KANAKKU BOOK

BY TEAM

MISCHIEF MANAGED

INTRODUCTION:

Overview:

- This application is designed to maintain **customers**, **payments** and their **purchases**.
- A retailer will be an **admin** of the application and each customer of the retailer's shop will be the **user**.
- Customers can create their account in the app by reading and agreeing to the **terms and conditions** of the shop.
- The retailer can **view the customer list** and **add/delete customers**.
- Once the customer created their account, they can **login to their account using their credentials.**
- They can **view their pending bills** and pay them.
- They can also **register a complaint** directly to the retailer.
- The retailer can **send SMS alerts** to the respective customers when they have bills unpaid for a long time.
- The admin maintains the data about **purchases** and the **payment details.**

Purpose:

- This app is used to **digitalise the payment and purchase details** for providing an easy way to access required details from the collection of payments.
- It makes the manual work of recording the details of payments and purchases easier by a mile.
- It isn't necessary to maintain a separate notebook for separate customers as the **need for paper and pen is completely eliminated.**

- The **human errors** while calculating bills and maintaining payments **are also eliminated** since everything in the app is digital.
- It enables the customers to **register complaints or clarify doubts without the necessity to go in-person to the shop**: everything can be done much more effectively via the app.

LITERATURE SURVEY:

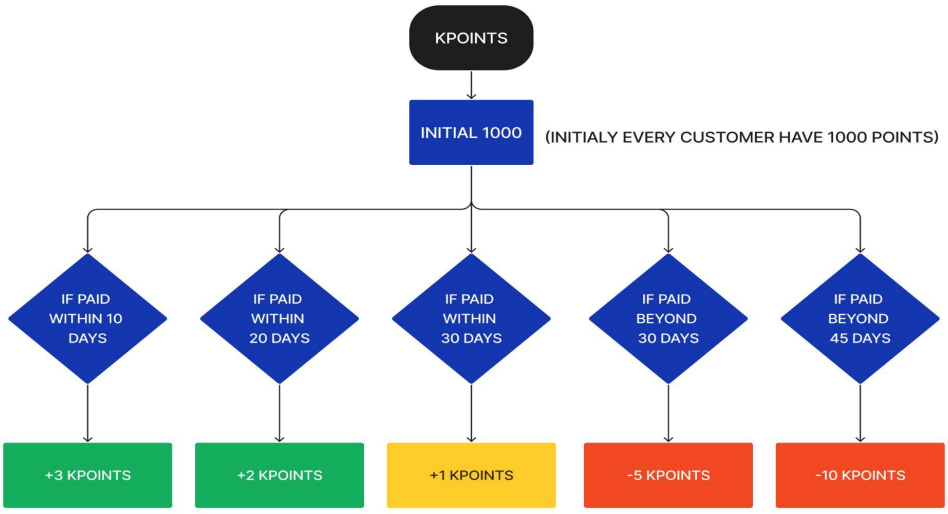
Existing system:

- The current way of keeping in track of purchases and payments is by recording everything in a notebook.
- It might even take separate notebooks for separate customers.
- It takes a lot of effort and space to maintain those details.
- It is also a tiresome job manually writing the details, amount, etc in those notebooks and are prone to human errors.
- If a retailer/customer wants to view a particular payment/purchase details from the past, it takes a lot of effort and time to manually go through every single one of the records and then find the required details.
- The detail is also susceptible to be carelessly missed by the person searching through it.
- It might also lead to data inconsistency.
- In cases of any queries or complaints, the customer must visit the shop and then clear doubts or register complaints.
- It is very time consuming for both the customer and the retailer.

Proposed System:

- The proposed system is to enter all those payment and purchase details in the app and digitalise everything.

- The retailer can **add/delete customers** to/from his shop.
- The retailer can **view the list of customers** in the shop.
- The retailer **can add the bill for the customer**s in the app and that will reflect in the **pending bills section** of the customer.
- The customer can **login using his credentials** and view the pending payments.
- The customer can also view the **total bills (pending + paid)** by a single click in one place.
- The customer can pay any pending bill via the pending bills section by **Razorpay**.
- The customers can **register complaints** via the app which will be sent directly to the retailer.
- The retailer can add customers easily by **scanning a QR code** assigned for the specific customer which reduces the workload.
- There is another feature called **Kpoints.**
- In this feature, each customer will be given an initial 1000 Kpoints. Based on whether the customer pays his due bills on time, the Kpoints will increase/decrease.
- The buffer amount is 500. When the due amount is more than 500, his Kpoints will change according to the following chart.



- When he pays his due bills on time/within a small span of time, his **Kpoints will increase.**
- When he takes a long time to pay his bills, his **Kpoints will decrease.**
- When a retailer adds a new customer, he will see the Kpoints of the customer that is **based on his previous payments to other shops.**
- Based on the Kpoints, the retailer can decide whether to add the customer to his shop or not.
- There is another feature called **Custom pay**, in which the customer can pay whatever amount he has for the time being and can pay the remaining bill whenever he finds cozy.
- It is like the customer can **pay the bill in segments** he finds comfortable.
- The retailer can **send SMS alerts** to the customer when a customer has a bill pending for a long time.
- The retailer can **view the complaints/queries** sent by the customers and can reply to the customers accordingly.
- The app contains a single **transition switch** to easily transition between **customer and retailer roles of a person** when a retailer is a customer of another shop.

Technology stack:

Flutter: It is used for implementing the front-end for our app. It works on both **android and ios.**

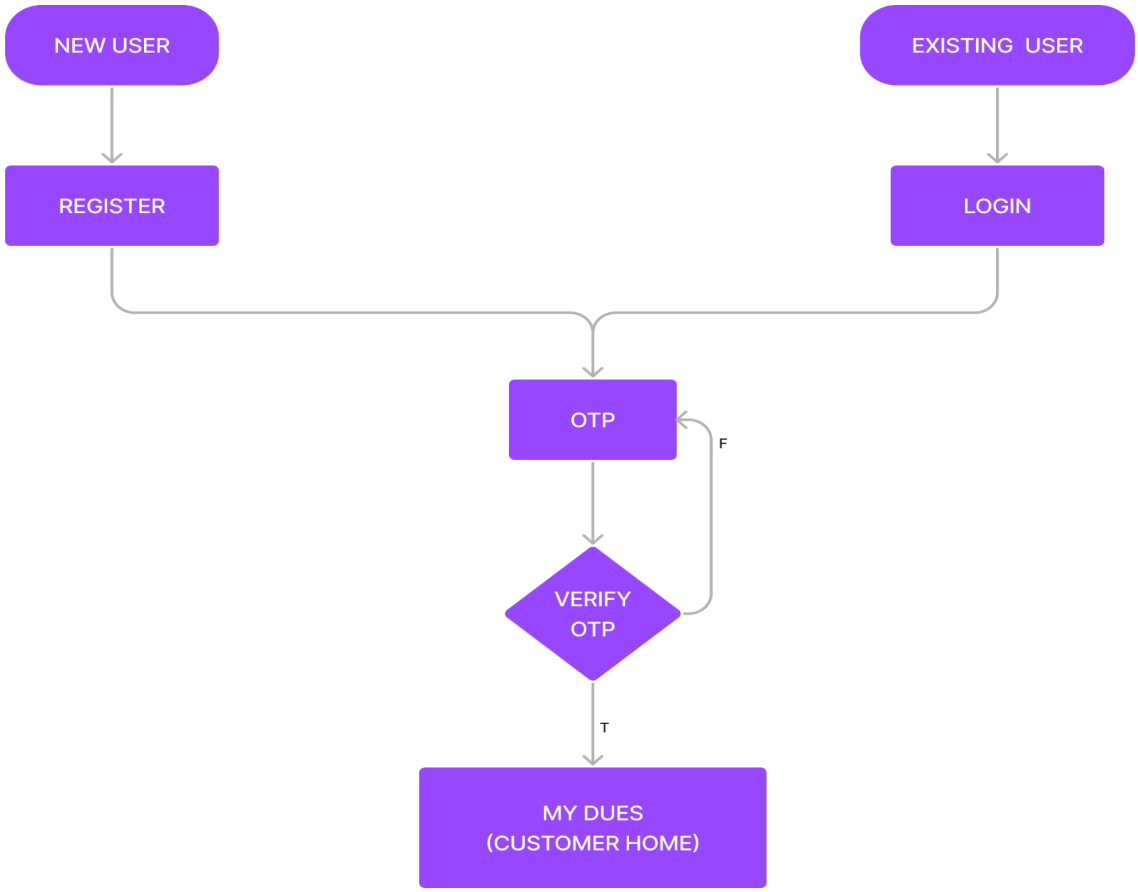
IBM-DB2: It is used for cloud and database.

Docker: It is used for delivering the app in packages called **containers** which allows the app to work on all the devices smoothly.

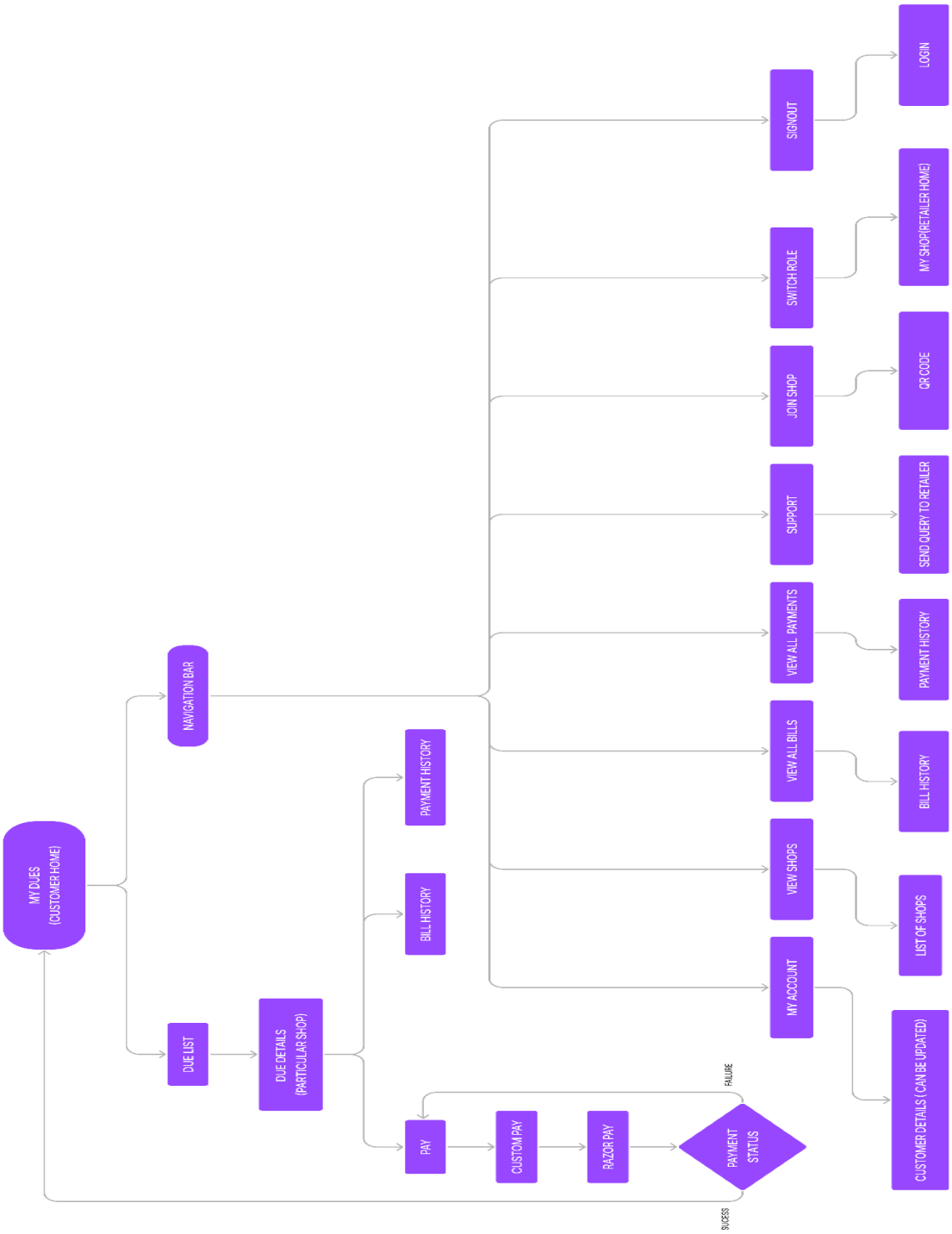
Openshift: It is used to deploy, scale and manage the container based application.

Flowchart:

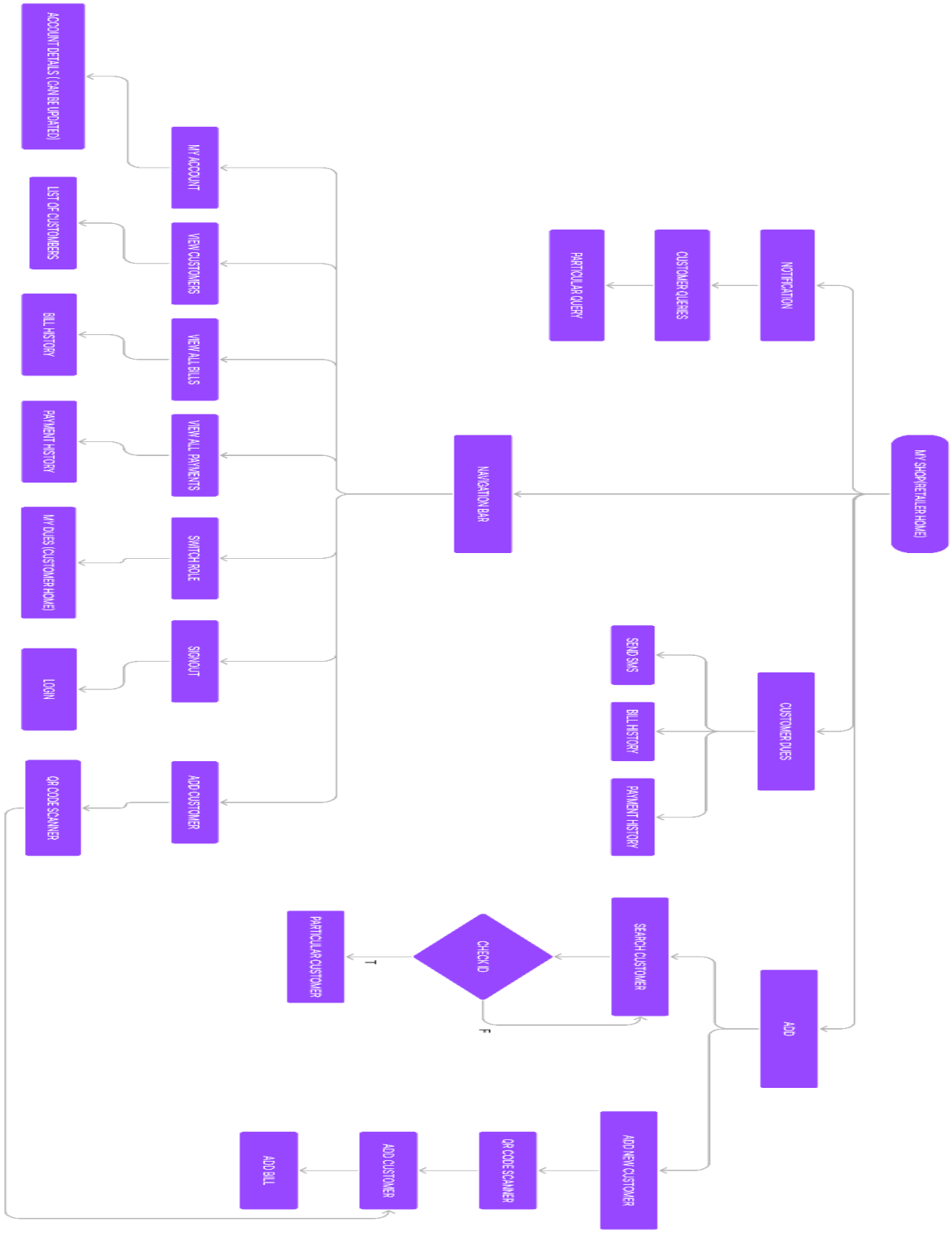
Authentication flow:



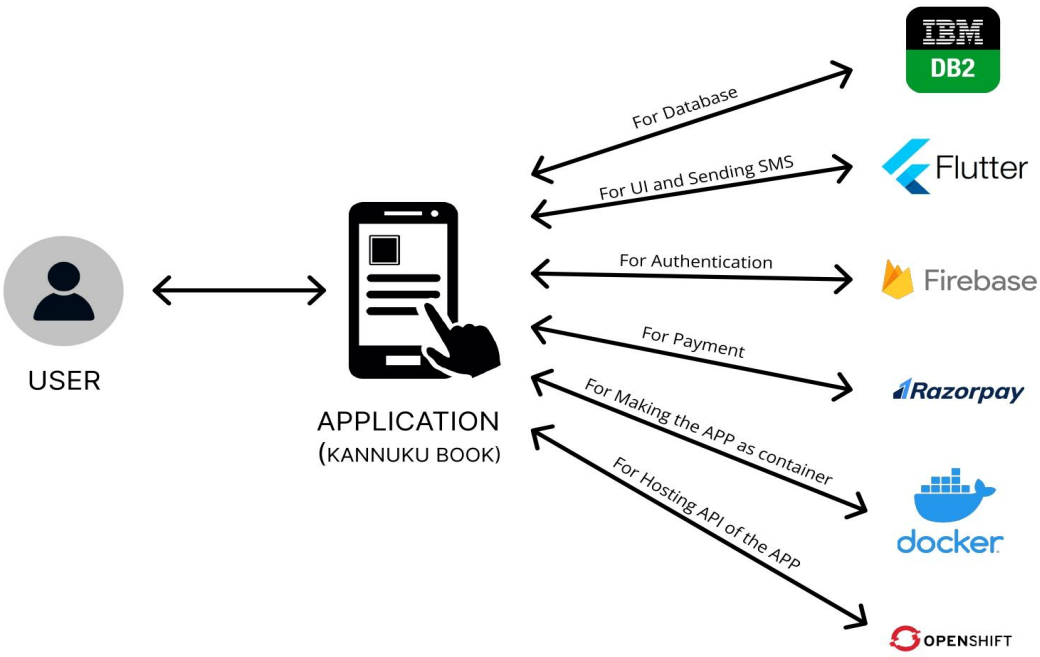
Customer flow:



Retailer flow:

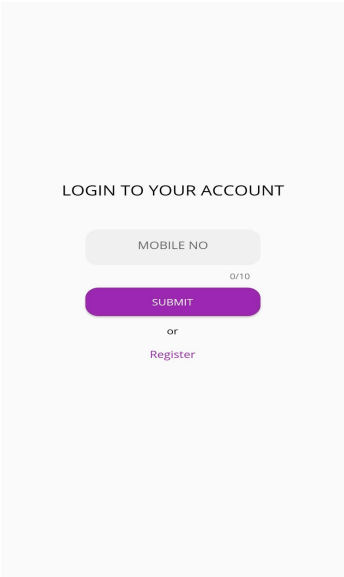


Block diagram:

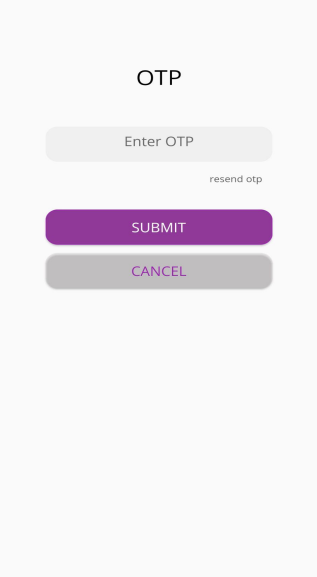


Output screenshots:

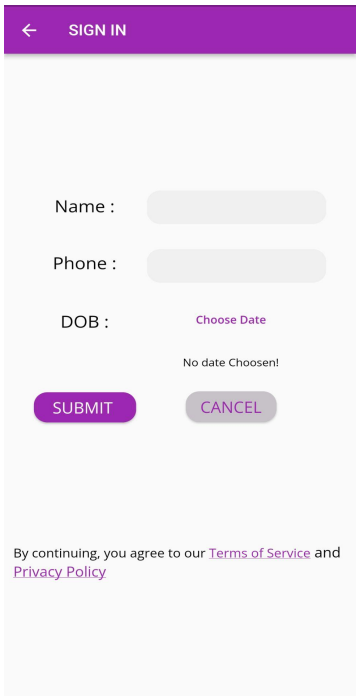
Login Page



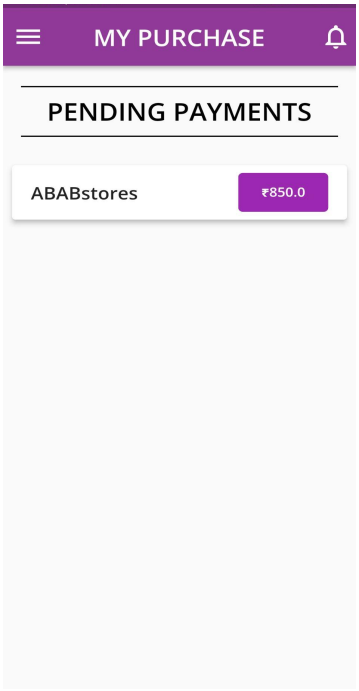
OTP



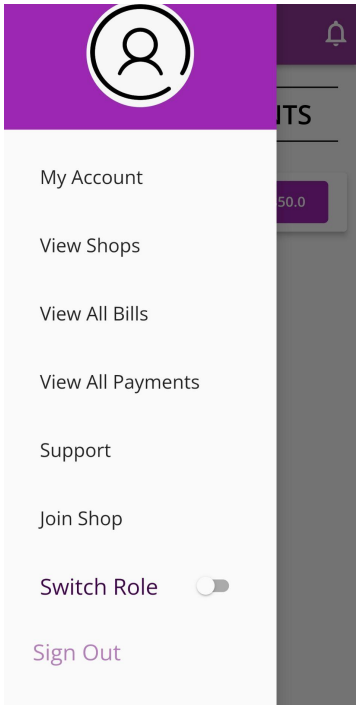
Sign in:



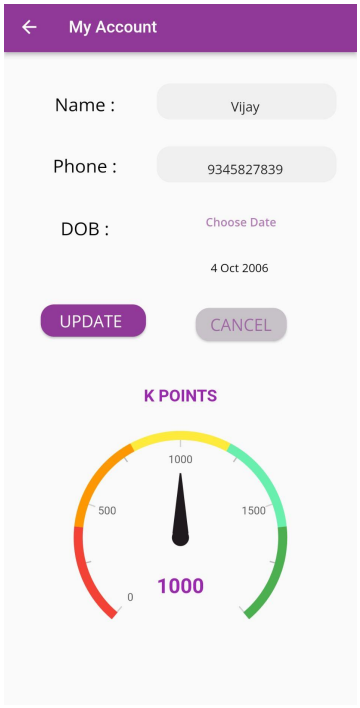
Customer home:



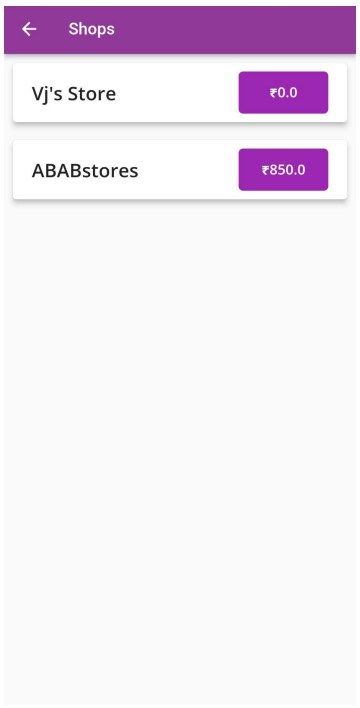
Customer Menu:



My account(Customer):

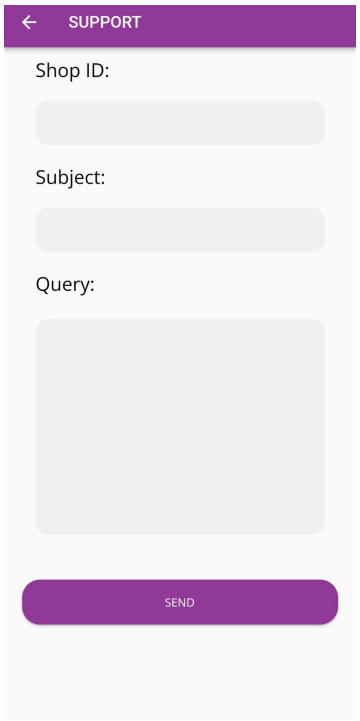


Customer’s Shops:

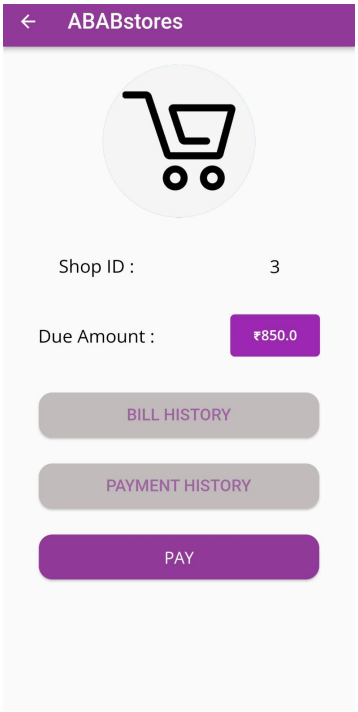


Join shop:

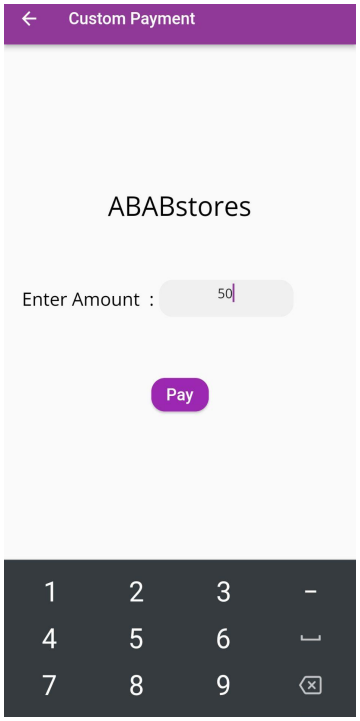
Support:



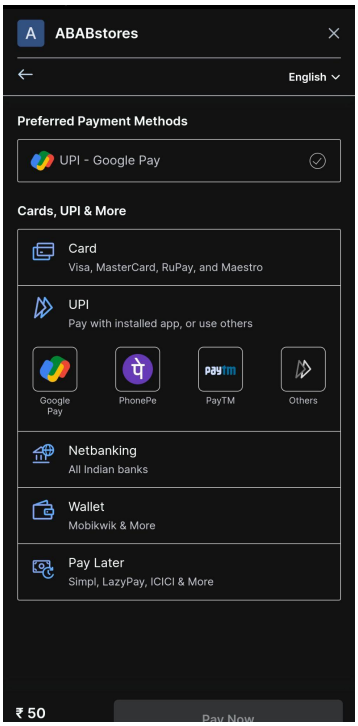
Payment:



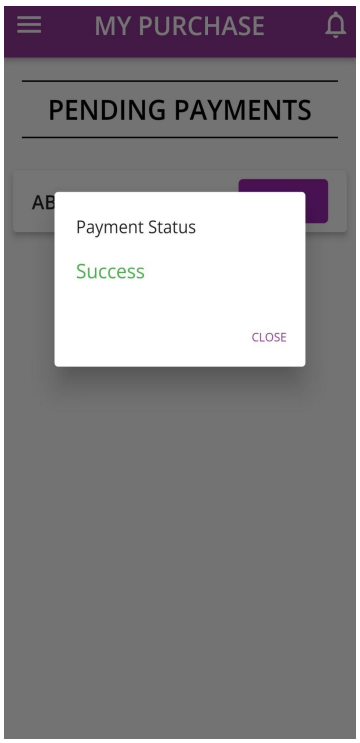
Custom pay:



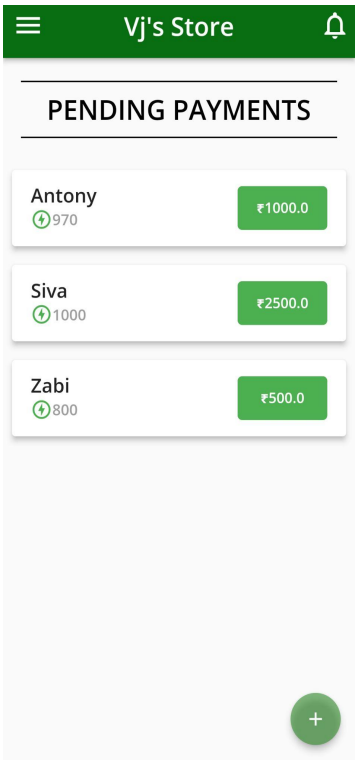
Payment gateway:



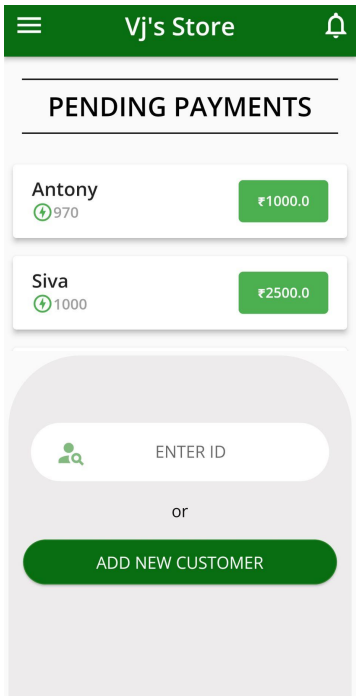
Payment success:



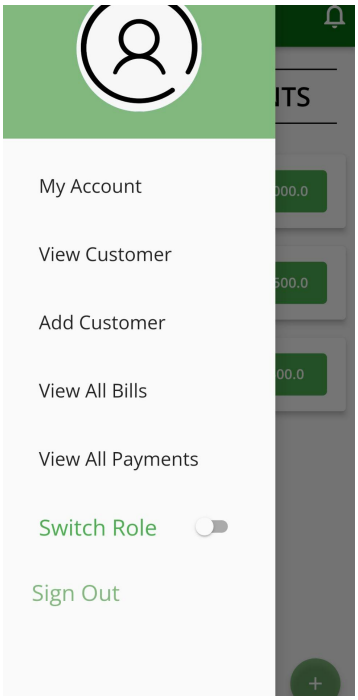
Retailer home:



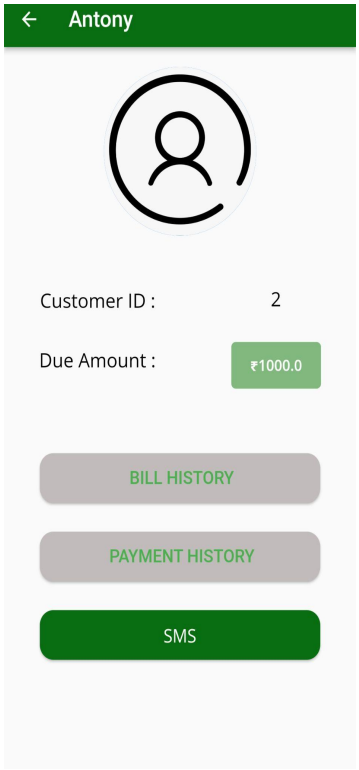
Add new customer/bill:



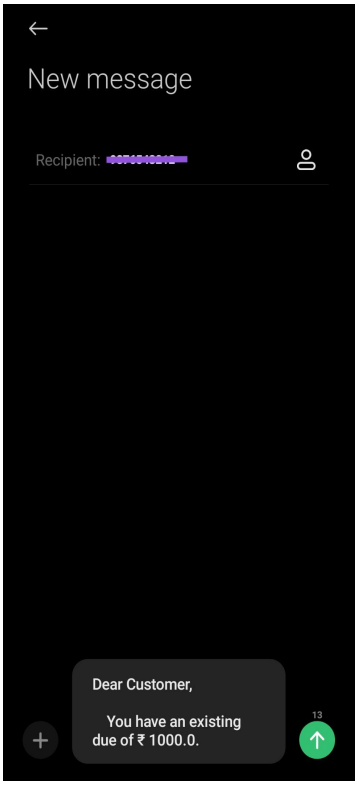
Retailer menu:



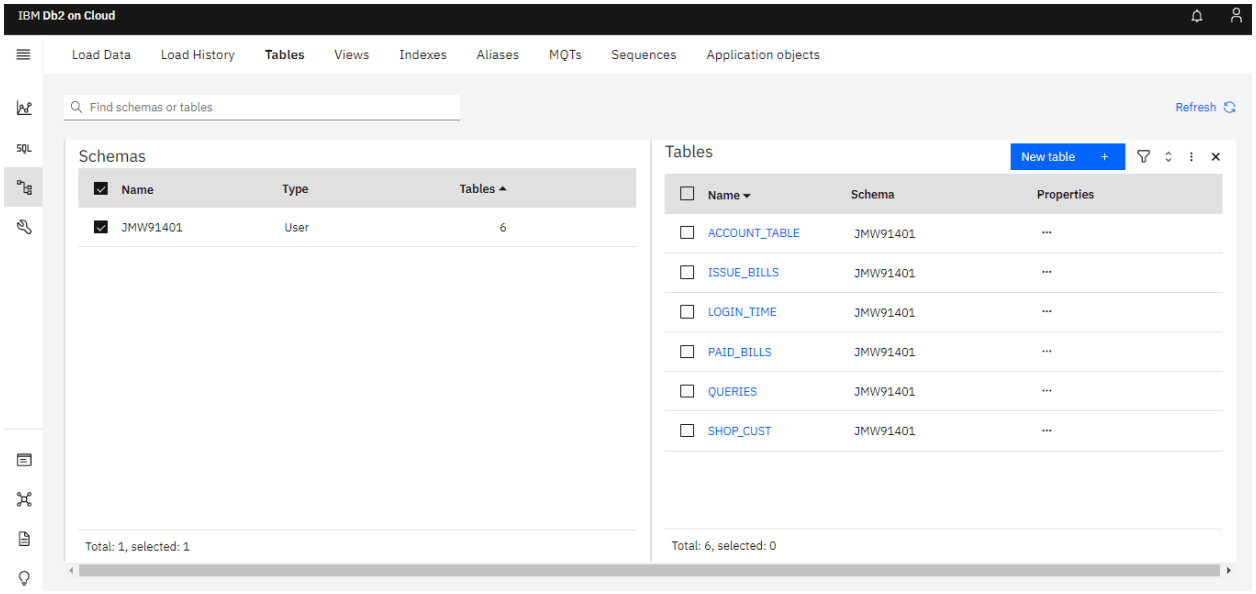
View Bill details:



SMS alert:



Database (DB2):



Advantages:

- This app is used to **digitalise the payments and purchase details.**
- The app has a **simple User Interface** for new users to navigate and learn the nooks and turns of the app.
- The user has to agree to the **terms and conditions** before registering into the app and read them thoroughly and agree if he finds those acceptable.
- Pens and paper are no longer needed to store the details of customers and their purchase and payment history.
- It enables the retailer to view his customers easily.
- **Searching for a specific customer's details** is so much easier through this app.
- It reduces the workload of both the customer and the retailer.
- It eliminates the human errors in calculation of bills.

- It eliminates the need for the customers to meet the retailer in-person to register any complaint or to clarify a query.
- The need for cash and coins are avoided as the app contains **UPI/Net banking/Debit/Credit card facilities**.
- Adding new customers to the retailer's shop becomes easier due to the **QR code feature.**
- The customers can **view all their bills from different shops** in a single place and can pay them easily directly through the app.

Applications:

- This app can be used in **small scale retail shops.**
- It can also be used in **wholesale shops.**

Conclusion:

- This app revolutionizes the digitization of payment and purchase history storage in the most efficient and user-friendly way possible.
- It reduces the workload and saves much time for the customer and the retailer.
- It enables the retailer to view all his customers and enables the customer to view all his **bills from different shops in a single place**.
- The customer can pay his bills directly through the app via **UPI/Net banking/Debit/Credit cards.**
- The customer can **view all his pending bills** and paid bills with much ease.
- The workload of the retailer is reduced much more through the feature of adding customers via **scanning a QR code.**

- The customer can easily **register complaints and clarify his doubts** directly using the app.
- The features **Kpoints** and **Custom pay** are added advantages to both the customer and the retailer.
- The retailer can notify the customer of any bills pending for a long time by **sending alerts through SMS.**

Future Scope:

- KYC verification
- Distributed database system

Bibliography:

- pub.dev
- stackoverflow.com
- github
- youtube.com