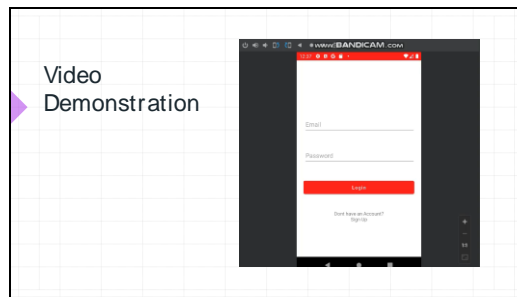


Slide 1

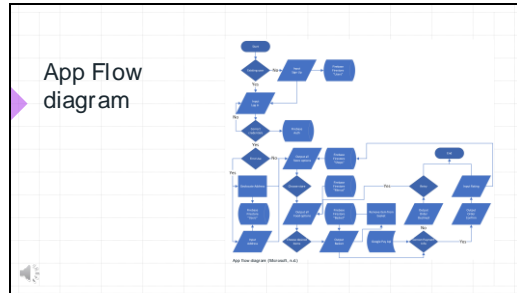


Hello this is the presentation for cmp 309 software development for mobile devices. The application im presenting is called foodhub. FoodHub is a food delivery app. it provides a selection of restaurants the user can choose from and place orders with. The application allows the user to pay for the order prior to it being accepted thus resulting in the user only needing to wait for the order to arrive, allowing them to enjoy their order faster. The application makes use of a database storing all required information.

Slide 2

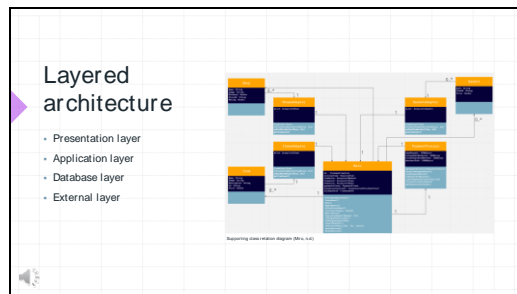


Slide 3



As seen from the video demonstration and the app flow diagram the application follows a linear path. This was chosen to allow the user to learn the application with more ease compared to an application that is structured with multiple nested paths.

Slide 4



The architecture the application followed was layered architecture. This architecture was mainly chosen for the efficient implementation and efficient testing this architecture provides. This choice is also justified by the nature of the drawbacks that come with this architecture. Since most of the drawbacks affect larger scale projects the above-mentioned drawbacks will not be as significant as the advantages provided. The layers of this application consist of. The presentation layer, which is mainly responsible for all the possible interactions the user may have with the application otherwise known as the user interface. The application layer, responsible for all functions that aid in handling processes within the application. The database layer where all data is handled. And the External layer consisting of any API or library used. The reduced dependency resulting from this architecture allowed for the development of each layer to be separate thus allowing for each layer to be easily tested by itself and once each layer was deemed to be working as desired, integrated with

the rest of the application with relative ease.

Slide 5

Authentication and new user Sign Up

- Firebase Authentication (Google, 2019b)

```
firebase.auth().signInWithEmailAndPassword(email, password).then(() => {  
  // Success! User is now signed in.  
}).catch((error) => {  
  // Handle the error here.  
});
```
- Firebase Firestore (Google, 2019a)

```
const user = firebase.auth().currentUser;  
const collectionRef = firebase.firestore().collection('users');  
collectionRef.doc(user.uid).set({  
  // Add user data here.  
});
```

The following are the key features of the application. Firstly, authentication of the user. The principal libraries used are Firebase authentication and Firebase firestore. Firebase authentication allows for the creation of accounts for users using the application and requires an email and a password. Once the account is created, the information is stored in the firebase database where the password is automatically encrypted and hashed. Using the `currentUser.uid` we can assign the value of the user id to a variable and input it in as the document id in the Firebase Firestore database to link the account to any additional information needed to be stored in the database. The Firebase Firestore library provides efficient CRUD operations for the manipulation of documents and collections in the firestore database.

Slide 6

Getting a user's address

```

// Location API (Google, 2020a)
addSuccessListener { location: Location? ->
    if (location != null) {
        //Get results and input the in the converter
        val latitude = location.latitude
        val longitude = location.longitude
        val timestamp = converter(latitude, longitude)
        return timestamp(timestamp)
    }
}

+ Geocoding API (Google, 2020b)
private fun convert(latitude: Double?, longitude: Double?) : String {
    //Create a geocoder
    val geocoder = Geocoder(context, Locale.getDefault())
    val timestamp = geocoder.fromCoordinates(latitude, longitude, timestamp)
    return timestamp.getLatitude() : String?
}

```

Another key feature of the application is the possibility of getting a user's address automatically under the condition that the user gives the application permission to access their precise location. The Google Play location API is used to get a location object. The object retrieved allows for the latitude and longitude to be extracted and assigned to values. Using those values the Geocoding API is then used to make that conversion from coordinates to address. Once the address is retrieved in the form of string it is shown to the user for validation

Slide 7

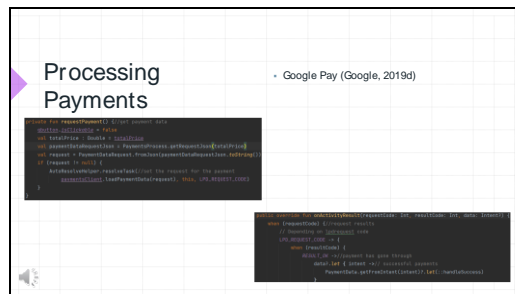
Recycler view with Firestore

[illegible]

The use of the recycler view library is crucial in the process of creating the graphical user interface that the user can interact with. It is used on three instances. Firstly, when displaying all store options. Secondly when displaying the menu items of each restaurant. And finally when displaying the contents of the users basket. In all of these cases recycler view was populated with data from the Firestore database. Each document in the database is instantiated as an object. All objects are then added to a list of objects and then each corresponding recycler view adaptor binds each said objects to the corresponding fields of the layout. Resulting in scrollable lists populated with the corresponding values retrieved from the database that the user can view or interact with. Finally, in certain cases images were used for the user interface. To achieve that the images themselves were stored in

Firebase Storage and the link to the images was contained within the other values in the firestore database. The link is then converted again to an image using Bitmap Factory

Slide 8



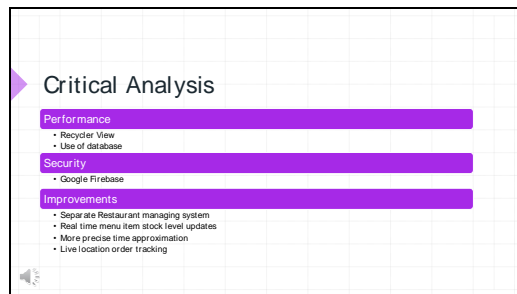
Finally google pay was used to allow for the processing of payments. Google pay API requires as an input a set of JSON objects strings and arrays which allow for the configuration of the payment system to match the preferences of the merchant. Once all the information is gathered a request to the API is made and based on the response certain functions were created to reflect the outcome of the payment process. In the second example we can see how success is determined and what function handles it.

Slide 9



To summarize, FoodHub is an application that allows the user to order food. This process is achieved through the combination of multiple smaller features that interact with each other and allow for the user to navigate through the application with relative ease without requiring significant prior knowledge. When interacting with each other the features have been designed to not leave any room for miscommunication within each feature by making data exchanges strictly for required data. This decision was made to minimize any potential bugs or errors and minimize any usability issues that these exchanges might encounter. The general usability of the app was considered throughout the entire process of development. Firstly, the user interface consists of elements that were designed with simplicity in mind focusing mostly on making the interface as simple to understand and learn as possible. Additionally, all inputs required by the application are checked for invalid input type which means that if the user either with malicious intent or by mistake inputs any values that might generate an error when the app is running the system will alert the user to modify what they inputted. This is a significant feature for an application due to the fact that it prevents many errors from occurring, making the usability of the system smoother. The compatibility of the app was considered when developing the application and thus the minimum sdk used was API level 29 meaning the application will be able to run on approximately 68% of android devices.

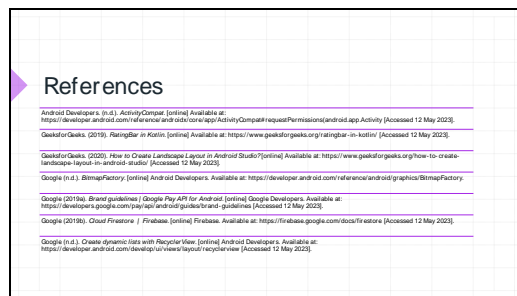
Slide 10



Throughout the entire development process performance was a significant concern. The use of recycler view allows the reuse of layout configurations allowing for smoother performance. Additionally, since all elements of data storage are done through databases no device storage is affected this is the case for most hardware resources. Lastly security was a significant concern during the development of the project. To increase the security measures were taken such as using the Firebase Database for all handling and storing of data. This decision was made due to the fact that the database is not SQL based therefore is not vulnerable to SQL injections. However, there is still room for improving the application. Firstly, one significant improvement would be the implementation of a system that allows the store to manage all the orders received, allowing for easier completion of work that has to be done as well as easier management of all daily processes. Furthermore, the above-mentioned system could additionally receive data from a database containing all stock

levels of the corresponding store thus allowing the system to automatically label items in the menu as out of stock. Finally, another improvement would be the implementation of an additional section of the application that the rider would use, this section would calculate the location of the rider and allow for a more precise order time estimation or even precise location tracking of a clients order.

Slide 11



Slide 12

