**STUDENT SIDE OF THE APP**

Screens screenshot of a login screen

Description automatically generated

The first page we get on opening the app is the Login Page, where the user first has to choose their profile, i.e whether they’re logging in as a Student or an Employee, in case they don’t have an account, they can click on the ***Sign Up*** button, which redirects them to the Sign Up page.

Here the User can sign up with their Name, Enrollment Number, email address, and create a password.

On clicking Sign Up, the details are stored in a Firebase Firestore database, which has a collection called users, where each user’s information is saved with the document ID equal to their enrollment number, as can be seen in the screenshot below.

A screenshot of a computer

Description automatically generated

After Signing Up, the user is redirected to the login page, where they can now log into the app, alternatively if the user already has an account, they can directly jump to the Login page by clicking on the ***Login*** button at the bottom.

***On Logging into the App:***

***Home Page:***

***Screens screenshot of a phone

Description automatically generated***

The user first lands on the Home screen, where they have all the services available listed in the form of a scrollable list, the information in this list of Product Cards is being pulled from a collection in the Firestore database, called “products”, which contains all the current available products listed.

Alternatively, the user can search for a particular service from the search bar available at the top.

The bottom navigation bar contains 3 items, the Home Page, that we just described, the Requests page, containing all the current pending requests of the user, and the profile page.

A screenshot of a phone

Description automatically generated

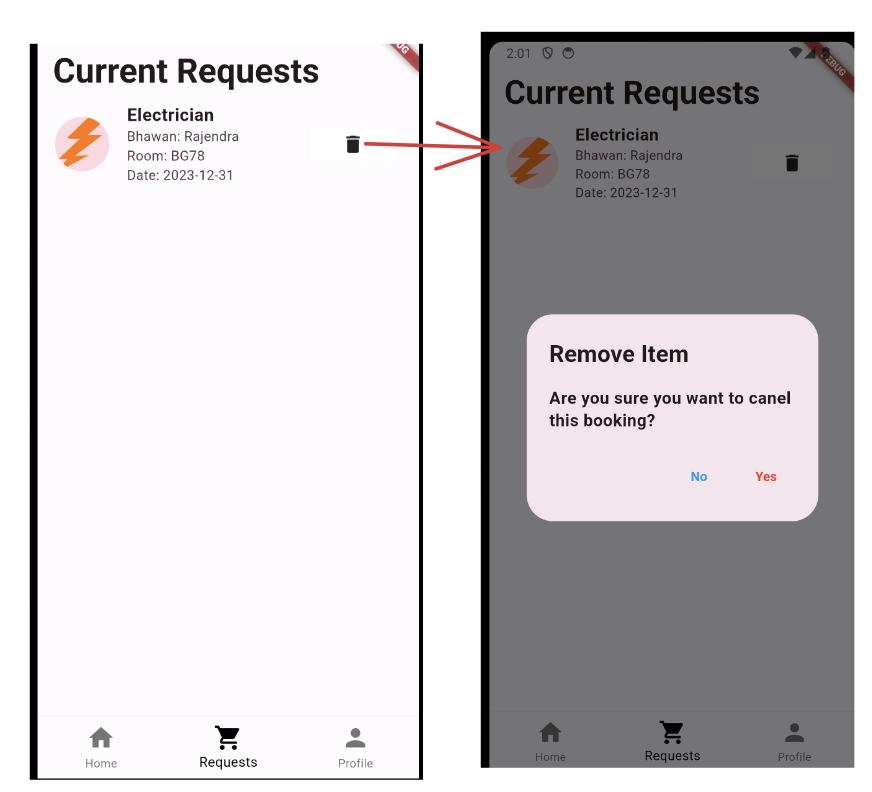
On clicking on any of the product cards, we are redirected to a product details page, which contains some info about the product.

From here, the user can either go back to the home page, or proceed to book the service, by clicking on the ***Book this Service*** button at the bottom, which redirects them to the booking page.

On the booking page, the user is required to enter the booking details, including where they want the service, i.e the hostel and room combination, along with the date of booking.

On picking all three, the user can proceed with the booking by clicking on ***Book it***, internally on clicking the button, this information along with the product information gets added to a field inside the user document, called the “*cart*”, which is a collection of all the current pending requests of the user, also this dictionary, or map in the form of key value pairs gets added to the requests collection of each and every service provider, providing that particular service, in this case it gets added to the requests collection of all the 3 electricians available, and a mail is sent to all three electricians that there has been a booking request made.

***Requests Page***

******

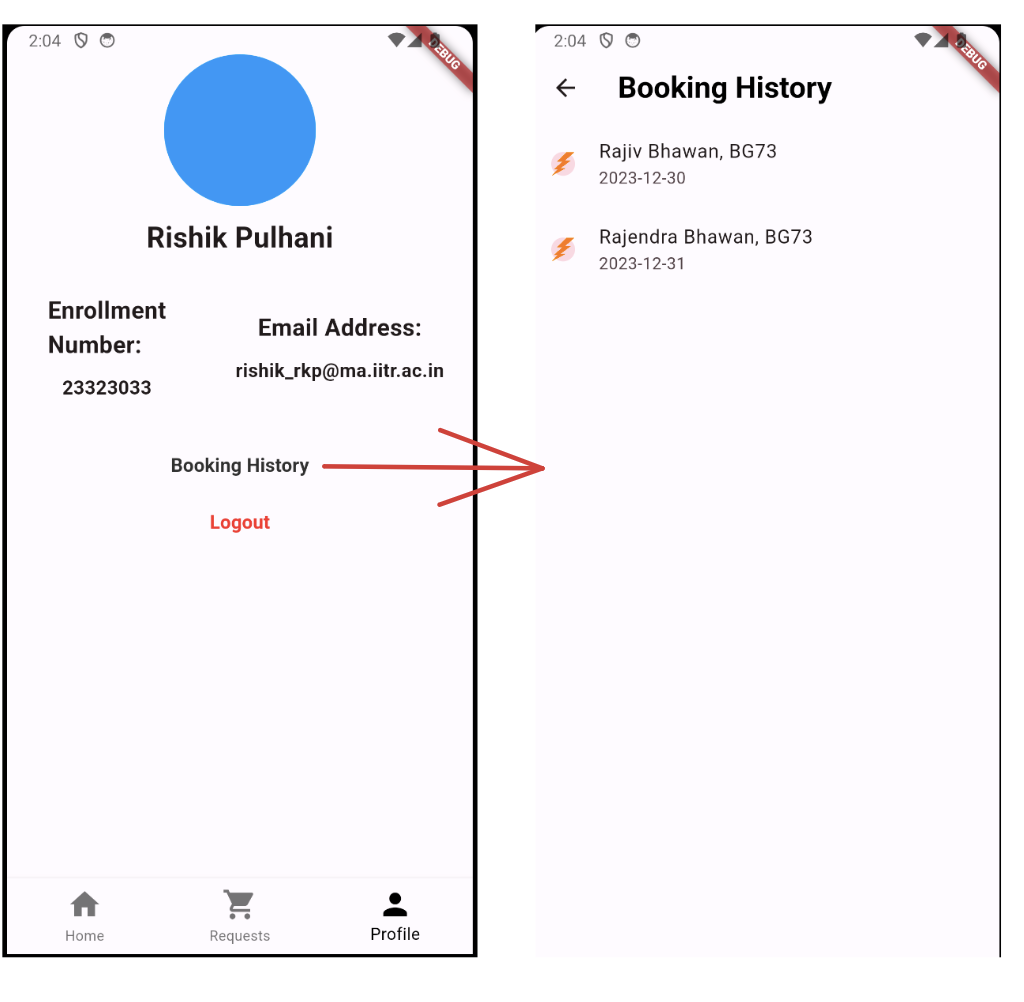
The requests page contains a list of all the current pending requests of the user, this information is obviously pulled from the “*cart”* item of the user, and is displayed using a List view builder in flutter.

Each list element contains information about the booking, and also has a cancel the request button beside it. On clicking this button, the user is shown an alert dialog box, to confirm whether they want to cancel the booking or not.

On clicking yes, the item is removed from the *“cart”* element of the user, and also is removed from the requests collection of every single service provider, in this case, it is removed from the requests collection of all the 3 electricians, and a mail is sent to all 3 saying that the booking request has been cancelled.

That pretty much wraps up the requests page.

***Profile Page***



The profile page contains information about the user, such as their enrollment number, their email address, and their profile picture, which is currently set to null.

There’s also an option to check the past bookings of the user, on clicking which we are redirected to the Booking History page, this page pulls information from the bookings collection inside the user’s document.

The last button present on the app is to Log out of the app.

***All of this pretty much wraps up the Student side of the app***

**EMPLOYEE SIDE OF THE APP**

**A screenshot of a phone

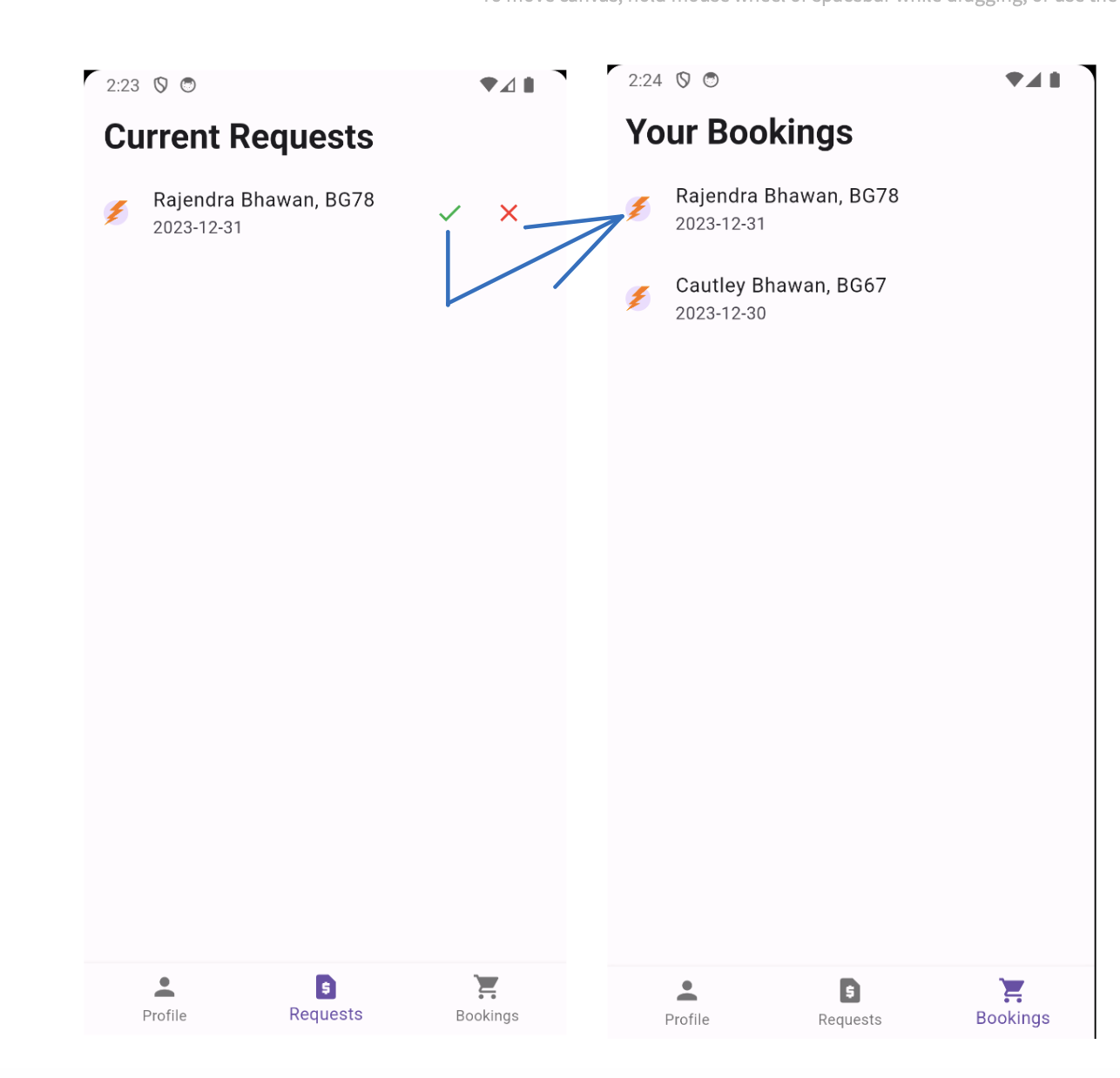
Description automatically generated**

The first page that we land upon on logging in as an employee, the credentials of whom are present inside the *“service\_providers“* collection in the Firestore database

This page contains some basic info about the employee, such as their name, their employee number, and their email address, and an option to log out of the app.

The bottom navigation bar contains three tabs, the Profile, that we just went over, the Requests tab, and the Bookings tab

***REQUESTS PAGE***

******

The requests page contains all the current booking requests of this employee, and requests are added to the requests collection of the employee when some user books a service on their app.

Here the employee is given two options, according to their choice or availability, an option to confirm the booking, denoted by a check mark

On pressing this, the item gets removed from the current requests page and moves to the “*Your Bookings”* page, and a mail is sent to the user who booked the service that their request has been successfully booked, and some details are shared so that the user can contact the service\_provider.

In addition to this, the request then gets removed from the requests page of all other electricians, so that no one else can accept an already accepted booking request.

Also, when the check mark is pressed, the item also moves from the “*cart”* of the user who booked the service to the bookings collection, and is therefore removed from the requests page of the user as well, and goes to the booking history list.

On the other hand, on pressing the cross button, or cancelling this request, the item gets removed from the requests collection, and automatically from the request page as well, and the service\_providers parameter in the user’s cart gets item decremented by one for this item. Additionally, a mail is sent to the user saying that this electrician has cancelled their request, and that they should wait by to see how other electricians respond to their request.

Now if, all the possible electricians decline the request, the service\_providers value becomes 0, and the item is then removed from the user’s cart,

***Bookings Page:*** The bookings page as described above just contains a list of the confirmed booking history of the employee, which is pulled from the bookings collection inside the service\_providers collection documents inside the Firebase database.

***That pretty much wraps up the functionality of this app,***

***In the future, I’m planning on adding a chat service, where the user can chat with the service provider,***

***And a feedback system, where the user can rate the service provider, and other features that I’m planning on adding after this submission.***