**Introduction:**

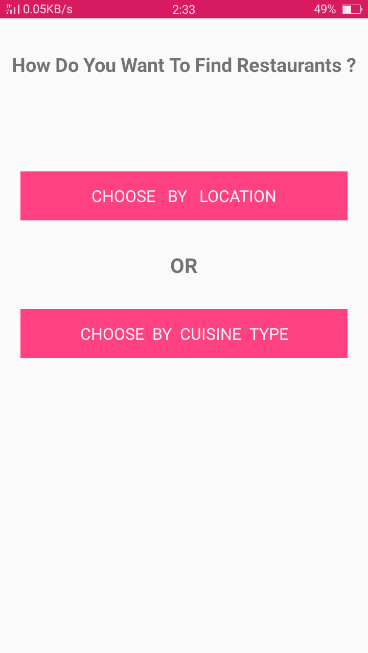
The main goal of this app was to make a hassle-free way of finding the best restaurant for users and make a smart reservation system.

Choosing a restaurant to eat can be a hassle especially if you don’t know what you are in the mood for or if you don’t have time and want to know what’s the best restaurant near you.

This is where Table4Me comes in. You can now find the best restaurant for you based on location or cuisine type and not only that Table4Me has a booking system that makes it very easy for you to see the restaurants sitting plan and reserve a table of your choice and actually see the tables that were already reserved.

With Table4Me restaurant owners can easily add their restaurants after getting their account approved and these restaurants will be shown to users based on their preferences. The restaurant owners will also get info about the reservation they receive from users.

There were several main objectives to developing the app. Image uploading and compression, making connection with a back end server, location determination for nearby feature, and finally making a centralized database on the back end server. Following is a sample screenshot of a stable version of the app :



**Research:**

Before developing the application, it was clear that the development required research for making various features of the app. This includes determining the restaurants that were going to be in search space, determining the types of restaurants and dividing them into categories, adding restaurants and their info in centralized database, location of restaurants, how reservations were going to be made and what information was needed to make reservations.

////////// Include the Rest from the previous Report

**App Design**

**Overall Design Aims**

The app’s UI was made from ideas by the aims and objectives of the project and constructed from the results of the research and analysis keeping in mind the android material design and simple user interface. The main features determined were:

* Determining what information was needed for Adding restaurant in our Firebase Database and how that info was going to be structured in the database.
* Storing and compressing Restaurant image and displaying that image to user over a HTTP connection from Firebase STORAGE.
* Making restaurants available to users in categories.
* Showing restaurants to users based on GPS/location.
* Allowing users to make a reservation for selected restaurant.
* Showing user’s the restaurant seating plan, the tables available, reserving the table of his/her choice and then booking those tables and notifying admin/vendor.

All the above mentioned was done keeping in mind the Google Material Design and using some external libraries such as Picasso, Firebase UI etc.

**User app Functional requirements.**

|  |  |
| --- | --- |
| Req.ID | Requirements Description |
| FR.01 | The app must allow user to show all the restaurants nearby their **location**. |
| FR.02 | The app must allow user to choose their favourite cuisine type. |
| FR.04 | The app must allow user to choose a **time** for when they want to book a table. |
| FR.05 | The app must allow user to give an option to choose how many people they would want to book table for. |
| FR.06 | The app must allow user to display availability of the tables inside the restaurant. |
| FR.07 | The app must allow user to pick the table of their choice and reserve it, only if they are available. |
|  | The app must allow users to enter any comments they have regarding the booking, in the ‘comment box’. |
| FR.08 | The app must display the contact details and other information e.g. address; of the restaurant if the users wish to contact the restaurant directly. |
| FR.08 | The app must allow users to enter their name, email and their phone number if the restaurant needed to contact them. |
| FR0.09 | The app must automatically send user a text and an email to saying that the booking has been booked. |

|  |  |
| --- | --- |
| Req.ID | Non-Functional Requirements Description |
| NFR.01 | **The app must load up information quickly.** |
| NFR.02 | **The app must look good and interactive.** |
| NFR.03 | **The app must work on all android mobile devices.** |
| NFR.04 | **The app must list all restaurant in specific areas.** |
| NRF.05 | **The app must allow users to book tables at any time and from anywhere.** |
| NRF.06 | **The app must allow admins to see the bookings from anywhere at any time.** |

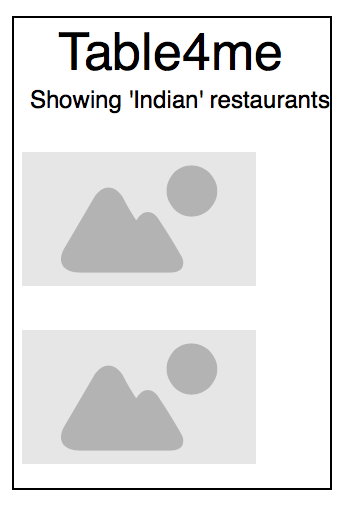
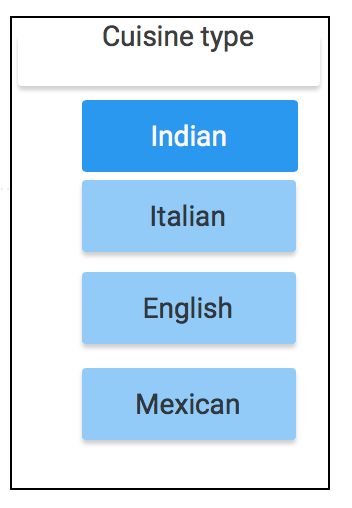
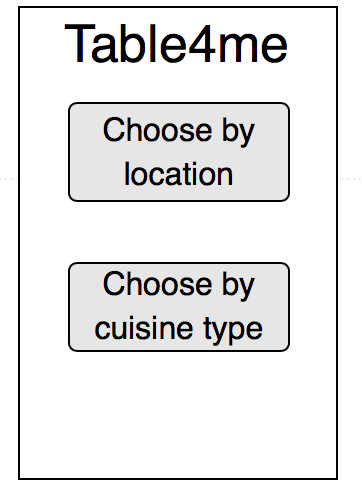
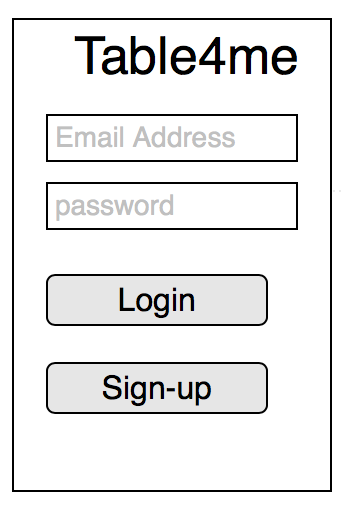
|  |  |
| --- | --- |
| **The user app will use firebase and have 7 screens.** | |
| **Home Screen** | * **Displays App Logo** * **Asks user for permission to use their location** |
| **Lists of Restaurants screen** | * **Displays options of different cuisine types of restaurants e.g. Indian, Chinese ,Italian etc.** * **Allows user to select the cuisine type they want to book a table for.** * **Displays all lists of restaurant nearby the location chosen by users.** * **Allows users to select the restaurant and automatically takes the user to Reservation times page.** |
| **Available Reservation times/date**  **e/number of peoples Screen** | * **Displays the availability times for that amount of bookings.** * **Allows users to select the time and date that they wish to reserve for.** * **Back button to go back if they wish to change the restaurant.** |
| **Restaurant Tables Available Screen** | * **Displays the table plans inside of the restaurant** * **Displays green and red for each tables; green for available to reserve and ‘red’ for table already reserved.** * **wouldn’t allow the users to select the tables that has been already reserved.** * **Would allow the users to select the available tables.** * **Offers the option which will notify the users if the table gets cancelled.** |
| **Reservation Confirm screen** | * **Displays the summary of the booking before clicking confirm button.** * **Option to clear the reservation and start a new one.** * **Confirm button to confirm the existing booking.** * **Allow users to enter their Phone number and email address for booking confirmation.** |
| **Firebase** | * **Maintains Real-time Database.** * **Makes connection Between the Restaurant app and Users app.** * **Updates the Tables availability after the reservation has been made or cancelled.** * **Uses cloud messaging to send emails for confirmation of bookings.** |

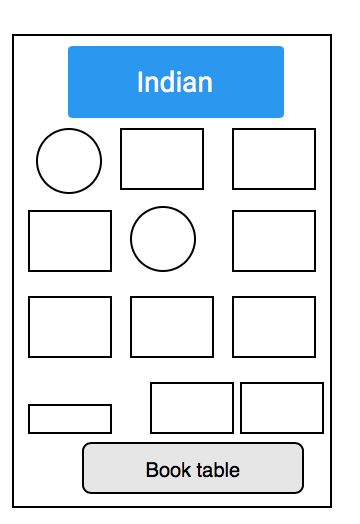
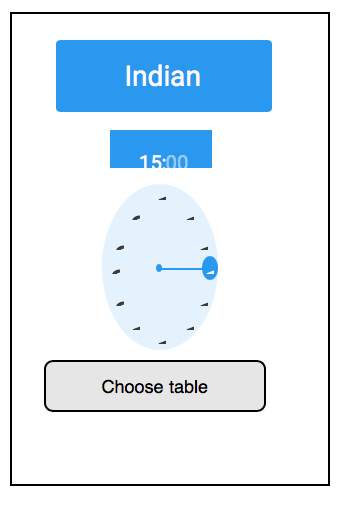
**Admin App Functional requirements**

|  |  |
| --- | --- |
| **The Admin app will use firebase and have 3 screens.** | |
| **Home Screen** | * **Allows admins to enter username and password.** * **Offers retry if the username or password is incorrect.** |
| **Your restaurant screen.** | * **Displays the tables that has been reserved as Green.** * **Displays the tables that aren’t reserved as Red.** * **Allow admins to go to previous and future bookings dates.** * **Allows admins to touch the reserved tables and see if any comments are left from the customers** * **Allow admins to manage and move tables around interactively.** * **Automatically updates with notification if the tables are reserved or cancelled.** |
| **Admin Account settings screen** | * **Allows admins to change, update their account details.** * **Allows admins to log out of their account.** |
| **Firebase** | * **Uses real-time database.** * **Makes connection within the Restaurant app and Users app.** * **Updates the Tables availability after the reservation has been made or cancelled.** * **Uses notifications to alert Admins.** * **Allows admins to send an confirmation email/text messages.** * **Uses cloud messaging to send emails for confirmation of bookings.** |

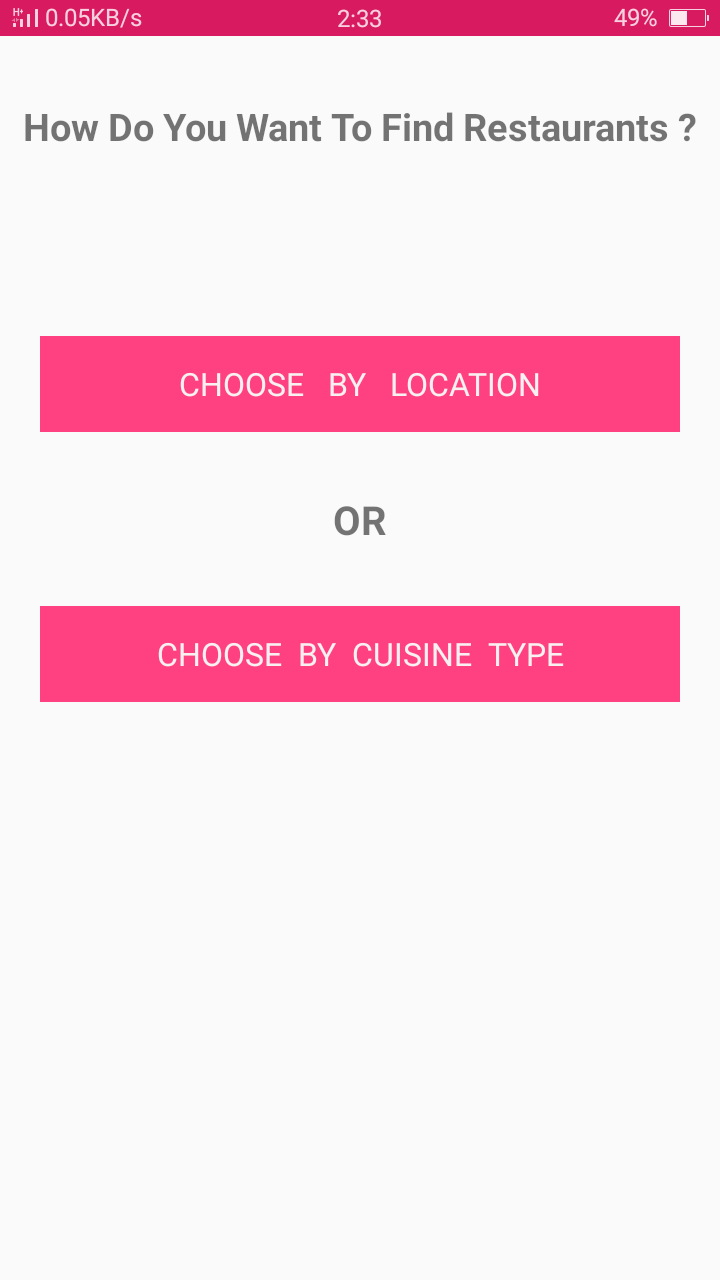
|  |  |
| --- | --- |
| Req.ID | Requirements Description |
| FR.10 | The App must allow admins to log in and log out of their accounts. |
| FR.11 | The app must allow admin to check the bookings made by the users. |
| FR.12 | The app must allow admin to show the users email and phone number. |
| FR.13 | The app must allow admin to edit and updates the tables. |
| FR.14 | The app must allow admin to whether accept the reservation or cancel it. |
| FR.15 | The app must allow admin to input the bookings manually. (bookings which aren’t made through the app). |

**USER INTERFACE DESIGN**

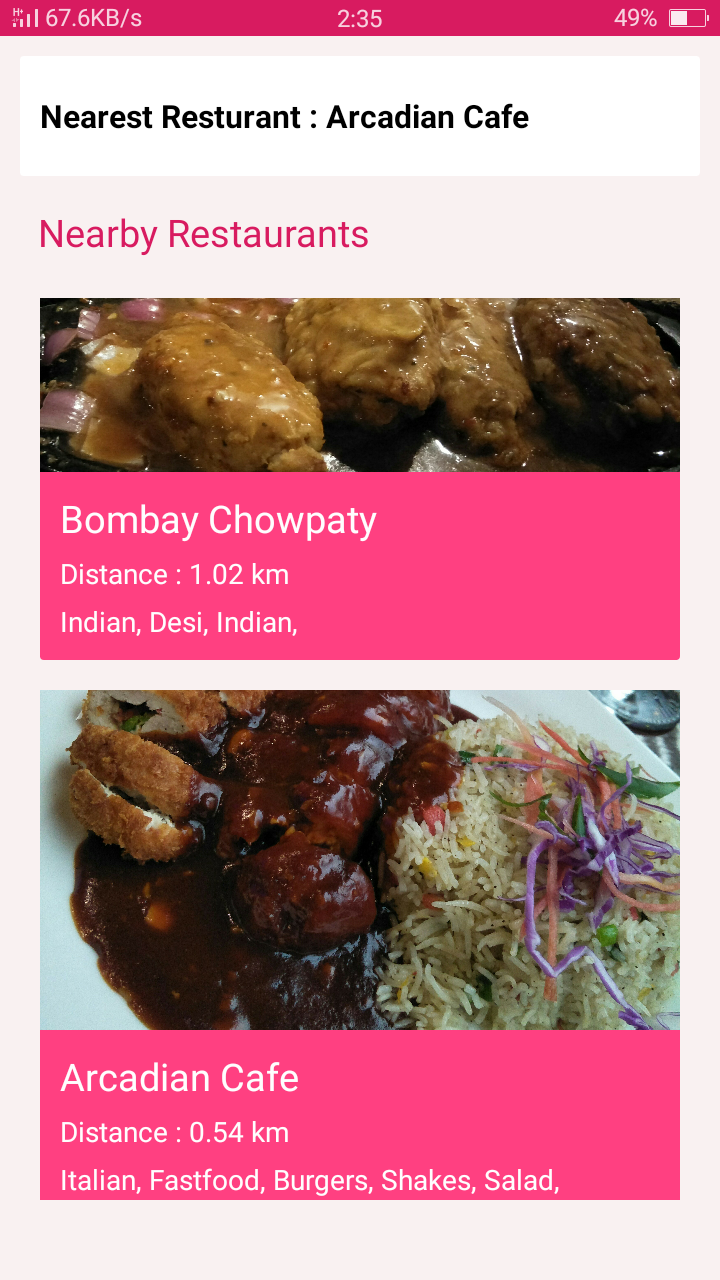




The app aims to give users a responsive, minimal and easy to use Interface using the latest google material design tools.



**Main Screen**



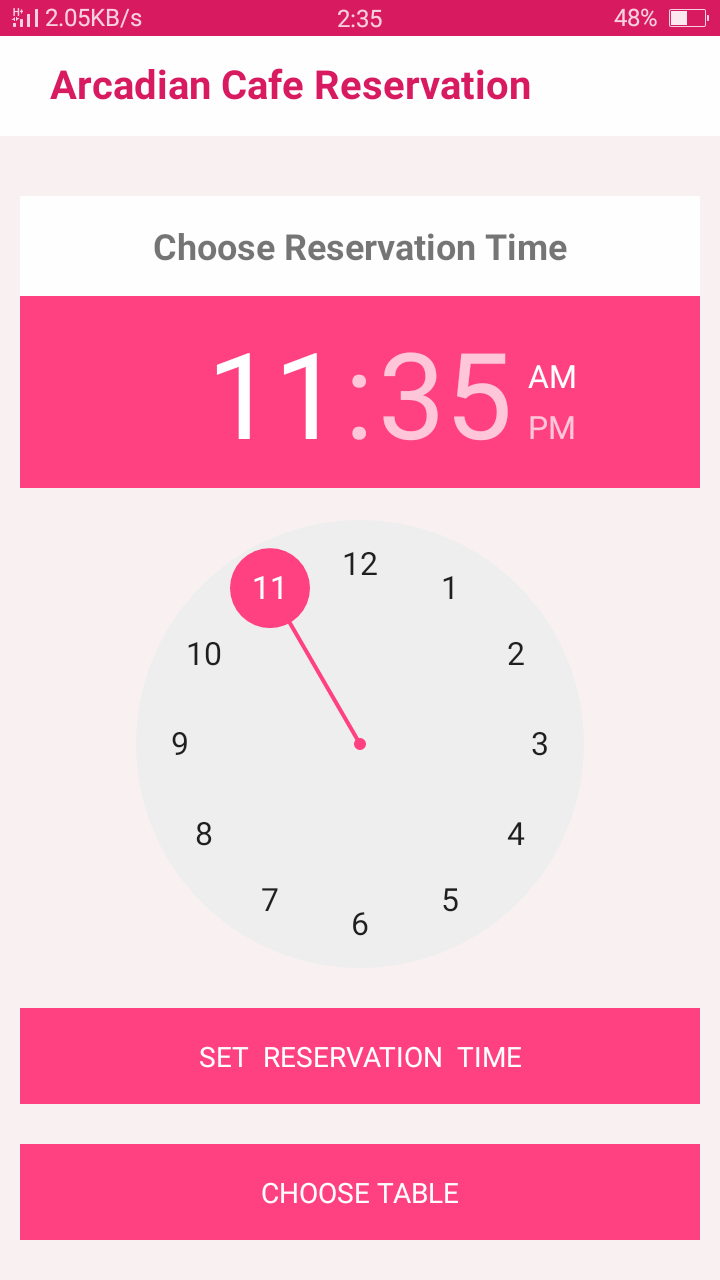
**Find Restaurants By Location Screen**

In the beginning the location screen did not show images of the restaurant and also didn’t show the exact distance of restaurant from users current location both these issues were resolved after some reviews from people.



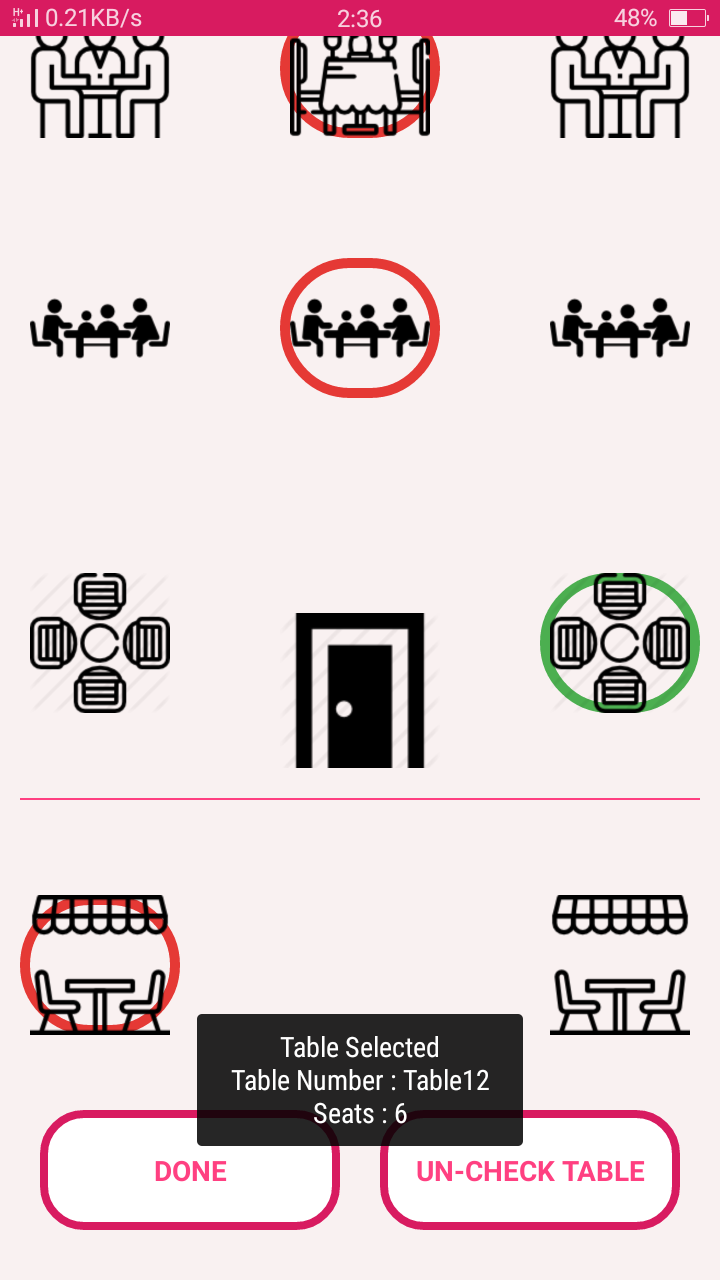
Find Restaurants By Cuisine Activity

In the initial design the find restaurants by cuisine used a recycler view but that was removed because that was just unnecessary and it resulted in a lot of extra code. Now all the design is in XML.



**Reservation Activity**

Initially the reservation activity didn’t set a reservation date that was later fixed and the time had to be set with a button after selecting it rom widget both these issues were resolved.

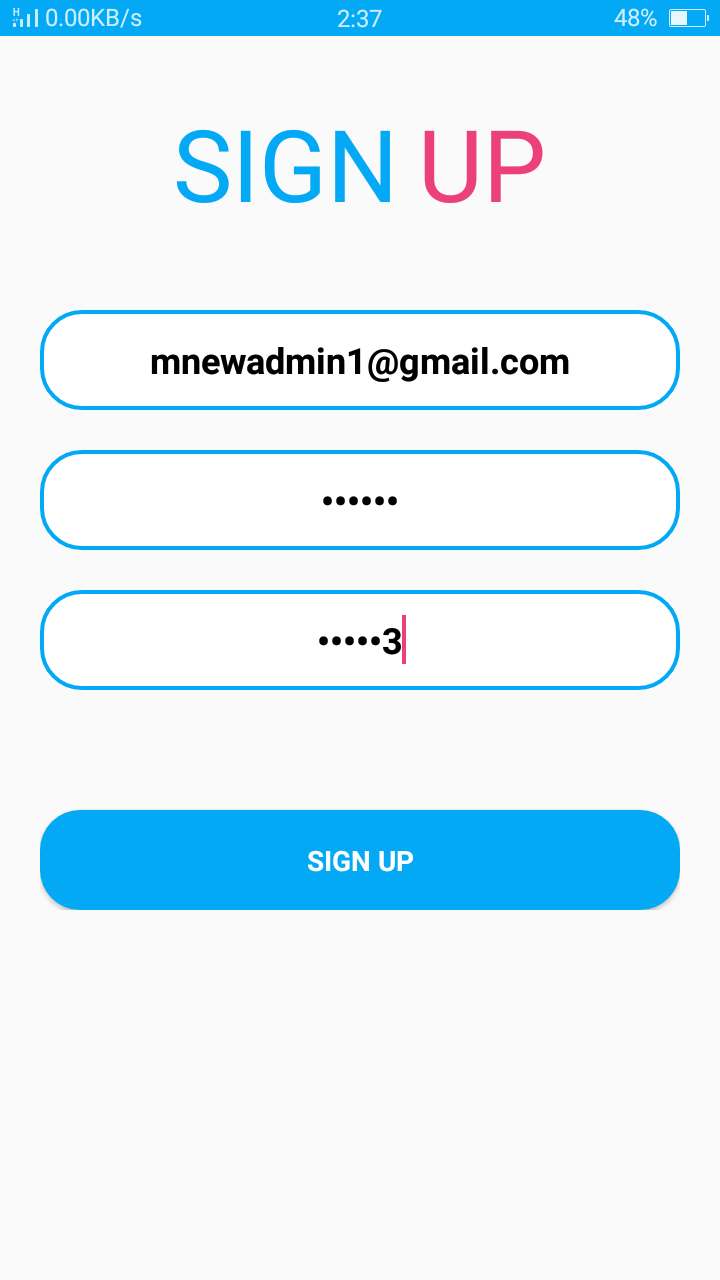


**Reserve Table of Choice Activity**

Initially the table reservation activity consisted of an Image and had an invincible checkbox that was shown when user tapped on that area to show table as selected but this was a problem because this did not took into account different screen sizes and it was really untidy.

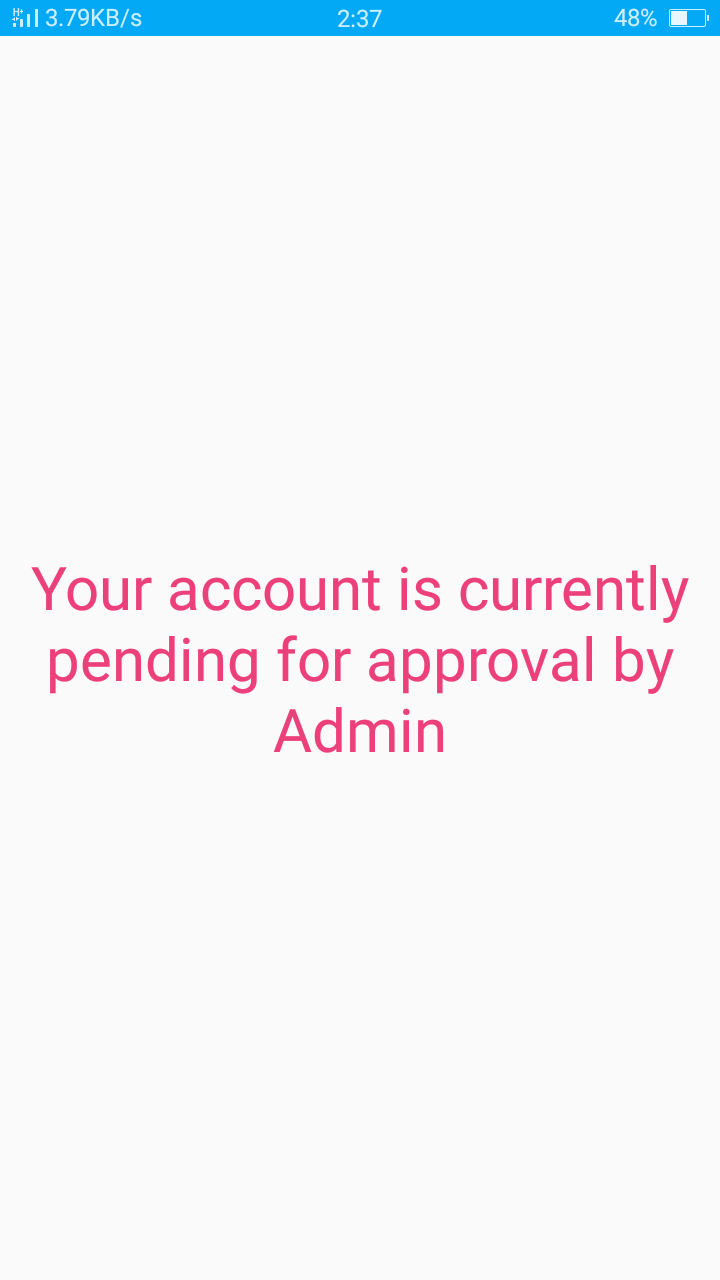
Now the app uses PNG images from drawable to show each individual table and a custom made circle shape drawable when table is reserved or if a table is already reserved

**Admin App UI Design**

****

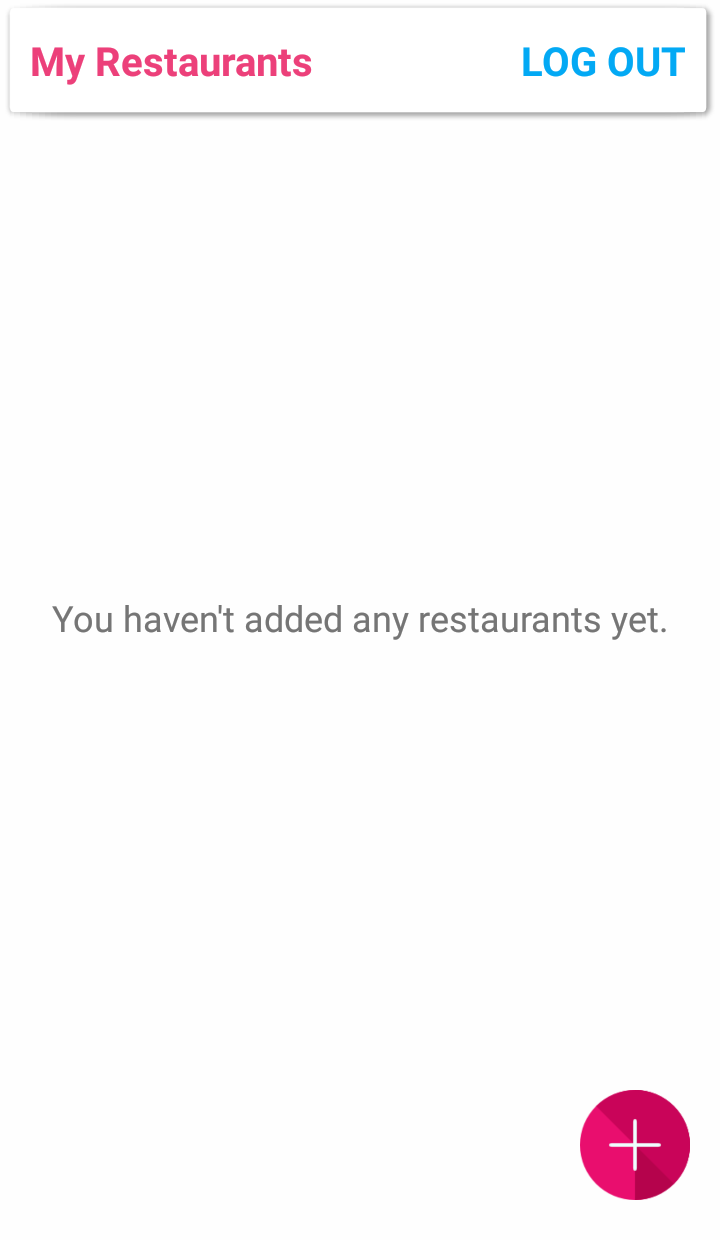
**Admin Sign Up Screen**

The admin signup screen uses Firebase Authentication to sign up a restaurant owner.



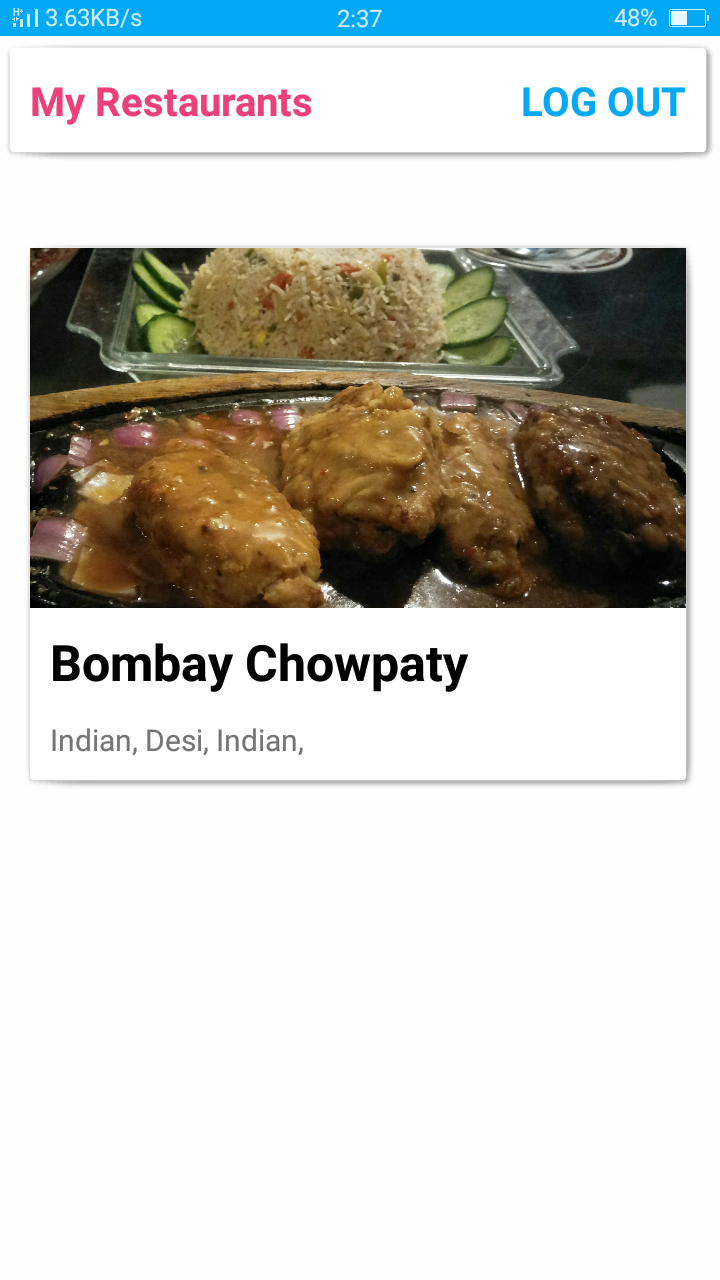
**Admin Account Approval**

Admin Cannot by pass this screen until his account is verified from Firebase Console. This screen was initially not included but after some consideration it was clear that this screen was necessary to make sure that verified restaurants are made available for users.



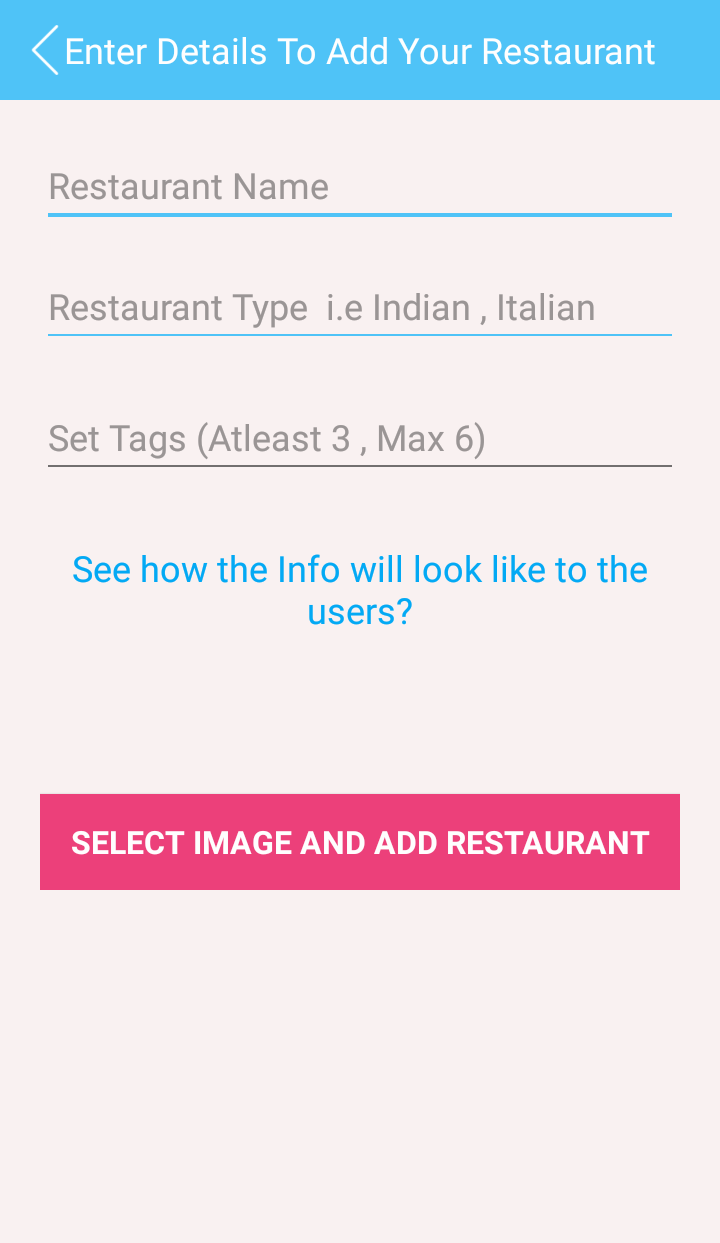
**Admin Home Screen**

This scree shows admin his restaurant



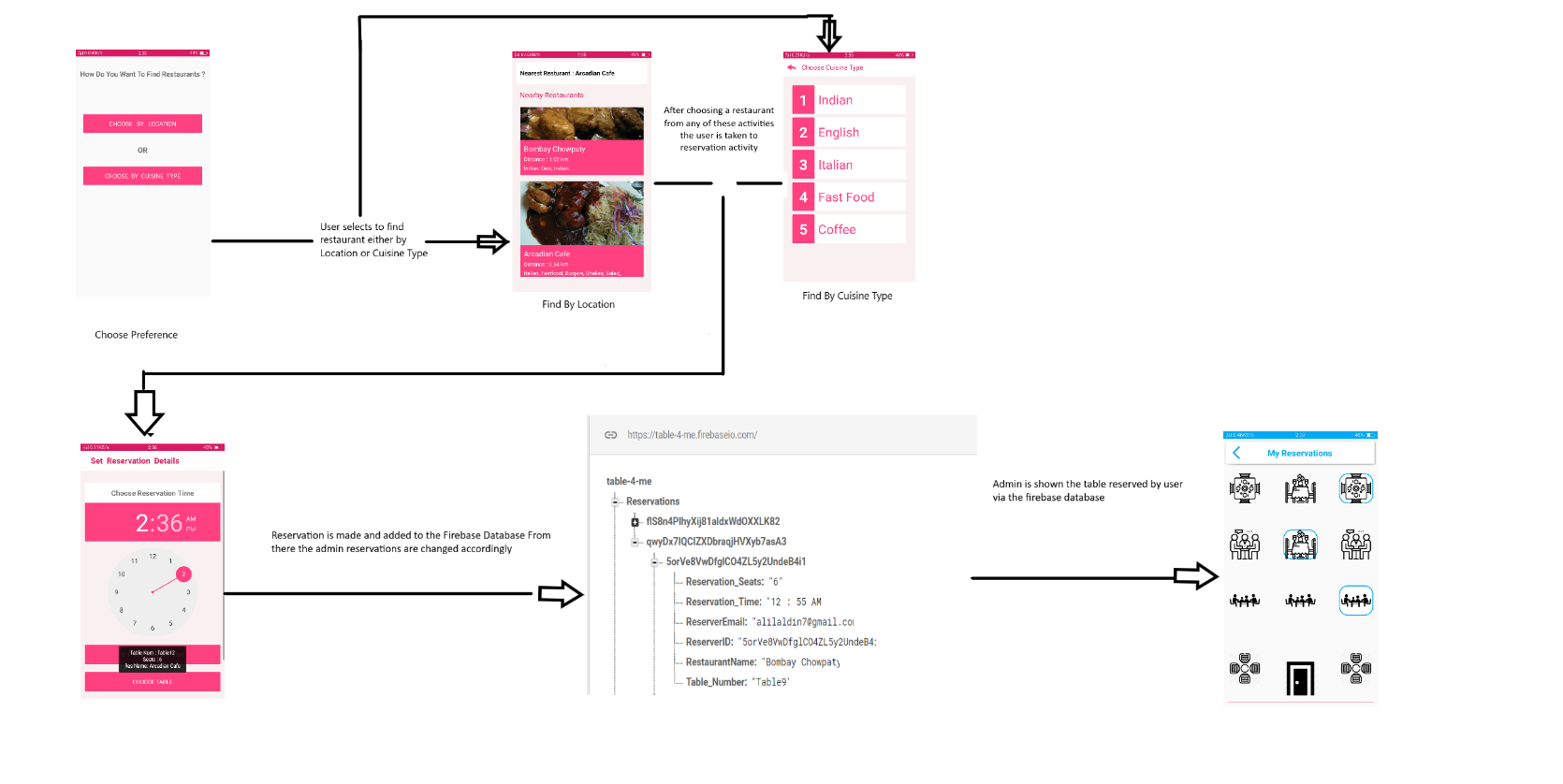
**Admin Home Screen**

(After Adding Restaurant)



**Admin Add New Restaurant Activity**

Auto Complete Text View was included in this screen to make sure that tags were set correctly.



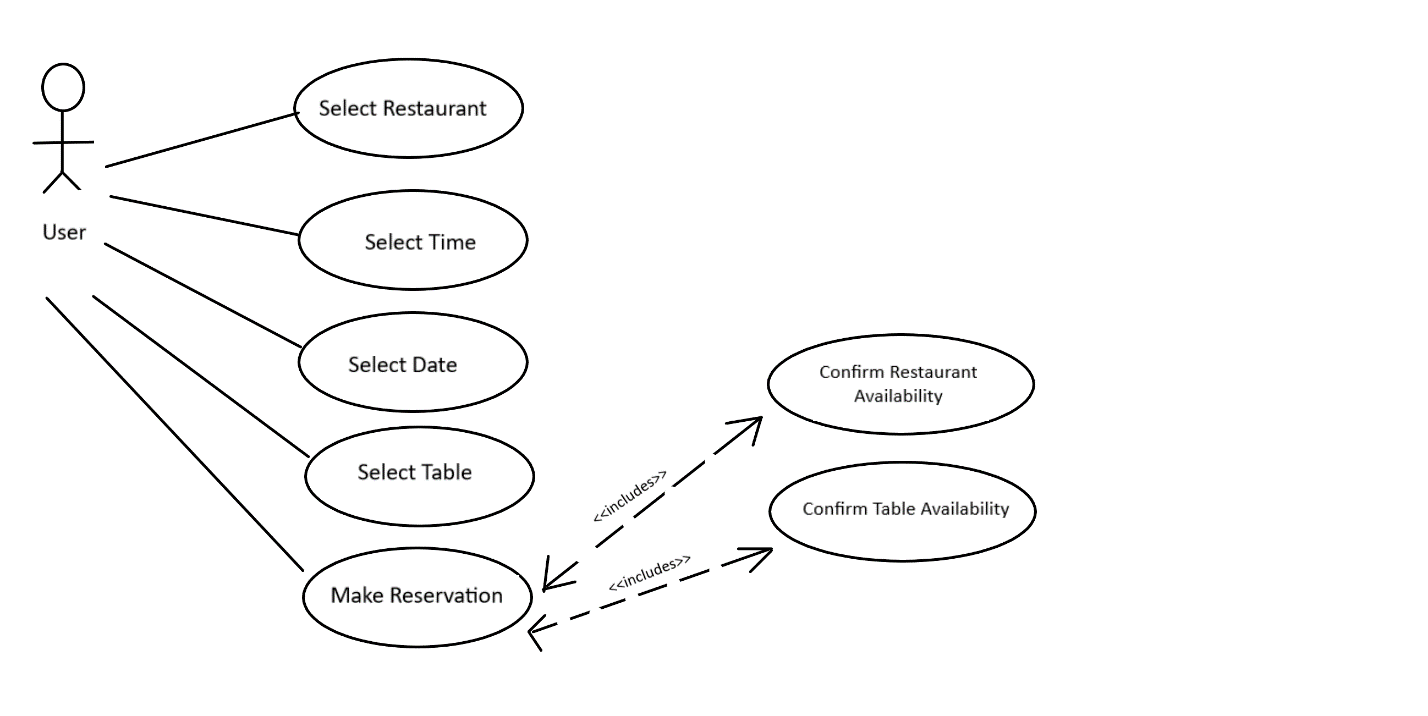
**Application Interaction Flow Diagram**

In the beginning there were two separate activities after for selection of restaurant by location and Cuisine type that was removed and user is navigated to the same Reservation activity.

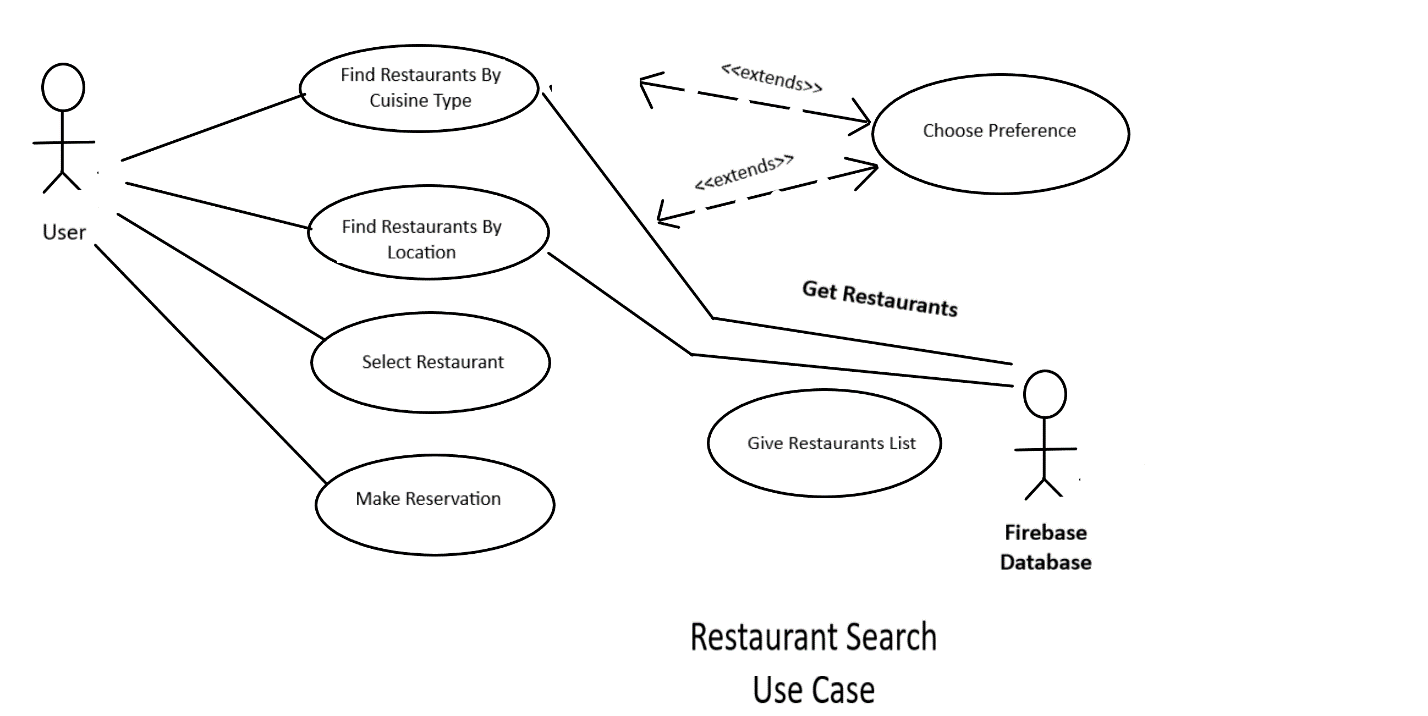
This also helped a lot to easily insert correct and meaning data into the Firebase Database. The Restaurant Id play a key role in determining the reservations and that was a good thing that these parameters were included from the start.

**USE CASES**

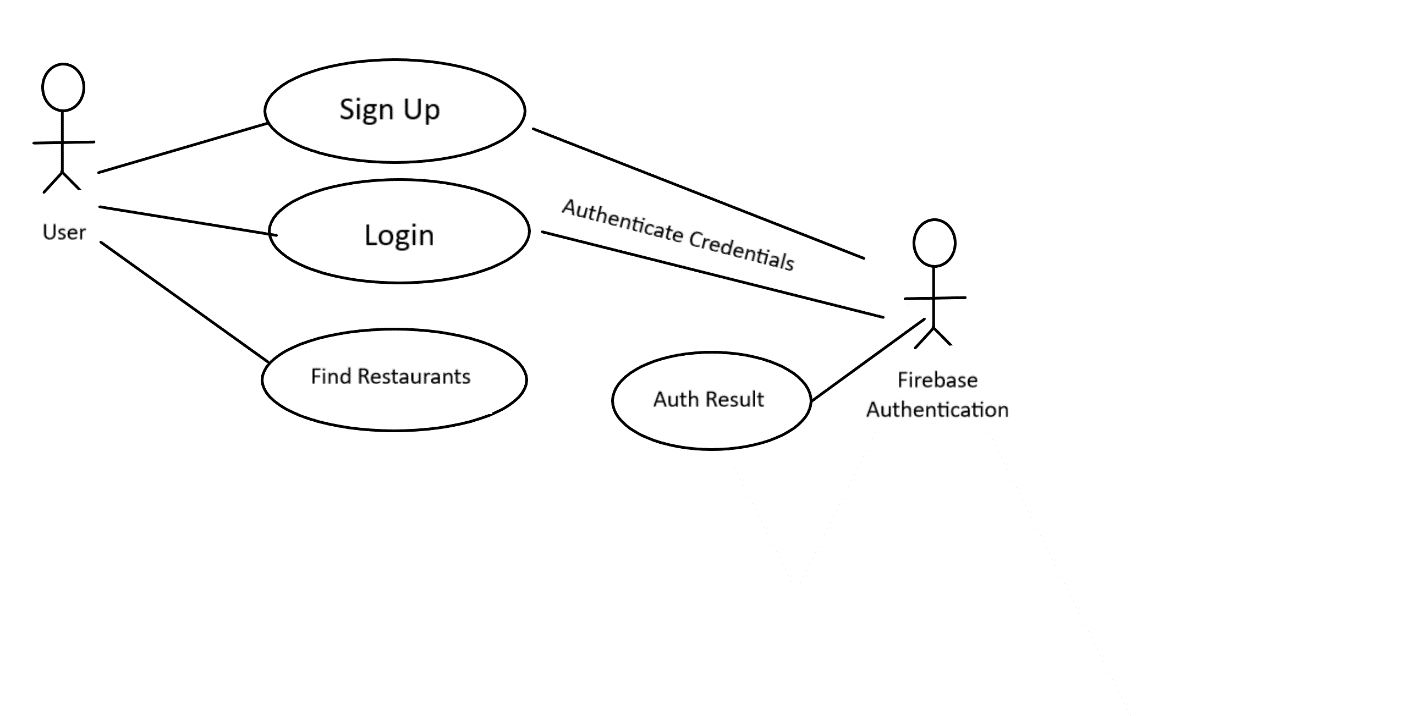
**Reservation Use Case**



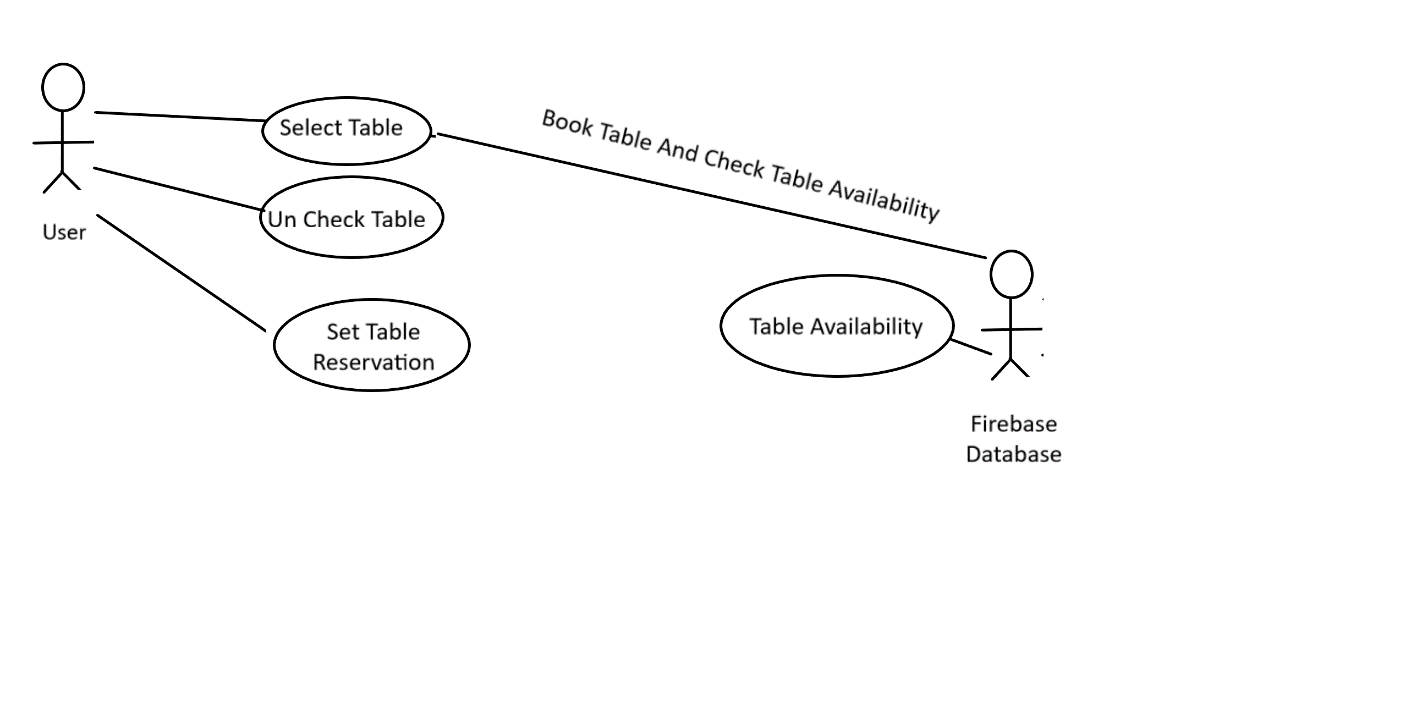
Restaurant Search Use Case



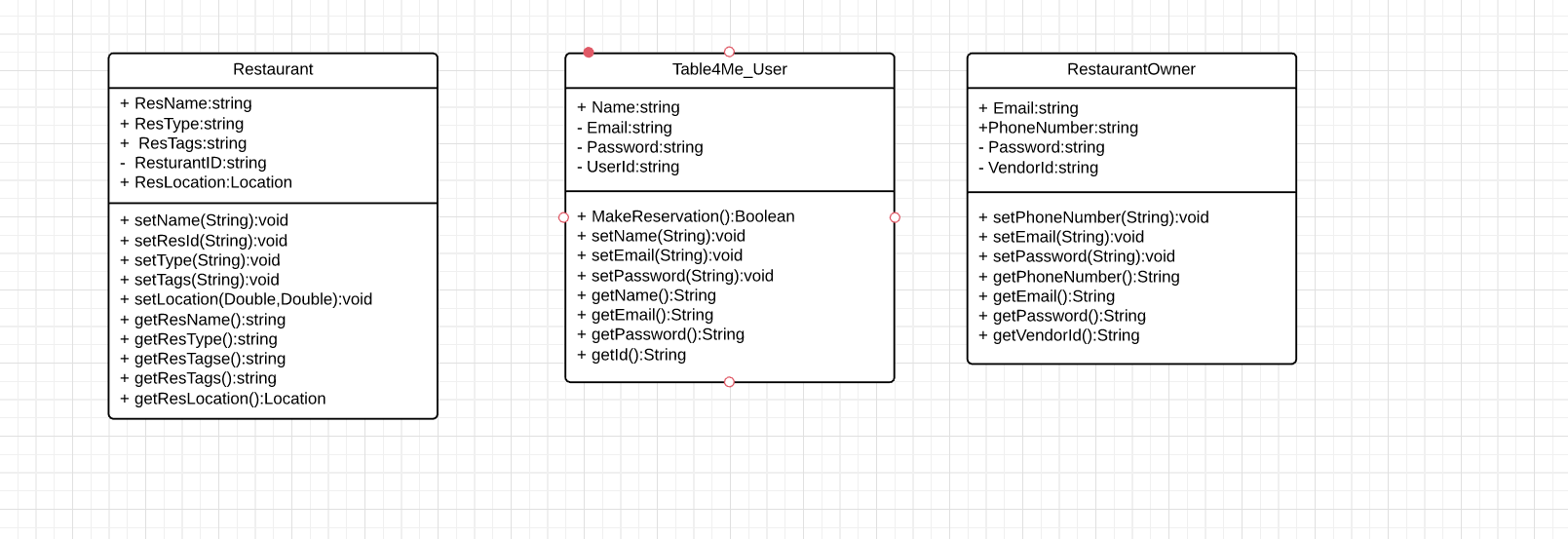
**User Authentication Use Case**



**Table Select Use Case**



CLASS DIAGRAM



Login and Signup were kept separate from the Main Classes and hence there was no problems from the start in keeping the different modules of the app separate from each other.

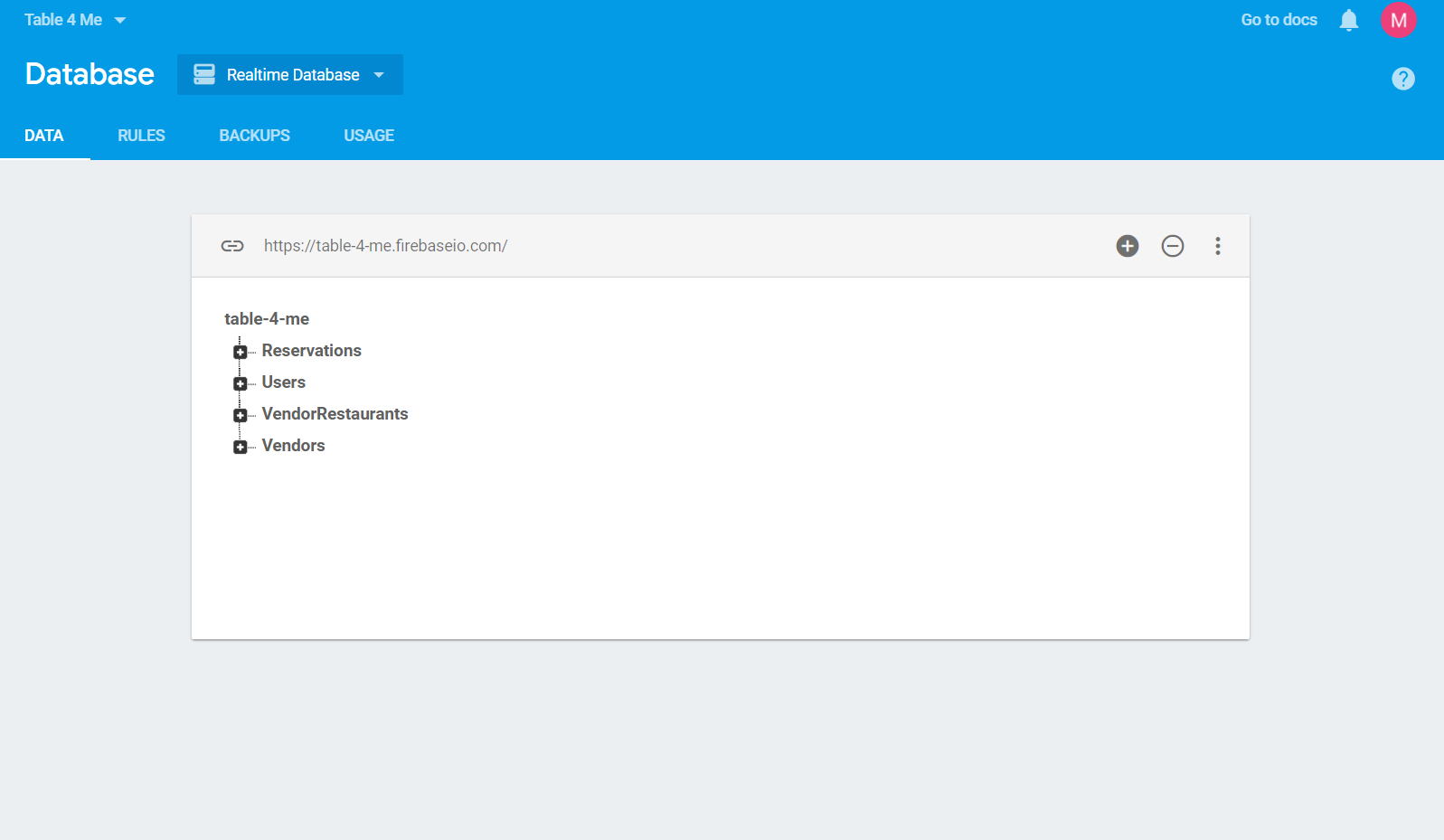
Since the User and Admin App were kept separate therefore they both have separate Classes and the only thing they have in common is the Restaurant Info class. Restaurant Owner adds his restaurant and that info is received in the user app.

**ER Diagram:**

For this project Firebase Database was used and since the Firebase Database uses Un Structured Data Sorting instead of Tables therefor it does not have any relationships between entities.

Database stores info in the form of nodes and each node can have sub child.

Following is the Firebase Database structure of this app :



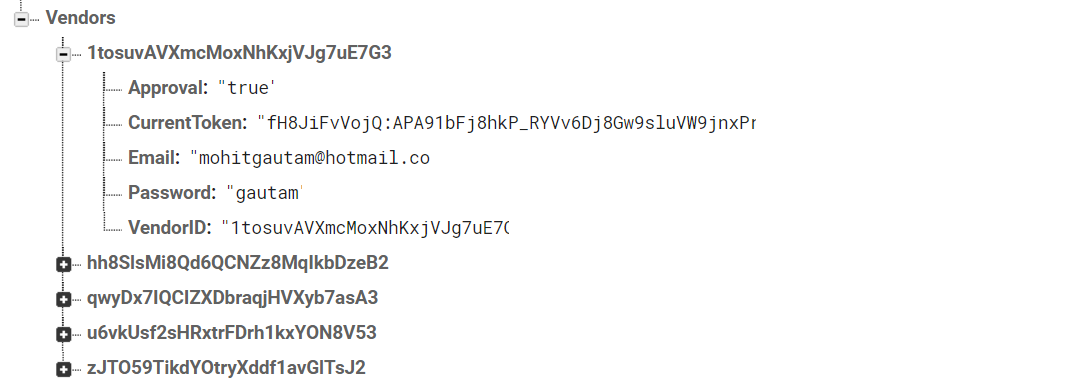
We have 4 nodes in our Database i.e. Users, Vendors, VendorRestaurants, and Reservations.

USERS



Users node has a sub node for each user and the name of each sub node is the ID of that particular user given to that at time of signing up by the Firebase Auth.

Vendors



This node contains info of all the restaurant owners and the name of each sub node is the Restaurant Owners ID which is also the ID of the Restaurant itself.

VendorRestaurant



This node has the info of all the restaurants . This info was taken via the admin app. The location coordinates are added by the Admin i.e. Control Team.

Reservations



This is a very complex structure each node belongs to the restaurant owner so that reservations are separate for all the restaurants. The second sub node is the User Id and this separates all the reservations of user.

**Programming Language and Architecture**

This is a native android application and therefore the programming language used was JAVA because it has the best support for android and it is Object Oriented.

The IDE used was Android Studio because it provides the best development environment and easy Firebase Integration. Use of external libraries and UI design is also very great in android studio.