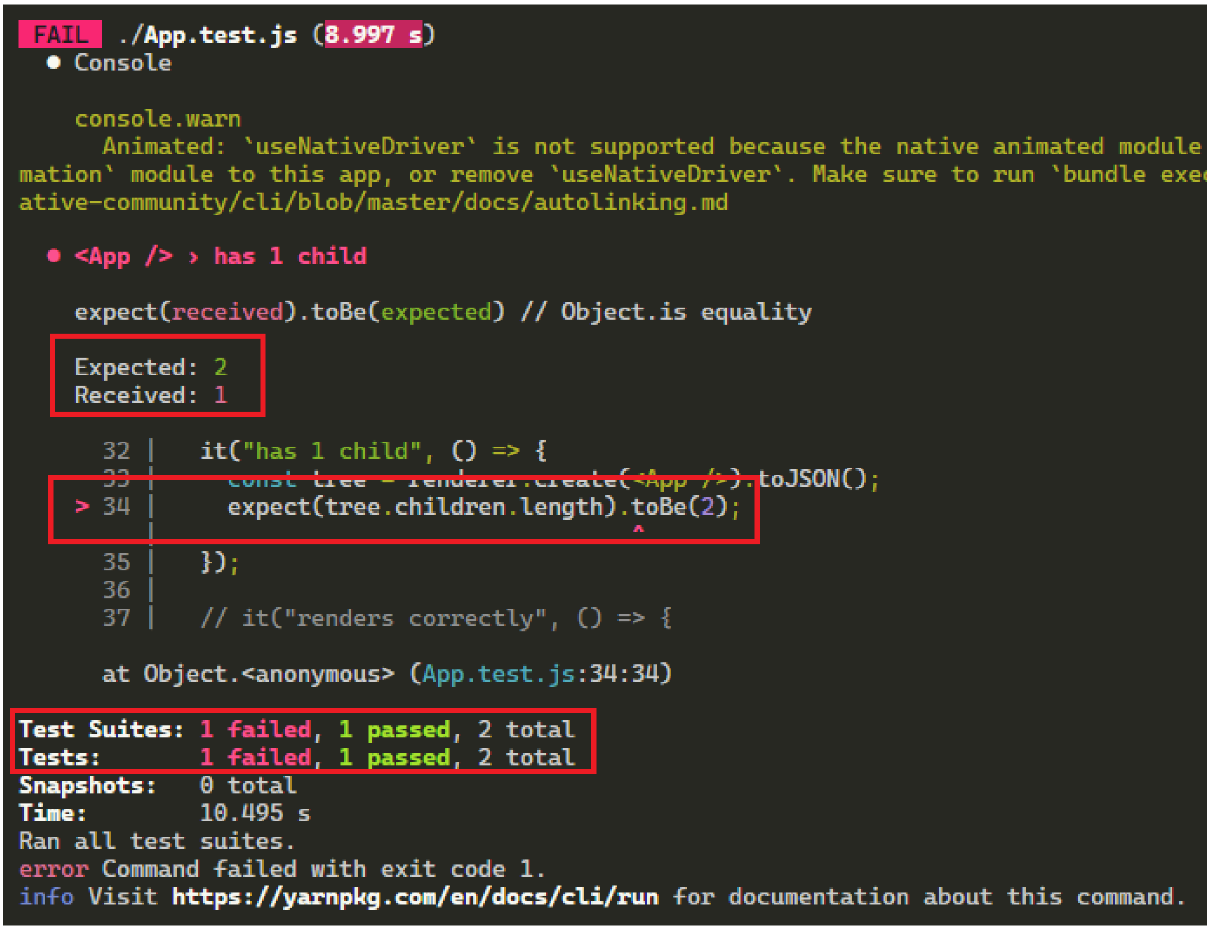
```
 PASS  ./App.test.js (9.259 s)
  ● Console

    console.warn
      Animated: `useNativeDriver` is not supported bec
mation` module to this app, or remove `useNativeDriver
ative-community/cli/blob/master/docs/autolinking.md

Test Suites: 2 passed, 2 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        10.853 s
Ran all test suites.
Done in 12.68s.
```

Result for Unit test: In the above image, we can see that there are no errors on the application when we are running the jest unit test

Next, we will try to make some changes to the testing code, changing the number 1 to 2

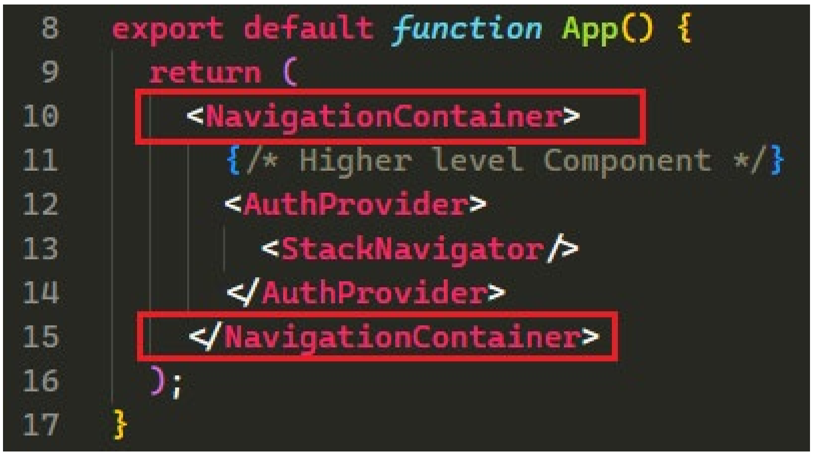


```
describe("<App />", () => {
  // it("renders correctly", () => {
  //   renderer.create(<App />);
  // }
  // );

  it("has 1 child", () => {
    const tree = renderer.create(<App />).toJSON();
    expect(tree.children.length).toBe(2);
  });

  // it("renders correctly", () => {
  //   const tree = renderer.create(<App />).toJSON();
  //   expect(tree).toMatchSnapshot();
  // });
});
```



```
 FAIL  ./App.test.js (8.997 s)
  ● Console

    console.warn
      Animated: `useNativeDriver` is not supported because the native animated module
mation` module to this app, or remove `useNativeDriver`. Make sure to run `bundle exe
ative-community/cli/blob/master/docs/autolinking.md

  ● <App /> › has 1 child

    expect(received).toBe(expected) // Object.is equality

    Expected: 2
    Received: 1

      32 |   it("has 1 child", () => {
      33 |     const tree = renderer.create(<App />).toJSON();
    > 34 |     expect(tree.children.length).toBe(2);
         |                                  ^
      35 |   });
      36 |
      37 |   // it("renders correctly", () => {

      at Object.<anonymous> (App.test.js:34:34)

Test Suites: 1 failed, 1 passed, 2 total
Tests:       1 failed, 1 passed, 2 total
Snapshots:   0 total
Time:        10.495 s
Ran all test suites.
error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
```

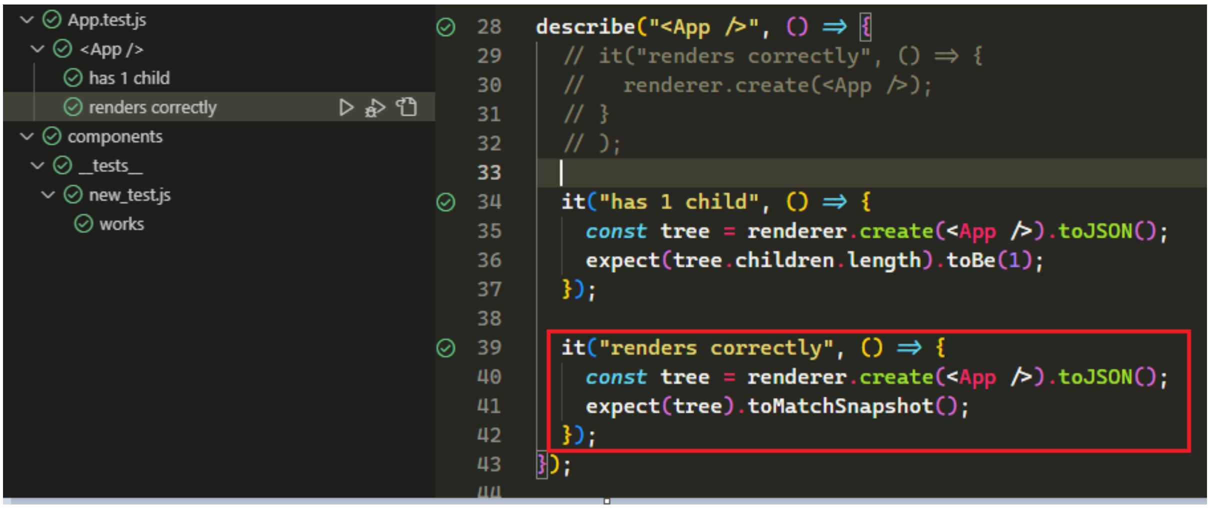
As we can see in the images above there are some errors displayed. Instead of receiving 1 child, we get the expected 2 which is wrong



```
export default function App() {
  return (
    <NavigationContainer>
      {/* Higher level Component */}
      <AuthProvider>
        <StackNavigator/>
      </AuthProvider>
    </NavigationContainer>
  );
}
```

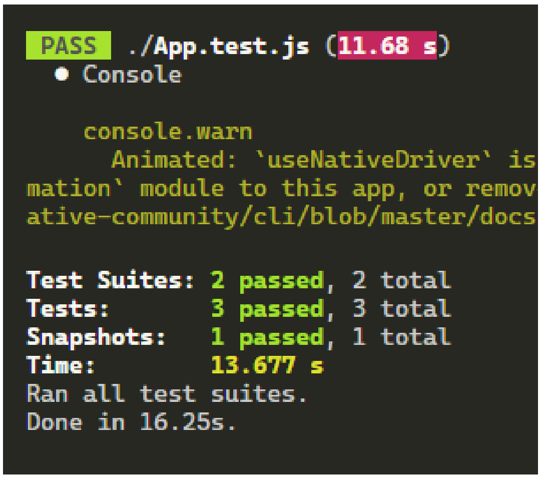
In the image above the function App is displayed where all the tags are wrapped with the <NavigationContainer> which counts as one child in React/React Native.

Result for Snapshot test**:**



```
App.test.js
  <App />
    has 1 child
    renders correctly
components
  _tests_
    new_test.js
      works
```

```
describe("<App />", () => {
  // it("renders correctly", () => {
  //   renderer.create(<App />);
  // }
  // );

  it("has 1 child", () => {
    const tree = renderer.create(<App />).toJSON();
    expect(tree.children.length).toBe(1);
  });

  it("renders correctly", () => {
    const tree = renderer.create(<App />).toJSON();
    expect(tree).toMatchSnapshot();
  });
});
```

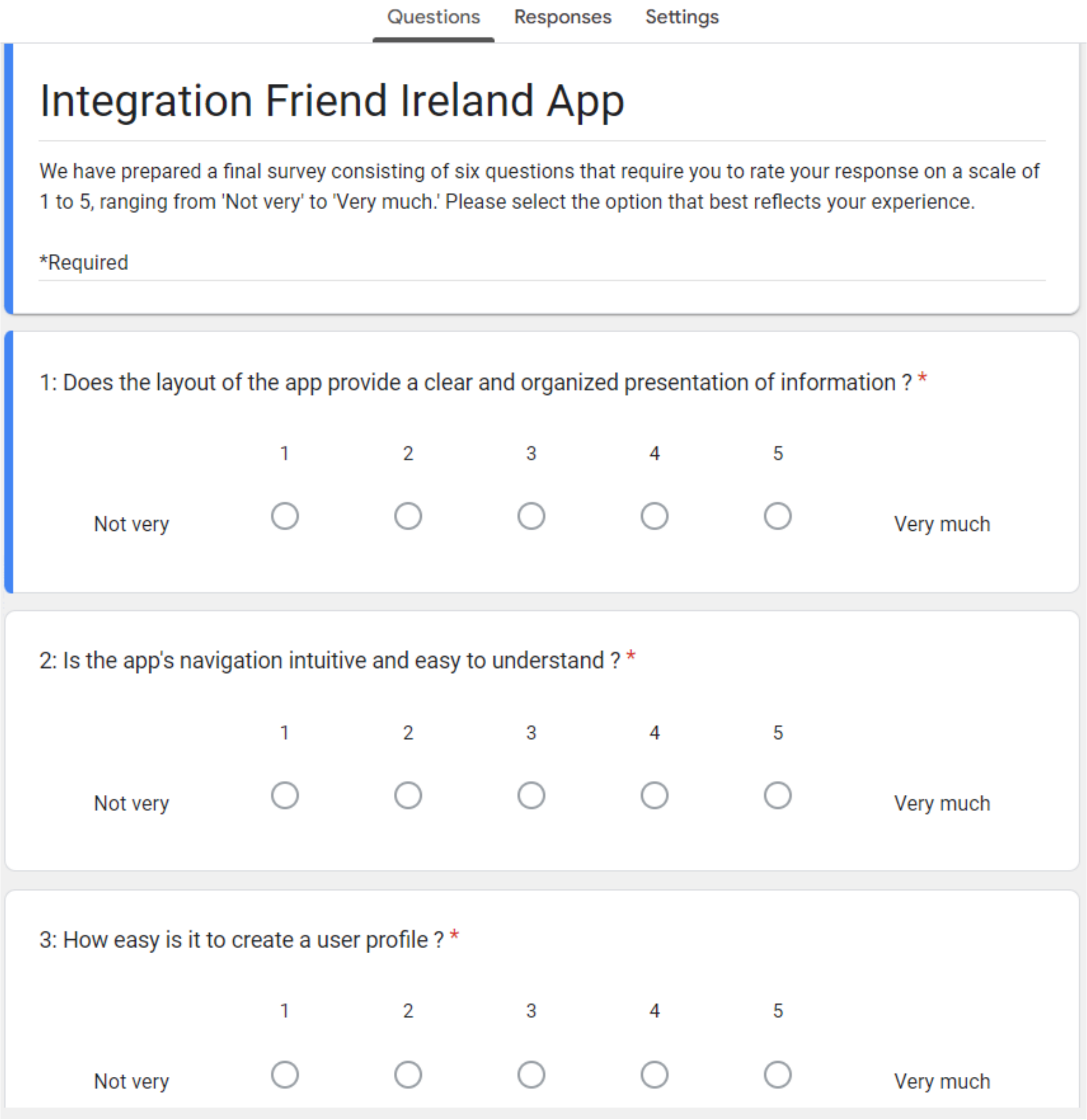


```
 PASS  ./App.test.js (11.68 s)
  ● Console

    console.warn
      Animated: `useNativeDriver` is
mation` module to this app, or remov
ative-community/cli/blob/master/docs

Test Suites: 2 passed, 2 total
Tests:       3 passed, 3 total
Snapshots:   1 passed, 1 total
Time:        13.677 s
Ran all test suites.
Done in 16.25s.
```

In the two images above we can see that the Snapshot test for the UI (user interface) is working as its supposed to work and with no errors

2.5.1. UI/UX Survey testing

I have conducted a survey among family, friends and acquaintances utilizing Google Forms. The survey comprised six questions, each requiring participants had to select their response on a scale ranging from 1 to 5 with No and Yes options. The scale begins with 1, representing "Not very," and concludes with 5, indicating "Very much."



Integration Friend Ireland App

We have prepared a final survey consisting of six questions that require you to rate your response on a scale of 1 to 5, ranging from 'Not very' to 'Very much.' Please select the option that best reflects your experience.

*Required

1: Does the layout of the app provide a clear and organized presentation of information ? *

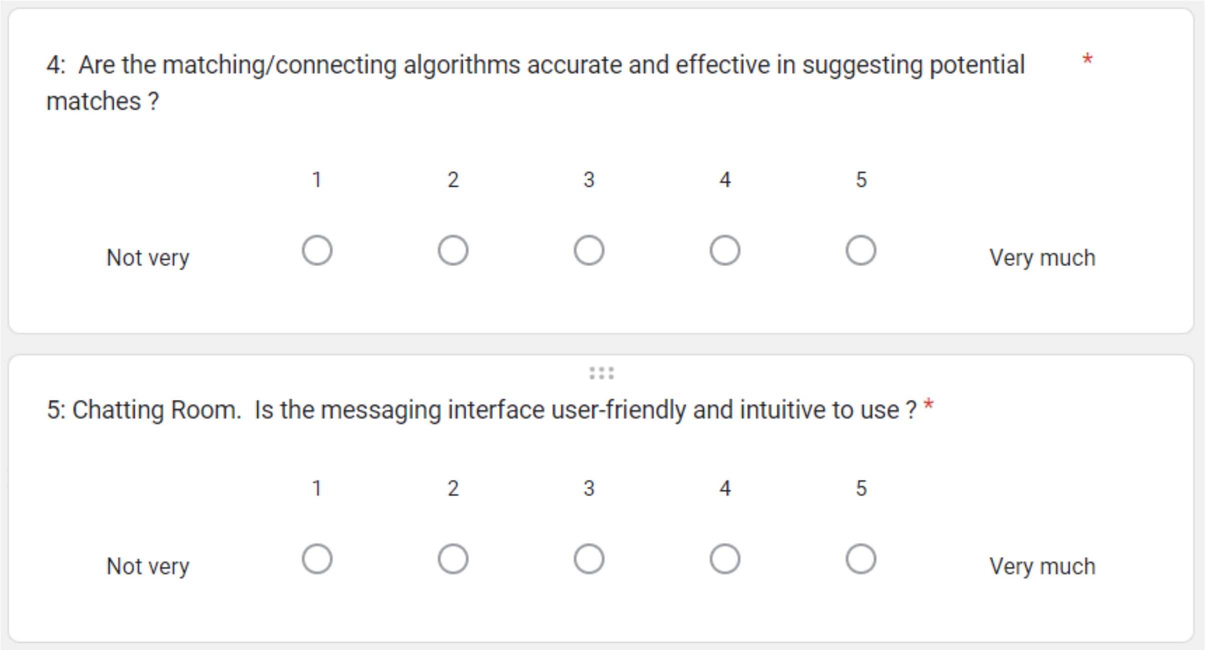
	1	2	3	4	5	
Not very	○	○	○	○	○	Very much

2: Is the app's navigation intuitive and easy to understand ? *

	1	2	3	4	5	
Not very	○	○	○	○	○	Very much

3: How easy is it to create a user profile ? *

	1	2	3	4	5	
Not very	○	○	○	○	○	Very much

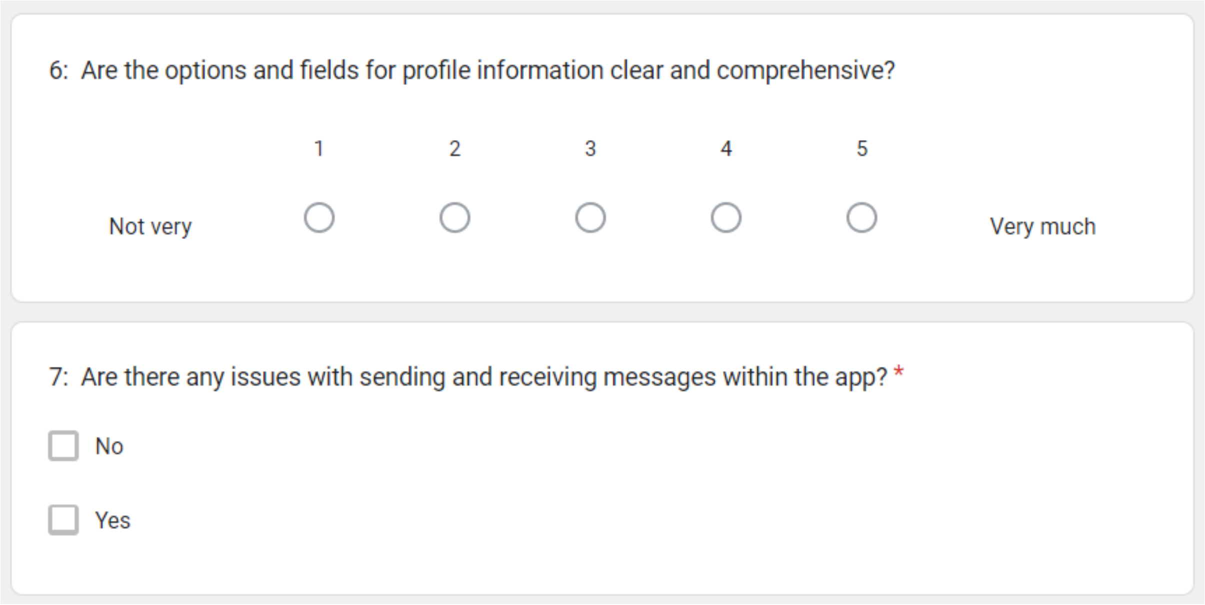


4: Are the matching/connecting algorithms accurate and effective in suggesting potential matches ? *

	1	2	3	4	5	
Not very	○	○	○	○	○	Very much

5: Chatting Room. Is the messaging interface user-friendly and intuitive to use ? *

	1	2	3	4	5	
Not very	○	○	○	○	○	Very much



6: Are the options and fields for profile information clear and comprehensive?

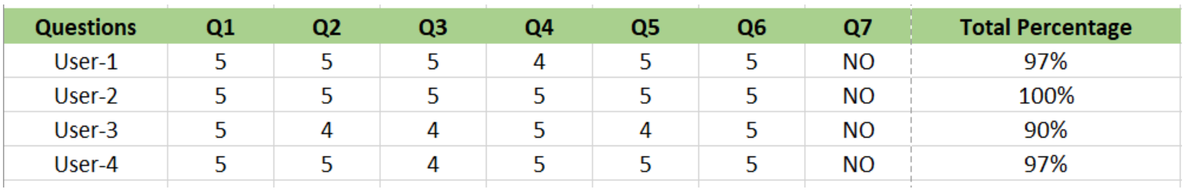
	1	2	3	4	5	
Not very	○	○	○	○	○	Very much

7: Are there any issues with sending and receiving messages within the app? *

□ No

□ Yes

UI/UX Survey results of four users



Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total Percentage
User-1	5	5	5	4	5	5	NO	97%
User-2	5	5	5	5	5	5	NO	100%
User-3	5	4	4	5	4	5	NO	90%
User-4	5	5	4	5	5	5	NO	97%

Google Form: https://forms.gle/P4811ysyxQoqoPEeA

2.6. Evaluation

The Integration Friend Ireland app was developed with a specific focus on the importance of integration within a functioning society. Recognising that integration is a two-way process, the app aimed to include both established Irish individuals, regardless of their ethnicity or place of birth. Drawing inspiration from dating and friendship apps, the app created a platform for connecting and learning from one another, similar to a buddy system. The objective was to foster mutual understanding and facilitate meaningful connections, without the pressure of finding a soulmate or lifelong friend.

In order to achieve comprehensive integration, it is not sufficient for newcomers alone to adapt and assimilate into their new environment. It is equally essential for established Irish individuals, including ethnic, naturalized, and EU citizens, to demonstrate a willingness to learn about and engage with newcomers. Acknowledging that social interaction is as vital as basic necessities like food, shelter, and warmth, the app aimed to facilitate contact between new and established Irish individuals, considering it the most crucial aspect of successful integration.

By creating an integration friend app, the project aimed to make these connections easier and more accessible. Recognising that humans possess both altruistic and selfish tendencies, the app sought to provide an opportunity for established Irish individuals to extend a helping hand, knowing that many would be willing to do so given the chance. Leveraging the familiarity of established Irish individuals with online dating, friendship apps, and senior companionship apps, the project highlighted the need for an integration friend app in Ireland.

While numerous organisations exist to support newcomers in Ireland, ranging from civil society to businesses and the public sector, many individual established Irish individuals wish to personally contribute without the hassle of multiple bureaucratic steps before connecting with a newcomer or an established Irish counterpart. The integration friend app aimed to empower users, allowing them to make their own decisions and initiate conversations within seconds of being matched, fostering immediate and meaningful interactions.

3. Conclusions

The Integration Friend Ireland app project has several advantages, strengths, and limitations that can be identified:

Advantages and Strengths:

1. Promotion of Integration: The project focused on the crucial aspect of integration within society, emphasising the importance of mutual understanding and connections between newcomers and established Irish individuals. By facilitating meaningful interactions, the app aimed to contribute to a more inclusive and cohesive community.
2. Easy and Accessible Platform: The app provided a user-friendly platform for individuals to connect and learn from each other, similar to a dating or friendship app. By leveraging the familiarity of such platforms, it ensured that the integration process could be initiated easily, allowing users to start chatting within seconds of being matched.
3. Personal Empowerment: The project aimed to empower both newcomers and established Irish individuals by giving them the autonomy to decide and engage in interactions based on their own preferences and interests. This personal empowerment aspect encouraged voluntary participation and a genuine desire to connect and learn from one another.

Limitations:

1. Reliance on User Participation: The success of the app heavily relies on the active participation of users. If there is a lack of engagement or limited user adoption, the effectiveness of the platform in promoting integration may be compromised.
2. Potential Language and Cultural Barriers: Despite the app's goal of fostering connections and understanding, language and cultural barriers may still pose challenges to effective communication and interaction. Overcoming these barriers may require additional resources or support mechanisms.
3. Limited Scope: The app primarily focused on facilitating connections and learning between newcomers and established Irish individuals. While this is a crucial aspect of integration, other dimensions, such as access to resources, employment opportunities, and legal support, may not be adequately addressed by the app alone. Collaborating with existing organizations and services may be necessary to provide comprehensive support.
4. Privacy and Safety Concerns: As with any social networking app, there are potential privacy and safety concerns associated with the integration friend app. Ensuring robust privacy measures, user verification, and moderation protocols would be crucial to address these concerns and maintain a safe environment for all users.

In conclusion, the Integration Friend Ireland app project offers several advantages, such as promoting integration, providing an accessible platform, and empowering individuals. However, limitations, such as reliance on user participation, language barriers, limited scope, and privacy concerns, need to be considered and addressed to maximize the app's effectiveness in achieving its integration goals.

4. Further Development or Research

With additional time and resources, the Integration Friend Ireland app project could evolve and expand in several directions:

1. Enhanced Features and Functionality: Further development could focus on enhancing the app's features and functionality to provide a more comprehensive and immersive user experience. This may include features like language translation capabilities, cultural exchange resources, integration-focused events or activities, personalized recommendations based on user preferences, and integration-related educational content.
2. Collaborative Partnerships: The project could explore collaborative partnerships with existing organizations, both within the public and private sectors, that are already engaged in supporting integration efforts. By leveraging these partnerships, the app could tap into additional resources, expertise, and networks, enabling a more holistic approach to integration and facilitating access to a wider range of support services for users.
3. Research and Evaluation: Additional research and evaluation efforts could be conducted to measure the impact and effectiveness of the app in promoting integration. This could involve gathering user feedback, conducting surveys or interviews, and analyzing usage data to gain insights into user experiences, identify areas for improvement, and measure the overall success of the app in achieving its integration goals.
4. Expansion to Other Regions: Once the app has established a successful model in Ireland, there may be potential for expansion to other regions or countries facing similar integration challenges. Adapting the app to different contexts, languages, and cultural nuances could help address integration needs in diverse communities and further amplify the positive impact of the project.
5. Integration with Existing Systems: The project could explore integration with existing systems or platforms that cater to newcomers or established Irish individuals, such as government databases, language learning platforms, or employment networks. By integrating with these systems, the app could offer a more streamlined and interconnected experience, facilitating access to a broader range of resources and services.

In summary, with additional time and resources, the Integration Friend Ireland app project could pursue avenues for enhanced features and functionality, collaborative partnerships, research and evaluation, expansion to other regions, and integration with existing systems.

These directions would further strengthen the app's impact and contribute to fostering successful integration within society.

5. References

[1] react native, “React Native,” react native, 2023. [Online]. Available: https://reactnative.dev/. [Accessed 5 Nvember- May 2022-2023].

[2] React, “React,” React, 2022-2023. [Online]. Available: https://legacy.reactjs.org/. [Accessed November-May 2022-2023].

[3] MDN Mozilla Developer Network, “JavaScript,” MDN Mozilla Developer Network, 1998. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript. [Accessed November-MAy 2022-2023].

[4] tailwindcss.com, “Get started with Tailwind CSS,” tailwindcss.com, 2017. [Online]. Available: https://tailwindcss.com/docs/installation. [Accessed November-May 2022-2023].

[5] firebase.google.com, “Add Firebase to your JavaScript project,” firebase.google.com, [Online]. Available: https://firebase.google.com/docs/android/setup. [Accessed November-May 2022-2023].

[6] React Native Firebase, “React Native Firebase/ Getting Started,” rnfirebase.io, 2017. [Online]. Available: https://rnfirebase.io/. [Accessed November-May 2022-2023].

[7] Microsoft, “Getting Started,” Microsoft, 29 April 2015. [Online]. Available: https://code.visualstudio.com/docs. [Accessed 1 November 2022].

[8] ANDROID STUDIO, “Download,” ANDROID STUDIO, [Online]. Available: https://developer.android.com/studio. [Accessed 1 November 2022].

6. Appendices

6.1. Project Proposal

6.1.1. Objective

Background

I chose this project because integration is important. The essential connection for newcomers with established Irish people, whether native or born elsewhere, is essential for good integration. Migrants can’t go through the process of integration on their own. Established Irish people are part of the process. We need to get better at connecting new Irish with established Irish to minimise division which leads to racism, hate and in a worst-case scenario, war. As a migrant myself, I realised that despite Ireland now being a multicultural country, people are better at integrating into different communities than the larger Irish community. To fully belong, a little bit of us must feel Irish. But how can we without connection to established Irish people who know about rights and responsibilities? Rights are about ourselves and responsibilities are about all of us. Established Irish have both and we must help new Irish to reach the same balance. All established Irish, whether ethnic or new Irish have the key to Ireland. This app will help established Irish give the key to Ireland to newcomers. Everyone in Ireland deserves the key to Ireland. Of course, no one has the key to the whole of Ireland, but all established Irish has a key to a part of Ireland. It's time to duplicate these keys by sharing knowledge, interests and love of living on this amazing island.

Shared happiness doubles happiness.

This app is an Integration Friend App; based on the same principles as a dating app, but it's for a different purpose. As integration is a two-way process, this app will help new and established Irish connect and learn from each other. To live together, we have to trust each other. Good integration creates trust.

6.1.2. State of the Art

The only similar thing I was able to find is the Department of Justice's integration website http://www.integration.ie/en/isec/pages/home which talks about integration but still doesn't involve established Irish people which is the missing ingredient in integration. We tend to put all our trust in the reputation of the friendliness of the Irish, presuming everyone will automatically help, but there's a difference between passive friendliness and active help. We can't risk division, hatred and racism by not involving everyday established Irish people in the integration journey. If newcomers and established Irish don't meet, we risk an "us v them" climate in Ireland, or in and out-groups. All newcomers deserve to become part of the Irish community too, whether that's a small special interest community like hikers in Wicklow or a large cultural community like the GAA which is found in all towns across Ireland.

This project stands out because it involves and trusts established Irish - native or born elsewhere to welcome newcomers and help with the integration process.

If successfully executed, this app could be used by Local Governments as an extra tool in the integration work.

6.1.3. Technical Approach

1. Research

The research uncovered the idea for the app itself. I considered many different types of apps until I decided to go with the idea of an integration friend app.

2. Idea

The idea for the integration app came about when brainstorming app ideas. I was thinking of problems and possible solutions, but the market was saturated with everything I could think of, coffee orders, pet sitting, and community giveaways until I thought of the difficulty of integrating into a new country. I checked the Department of Justice's website, and yes, they talk about integration, but still as a one-way process, so I thought, that's my app. An integration friend Ireland app. As the name suggests, it's a friend during different stages of integration, usually the beginning, but many also need help later on in their integration journey. Integration is not just about settling into a new country, it's discovering a new self. An established Irish can help a newcomer to figure out what their unique flavour of Irishness will be.

3. Identification of the problem

As established Irish, we often leave it to the newcomer to integrate on their own, to make the necessary contact to become part of Ireland. We fail to acknowledge that integration is a two-way process. The government leave it to professional bodies and organisations to help with the integration, but newcomers need to connect with every day established Irish people of all nationalities to integrate fully. The connections made via this app will teach both parties about rights and obligations. Friendships are a bonus, the goal is integration to the extent that we understand and accept each other.

4. Functionality

The plan is to match a newcomer and an established Irish according to location and introduction of themselves - who they are, what they are looking for/what they can help with and some interests. When they are matched, a chat will appear and they can make their own arrangements. There is no obligation for either party to become lifelong friends. The app is called Integration Friend Ireland, but the bond of friendship, in this case, is helping the newcomers settle into Ireland and for the established person to learn about a different culture.

5. Design

The design will in part be similar to a dating app, only this app will match people according to geography, interests, availability and also the option of a once-off meet-up. This is not an app for meeting new friends, if it happens, fine, but it's not the end goal. The initial contact will be via chat in the app and from there, people can decide to meet for a coffee/lunch/walk and learn what's on their minds.

6. Testing

I will test the app by inserting test profiles and matching these profiles according to

1. Geography
2. Interests
3. Availability

7. Problem Assessment

If the app crashes with too many matches, I'll have to figure out why and solve it

8. Feedback

6.1.4. Technical Details

For this project, I will develop an android application and a database. I will be using a number of different programming languages and software to accomplish the design and performance of this project.

The programming languages which will be used are: React/React Native, JavaScript, CSS(Tailwind), Firebase or MongoDB, also IDE android studio, and VSCode (Visual Studio Code).

I will need to learn all the programming languages mentioned above from scratch.

- **React/React Native:** It is a framework for developing mobile apps for android and iOS by using one common language JavaScript, with react native the app will have a smooth, responsive user interface
- **JavaScript:** This will be used for modifying the content and for functionality
- **CSS(Tailwind):** For styling the application
- **Firebase or MongoDB:** One of these databases will be used for communicating with the Application and data to be stored
- **For hosting:** When the application is accomplished can be hosted on the Google store or App Store
- **Heroku:** For demonstration purposes

6.1.5. Special Resources Required

I will use made-up data.

6.1.6. Project Plan

Phase 1 Pre-Planning

Step 1: Write down the idea with as much detail as possible.

Who is it for? What will it do? Why should people care?

Why does the world need an app like this? Will it solve a problem? Will there be a need for money transactions? Will it make Ireland a better place to live? How? Is there a business case for the app?

Step 2: Research - is there already a similar app on the market and if so, how will this app be better?

Phase 2: Brainstorming

Involve others to get ideas for how it could work/look when ready as I'm working on my own. Determine if this idea is feasible, is it too big, or too small

How much will it cost? Where will I host it? Who is going to do the design? What's the timeline and budget?

Make rough sketches of how the app will look. Uncover usability issues.

Phase 3: Technical Assessment

It's not just about interaction and visuals, but will the backend support functionality?

Get usable data. Source a Public API, or consider building my own abstraction layer

What devices will the app go on? What are the different requirements?

Phase 4: Prototype

Build a prototype to get the concept of the app to users. I will use rough wireframes. Feedback is important at this point.

Phase 5: Design Before Development

Design for user experience UX. The design comes before the code. Create the interaction architecture

The user interface UI will create the look and feel of the app. At this stage, the visual direction is important. How will the app feel and flow?

Phase 6: Build the App

It's time to build using agile development to adapt to change quickly.

Phase 7: Testing

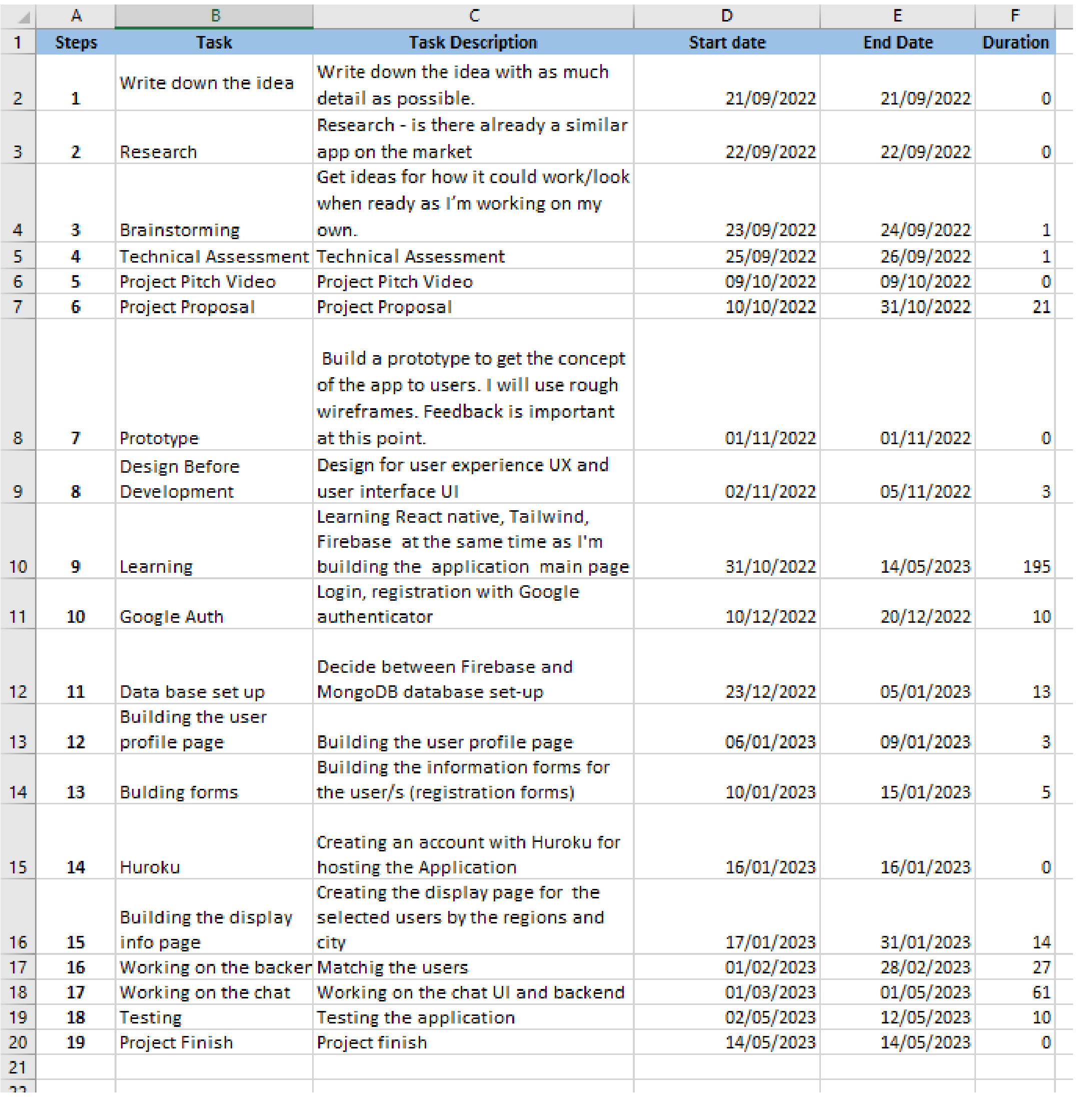
Will the app pass the user acceptance testing? Give the app to testers to see if the app works.

Beta testing is also important. Feedback is crucial to find out if the app works in a real world environment.

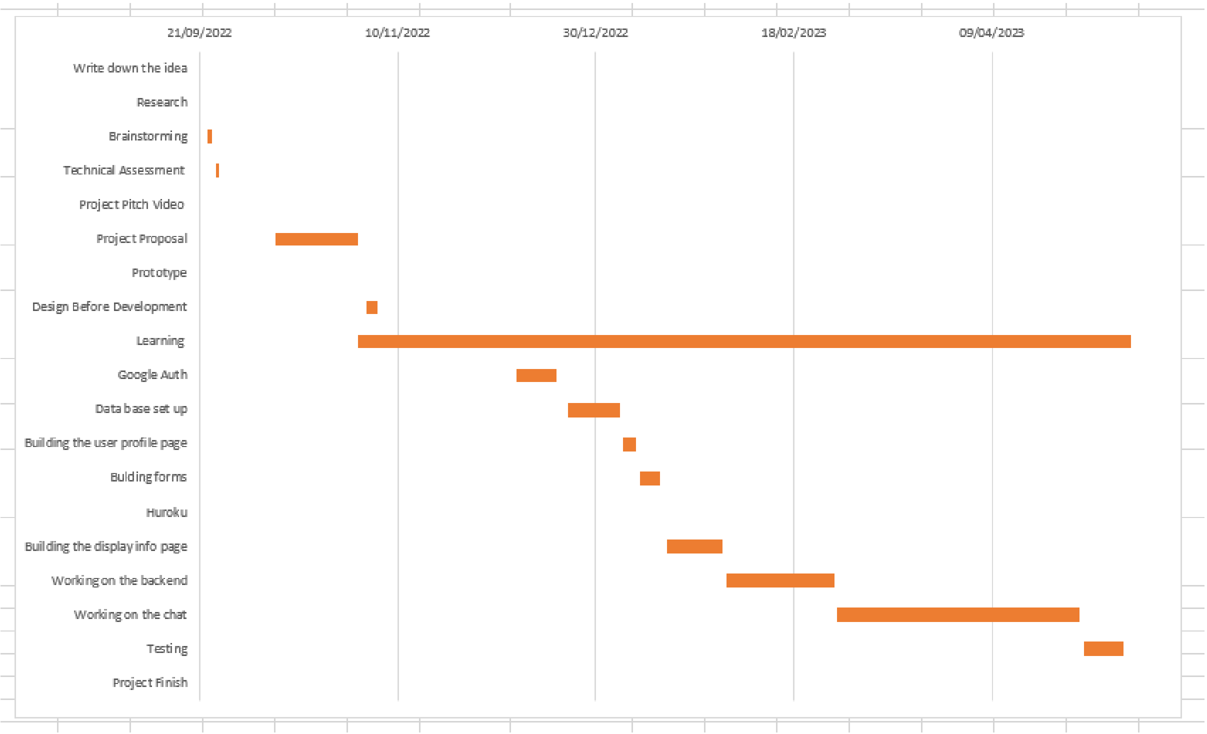
Phase 8: Launch

Submit the app and celebrate, I have built an app.

Project Plan (Gantt Chart)



	A	B	C	D	E	F
1	**Steps**	**Task**	**Task Description**	**Start date**	**End Date**	**Duration**
2	**1**	Write down the idea	Write down the idea with as much detail as possible.	21/09/2022	21/09/2022	0
3	**2**	Research	Research - is there already a similar app on the market	22/09/2022	22/09/2022	0
4	**3**	Brainstorming	Get ideas for how it could work/look when ready as I'm working on my own.	23/09/2022	24/09/2022	1
5	**4**	Technical Assessment	Technical Assessment	25/09/2022	26/09/2022	1
6	**5**	Project Pitch Video	Project Pitch Video	09/10/2022	09/10/2022	0
7	**6**	Project Proposal	Project Proposal	10/10/2022	31/10/2022	21
8	**7**	Prototype	Build a prototype to get the concept of the app to users. I will use rough wireframes. Feedback is important at this point.	01/11/2022	01/11/2022	0
9	**8**	Design Before Development	Design for user experience UX and user interface UI	02/11/2022	05/11/2022	3
10	**9**	Learning	Learning React native, Tailwind, Firebase at the same time as I'm building the application main page	31/10/2022	14/05/2023	195
11	**10**	Google Auth	Login, registration with Google authenticator	10/12/2022	20/12/2022	10
12	**11**	Data base set up	Decide between Firebase and MongoDB database set-up	23/12/2022	05/01/2023	13
13	**12**	Building the user profile page	Building the user profile page	06/01/2023	09/01/2023	3
14	**13**	Bulding forms	Building the information forms for the user/s (registration forms)	10/01/2023	15/01/2023	5
15	**14**	Huroku	Creating an account with Huroku for hosting the Application	16/01/2023	16/01/2023	0
16	**15**	Building the display info page	Creating the display page for the selected users by the regions and city	17/01/2023	31/01/2023	14
17	**16**	Working on the backer	Matchig the users	01/02/2023	28/02/2023	27
18	**17**	Working on the chat	Working on the chat UI and backend	01/03/2023	01/05/2023	61
19	**18**	Testing	Testing the application	02/05/2023	12/05/2023	10
20	**19**	Project Finish	Project finish	14/05/2023	14/05/2023	0
21						



6.1.7. Testing

I will ask family members to test the application for UI, UX and errors

1.1. Reflective Journals

1.2. Other materials used

Any other reference material used in the project for example evaluation surveys etc.

6.2. Reflective Journals

October

Supervision & Reflection Template

Student Name	Vitalie Brescanu
Student Number	x19241011
Course	BSHCSD4
Supervisor	Lisa Murphy

Month: October

What?
Reflect on what has happened in your project this month?
I came up with the idea for the app and started researching similar apps, but as I couldn't find any, I looked into how functionality on how to match people for a specific purpose which in my case is for integration help.

So What?
Consider what that meant for your project progress. What were your successes? What challenges still remain?
I was happy to finally settle for an idea as I had discarded so many due to saturation in the market. The succes this month was that I wrote the proposal, pitched the idea via video and it was approved with some modifications -explain the functionality in more detail.

Now What?
What can you do to address outstanding challenges?
Think deeper about how the app will help people.

Student Signature	

November

Supervision & Reflection Template

Student Name	Vitalie Brescanu
Student Number	x19241011
Course	BSHCSD4
Supervisor	Lisa Murphy

Month: November

This month I met up with my supervisor on MS Teams and had a general chat about the application. I received feedback and I was asked for suggestions about what else to add to the application to make it more complex. I have since decided to add an interactive Map to the application. Also, the wireframe has been created for the application. The Firebase database was selected for this application and an account for the Firebase was created. The Firebase will help to make the map easier in the future using JSON objects. I'm continuing to learn React Native, Tailwind CSS and Google authenticator.
I've started and I'm progressing slowly. A lot of challenges still remain as I have a lot of things to learn. I'm struggling to learn React Native and Firebase which are challenging programs and totally new for me. Remaining challenges are still to create, design the database and Google Authenticator
The plan for the next month is to create the Firebase database and connect it with the application. I will also build the main screen page. I will try to make the prototype application work as planned. When it works the user will be able to see the main screen and let the user login with their Google account.

Student Signature	

December

Supervision & Reflection Template

Student Name	Vitalie Brescanu
Student Number	x19241011
Course	BSHCSD4
Supervisor	Lisa Murphy

Month: December

In December, I set up a project for the application with Google Cloud and Google Firebase, and I also completed the main screen / welcome screen of the application. The login with a Google account is almost finished. I encountered some small issues with the Google authenticator, I can't see the client_id, it comes up as "render error" and I'm still working on it. To move between screens on the application, I have created two more screen pages to navigate through these screens on the app.

For this month, what was planned has been achieved except for the login with a Google account. All the documentation is explained and a lot of code is deprecated, taking a lot of time for the research. There has been a lot of challenges connecting the application with Google Cloud and Google Firebase and I still have a lot to learn. What hasn't been finished this month will be sorted out next month. The home screen page was completed and also navigation to other pages. The database has been created on Firebase, but the application has not been connected to the database becasue first I need to solve the issue by login in with a Google account into the application.

The plan for next month is to continue to work on the login with the Google account connecting to the application, also connecting with the database, building the user profile screen page, creating the user profile form, and saving the input values from the user profile form into the database.

Student Signature	

January

Supervision & Reflection Template

Student Name	Vitalie Brescanu
Student Number	x19241011
Course	BSHCSD4
Supervisor	Lisa Murphy

Month: January

This month I have been concentrating on finishing of the functionality for login with the Google account which will allow the user to login with Google credentials and the user value to be saved into the database. I also worked on building the user profile form/screen page where the user can update the profile. As well, I worked on building the Home screen page where the user data from the database will be displayed.

Plans for January: login with the Google account and everything mentioned in the section above has been accomplished. Everything is connected to the Firebase database, the user can login and logout, update the user profile on the application with saving the data to the database. There are still a lot of challenges and a lot to learn. I am encountering a lot of challenges with react native libraries where there is lots of libraries deprecations and the documentation on the expo React Native website is hard to understand.

Also, the design of the user profile form and the screen pages were challenging. I need to study Tailwind CSS more in-depth and how to use it for styling content. The same with JavaScript, Goole Firebase, and expo React Native, but in the end, the job was done.

For next month the plan is to continue working on the application, finishing with the Home screen page that will render a stock of cards with the user info will be displayed where the info will be retrieved from the database. Also working on the swiping algorithm where the user can swipe left for not accepting a friend or swipe right to accept a friend.

The biggest challenge for next month will be working on the swiping algorithm, but it’s an exciting challenge.

Student Signature	

February

Supervision & Reflection Template

Student Name	Vitalie Brescanu
Student Number	x19241011
Course	BSHCSD4
Supervisor	Lisa Murphy

Month: February

In February, I was working on the Home screen page where a list/deck of cards is displayed. The card displays the info of a registered user. I also created the ChatRow screen page where after the matches has happened it will be redirected to this page and display a list with the matched users. I did work on the code to fetch the user data from the database to display on the cards.

I also worked on the matching algorithm, a process that will take two sets of data and determine which elements/users from each set correspond or match with each other. I checked similar applications with the same swiping functionality to see how it works. I searched different blogs and videos on this swiping algorithm.

This month I succeeded to finish everything mentioned in the above section, connecting and fetching the data with the database and displaying the user info on the card. I finished the functionality of swiping the card left or right which is now working fine. I felt very happy about this. The biggest challenge I encountered was the matching algorithm – matching the two users to become friends and after moving to the ChatRow screen page. The on-screen pages of the application is getting better with tailwind CSS. I am feeling very satisfied with what I have done so far and confident that I can accomplish everything that is left to be done.

I'm continuing to learn JavaScript on the Firebase and expo React Native

For March, the plan is to accomplish finish the matching algorithm for matching two users if they want to be friends and display the matched user on the ChatRow screen page with designing using the tailwind CSS. I'll also work on modifying the user profile form, adding checkboxes, radio buttons, and <textarea> field.

Student Signature	

March

Supervision & Reflection Template

Student Name	Vitalie Brescanu
Student Number	x19241011
Course	BSHCSD4
Supervisor	Lisa Murphy

Month: March

What? This month I continued to work on the matching algorithm for displaying the users that were matched on the ChatRow screen. I also worked on modifying the user profile form, adding checkboxes, radio buttons, <textarea> field, and adding a button for selecting an image from the device. I had some issues creating the radio buttons and the checkboxes, and getting the value of the radio button saved on the Firebase database to display on the user card info. I spent a lot of time working on the matching functionality, and displaying the matching user on the CharRow seen page
So What? This month I managed to finish almost everything that was planned, except for getting the radio button value saved on the Firebase database and displaying it on the user's card. I'm continuing to work on this issue. This is the biggest challenge that I've encountered this month. I worked on designing the ChatRow screen to display the matching users which I also finished. I'm continuing to check on other similar application to see how do they work. I'm also continuing to learn JavaScript, tailwind CSS, Firebase and expo React Native.
Now What? The plan for April is to finish the ChatRow (chat list) screen and a Message Screen page with functionality to send messages between the users. The plan is also to work on retrieving the radio button values form the Firebase database and display it on the user card and validation of the of the user profile

Student Signature	

April

Supervision & Reflection Template

Student Name	Vitalie Brescanu
Student Number	x19241011
Course	BSHCSD4
Supervisor	Lisa Murphy

Month: April

What?

April was a very intensive month for me. I worked on what I had planned, including continuing to develop the user profile form. I decided to replace the radio button on the user profile with the select option method. Additionally, I worked on the ChatRow screen algorithm to fix a small error, which will display connected users, and on the chat room screen, sending, and receiving the messages making sure that the connection and saving to Firebase database includes the chat messages too.

I started working on the AES encryption and decryption algorithm using the CryptoJS for the user data protection

So What?

This month, I finished working on the user profile replacing the radio buttons with the select option method and it's working well, the date is saving and retrieving form the Firebase, and it displays the info perfect on the user card UI. The chat room screen and the sending messages between the users is also done. It was a little bit of a challenge with the chatting algorithm but I managed to finish it. Almost all the necessary functionality works so far except encryption and decryption of user data.

The AES encryption and decryption algorithm are a little bit of a headache, despite following all the steps in the documentation instructions I am getting an error. I need to do more research on this.

The design UI application looks good. I am happy so far.

Now What?

Next step is to work on the delete user account, integration Map screen, and finishing the AES encryption and decryption of user data. But because the time is so restricted, I will concentrate the rest of the coding and on finalising the technical report. As we have finished all the subjects and TABA's, I will have more time to work on both coding and documentation to finish the project. In case of time constraints, the documentation is more important.

Student Signature	

Additional resource

Learning material:

https://reactnavigation.org/docs/getting-started

https://rnfirebase.io/

https://docs.expo.dev/guides/using-firebase/

https://docs.expo.dev/tutorial/create-your-first-app/

https://reactnative.dev/

https://www.tutorialspoint.com/firebase/index.htm

https://www.w3schools.com/js/

https://developer.mozilla.org/en-US/docs/Web/JavaScript

react-native-deck-swiper

https://github.com/alexbrillant/react-native-deck-swiper

https://www.npmjs.com/package/react-native-deck-swiper