CS4048 Project Proposal

1 Group Members

Project Name – Forward Food
Kerry Houlihan / 19144946
Jason Redmond / 19274807
Eric Testu /21179565

Main Idea

We as a group decided to do a charity/ takeaway food app called Forward Food. We noticed that there is a lot of waste in the restaurant / takeaway industry at the end of the day. Forward Food allows businesses to sign up and post unwanted produce that then people can collect for free or for a discounted price. The typical users would be that of people from an impoverished background or people on a tight budget student etc. If you are a user, you will have to sign up after which you will be taken to the main page of the app where all the offers are. You can then reserve food at different places , which will be placed in your orders. You will also have an option to see where the business is with the help of google maps. If you choose to sign up as a business, you will be able to see your offers and orders for the offers. you can also change/edit your store.

2 User Stories

1. As a person from an impoverished background, I want to be able to get free/discounted food so that I can eat.

2. As a restaurant owner   I want to give away left over produce/food so that I can help people from a less fortunate background.

3. As a student I want to get discounted produce/food so that I can save up for my tuition fees.

4. As a takeaway owner   I want to sell produce at a discounted price so that I don't have to throw it away before closing.

5. As a charitable person I want to collect free/discounted food from restaurants   so that I can go and give to it the homeless /homeless shelter.

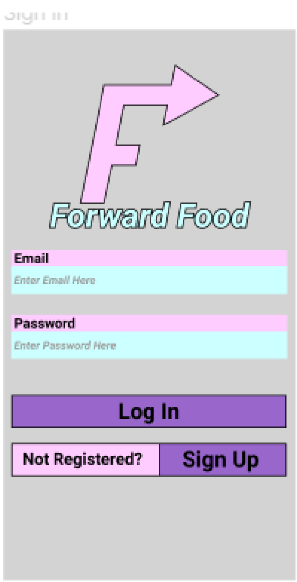
3 Technology

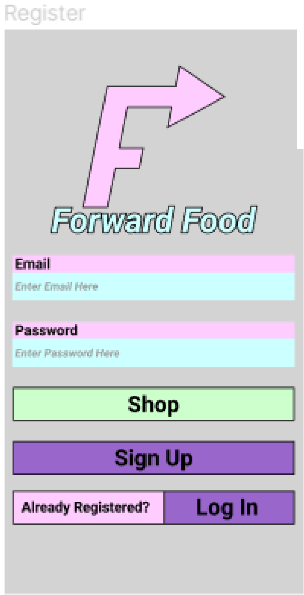
List any APIs and frameworks you intend to use (e.g., database, location-based functions, authentication, …)

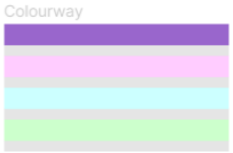
API/framework	**Purpose**
Firebase	Firestore Database and Authentication - Store user/business login and data.
Google Maps API	To show users where businesses are.

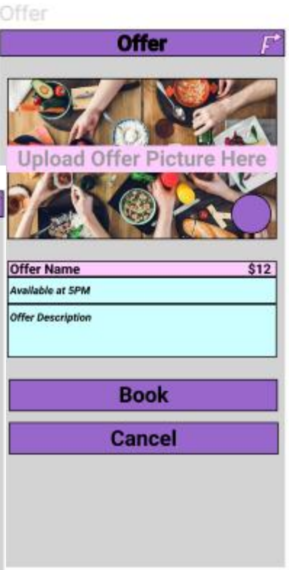
4 Navigation Structure

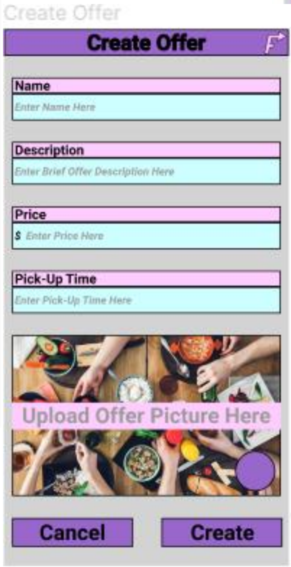
Overview







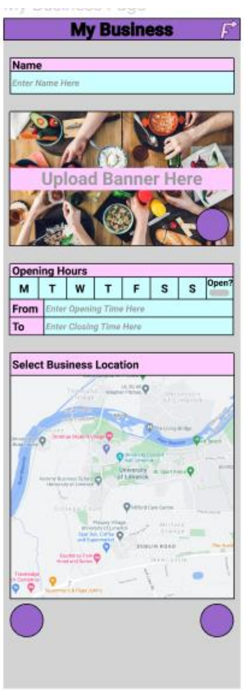




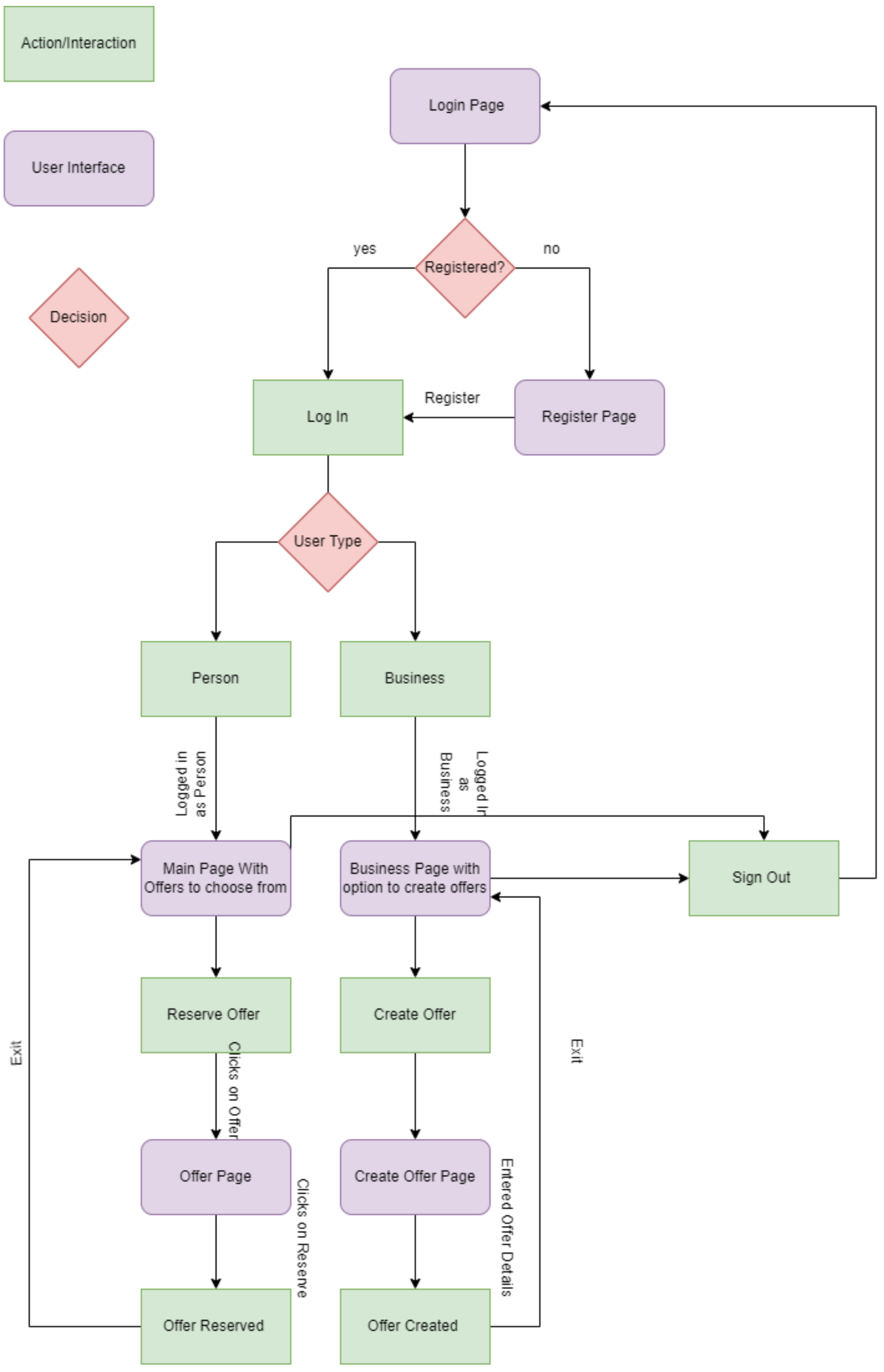
Sign In and Register: Once opening the app the user will be greeted with a sign in page where they have the choice to sign in. If they are not registered, they can Sign Up with their email. They must provide a user type here which can be Shop or a User/ Person.

Business: If the user registered as a Business they will be greeted with their business page where they can edit / set up their business. Once that's done they have an option to create an offer.

Person/User: If the user registered as a Person they then will be greeted with a page will all the offers available at the time. When they click on the offer a description of the offer will come up with the location and time of the pickup.



Flow Chart


Action/Interaction
User Interface
Decision
Login Page
Registered?
yes
no
Log In
Register
Register Page
User Type
Person
Business
Logged in as Person
Logged In as Business
Main Page With Offers to choose from
Business Page with option to create offers
Sign Out
Reserve Offer
Create Offer
Clicks on Offer
Exit
Exit
Offer Page
Create Offer Page
Clicks on Reserve
Entered Offer Details
Offer Reserved
Offer Created

Description

Write a brief description (20 to 100 words) to every node in your diagram and explain what information is given in this particular part and which other nodes the user can reach from there.

Login Page: if user is registered then they can log in using their email and password. If the are not registered they can register with an email and password.

Main Page: If the logged in person is of type user, then the main page of offers available will be shown to them. If they are a business, then their business page will be shown with option to edit their page and manage their offers including creating new offers.

Offer Page: If a user decides to click on an offer, they like from the main page they will be greeted with the Offer page which will have the description , price and location of the pickup. They then can reserve the offer. once the offer is reserved, they can then exit the offer and look for other products they would like to purchase.

Create Offer Page: If a business decides that they want to create a new offer then they will be greeted with the create offer page where they can upload a picture , set the price , location and details of the offer. Once the offer is created then they can exit back to the main business page.

5. Design and Development Process

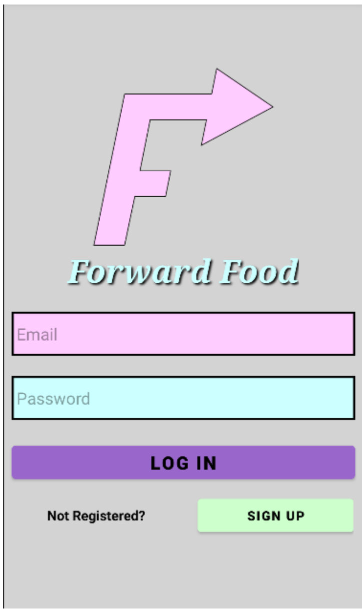
Getting Started

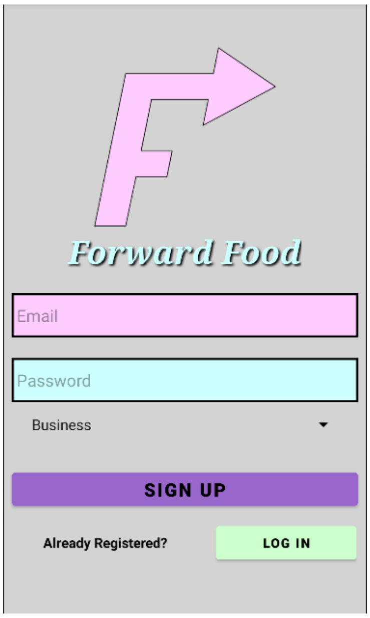
After having decided what app we were going to make it was time to set it up. So, we created a Figma Storyboard and an Android Studio Project. We used the storyboard to get a better understanding of how our app will look like and what design features will make our app more user friendly. After agreeing on design features such as colour schemes and layouts , we made a GitHub repository that we uploaded our Android Studio Project to. Here everyone made their own branch so that we could later merge the branches once it had been approved by the group.

FireBase – Authentication

After having our GitHub repository set up , we decided to each implement one thing that we had previously agreed upon in one of our group meetings. I decided to do authentication and the database for this app , as around this time we had already had a look at Firebase in one of our labs . So, I set about creating a project on Firebase which would be responsible for authentication and storing the user's data. Once that was done, I linked the Firebase project to the Android Studio project. The first thing I worked on was the authentication as this would be the initial step every user would have to go through. I set up the login page shown in the lab slides just to confirm that everything was in working order and that I was able to log in and register. Although the included log in page worked great there was no way to edit the layout to our teams liking so I decided to make new login and register activities.

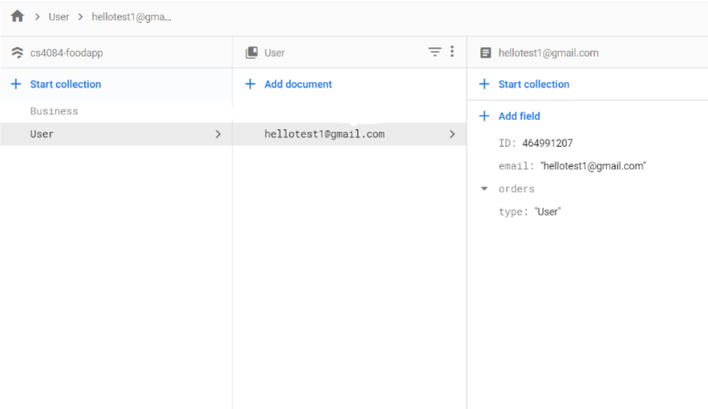
With the new login and register pages we now had a customised look to our project. At this time, I also added a spinner in the sign-up window where the user that is registering would have to choose between Business and Customer. When the user clicks on the Sign-Up button they are signed up and based on their type they are then saved into the appropriate collection within our database.



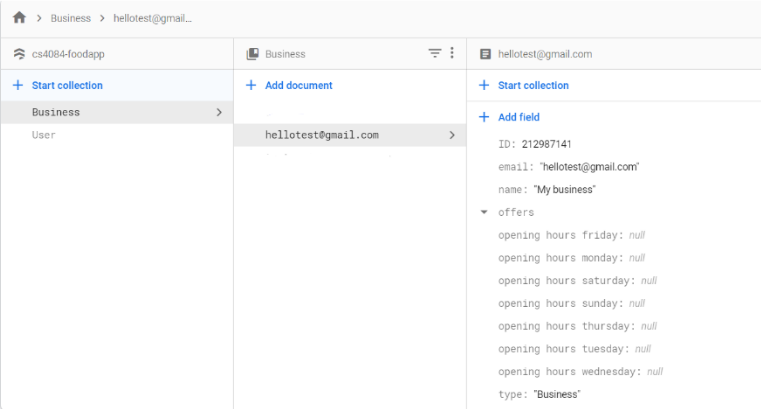


FireBase – FireStore Database

This brings me onto the next part of my development process which was the database. Once the user registered successfully, I would create a document with the document ID being their email. this was it was easy to find any user as we could just search for a document that matches their email in the collection. Upon successful registration it would also create field in the users document relevant to their type

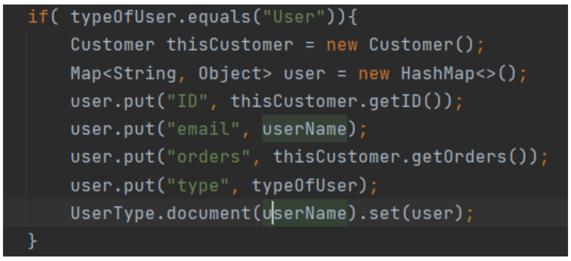


ID: 464991207
email: "hellotest1@gmail.com"
orders
type: "User"

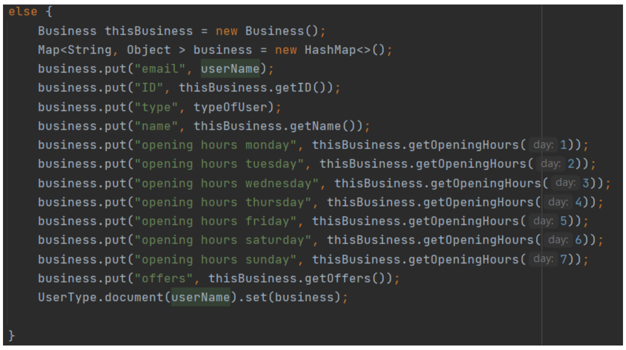


ID: 212987141
email: "hellotest@gmail.com"
name: "My business"
offers
opening hours friday: null
opening hours monday: null
opening hours saturday: null
opening hours sunday: null
opening hours thursday: null
opening hours tuesday: null
opening hours wednesday: null
type: "Business"

Above are examples of how these users' data would be stored after successful registration. This was achieved by either creating an instance of a Customer or a Business then using their set / get methods to save them into their specific documents.



```
if( typeOfUser.equals("User")){
    Customer thisCustomer = new Customer();
    Map<String, Object> user = new HashMap<>();
    user.put("ID", thisCustomer.getID());
    user.put("email", userName);
    user.put("orders", thisCustomer.getOrders());
    user.put("type", typeOfUser);
    UserType.document(userName).set(user);
}
```



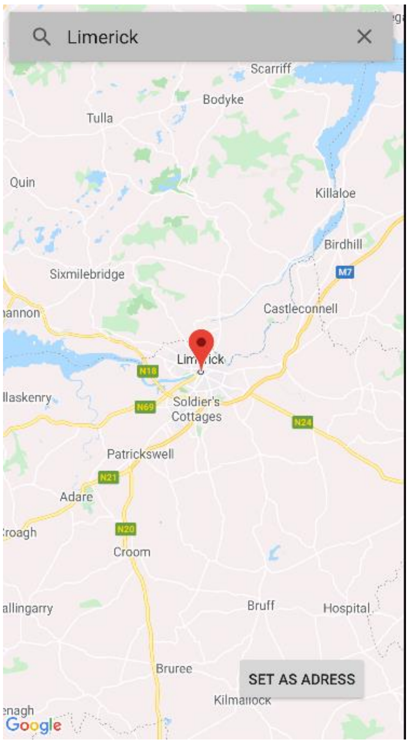
```
else {
    Business thisBusiness = new Business();
    Map<String, Object > business = new HashMap<>();
    business.put("email", userName);
    business.put("ID", thisBusiness.getID());
    business.put("type", typeOfUser);
    business.put("name", thisBusiness.getName());
    business.put("opening hours monday", thisBusiness.getOpeningHours( day: 1));
    business.put("opening hours tuesday", thisBusiness.getOpeningHours( day: 2));
    business.put("opening hours wednesday", thisBusiness.getOpeningHours( day: 3));
    business.put("opening hours thursday", thisBusiness.getOpeningHours( day: 4));
    business.put("opening hours friday", thisBusiness.getOpeningHours( day: 5));
    business.put("opening hours saturday", thisBusiness.getOpeningHours( day: 6));
    business.put("opening hours sunday", thisBusiness.getOpeningHours( day: 7));
    business.put("offers", thisBusiness.getOffers());
    UserType.document(userName).set(business);

}
```

Later on, I would work on our FireBaseLink class which was responsible for getting user and business data from the database some of these methods included:

```
getUserType() // gets user type from database
getCurrentBusiness() //gets all data from currently logged in business
updateBusiness() //updates a businesses page / information
getCurrenCusotmer() //gets all data from currently logged in customer
updateCustomer() //updates customer data
bookOffer() // books an offer (only works when userType is that of customer)
createOffer() //creates an offer ( only works when userType is that of business )
```

Google Maps

while working on the Firebaselink class I also decided to set up the Google Maps API for our project. First thing I had to do was a set up a Google Console Developer account , then I had to enable maps SDK within the account. After enabling google play services SDK tool in Android Studio I created a project in Google Cloud Platform , now all I that was left was to generate an API key and add that to the project's android manifest. Initially I had only a simple working example of google maps which just included a marker placed at Sidney. After looking at the lab examples provided at other ones I could find I thought it would be best if we had a search view so businesses could search and save their addresses as well. Then these addresses will be displayed when a user is looking at an offer.



My Other Developmental Processes & Development in General

Things I was tasked with other than the ones listed above includes , was working on the Business and Customer classes.

Other processes which were given to other team member included:

- Designing the UI – Jason
- Implementing Offer and Business functionality – Ericu

- Implementing Bookings and Main page – Kerry

Designing the UI

Jason was tasked with designing the User Interface as he had previous experience with User interface design.

Implementing Offer and Business functionality

Ericu was tasked with this as it was basically the core of our project, and he has had the most experience with java therefore he seemed like the best person for this task.

Implementing Bookings and Main page

Kerry was tasked with this part of the project as she had previous experience with java and also a good understanding of what the main page and booking are expected to look like.

Conclusion and Reflection

During the development of this app, I feel like I've learned quite a lot of thing about FireBase , Google Maps API and Android Studio in general. Some problems I've encountered was relating to the database specifically reading and writing data to documents. During the early stages of development, I would have trouble reading and writing some data types like integers or structures like lists. I've overcame this by doing research and reading forums and documentations , which in most cases fixed my issue.

Although we did have weekly meetings as a group, one thing that was hard for us was time management. This was because we were all doing different courses which had different modules therefore it was hard for us to spend the same amount of time on the project at any given moment. Although on an individual level we had managed to do all the parts we had originally wanted to(this includes Google maps , basic classes , Authentication , UI elements , The Database) due to time constraints, we could not finish implementing every part of the project we had previously planned on. Therefore, if given the opportunity I would like to finish implementing every feature.

video demo: https://youtu.be/ve38FwVBkog