PARLOR BEACON

GitHub Username: agarwal.mayank32

PARLOR BEACON

Description

As the name suggests, it is a beacon (search) service for parlors and salons of all types. The main motive of this project is to increase convenience among the masses for a fluid service. They do not have to come to a particular parlor and wait in line for their turn; that's a hassle. Instead through this Android-based application, we intend to aid them in simplifying this experience. Registered customers can book an appointment beforehand and get their time slot fixed for their particular needed service. This will not only save time for the customers but will also help in the smooth working of salons.

Intended User

- Shopkeepers who run parlors or Salons
- People / Customers who want to use salon services.

Features

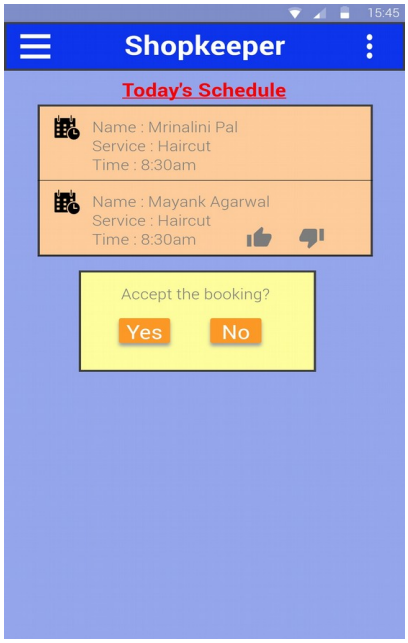
The main features of this app are:

- Keep a record of information of Salons and Customers.
- Allow Shopkeepers to add employee details and salon services.
- Allow Shopkeepers to maintain history of their customers and reviews.
- Allow Shopkeepers to accept or reject customer bookings according to their shop schedules.

- Allow Customers to search and book services according to their needs.
- Allow Customers to give rating and reviews to the shop services provided to them.

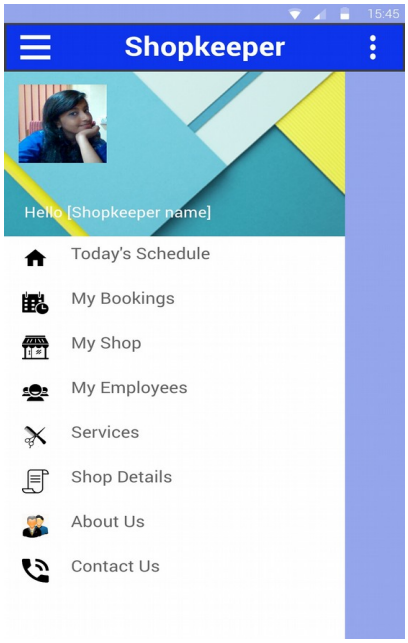
User Interface Mocks

Screen 1



This is the first Page which opens after the shopkeeper has registered/logged in successfully .

Screen 2



This is the navigation drawer menu which can be opened after shopkeeper has successfully logged into the application.

Screen 3

This activity shows the previous bookings to the shopkeeper.

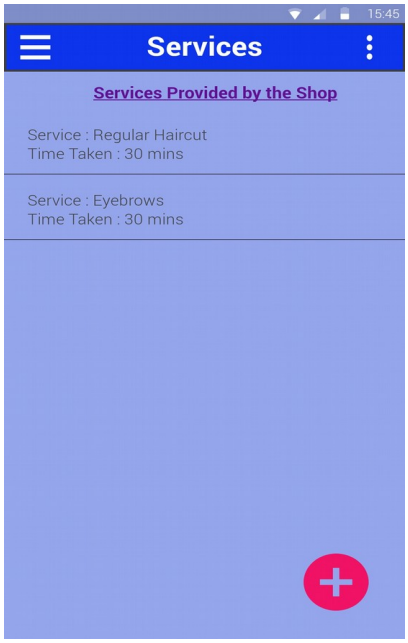
Screen 4

This activity shows all the pictures added by the shopkeeper.

Screen 5

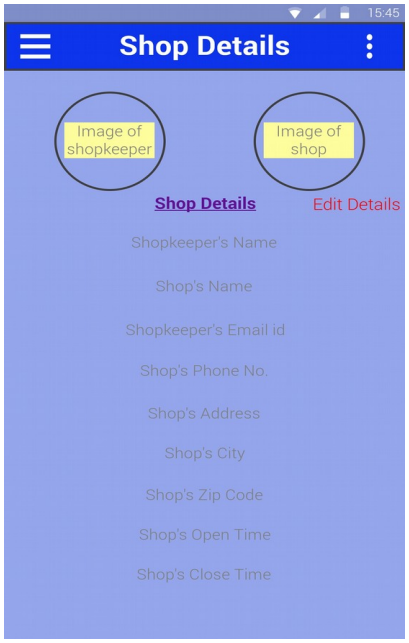
This activity shows the details of the employees working in the shop to the shopkeeper.

Screen 6



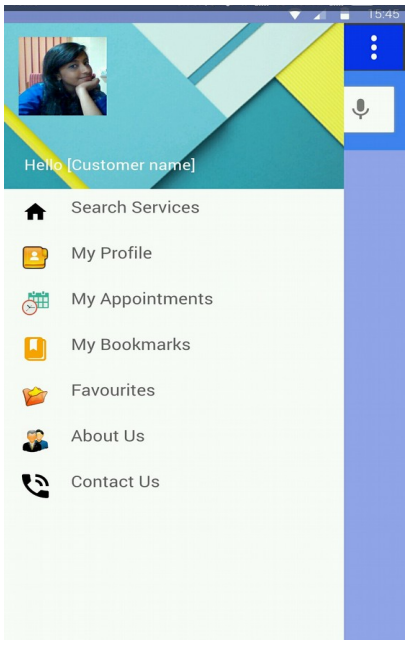
This activity shows the list of services and their details provided by the shop to the shopkeeper.

Screen 7



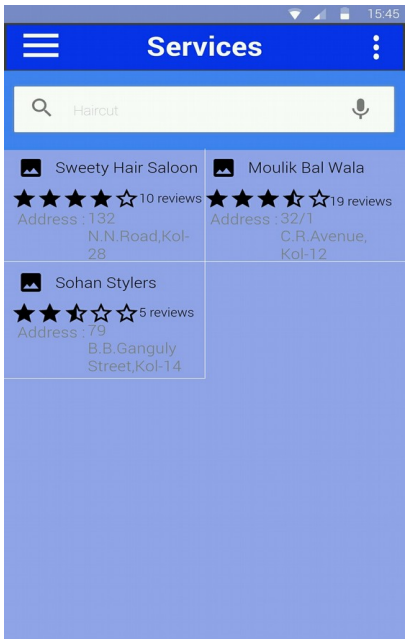
This activity shows the details of the shop to the shopkeeper.

Screen 8



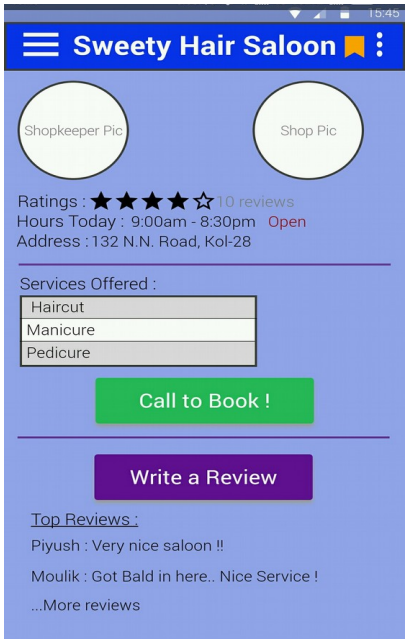
This is the navigation drawer menu which is shown to the customers.

Screen 9



This activity shows the number of shops available to the customer according to his search.

Screen 10

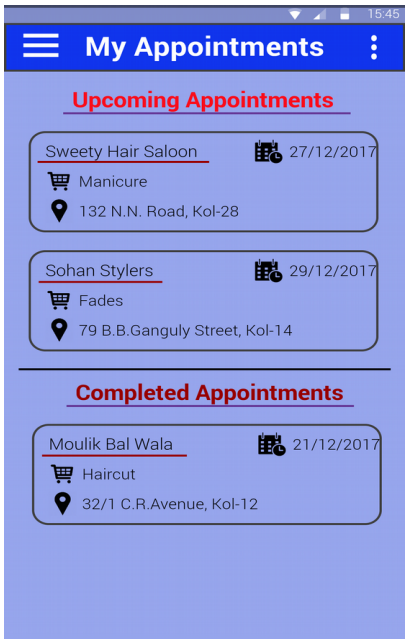


This activity shows the details of the shop selected by the customer.

Screen 11

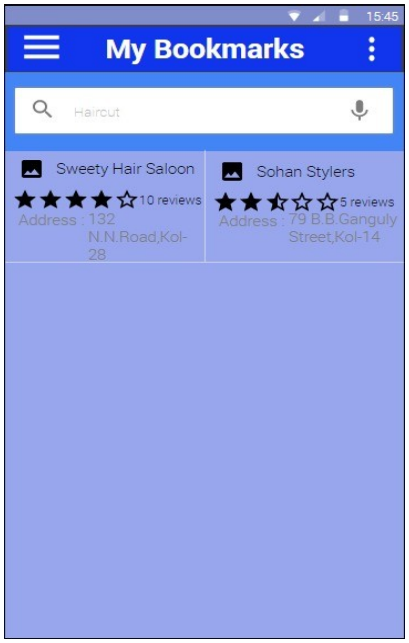
This activity shows personal details of the customer.

Screen 12



This activity shows the previous and upcoming appointments to the customer.

Screen 13



This activity shows the shops bookmarked by the customer.

Screen 14

This activity shows the shops that the customer has visited often.

Key Considerations

How will your app handle data persistence?

App will store data in

- PostrgreSQL Database of server and
- Local SQlite Database using Content Provider.

Describe any corner cases in the UX.

- No Employee data added : Show message
- No Service added : Show Message
- No Picture added : Show Message
- No internet connection : Show Message
- No Bookmarks : Show Message
- No Favourites : Show Message

Describe any libraries you'll be using and share your reasoning for including them.

The libraries which will be used are

- **Glide** to handle the loading and caching of images.
- **Butterknife** to reduce the boilerplate code required for binding views.
- **Android Design Support** to include material design.
- **Volley** to fetch and send data to server.

Describe how you will implement Google Play Services.

The app will contain google ads thereby including Google Play Service.

Next Steps: Required Tasks

This is the section where the main features of the app (as declared above) are taken and decomposed into tangible technical tasks that can be completed incrementally until the final app is created.

Task 1: Project Setup

Setup the project and configure

- Create skeleton app and link to Github repo.
- Import all the libraries needed.
- Create Database in the server.
- Create Mockup for the basic idea.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LOGIN and SIGNUP activity.
- Build UI for all the other activities required.

Task 3: Implement Functionalities along with importing libraries

- Impose the Functions on the built UI .
- Use the functions from imported libraries.
- Control the errors and exceptions.

Task 4: Observe the Data Flow and Logic Checking

- Observe how the specific data gets parse from one function to other function and from one activity to another activity.
- The data should follow the logic that got implemented as decided.

Task 5: Check on Simulator and build SIGNED APK

- Check the full app and build the SIGNED APK of it.
- Control the errors and exceptions.