

# **Flutter Mobile Application Development**

Project report submitted in partial fulfillment of the requirement for the degree of  
Bachelor of Technology

in

**Computer Science and Engineering/Information Technology**

By

(Abhishek Samar Yadav (191280))

Under the supervision of

(Dr. Shubham Goel)

to



Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology Waknaghat, Solan-173234,  
Himachal Pradesh**

**Certificate**

## **Candidate's Declaration**

I hereby declare that the work presented in this report entitled “ .....” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **(Supervisor name)** (Designation and Department name).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Student Name, Rollno.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Supervisor Name

Designation

Department name

Dated:

**Plagiarism certificate**

**(should be signed by supervisor, student, and LRC officials)**

### **Acknowledgement**

I would like to offer our sincere gratitude to our esteemed and knowledgeable mentors, DR. Shubham Goel, for their invaluable assistance and direction, as well as for the inspiration they have provided for us to finish the project.

We are also grateful to our respected HOD Dr. Vivek Seghal and all other teachers for providing us assistance in various stages during the course of our project.

(Student Signature)

Project Group No. :

Student Name: Abhishek Samar Yadav

Rollno.:191280

# Table of Contents

<b>Introduction</b>	
<b>Literature Survey</b>	
<b>System Development</b>	
<b>Performance Analysis</b>	
<b>Conclusions</b>	
<b>References</b>	
<b>Appendices</b>	

## **List of Abbreviations:**

Very Less (almost zero) abbreviations are used in this Major Project Report.

## **List of Figures:**

Various images and screenshots are inserted in this report ,  
all are in the output section.

Page Number:

Page Number:

## Abstract

In today's environment and age, developing cross-platform mobile applications is a top priority. Developers must choose between creating the same application repeatedly for different OSs (operating systems) or settling for a subpar alternative that sacrifices native speed and accuracy for portability. A high-performance and more dependable mobile application can be created using the open-source Flutter SDK for platforms like iOS and Android. Just-in-time compilation, which executes the computer code that includes compilation during programme execution at run time rather than before execution, is one of the key components of the Flutter framework. This often consists of translating source code, also known as bytecode, into machine code that is invariably executed.

Compilation done in advance High-level programming languages like C and C++, as well as intermediate representations like Java bytecode and NET Framework Common Intermediate Language (CIL) code, are all compiled.

So that the following binary file can run natively on the target system, into native system-dependent machine code. Hot reload, a feature of Flutter, makes it simple to experiment, create user interfaces, add functionality, and repair errors.

Updated source code files are inserted into the active Dart Virtual Machine (VM) to perform hot reloading.

The Flutter framework dynamically reconstructs the widget tree once the VM updates classes with the new versions of fields and functions, enabling you to quickly see the unique effects of your modifications. Targeting the leading mobile operating systems, such as Android and iOS, Flutter provides a GPU rendering and user interface solution that is driven by native ARM code.

Cross-Platform mobile application development, IDE, Android and iOS development, Flutter, and Dart are some of the keywords.

# Chapter-1

## Introduction

Aim of this project Report is to present the Work/Projects which I have done in my organization during my Condensed Semester. I was placed as /my designation was Flutter Mobile Application Developer. Responsibilities which were assigned to me were , writing clean and structured code for frontend of mobile application cross platform (Ios & Android) in Flutter framework using Dart Language, Integrating third party packages and Restful APIs( as backend of applications were developed in different languages and frameworks). Working on this profile and with these responsibilities I have worked on three mobile applications:

**Rentswale:** an ecommerce application to connect users who want products on rent and vendors who want to list their products on rent. Front end of the application was developed in Dart language using the Flutter framework , backend was developed using PHP and for database MySQL was used.

**Authentic Guards:** Android application developed for a client ,to maintain the regular updates of their guards, track their activities , salary slip, and their client login functionality to track guards which are provided by the organization for their services. Frontend was developed using flutter framework , backend was developed using PHP.

**Rk Construction:** Android application developed for the client, mostly features of construction ERP , with multiple logins(i.e, for engineers, supervisors, purchase store etc) integrated with super web admin Panel of Construction ERP. Mobile application was developed in Flutter framework using dart language(Frontend), with multiple third party packages, integration of restful APIs to communicate with backend and database, backend was developed in PHP.



## **Problem Statement:**

### **Rentswale:**

Problem Statement for this project was to develop a fully functional E-commerce mobile app, which will be published on both Google Play Store and Apple App Store. a web admin panel to access both vendors and customers activity. Various features were asked to add in app like, fetching live location, a separate module in the same app so that new vendors can post their products, payment gateway integration, image picker, file picker, in app call and chat feature, state management of different components/features, dependent dropdowns, product filtering based on location, category, subcategory.

### **Authentic Guards:**

Problem Statement for this project was to develop a fully functional mobile app for the Security/Guards providing organization, which will be published on Google Play Store. App must contain features which will access the live location of registered guards. Guards will login in the app only if he/she is in a particular range. In the same app their client login credentials, so that they can check the details of Guards available for their services. App must have features like live location access, image picker, file picker, multi login system, alert boxes etc.

### **Rk Construction:**

Problem Statement for this project was to develop a fully functional mobile app for the construction company with ERP features, which will be used by company registered employees only. Mobile app must contain multi login features (i.e. in the same app engineers, supervisor, store manager etc) will login and different screens and data will be visible to them based on their job roles. Integrating biometric device, multiple features of construction ERP.

## **Objectives :**

Main objective of this Project Report is to give a detailed description of the projects and works which I have done and completed during my condensed semester duration in Softhub Technologies as a Flutter Application Developer. I have worked on three mobile applications development projects. Problem Statements for the three projects have been described above.

### **Objectives of the projects are explained as below:**

**Rentswale:** The objective of this project was to provide a platform where persons who want to list their things on rent and persons who want things on rent can connect. Sellers and Buyers must be from the same city, Sellers will have to buy a subscription plan for posting their things on rent. Services for customers (only buyers ) will be free, they will be able to access contact numbers and can chat to potential sellers nearby, after registering with the app and done with the KYC process. Sellers can list their products/things in categories like real state, electronics, transport, customs, event management products, furniture, services like plumbing carpentering. later they will be filtered based on subcategory. Overall objective was to develop a fully functional cross platform mobile application that is efficient, bug free, flow should be smooth, scalable.

### **Authentic Guards:**

The objective of this project was to develop an Android Mobile Application for a Security Service providing company. The main objective of this application was to access the live location of their security guards and mark their attendance based on their location. If guards are in the specified location/radius of their assigned duties , they will be able to mark their attendance else they will be marked absent. In the same app there is client/customer login , objective of this was to provide functionality to check details of security guards available for their service.

Overall objective of this project was to develop a mobile application which helps companies in tracking their employees daily activities, increasing efficiency, making various processes automatic, building a strong and trusting relationship with their clients by providing them access to their employees who are available for their services.

## **RK Construction:**

The objective of this project was to develop a mobile application for a construction company only for their employees. As there are multiple types of employees work in a construction company like engineers, supervisor, purchase store manager, instructor and all these have different job responsibilities. With this in mind the main objective of this was to create separate login for each, and after login different users will have different user interfaces and functionality based on their job roles and responsibilities. Integrating restful apis in a manner so that most features of construction ERP integrated with super admin panels functionality works. like, order received, order approved, material requirements, purchase items, etc etc. As this project was very big in length, I was assigned to create the user interface of the applications , basically ui designing.

The overall objective was to develop an efficient mobile application which will have various functionalities and features of construction ERP.

## **Methodology:**

All the above mentioned three applications are developed using Dart language in Flutter Framework for frontend, backend was developed in Codigentor using PHP language and for database MySQL has been used. Methodologies used in all the above mentioned applications are almost similar:

1. Gathering all requirements of the client i.e, listing of all the required features in the application.
2. Preparing SRS document for the entire project, creating flowchart of the entire project structure.
3. Creating a roadmap of backend development , database connection, frontend development, sdk requirements , third party packages requirements, servers, testing softwares.
4. Wireframing of the UIs, selection of templates, designs , colors, fonts, images and other required assets to develop an attractive, efficient, responsive frontend for the users.
5. Developing the backend ,connection to the database, creating Restful APIS for integrating to the frontend , so that the user can send requests to get data from the database and receive responses from the server.
6. Testing Restful APIs in PostMan Software to understand the responses from the backend and creating models.
7. Integrating the restful apis in front end and managing states of application as per the responses from the server and request from users.
8. Testing the developed application on various testing platforms both manual and automatic.

As this is the general methodology which we use, but as these all above mentioned projects were developed for real time business purposes.

So we followed a more professional methodology while developing these applications , clients feedback and updates were taken at various stages of development. therefore organization used the professional methodology i.e, AGILE Methodology.

### **AGILE Methodology:**

In order to manage a project, the Agile methodology separates it into several phases. Continuous cooperation with stakeholders and improvement at every level are necessary. Once the job is underway, teams cycle through a process of planning, performing, and evaluating.

## Organization :

As all the above mentioned projects are developed and analyzed by the team of service based company in which i was placed and currently working as Junior Flutter Application Developer.

Name of Organization: **SoftHub Technologies Pvt, Ltd.**

Office No. 108-Yash 101 sus Behind Mercedes-Benz off Mumbai pune highway  
Baner,Pune.

Email id: [info@softhubtechno.com](mailto:info@softhubtechno.com)

Project team: All the above mentioned projects are developed in collaboration with different Tech Stack teams. Backend team, testing team , were others employees of the organization. **Frontend is entirely developed and designed by me in the Flutter Framework using Dart Language.**

Projects Development were organized in the following order:

- . Project manager/Business analyst were responsible for gathering all the requirements and inputs from clients .
- . Discussion with different teams to start development by breaking the project into frontend , backend and database management.
- .Frontend development /user interface development based on ui design provided by clients/ designing team, integrating Restful apis, integrating third party packages.
- . Getting feedback from the testing team and working on bugs and changes asked.

# Chapter-2

## LITERATURE SURVEY 1 :

2019 July 23–26, Pilsen, Czech Republic: Proceedings of the International Conference on Industrial Engineering and Operations Management.

Development of mobile applications: A thorough and methodical examination of the literature.

Institute of Science & Technology Hanif SRM email: [hanif\\_sh@srmuniv.edu.in](mailto:hanif_sh@srmuniv.edu.in).

Science and Technology Institute of S. Jagadeesan [@ktr.srmuniv.ac.in](mailto:@ktr.srmuniv.ac.in), or Jagadeesan.

Vinayak A. Drave Department of Industrial & Management Engineering IIT Kanpur, 208016, India [vinayak@iitk.ac.in](mailto:vinayak@iitk.ac.in) .

Priyanka C. Bhatt Learning Resource Centre Bennett University, Times of India Group 201310, India [priyanka.bhatt@bennett.edu.in](mailto:priyanka.bhatt@bennett.edu.in) .

### Abstract

Mobile penetration has reached a substantial level in the current era of e-commerce and has been increasing dramatically over the past ten years. A new perspective on customer service has emerged thanks to the mobile application. Researchers' attention has shifted towards development as a result of the rise in these applications, and they have discovered numerous problems related to it. The large body of scholarly literature that has amassed in the last ten years is the subject of this investigation. A thorough and organized evaluation of the available literature was conducted in the newly-emerging field of mobile application development. After examining a large body of literature from the largest database, "Scopus," a total of 26 pertinent journal publications were taken into consideration for the review after going through several stages of filtration.

This article provides the organisation with a broad understanding of the field of mobile development and gives it scope and direction. The study's findings offer insightful analysis and recommendations for the future in this emerging subject.

Keywords: Android Operating System, Framework, Collaborative System, Online Services.

### Research approach:

In this section, we've spoken about the approach used for the literature review procedure. A thorough and complete examination of the available literature was conducted in the field of developing mobile applications. The methods outlined by (Tranfield et al. 2003; Dubey et al. 2017) served as the foundation for the systematic and thorough literature review.

## **Introduction:**

In the past ten years, a revolution has been sparked by the development of mobile devices and applications in numerous industries. The initial application was in the marketing, advertising, and various service sectors; later, it was expanded into the healthcare and insurance sectors, leaving no area or company untouched. The research community was motivated to comprehend all of the vertices in the niche by the exponential rate of application development. A previous study (Lee et al., 2014) developed an architecture that primarily loads the burden of a mobile device onto an intelligent cloud, taking into mind the importance of productive execution of mobile applications.

The framework includes a runtime infrastructure for the organisation in the cloud as well as a programming tools that promotes the creation of mobile applications capable of enabling computation loading. The device-independent mobile application generation (DIMAG) framework, created by Miravet et al. (2013), shows how the specifics of client-server mobile applications might be a suitable method for developing both the client and server sides of native applications. Yusoff et al. (2016) investigated the challenges and limitations faced by software and requirements engineers while generating security requirements and security attributes. Two tests using inexperienced requirements engineers (REs) focused on manually extracting security attributes from a variety of needs situations.

The investigation shows that eliciting security qualities is particularly difficult for inexperienced REs and that they require support, especially automation support. A comparable survey instrument based on Apple's general user experience guidelines was conceptualised by Hoehle and Venkatesh (2015) to aid in such an endeavour. The Mobile Infrastructure Analytics System (MIAS) was introduced by Ramakrishna et al. (2017). It aids in effectively identifying and examining application faults in a distributed domain by thoroughly examining application and network activity across client devices, application servers, database servers, etc. According to (Dar et al., 2018), precondition designing is the fundamental step in characterising partners' needs and requirements for any software development.

## **References:**

- Ahmad, A., Li, K., Feng, C., Asim, S. M., Yousif, A., & Ge, S. (2018). An Empirical Study of Investigating Mobile Applications Development Challenges. *IEEE Access*, 6, 17711–17728. <https://doi.org/10.1109/ACCESS.2018.2818724>
- Bartin, B., Ozbay, K., & Yang, H. (2018). Evaluation framework for mobile ticketing applications in public transit: a case study. *IET Intelligent Transport Systems*, 12(9), 1166–1173. <https://doi.org/10.1049/ietits.2018.5248>
- Bergvall-Kåreborn, B., & Howcroft, D. (2011).

## LITERATURE SURVEY 2 :

International Research Journal of Engineering and Technology (IRJET)

e-ISSN: 2395-0056 , p-ISSN: 2395-0072 , Volume: 08 Issue: 04 | Apr 2021 [www.irjet.net](http://www.irjet.net) .

### Mobile Application Using Flutter (Know Your Ride)

1. Prof. Shital Agrawal, Dept. of Information Technology Engineering, Armiet, Maharashtra, India
2. Manish Patil, Dept. of Information Technology Engineering, Armiet, Maharashtra, India
3. Manoj Kumar, Dept. of Information Technology Engineering, Armiet, Maharashtra, India
4. Khan Aatif, Dept. of Information Technology Engineering, Armiet, Maharashtra, India

### Abstract:

– Flutter is Google’s UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. In this project, we will make a mobile application using FLUTTER named Know Your Ride. This app will provide detailed information about vehicles. We used flutter for frontend and Firebase for backend. The Admin Panel is made using Python Django. ChatBot is also available for quick question answers related to the vehicles.

**Key Words:** Flutter, FireBase, Django, ChatBot, Admin Panel.

### INTRODUCTION:

Flutter is a Google mobile UI framework that is free and open source that offers developers a quick and expressive approach to create native apps for both iOS and Android. There are numerous tools available for developing mobile applications, like Python Kivy, React Native, and Android Studio utilising Java or Kotlin. The only framework with a mobile SDK that offers reactive styles without utilising a Javascript bridge is flutter. The SDK is open source and available for free, allowing developers to experiment with and build robust tracking apps. It is the inspiration behind the flutter-based programmes and user interfaces. Flutter uses the GPU, builds from a single codebase, and accesses platform APIs and services. It also compiles directly to native arm code. Therefore, for this project, we used Flutter.

### REFERENCES

- [1] Marco L. Napoli. "Beginning Flutter: A Hands On Guide to App Development."
- [2] Alessandro Biessek. "Flutter for Beginners: An Introductory Guide to Building Cross-platform Mobile Applications with Flutter and Dart 2."
- [3] Dzenan Ridjanovic and Ivo Balbaert. " Learning Dart."
- [4] <https://flutter.dev/docs>.
- [5] <https://flutter.dev/docs/reference/tutorials>
- [6] <https://dart.dev/guides/language/language-tour>.
- [7] <https://cloud.google.com/dialogflow/docs>.
- [8] <https://firebase.google.com/docs/guides>



## LITERATURE SURVEY 3 :

International Research Journal of Modernization in Engineering Technology and Science  
e-ISSN: 2582-5208

Volume:02/Issue:08/August-2020 Impact Factor- 5.354 [www.irjmets.com](http://www.irjmets.com)

### APPLICATION DEVELOPMENT USING FLUTTER

1. Aakanksha Tashildar, 2. Nisha Shah, 3. Rushabh Gala, 4. Trishul Giri, 5. Pranali Chavhan.

SPPU, Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.

#### ABSTRACT:

In today's environment and age, developing cross-platform mobile applications is a top priority. The only options available to developers are to create identical applications repeatedly for different OSs (operating systems) or to settle for a subpar, similar solution that sacrifices native accuracy and performance for portability. A high-performance and more dependable mobile application can be created using the open-source Flutter SDK for platforms like iOS and Android. Just-in-time compilation, which executes the computer code that includes compilation during programme execution at run time rather than before execution, is one of the key components of the Flutter framework. This usually involves converting source code, often known as bytecode, into machine code that is invariably executed. AOT compilation, also known as ahead-of-time compilation, converts high-level programming languages like C or C++, as well as intermediate representations like Java bytecode or NET Framework Common Intermediate Language (CIL) code, into native system-dependent machine code so that the resulting binary file can run natively. Hot reload, a feature of Flutter, makes it simple to experiment, create user interfaces, add functionality, and repair errors. Updated source code files are inserted into the active Dart Virtual Machine (VM) to perform hot reloading. The Flutter framework dynamically reconstructs the widget tree once the VM updates classes with the new versions of fields and functions, enabling you to quickly see the unique effects of your modifications. Flutter focuses on popular mobile OSes like Android and iOS.

**KEYWORDS:** Cross-Platform Mobile application development, IDE, Android development, iOS development, Flutter, Dart

#### PROPOSED METHODOLOGY:

##### System Overview:

For the Billing and Reward system, we have created a mobile application based on Flutter. The redeem point or point system places a strong emphasis on the idea that the more money you spend, the more points you receive in return, which can then be redeemed for future purchases. Each time a consumer makes a purchase, they receive a specific number of points based on the amount they spent on the item.

## **DART:**

Every application in Flutter is created using Dart. A programming language named Dart has been created and is maintained by Google. It is widely used at Google, and it has proven to be capable of producing large-scale web applications like AdWords. Dart was initially created to succeed and replace JavaScript. As a result, it incorporates the majority of the core features of ES7, JavaScript's upcoming standard, including the "async" and "await" keywords. Nevertheless, Dart includes a Java-like syntax to appeal to developers who are unfamiliar with JavaScript. Even though only a few other systems use reactive views, the Flutter application updates the view tree with each new frame. This behaviour has the downside of producing a large number of objects, some of which may only last for a single frame. Given that Dart is a contemporary programming language, it is enhanced with "Generational Garbage Collection" to handle this issue in memory.

## **INTRODUCTION:**

Our daily lives are becoming more and more impacted by mobile applications. Since November 2016, mobile devices have generated more network traffic (48.19%) than desktops or laptops (47%). A mobile application must get familiar with both the Android and iOS platforms in order to provide it to the majority of consumers. Due to the stark differences between these two platforms, it is frequently necessary to build alternative skill sets. For instance, Object-C or Swift for iOS and Java or Kotlin for Android. Because of this, businesses and developers frequently struggle to handle the complexity of creating cross-platform applications. React Native is an open-source, cross-platform JavaScript framework that Facebook unveiled on March 15 in an effort to address the aforementioned issue. React framework is widely used by developers due to its simplicity and effectiveness in the development process. In the second half of 2016, Google also unveiled the Flutter mobile SDK. Flutter applications, which were inspired by React Native, may also run equally on both platforms, hence lowering the cost and complexity of app development for iOS and Android. Only Google was employing Flutter for business applications at the time this report was being written (August 2017), as it had been totally created from scratch.

Cross-platform frameworks similar to React Native and Flutter have been discussed and used by many different firms in the past. However, neither one is sufficient to meet the need for industrial development. React Native and Flutter, which are supported by Facebook and Google and grab attention despite the ineffectual predecessors, inspire confidence in the future.

## **REFERENCES:**

[1] Wenhao Wu's March 2018 thesis, "React Native vs. Flutter: Cross-Platform Mobile Application Framework."

Flutter Clean Architecture Package: A Clean Approach to Flutter Development, IEEE 2019, Eduardo Colemanares and Shady Boukhari.

# Chapter-3

## SYSTEM DEVELOPMENT:

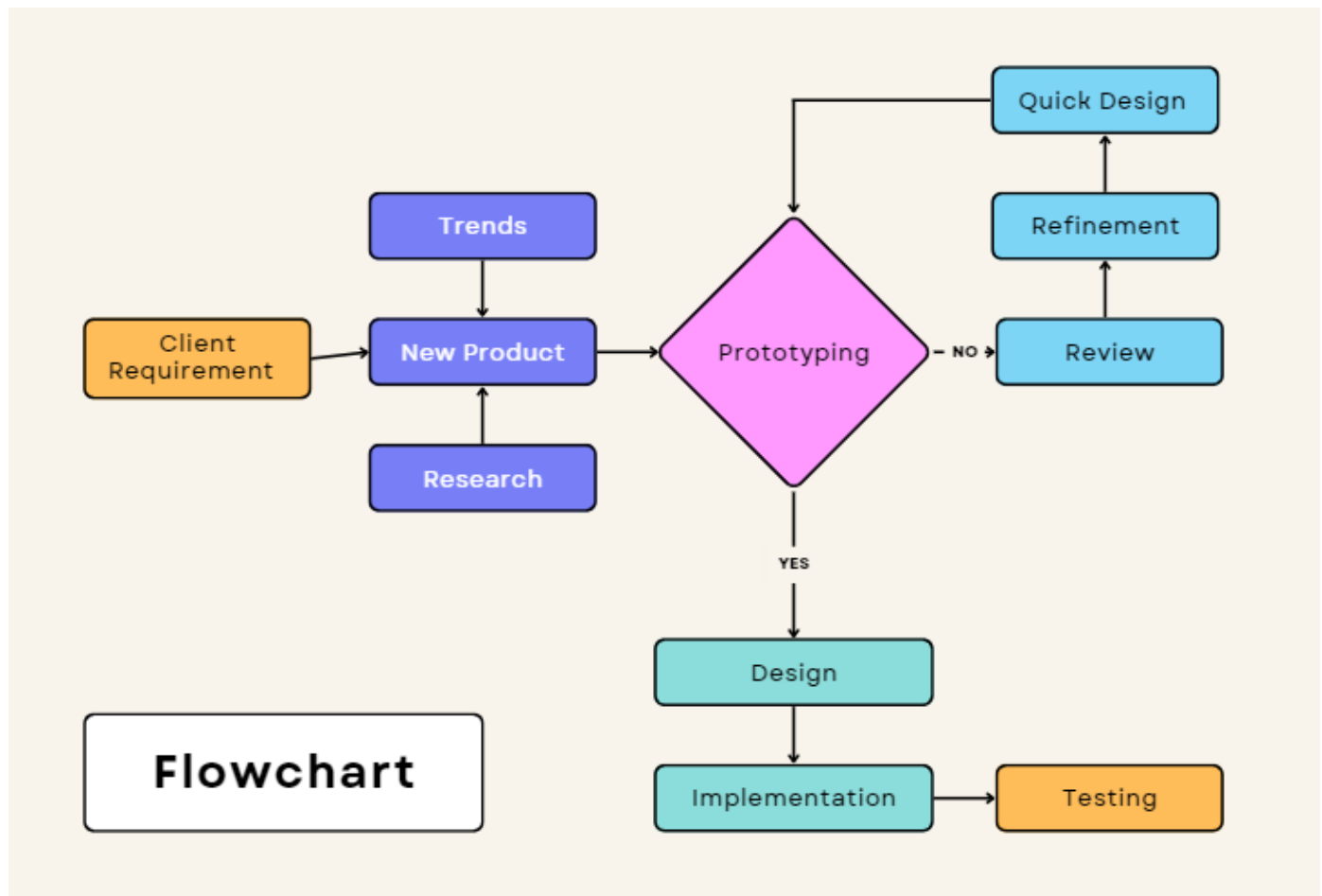
As all the above mentioned projects are real time/live mobile applications business projects. So various teams were involved and everyone had their specific responsibilities.

Software Requirement specifications(SRS) / Technical Requirements/ Designing / Backend were done by different persons and I entirely worked on the front end development part.

### Analysis/Design:

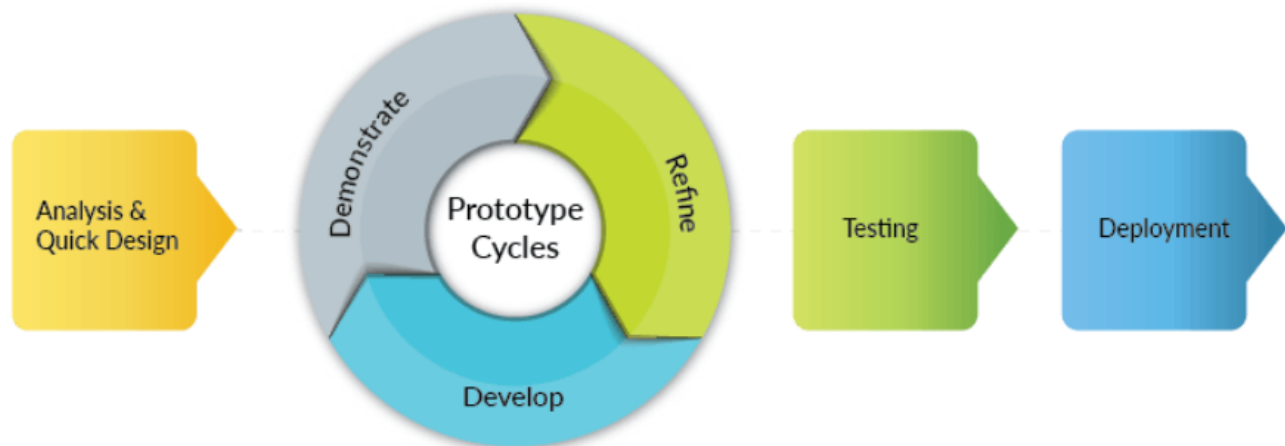
For all the above mentioned projects, Analysis and clients requirements were gathered by the business team. Flow of application, User interface Designing , Backend , Database ,Server, hosting etc things were all decided and done by the technical head after discussion with the client.

### Flow Chart:



## Rentswale Model Development:

### Rapid Application Development Model



*Figure 8 Rapid Application Development Model*

For system development, I opted to use the RAD (Rapid Application Development) methodology. For my capstone project, it was the best methodology because it allows for the speedy development and delivery of a high-quality solution. Actually, throughout the entire software development process, I was in continual contact with the client. I separately developed each component of the project, tested it, and asked the client (Rentswale) for feedback on what should be changed or improved. I was able to save a boatload of time because to the quick feedback from this exchange. Thus, the user was involved in the design process.

Due to the project's online store component, it is also critical to satisfy the business's needs, which is in keeping with the core principles of the Rapid Application Development approach. As a capstone project, we also have deadlines that we must adhere to. One of the key RAD ideas is this. There are deadlines or "timeboxes" in place.

The main advantages of the RAD model are:

The operational version of an application is completed much faster than with Waterfall, Incremental, or Spiral frameworks.

- Because RAD creates systems more quickly and with a commercial focus, it frequently generates systems at a cheaper cost.

focuses on crucial system components from the perspective of the user.

## Software Architecture:

As the project's system architecture, I went with a three-tier application design. Due to its four main benefits, the bulk of which are also included as non-functional needs for the project:

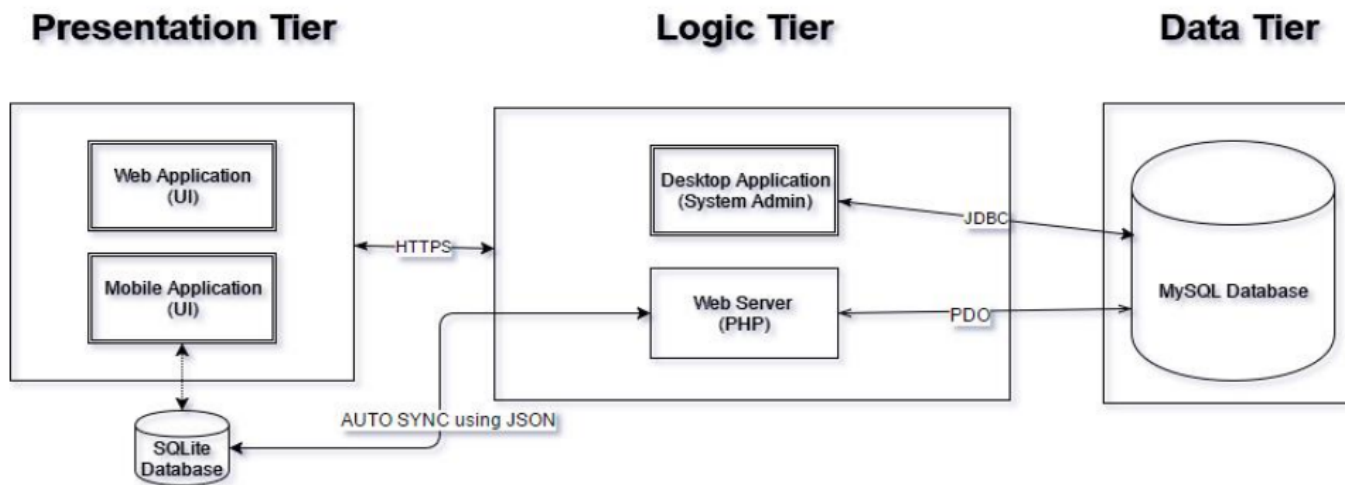
it's upkeep. Due of each layer's independence from the others, updates or changes can be done without affecting the programme as a whole.

**Scalability:** Because tiers are based on the deployment of layers, a relatively simple application can be scaled out.

**Flexibility:** Because each layer can be adjusted or scaled separately, flexibility is increased. Applications can take use of the modular architecture of enabling systems by using easily scaled components, which raises availability.

However, the system architecture I selected is similar to the RESTful API architecture. Clients connect to the presentation tier's web or mobile application to place orders. Before saving them in the data tier, the administrator processes the orders in the logic tier. The synchronisation of the SQLite database and MySQL database is done automatically to add new products to the mobile application.

### System Architecture:



# Use case Diagram:

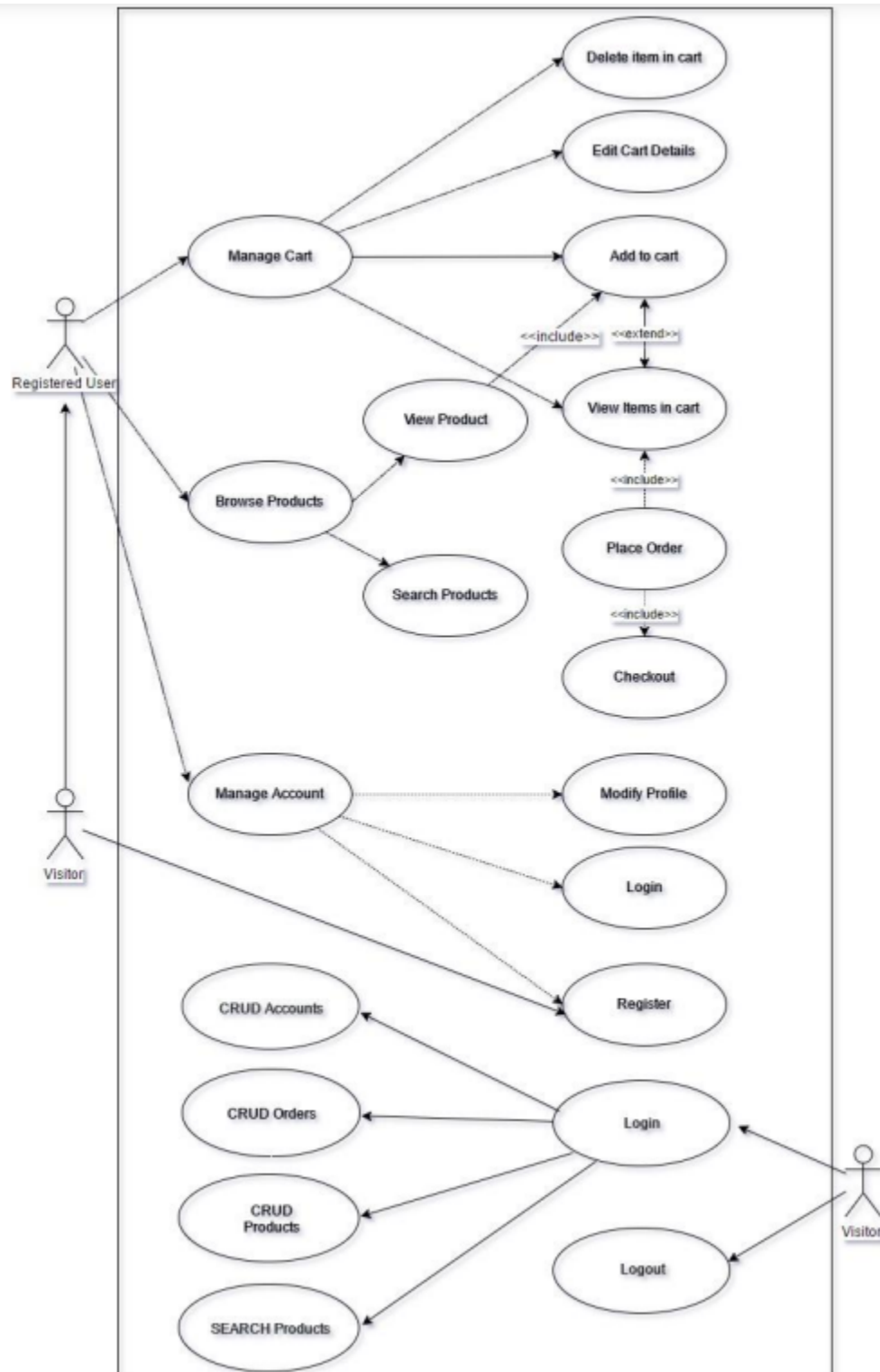


Figure 11 Use Case Diagram

The use case graphic shown above depicts how a mobile user interacts with the interaction between the administrator and the desktop programme, as well as the mobile application's operating system. The graphic displays the many system-user-specific actions and

relation to one another. The four main operations that the system administrator carries out on the system's numerous components are referred to as CRUD. Actors include: The administrator has the ability to create, read, change, and delete accounts, orders, and products.

- Visitor: Makes an account in order to browse, look for, and purchase items.
- Registered User: He must first log in in order to create a cart and add products to it.

### Data Model:

**Entity Relationship Diagram (ERD):**

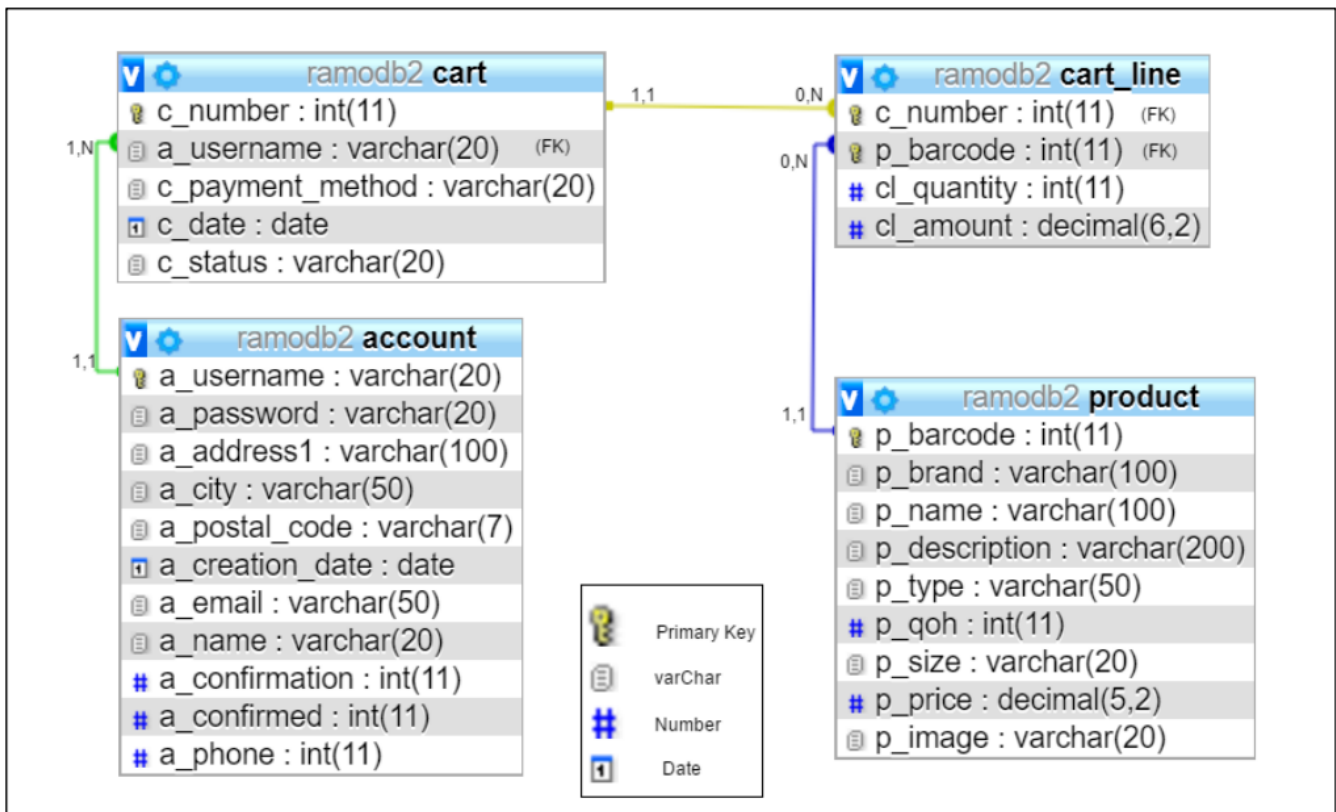
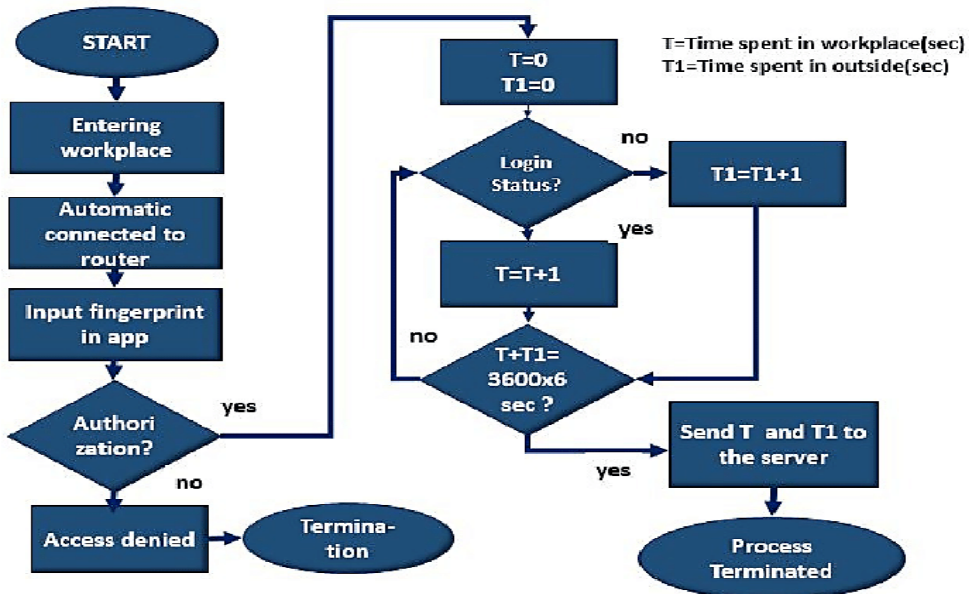
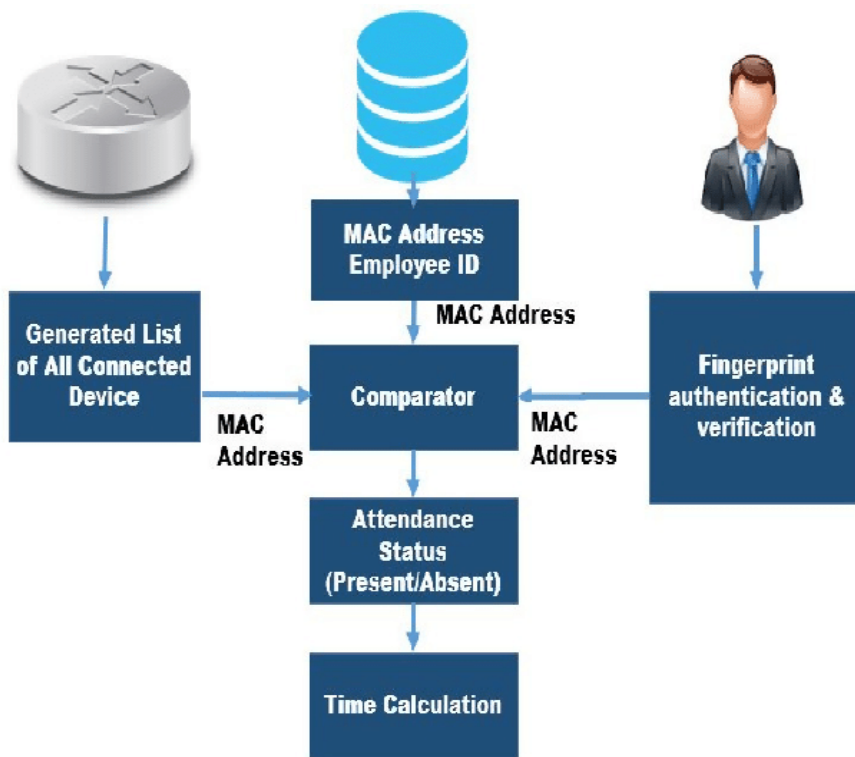
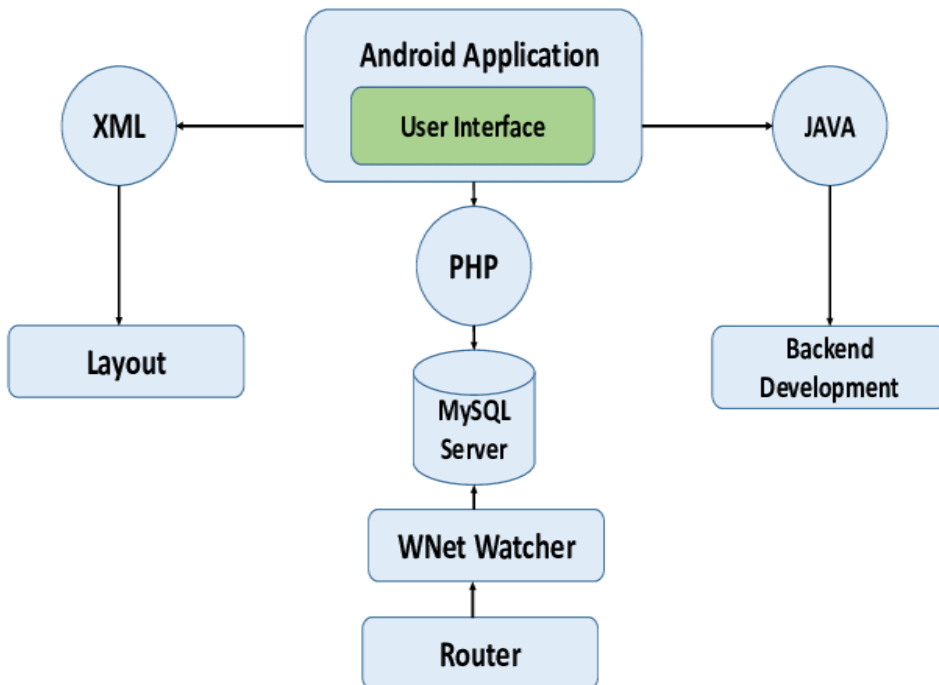
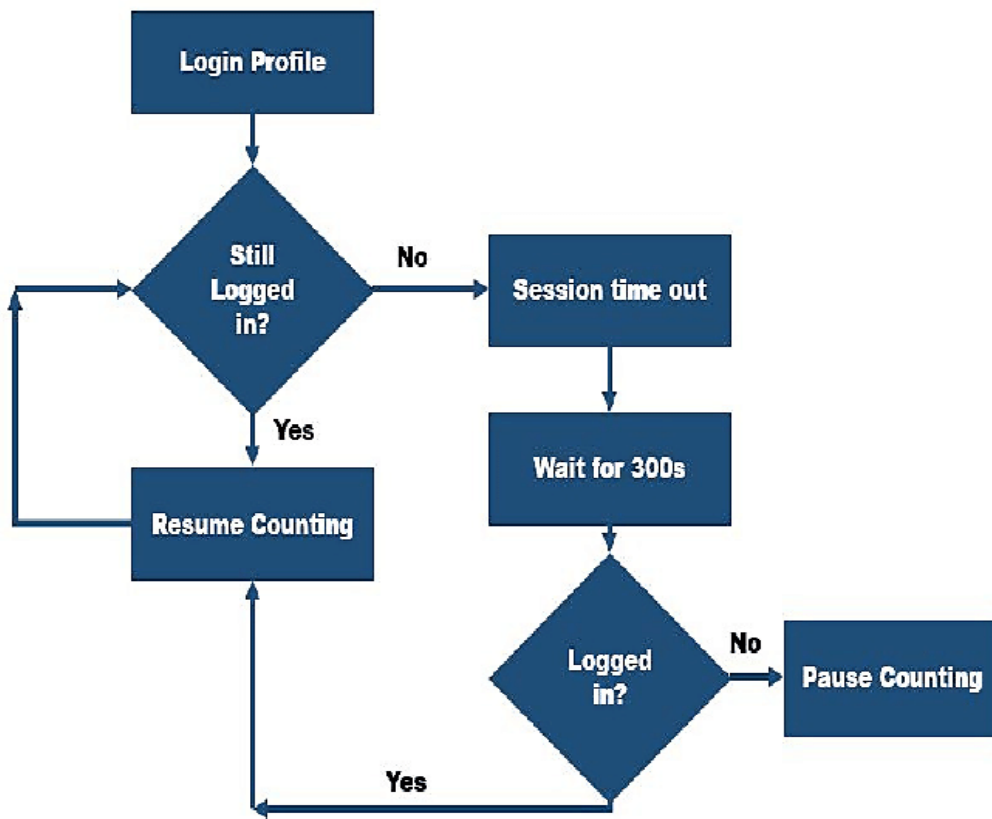


Figure 13 Entity Relationship Diagram Made using PHPMYAdmin

## Authentic Guards:

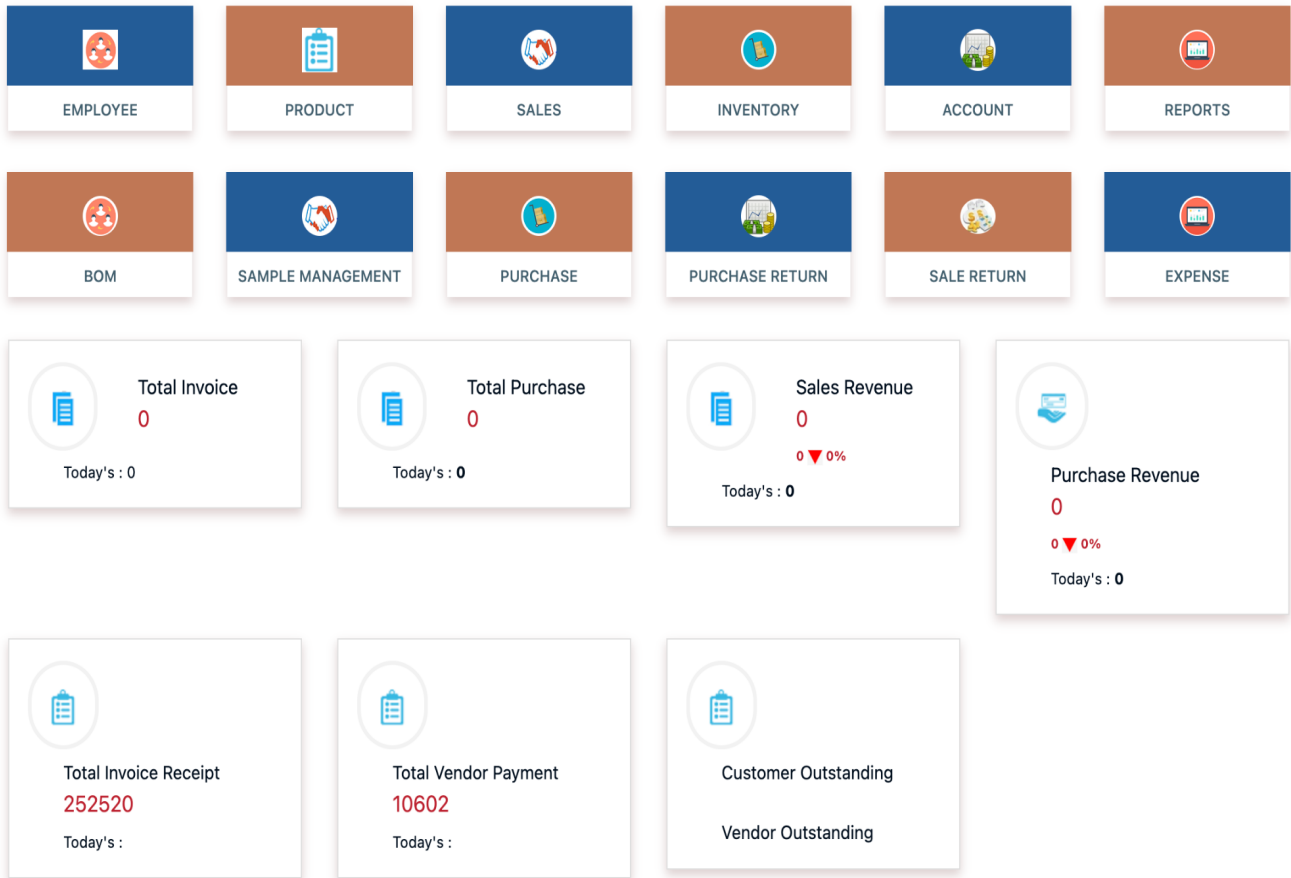






# Rk Construction:

## Dashboard



- Dashboard

---

- Master Forms >

---

- Employee

---

- BOM

---

- AMC

---

- Sample Item Management >

---

- Sales >

- Purchase >

---

- Payment >

---

- Sale Return (Credit Note)

---












- Purchase Return (Debit Note)











---

- Reports >

---

- Logout

-  Dashboard
-  Master Forms ▼
  -  Designation
  -  Number Plan
  -  Customer
  -  Vendor
  -  Unit
  -  Make Name
  -  Add Category
  -  Raw Material
  -  Add FG Category

-  FG Master
-  Architech
-  Add Bank Details
-  Payment Terms
-  Delivery Period
-  Warranty
-  Terms & Conditions
-  Expense
-  Customer Opening balance
-  Vendor Opening balance

**Designation**










Designation

Submit

**Designation List**

Show  entries

Search:

Sr	Designation	Edit	Delete
1	AREA SALES MANAGER		<span style="color: red;">✕</span>
2	TECHNICAL INCHARGE		<span style="color: red;">✕</span>
3	CEO		<span style="color: red;">✕</span>
4	MD		<span style="color: red;">✕</span>
5	SALES EXECUTIVE		<span style="color: red;">✕</span>
6	Accountant		<span style="color: red;">✕</span>
7	purchase		<span style="color: red;">✕</span>
8	Design Engineer		<span style="color: red;">✕</span>
9	Tool Room Supper Wiser		<span style="color: red;">✕</span>

### Sample Item Management

Sample Order Number	<input type="text"/>	Sample Order Date	<input type="text" value="2023-05-12"/>
Customer Code	<b>Direct Sale( Please Select Customer Code)</b> <input type="text" value="Please Select Customer"/>	Customer Name	<input type="text"/>
Company Address	<input type="text"/>	Contact Detail	<input type="text"/>
Payment Terms	<input type="text" value="Select Payment Terms"/>	Shipping and Handling Charges (optional)	<input type="text"/>
Warranty	<input type="text"/>	Terms & Condition	<input type="text" value="Select"/>
Remark	<input type="text"/>	Currency	<input type="text" value="INR"/>
Delivery Terms	<input type="text"/>	Place Of Delivery	<input type="text"/>
Sample Received Date	<input type="text"/>		

Sr No	FG Category	FG Name	FG Code	HSN	Qty	Price	Discount (%)	S Total	SGST (%)	CGST (%)	IGST (%)	Total	
0	<input type="text" value="Select Categor"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Save"/>

### Purchase Order

Purchase Order Number	<input type="text"/>	Purchase Order Date	<input type="text" value="2023-05-12"/>
Vendor Reference	<input type="text"/>	Vendor Reference Date	<input type="text"/>
Vendor Name	<input type="text" value="Select Vendor"/>	Company Address	<input type="text"/>
Contact Detail	<input type="text"/>	Payment Terms	<input type="text" value="Select Payment Terms"/>
Terms & Conditions	<input type="text" value="Select"/>	Delivery Terms	<input type="text"/>
Remark	<input type="text"/>	Currency	<input type="text" value="INR"/>

Sr No	Category	Sub Category	RM Code	Price	Quantity	Discount%	Amount	SGST (%)	CGST (%)	IGST (%)	Total	
0	<input type="text" value="Select Ca"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Save"/>

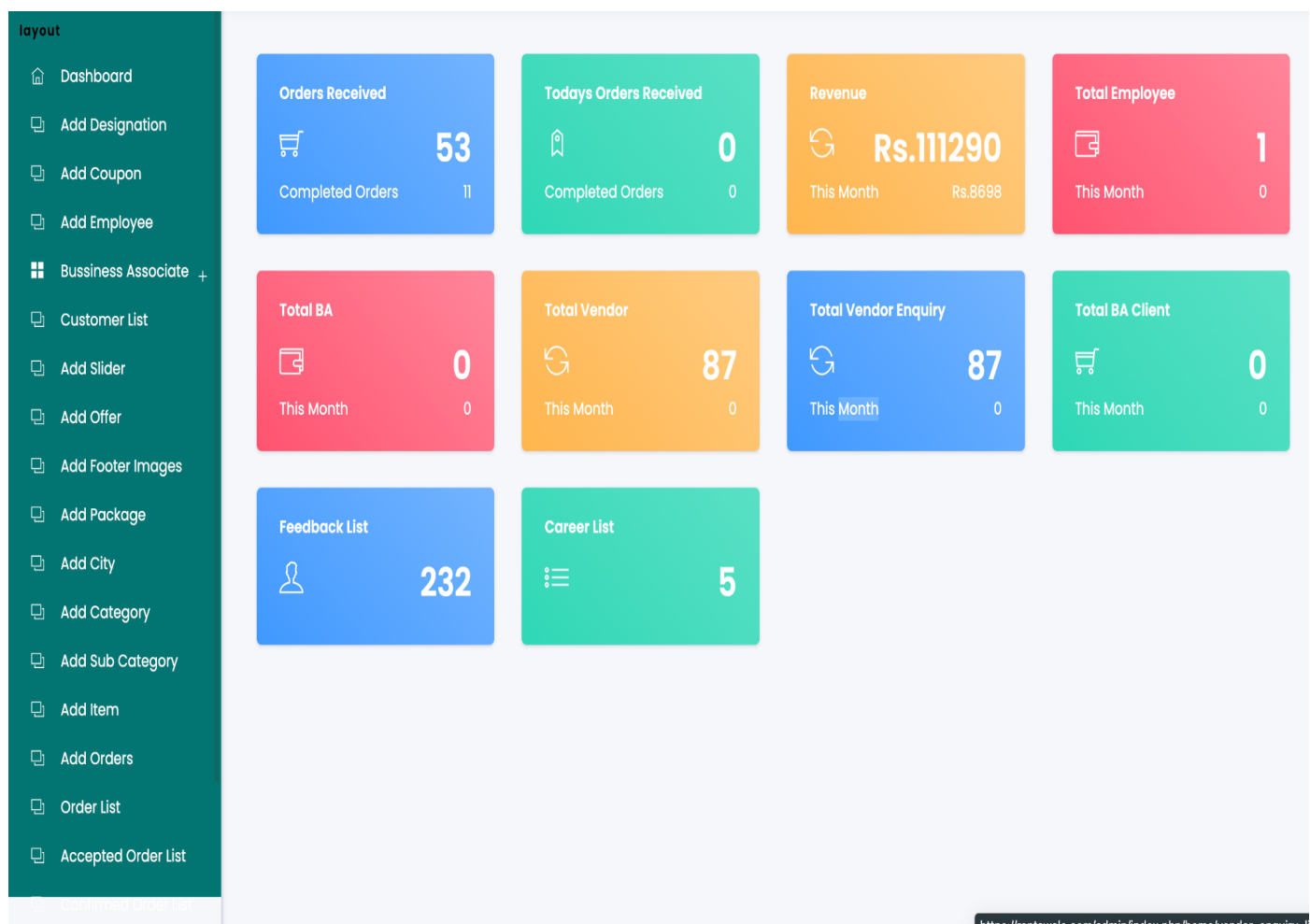
## Mobile Application Development Process:

For all the above mentioned projects , developing the frontend of the mobile applications in Flutter framework using Dart language was my job responsibility. Frontend of the mobile application, was developed based on the webadmin panel of the project. Screens and Designs , StateManagement , Data flow, Restful APIS integration , local Database management, file picker , image pickers, geolocator, geolocation etc , different features and packages were added on frontend and managed accordingly..

## Rentswale Mobile application Development:

As mentioned above Rentswale is an ecommerce mobile application. Backend was developed using PHP codeigniter and frontend was developed using Dart language in Flutter framework, for Database MySQLite was used.

## Web admin panel :



### ADD PACKAGE

Package name  Amount

Description  No. of post

Days

### Package List

Show

10

entries

Search:

Sr	Package	Amount	Description	No.of post	Days	Action
1	Free Package	0	testing_001	05	30	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Package 4	1	hi	2	20	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Package 3	2000	csd fhdhdf	30	90	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

### ADD CATEGORY

Name  Upload Image  No file chosen

### Category List

Show

10

entries

Search:

Sr	Category Name	Image
1	Services	
2	Furniture	
3	Medical Equipment	

### ADD SUB CATEGORY

Category  Sub Category

Upload Image  No file chosen

### Sub Category List

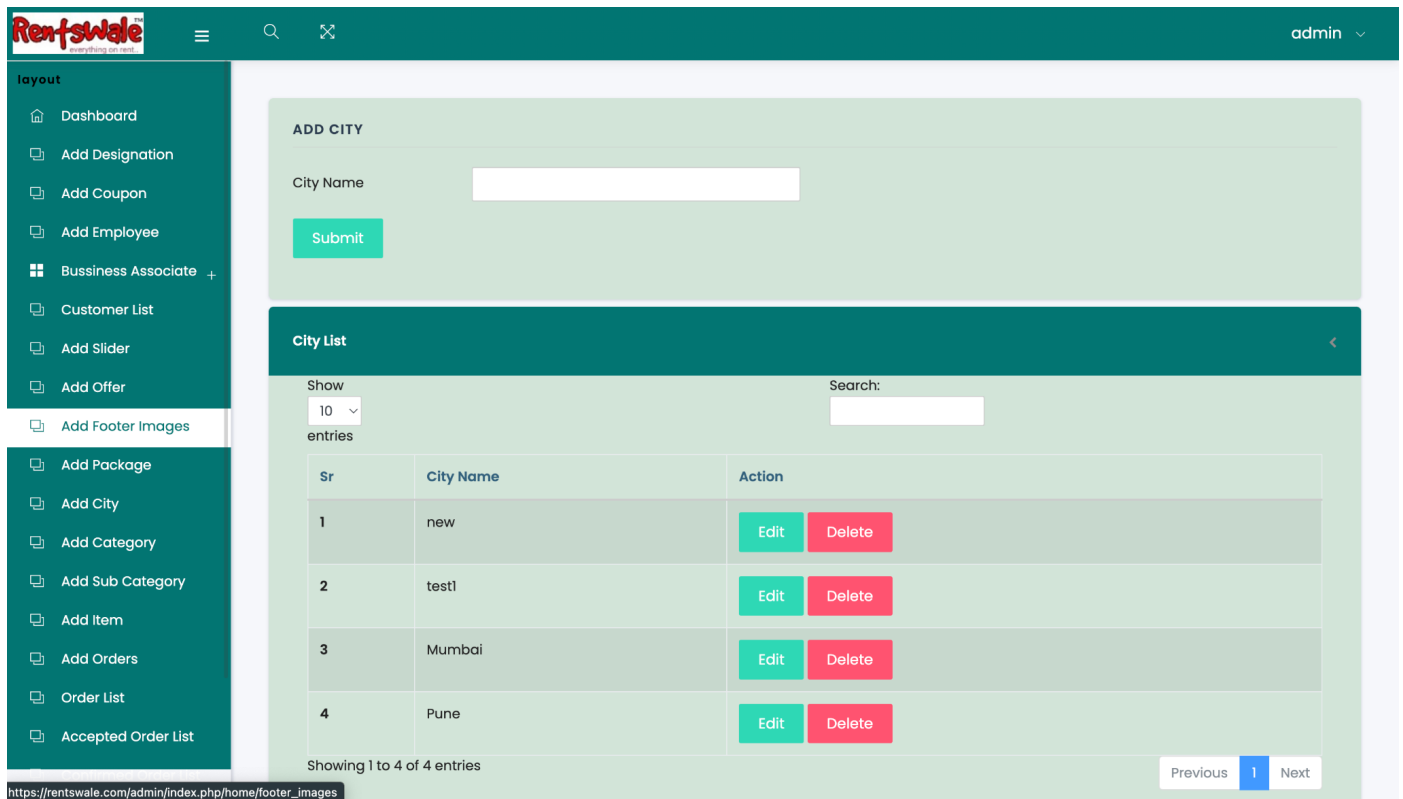
Show

10

entries

Search:

Sr	Category Name	Sub Category	Action
1	Electronics	UPS & Genset	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Electronics	Water Purifier	<input type="button" value="Edit"/> <input type="button" value="Delete"/>



Based on the above web admin panel restful apis were developed by the backend developers so that, Admin can perform crud operations on frontend.,

Backend : PHP,

Database: MYSQL

**Frontend Development:**

Language: Dart

Framework: Flutter

API Testing Platform : Postman

Software for application development: Android studio

Third party Packages used:

**http: ^0.13.6**

An extensible library for sending HTTP requests that is future-based.

This package contains a set of high-level classes and functions that make it simple to consume HTTP resources. It supports browsers, desktops, and mobile devices and is cross-platform.

```
import 'package:http/http.dart' as http;

var url = Uri.https('example.com', 'whatsit/create');
var response = await http.post(url, body: {'name': 'doodle', 'color': 'blue'});
print('Response status: ${response.statusCode}');
print('Response body: ${response.body}');

print(await http.read(Uri.https('example.com', 'foobar.txt')));
```

```
var client = http.Client();
try {
  var response = await client.post(
    Uri.https('example.com', 'whatsit/create'),
    body: {'name': 'doodle', 'color': 'blue'});
  var decodedResponse = jsonDecode(utf8.decode(response.bodyBytes)) as Map;
  var uri = Uri.parse(decodedResponse['uri'] as String);
  print(await client.get(uri));
} finally {
  client.close();
}
```

### **shared\_preferences:**

wraps simple data's platform-specific persistent storage (SharedPreferences on Android, NSUserDefaults on iOS and macOS, etc.). Because there is no assurance that writes will be saved to disc after returning and because data may be written to disc asynchronously, this plugin shouldn't be used to save important data.

The data types `int`, `double`, `bool`, `String`, and `ListString>` are supported.

Support SDK 16+ for Android iOS Linux MacOS Web11.0+ Any 10.11+ Any Any



```

// Obtain shared preferences.
final SharedPreferences prefs = await SharedPreferences.getInstance();

// Save an integer value to 'counter' key.
await prefs.setInt('counter', 10);
// Save an boolean value to 'repeat' key.
await prefs.setBool('repeat', true);
// Save an double value to 'decimal' key.
await prefs.setDouble('decimal', 1.5);
// Save an String value to 'action' key.
await prefs.setString('action', 'Start');
// Save an list of strings to 'items' key.
await prefs.setStringList('items', <String>['Earth', 'Moon', 'Sun']);

```

```

// Try reading data from the 'counter' key. If it doesn't exist, returns null.
final int? counter = prefs.getInt('counter');
// Try reading data from the 'repeat' key. If it doesn't exist, returns null.
final bool? repeat = prefs.getBool('repeat');
// Try reading data from the 'decimal' key. If it doesn't exist, returns null.
final double? decimal = prefs.getDouble('decimal');
// Try reading data from the 'action' key. If it doesn't exist, returns null.
final String? action = prefs.getString('action');
// Try reading data from the 'items' key. If it doesn't exist, returns null.
final List<String>? items = prefs.getStringList('items');

```

### **provider:**

The tool for managing global states that is currently officially recommended is called Provider, and it was co-written by Flutter Team. We have to create a dependency on it before we can use it. The most recent version of Provider as of this writing is 4.0.4:

We focus on three ideas in particular when using Provider:

ChangeNotifier: the location of the actual data (state)

Wherever data (status) is provided in the widget tree, the relevant ChangeNotifier will be created in it using the ChangeNotify Provider.

Consumer: In which widgets in the tree should data (state) be used.

## Create your own ChangeNotifier:

```
class CounterProvider extends ChangeNotifier {
  int _counter = 100;
  int get counter {
    return _counter;
  }
  set counter(int value) {
    _counter = value;
    notifyListeners();
  }
}
```

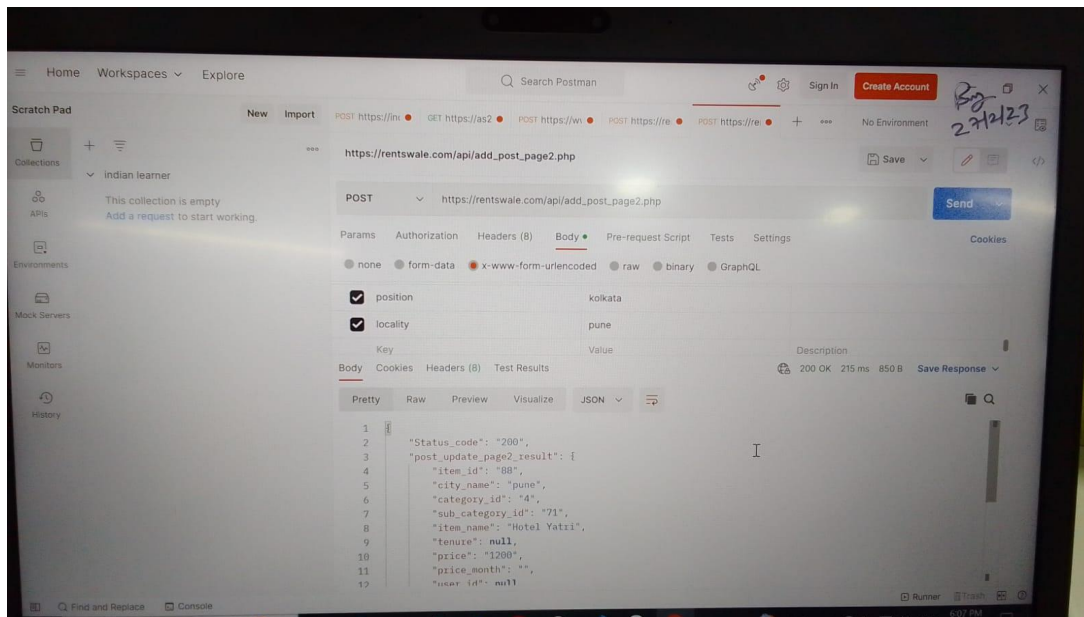
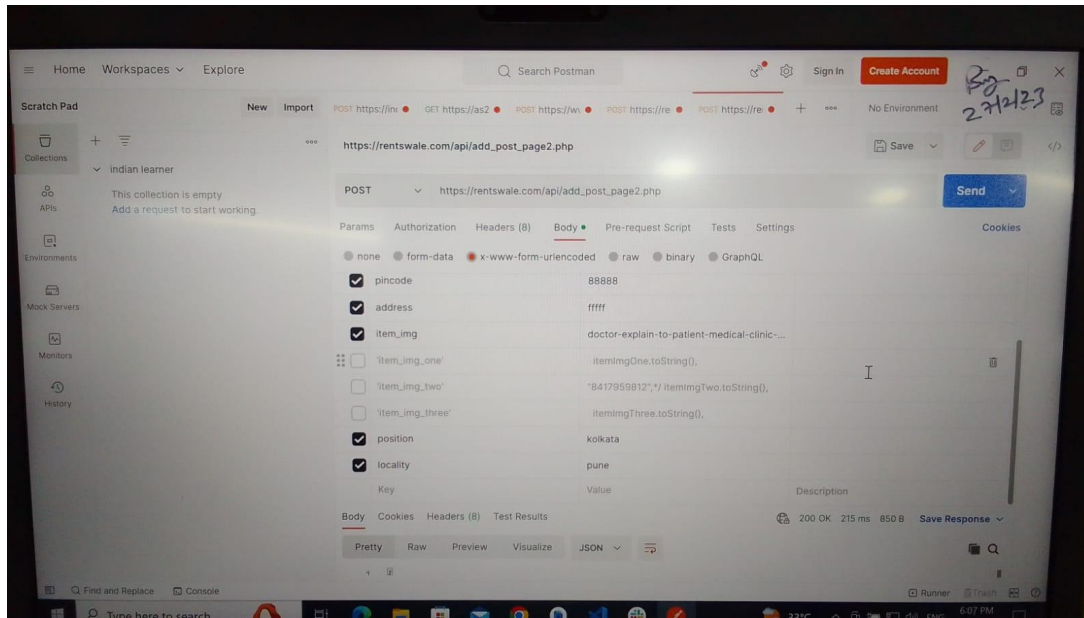
## Insert ChangeNotifier in the Widget :

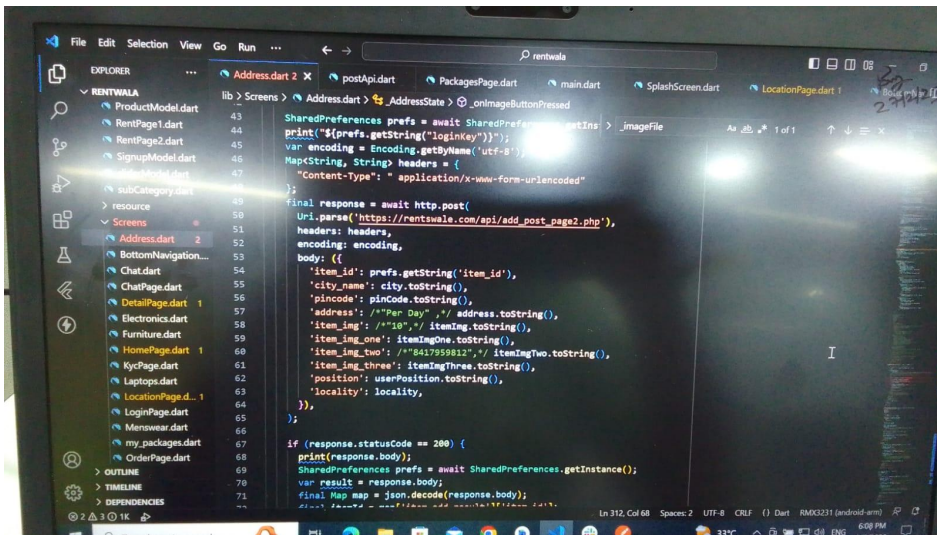
```
void main() {
  runApp(ChangeNotifierProvider(
    create: (context) => CounterProvider(),
    child: MyApp(),
  ));
}
```

## Use Consumer to import and modify the state on the home page:

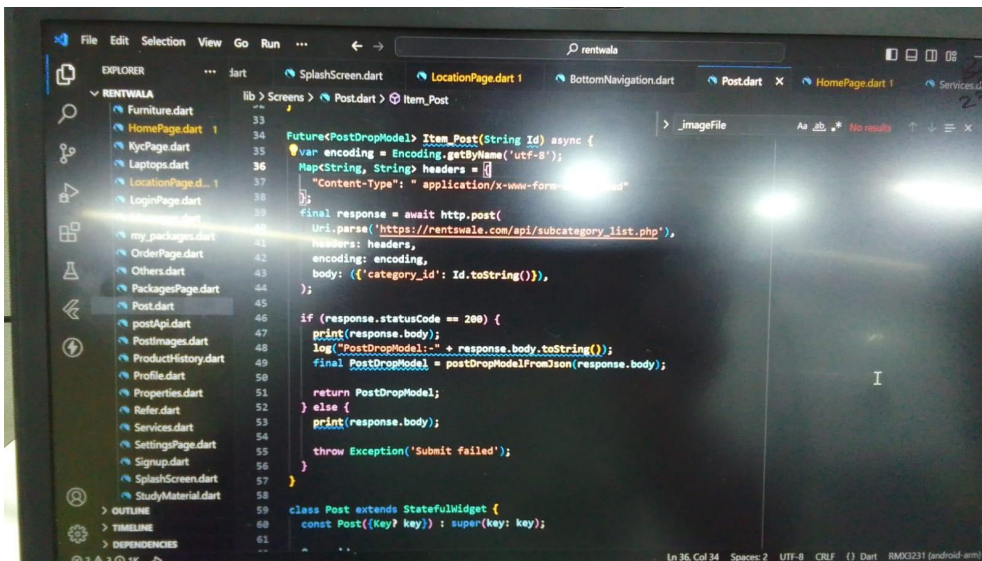
```
class HYHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Column Test"),
      ),
      body: Center(
        child: Consumer<CounterProvider>(
          builder: (ctx, counterPro, child) {
            return Text("Current Value:${counterPro.counter}", style: TextStyle(fontSize:
            )
          ),
        ),
      ),
      floatingActionButton: Consumer<CounterProvider>(
        builder: (ctx, counterPro, child) {
          return FloatingActionButton(
            child: child,
            onPressed: () {
              counterPro.counter += 1;
            },
          );
        },
      ),
      child: Icon(Icons.add),
    );
  }
}
```

# Real Time Implementation :

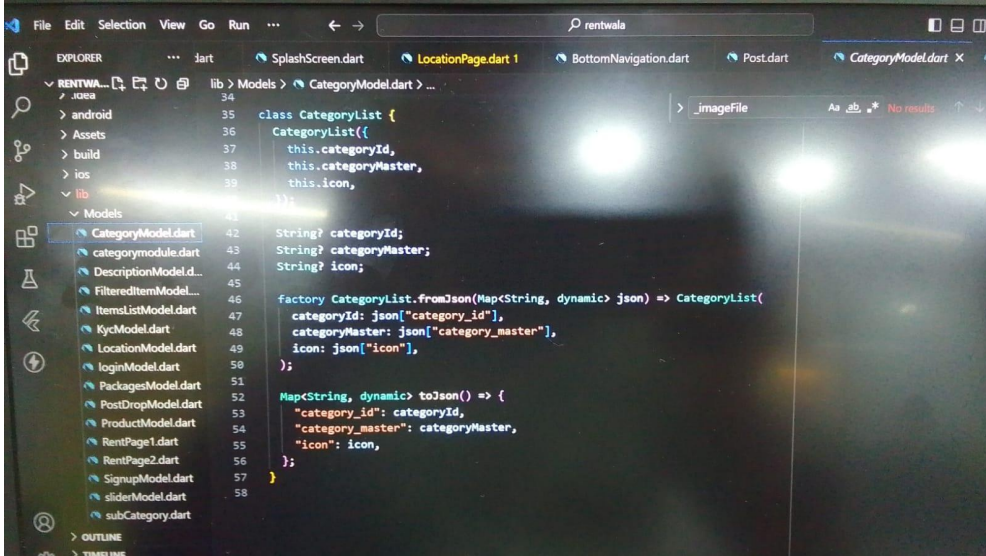




```
AddressState > _onImageButtonPressed
SharedPreferences prefs = await SharedPreferences.getInstance();
print("${prefs.getString("loginkey")}");
var encoding = Encoding.getByName("utf-8");
Map<String, String> headers = {
  "Content-Type": "application/x-www-form-urlencoded"
};
final response = await http.post(
  Uri.parse('https://rentswale.com/api/add_post_page2.php'),
  headers: headers,
  encoding: encoding,
  body: ({
    'item_id': prefs.getString('item_id'),
    'city_name': city.toString(),
    'pincode': pinCode.toString(),
    'address': /*"Per Day"*/ address.toString(),
    'item_img': /*"10"*/ itemImg.toString(),
    'item_img_one': itemImgOne.toString(),
    'item_img_two': /*"217959811"*/ itemImgTwo.toString(),
    'item_img_three': itemImgThree.toString(),
    'position': userPosition.toString(),
    'locality': locality,
  })),
);
if (response.statusCode == 200) {
  print(response.body);
  SharedPreferences prefs = await SharedPreferences.getInstance();
  var result = response.body;
  final Map map = json.decode(response.body);
}
```



```
Post.dart > Item_Post
Future<PostDropModel> Item_Post(String Id) async {
  var encoding = Encoding.getByName('utf-8');
  Map<String, String> headers = {
    "Content-Type": "application/x-www-form-urlencoded"
  };
  final response = await http.post(
    Uri.parse('https://rentswale.com/api/subcategory_list.php'),
    headers: headers,
    encoding: encoding,
    body: ({'category_id': Id.toString()}),
  );
  if (response.statusCode == 200) {
    print(response.body);
    log("PostDropModel:-" + response.body.toString());
    final PostDropModel = postDropModelFromJson(response.body);
  }
  return PostDropModel;
} else {
  print(response.body);
  throw Exception('Submit failed');
}
}
class Post extends StatefulWidget {
  const Post({Key? key}) : super(key: key);
}
```



```
CategoryModel.dart > CategoryModel
class CategoryList {
  CategoryList({
    this.categoryId,
    this.categoryMaster,
    this.icon,
  });
  String? categoryId;
  String? categoryMaster;
  String? icon;
  factory CategoryList.fromJson(Map<String, dynamic> json) => CategoryList(
    categoryId: json["category_id"],
    categoryMaster: json["category_master"],
    icon: json["icon"],
  );
  Map<String, dynamic> toJson() => {
    "category_id": categoryId,
    "category_master": categoryMaster,
    "icon": icon,
  };
}
```

```
EXPLORER    ... dart    SplashScreen.dart    LocationPage.dart 1    BottomNavigation.dart    Post.dart    categorymodule.dart X

lib > Models > categorymodule.dart > CategoryModule
> .idea
> android
> Assets
> build
> ios
> lib
  > Models
    CategoryModel.dart
    categorymodule.dart
    DescriptionModel.d...
    FilteredItemModel...
    ItemsListModel.dart
    KycModel.dart
    LocationModel.dart
    loginModel.dart
    PackagesModel.dart
    PostDropModel.dart
    ProductModel.dart
    RentPage1.dart
    RentPage2.dart
    SignupModel.dart
    sliderModel.dart
    subCategory.dart
  > OUTLINE
  > TIMELINE
  > DEPENDENCIES

1 class CategoryModule {
2   String? statusCode;
3   String? message;
4   List<ProductList>? productList;
5
6   CategoryModule({this.statusCode, this.message, this.productList});
7
8   CategoryModule.fromJson(Map<String, dynamic> json) {
9     statusCode = json['status_code'];
10    message = json['message'];
11    if (json['product_list'] != null) {
12      productList = <ProductList>[];
13      json['product_list'].forEach((v) {
14        productList!.add(new ProductList.fromJson(v));
15      });
16    }
17  }
18
19  Map<String, dynamic> toJson() {
20    final Map<String, dynamic> data = new Map<String, dynamic>();
21    data['status_code'] = this.statusCode;
22    data['message'] = this.message;
23    if (this.productList != null) {
24      data['product_list'] = this.productList!.map(v => v.toJson()).toList();
25    }
26    return data;
27  }
28 }
29
30 class ProductList {
```

```
EXPLORER    ... dart    SplashScreen.dart    LocationPage.dart 1    BottomNavigation.dart    Post.dart    PackagesModel.dart X

lib > Models > PackagesModel.dart > PackageList
> .idea
> android
> Assets
> build
> ios
> lib
  > Models
    CategoryModel.dart
    categorymodule.dart
    DescriptionModel.d...
    FilteredItemModel...
    ItemsListModel.dart
    KycModel.dart
    LocationModel.dart
    loginModel.dart
    PackagesModel.dart
    PostDropModel.dart
    ProductModel.dart
    RentPage1.dart
    RentPage2.dart
    SignupModel.dart
    sliderModel.dart
    subCategory.dart
  > OUTLINE
  > TIMELINE
  > DEPENDENCIES

35 class PackageList {
36   PackageList({
37     this.packageId,
38     this.packageName,
39     this.description,
40     this.amount,
41     this.noOfPost,
42     this.days,
43   });
44
45   String? packageId;
46   String? packageName;
47   String? description;
48   String? amount;
49   String? noOfPost;
50   String? days;
51
52   factory PackageList.fromJson(Map<String, dynamic> json) => PackageList(
53     packageId: json["package_id"],
54     packageName: json["package_name"],
55     description: json["description"],
56     amount: json["amount"],
57     noOfPost: json["no_of_post"],
58     days: json["days"],
59   );
60
61   Map<String, dynamic> toJson() => {
62     "package_id": packageId,
63     "package_name": packageName,
64     "description": description
```

## Authentic Guards:

Third party Packages and Plugins used:

### Geolocator:

A Flutter geolocation plugin that makes it simple to use platform-specific location services (FusedLocationProviderClient, or LocationManager on Android and CLLocationManager on iOS if not available), now available.

### Features

Obtain the most recent location;

Obtain the device's current position;

receive ongoing location updates;

Verify the device's location services are turned on;

Determine the distance between two geocoordinates (in meters);

Bearing between two geo coordinates should be calculated.

```
import 'package:geolocator/geolocator.dart';

/// Determine the current position of the device.
///
/// When the location services are not enabled or permissions
/// are denied the `Future` will return an error.
Future<Position> _determinePosition() async {
  bool serviceEnabled;
  LocationPermission permission;

  // Test if location services are enabled.
  serviceEnabled = await Geolocator.isLocationServiceEnabled();
  if (!serviceEnabled) {
    // Location services are not enabled don't continue
    // accessing the position and request users of the
    // App to enable the location services.
    return Future.error('Location services are disabled.');
```

## Image picker:

a Flutter plugin for iOS and Android that allows users to select photos from their photo library and take brand-new photos using the camera.

---

```
dependencies:  
  flutter:  
    sdk: flutter  
  image_picker: ^0.6.7+6
```

---

## Configure Android:

```
<application  
    android:requestLegacyExternalStorage="true"  
    android:name="io.flutter.app.FlutterApplication"  
    android:label="xxxxxx"  
    android:icon="@mipmap/launcher_icon">  
  <activity>  
    ...  
    ...  
  </activity>  
</application>
```

---

## Pick Image From Gallery:

```
/// Get from gallery  
_getFromGallery() async {  
  PickedFile pickedFile = await ImagePicker().getImage(  
    source: ImageSource.gallery,  
    maxWidth: 1800,  
    maxHeight: 1800,  
  );  
  if (pickedFile != null) {  
    File imageFile = File(pickedFile.path);  
  }  
}
```

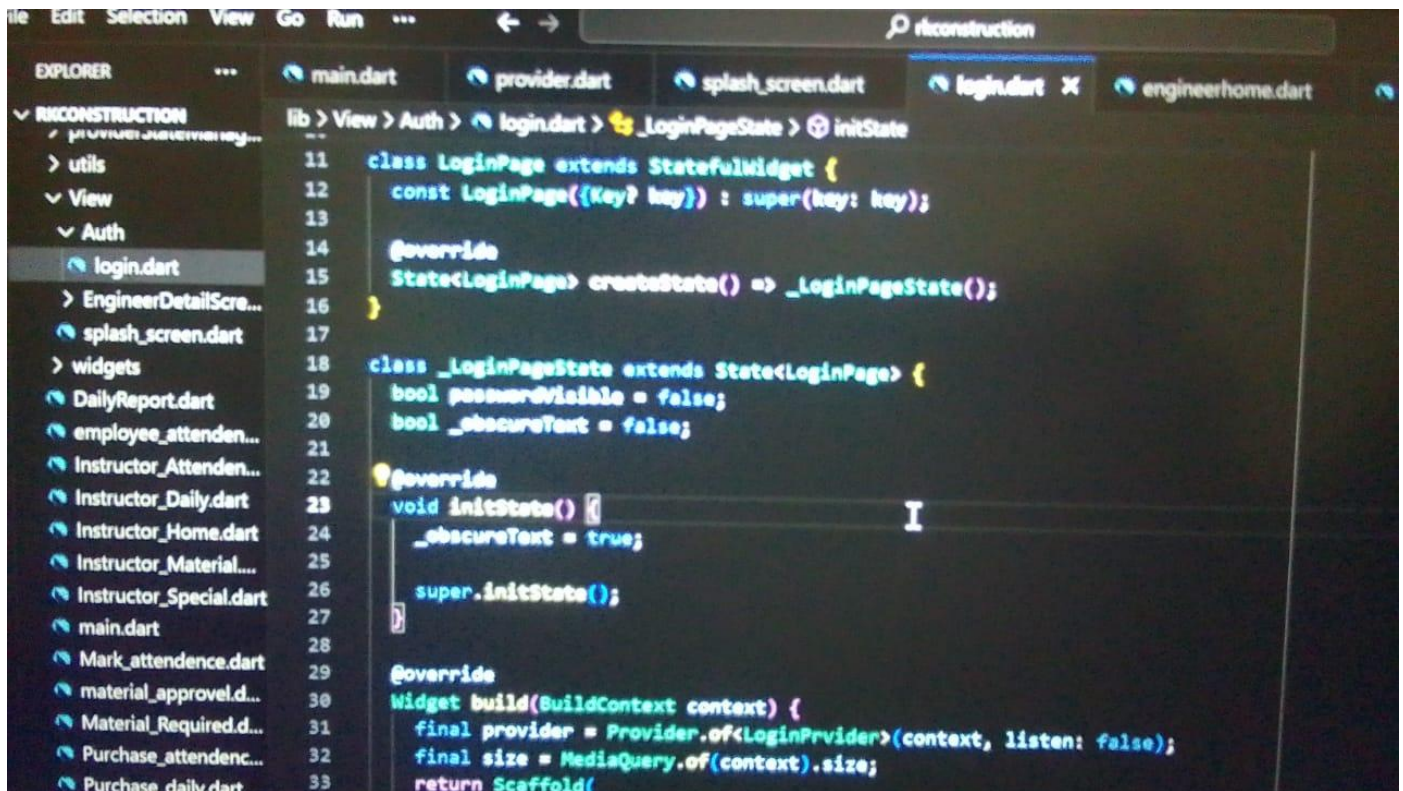
---

## Pick Image From Camera;

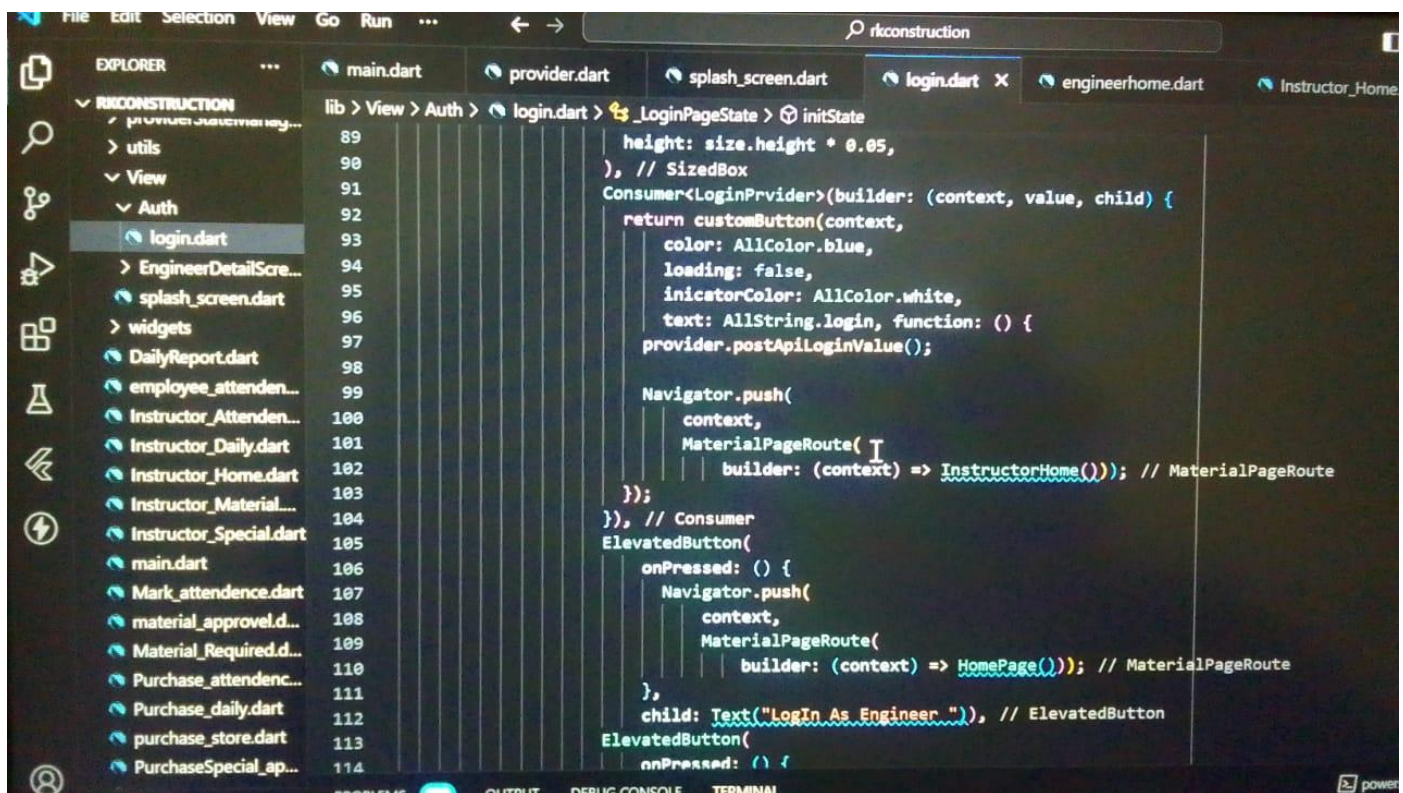
```
/// Get from camera  
_getFromCamera() async {  
  PickedFile pickedFile = await ImagePicker().getImage(  
    source: ImageSource.camera,  
    maxWidth: 1800,  
    maxHeight: 1800,  
  );  
  if (pickedFile != null) {  
    File imageFile = File(pickedFile.path);  
  }  
}
```

---

## RK Construction:



```
lib > View > Auth > login.dart > _LoginPageState > initState
11 class LoginPage extends StatefulWidget {
12   const LoginPage({Key? key}) : super(key: key);
13
14   @override
15   State<LoginPage> createState() => _LoginPageState();
16 }
17
18 class _LoginPageState extends State<LoginPage> {
19   bool passwordVisible = false;
20   bool _obscureText = false;
21
22   @override
23   void initState() {
24     _obscureText = true;
25     super.initState();
26   }
27
28   @override
29   Widget build(BuildContext context) {
30     final provider = Provider.of<LoginPrvider>(context, listen: false);
31     final size = MediaQuery.of(context).size;
32     return Scaffold(
33
```



```
89     height: size.height * 0.05,
90   ), // SizedBox
91   Consumer<LoginPrvider>(builder: (context, value, child) {
92     return customButton(context,
93       color: AllColor.blue,
94       loading: false,
95       indicatorColor: AllColor.white,
96       text: AllString.login, function: () {
97         provider.postApiLoginValue();
98
99         Navigator.push(
100           context,
101           MaterialPageRoute(
102             builder: (context) => InstructorHome()); // MaterialPageRoute
103       ));
104   } // Consumer
105   ElevatedButton(
106     onPressed: () {
107       Navigator.push(
108         context,
109         MaterialPageRoute(
110           builder: (context) => HomePage()); // MaterialPageRoute
111       ),
112       child: Text("LogIn As Engineer."), // ElevatedButton
113     ElevatedButton(
114     onPressed: () {
```



```
EXPLORER    ...    main.dart    provider.dart    splash_screen.dart    login.dart X    engineerhome.dart    Instructor_H...

RKCONSTRUCTION
  > provider
  > utils
  > View
  > Auth
    login.dart
  > EngineerDetailScre...
  splash_screen.dart
  > widgets
    DailyReport.dart
    employee_attenden...
    Instructor_Attenden...
    Instructor_Daily.dart
    Instructor_Home.dart
    Instructor_Material...
    Instructor_Special.dart
    main.dart
    Mark_attendance.dart
    material_approvel.d...
    Material_Required.d...
    Purchase_attendenc...
    Purchase_daily.dart
    purchase_store.dart

lib > View > Auth > login.dart > _LoginPageState > initState
30  Widget build(BuildContext context) {
31    final provider = Provider.of<LoginPrvider>(context, listen: false);
32    final size = MediaQuery.of(context).size;
33    return Scaffold(
34      body: SafeArea(
35        child: Center(
36          child: Column(
37            mainAxisAlignment: MainAxisAlignment.center,
38            children: [
39              Text("Login"),
40              SizedBox(
41                height: size.height * 0.05,
42              ), // SizedBox
43              Container(
44                height: size.height * 0.50,
45                width: size.width * 0.80,
46                decoration: BoxDecoration(
47                  borderRadius: BorderRadius.circular(30),
48                  border: Border.all(), // BoxDecoration
49                child: Column(
50                  children: [
51                    SizedBox(
52                      height: size.height * 0.05,
53                    ), // SizedBox
54                    TextInput(
```

```
File Edit Selection View Go Run ...    rkconstruction
EXPLORER    ...    main.dart    provider.dart    splash_screen.dart    login.dart X    engineerhome.dart    Instructor_Home.dart

RKCONSTRUCTION
  > provider
  > utils
  > View
  > Auth
    login.dart
  > EngineerDetailScre...
  splash_screen.dart
  > widgets
    DailyReport.dart
    employee_attenden...
    Instructor_Attenden...
    Instructor_Daily.dart
    Instructor_Home.dart
    Instructor_Material...
    Instructor_Special.dart
    main.dart
    Mark_attendance.dart
    material_approvel.d...
    Material_Required.d...
    Purchase_attendenc...
    Purchase_daily.dart
    purchase_store.dart
    PurchaseSpecial_ap...

lib > View > Auth > login.dart > _LoginPageState > initState
132  class TextInput extends StatelessWidget {
133    final double? width;
134    final double? height;
135    final Widget? prefixIcon;
136    final Widget? suffixIcon;
137    final TextEditingController? textEditingController;
138    bool? obscureText;
139    final String? hintText;
140
141    TextInput(
142      {Key? key,
143        this.width,
144        this.height,
145        this.textEditingController,
146        this.prefixIcon,
147        this.suffixIcon,
148        this.obscureText = false,
149        this.hintText})
150      : super(key: key);
151
152    @override
153    Widget build(BuildContext context) {
154      return SizedBox(
155        width: width,
156        height: height,
157        child: TextFormField(
```

```
lib > View > splash_screen.dart > _SplashScreenState > initState
81 @override
82 Widget build(BuildContext context) {
83   screenWidth = MediaQuery.of(context).size.width;
84   screenHeight = MediaQuery.of(context).size.height;
85   return Scaffold(
86     backgroundColor: AllColor.white,
87     body: Center(
88       child: Container(
89         height: screenHeight,
90         width: screenWidth,
91         color: AllColor.white,
92         child: Stack(
93           children: [
94             Container(
95               height: screenHeight,
96               width: screenWidth,
97               child: Center(
98                 child: SizedBox(
99                   height: animation.value,
100                  width: animation.value,
101                  child: Image.asset("Assets/Images/RK Engineering 2.png"), // SizedBox
102                ), // Center
103              ), // Container
104            ],
105          ), // Stack
106        ), // Container
```

```
lib > View > splash_screen.dart > _SplashScreenState > initState
55 @override
56 void dispose() {
57   animationController.dispose();
58   super.dispose();
59 }
60
61 @override
62 void initState() {
63   // TODO: implement initState
64   super.initState();
65
66   animationController =
67     AnimationController(vsync: this, duration: Duration(seconds: 3));
68
69   animation = Tween(begin: 0.0, end: 230.0).animate(animationController);
70   animationController.addListener(() {
71     setState(() {});
72   });
73   animationController.forward();
74
75   Future.delayed(Duration(seconds: 3)).then((value) {
76     Navigator.of(context).pushReplacement(
77       CupertinoPageRoute(builder: (BuildContext context) => LoginPage());
78     );
79   });
80 }
```

EXPLORER ... main.dart provider.dart splash\_screen.dart login.dart engineerhome.dart Instructor\_Ho

lib > Instructor\_Home.dart > InstructorHome

```

33   body: SingleChildScrollView(
34     child: Column(
35       mainAxisAlignment: MainAxisAlignment.start,
36       crossAxisAlignment: CrossAxisAlignment.start,
37       children: [
38         SizedBox(
39           height: size.height * 0.01,
40         ), // SizedBox
41         SizedBox(
42           child: Center(
43             child: Text(
44               "Heroes of the week",
45               style: TextStyle(fontWeight: FontWeight.bold, fontSize: 20),
46             ), // Text
47           ), // Center
48         ), // SizedBox
49         Padding(
50           padding: const EdgeInsets.all(8.0),
51           child: SizedBox(
52             height: 120,
53             width: double.infinity,
54             child: ListView.builder(
55               scrollDirection: Axis.horizontal,
56               shrinkWrap: true,
57               itemBuilder: (context, index) {
58                 return SizedBox(

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\Wetizens\Desktop\Raaj\Rk\_Construction File\rkconstruction>

File Edit Selection View Go Run ... rkconstruction

EXPLORER ... main.dart provider.dart splash\_screen.dart login.dart engineerhome.dart Instructor\_Ho

lib > Instructor\_Home.dart > InstructorHome

```

7   class InstructorHome extends StatefulWidget {
8     const InstructorHome({Key? key}) : super(key: key);
9
10  @override
11  State<InstructorHome> createState() => _InstructorHomeState();
12  }
13
14  class _InstructorHomeState extends State<InstructorHome> {
15    String? _selectedValue1;
16    String? _selectedValue2;
17    @override
18    Widget build(BuildContext context) {
19      final size = MediaQuery.of(context).size;
20      return Scaffold(
21        appBar: AppBar(
22          leading: Icon(Icons.arrow_back, color: Colors.black),
23          backgroundColor: Colors.white,
24          title: Text(
25            "Instructor Interface",
26            style: TextStyle(
27              fontWeight: FontWeight.bold,
28              color: Colors.black,
29            ), // TextStyle
30          ), // Text
31          centerTitle: true,
32          // AppBar

```

```
File Edit Selection View Go Run ... rkconstruction
EXPLORER
  RKCONSTRUCTION
    provider.dart
    utils
    View
    Auth
    login.dart
    EngineerDetailScre...
    splash_screen.dart
    widgets
    DailyReport.dart
    employee_attenden...
    Instructor_Attenden...
    Instructor_Daily.dart
    Instructor_Home.dart
    Instructor_Material...
    Instructor_Special.dart
    main.dart
    Mark_attendance.dart
    material_approvel.d...
    Material_Required.d...
    Purchase_attendenc...
    Purchase_daily.dart
    purchase_store.dart
  providerStateManagement > provider.dart > LoginPrvider > postApiLoginValue
23 postApiLoginValue() {
24   postApiLogin().then((responsejson) async {
25     if (responsejson == null) {
26       log("No Data Found");
27       loaderResponse(false);
28     } else {
29       try {
30         loaderResponse(true);
31         notifyListeners();
32         final LoginModel1 = LoginModel.fromJson(jsonDecode(responsejson));
33         Map<String, dynamic> jsonData = jsonDecode(responsejson);
34
35         sharedPreferences.setString(
36           AllSharedPreferencesKey.phone,
37           jsonData["result"]["phone"].toString() == 'null'
38             ? ""
39             : jsonData["result"]["phone"].toString());
40
41         sharedPreferences.setString(
42           AllSharedPreferencesKey.user_type,
43           jsonData["result"]["user_type"].toString() == 'null'
44             ? ""
45             : jsonData["result"]["user_type"].toString());
46
47         sharedPreferences.setString(
48           AllSharedPreferencesKey.login_id
```

```
File Edit Selection View Go Run ... rkconstruction
EXPLORER
  RKCONSTRUCTION
    provider.dart
    utils
    View
    Auth
    login.dart
    EngineerDetailScre...
    splash_screen.dart
    widgets
    DailyReport.dart
    employee_attenden...
    Instructor_Attenden...
    Instructor_Daily.dart
    Instructor_Home.dart
    Instructor_Material...
    Instructor_Special.dart
    main.dart
    Mark_attendance.dart
    material_approvel.d...
    Material_Required.d...
    Purchase_attendenc...
    Purchase_daily.dart
    purchase_store.dart
  providerStateManagement > provider.dart > LoginPrvider > postApiLoginValue
1  import 'dart:convert';
2  import 'dart:developer';
3
4  import 'package:flutter/material.dart';
5  import 'package:rkconstruction/apiData.dart/loginApi.dart';
6  import 'package:rkconstruction/main.dart';
7  import 'package:rkconstruction/model/loginModel.dart';
8  import 'package:shared_preferences/shared_preferences.dart';
9
10 import '../utils/shearpreference/shearpreference.dart';
11
12 class LoginPrvider with ChangeNotifier {
13   bool _loader = false;
14
15   bool get loader => _loader;
16
17   loaderResponse(bool value) {
18     log("check:-" + value.toString());
19     _loader = value;
20     notifyListeners();
21   }
22
23   postApiLoginValue() {
24     postApiLogin().then((responsejson) async {
25       if (responsejson == null) {
26         log("No Data Found");
```

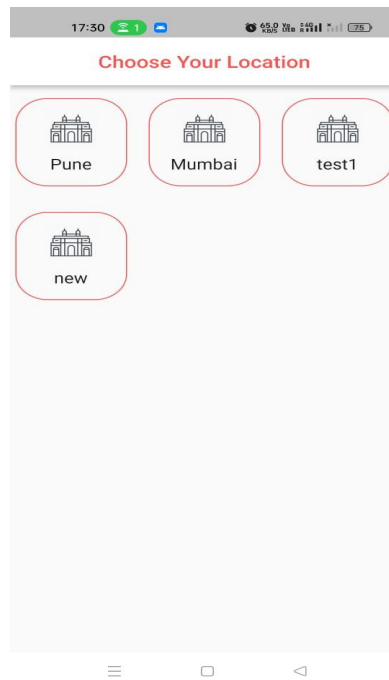
# CHAPTER: 4

## OUTPUTS & RESULT ANALYSIS :

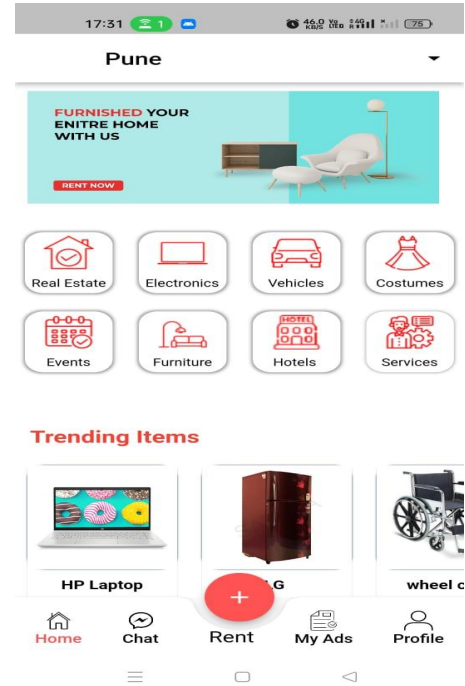
Rentswale Mobile Application Frontend:  
Splash screen:



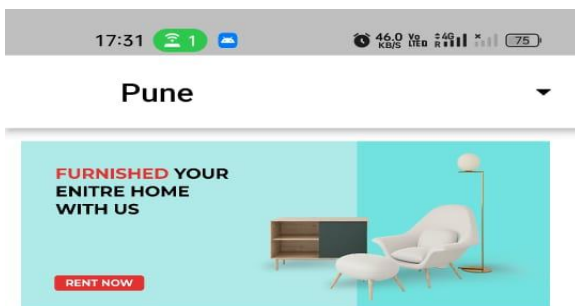
Company Logo with timer 3 sec



Location Screen to choose city



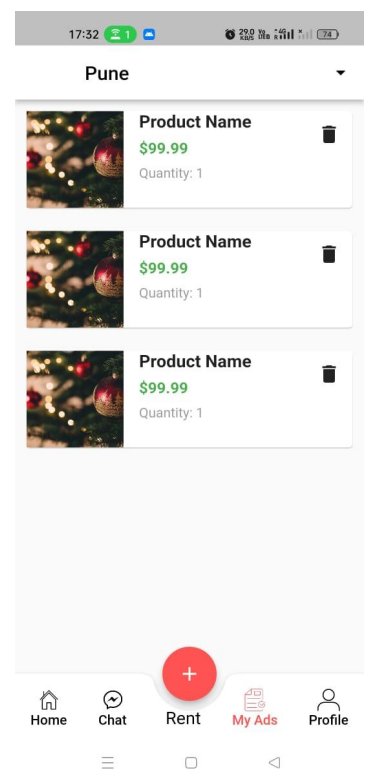
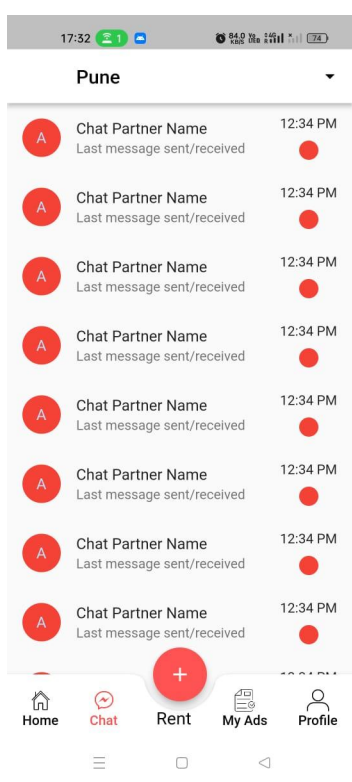
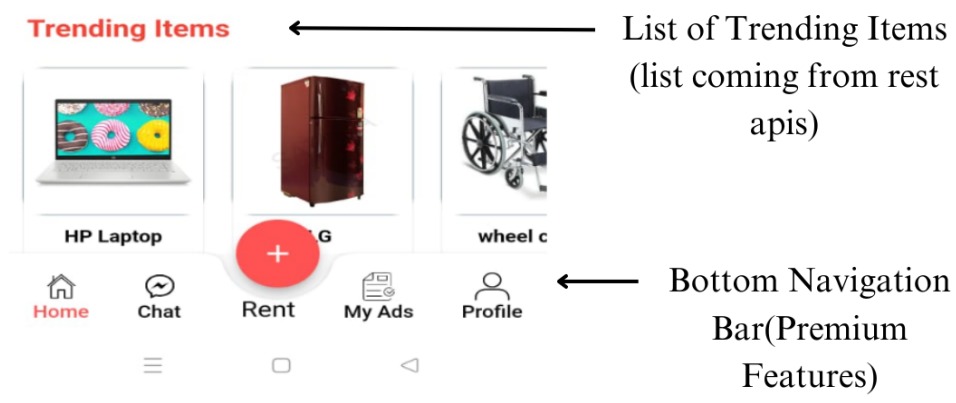
Home Screen

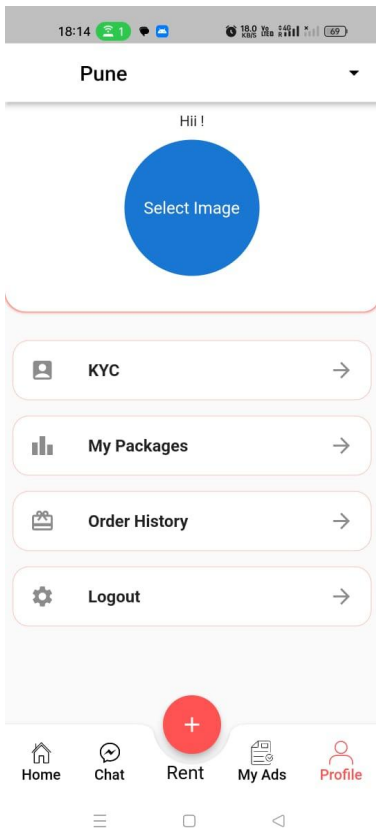


Slider Images

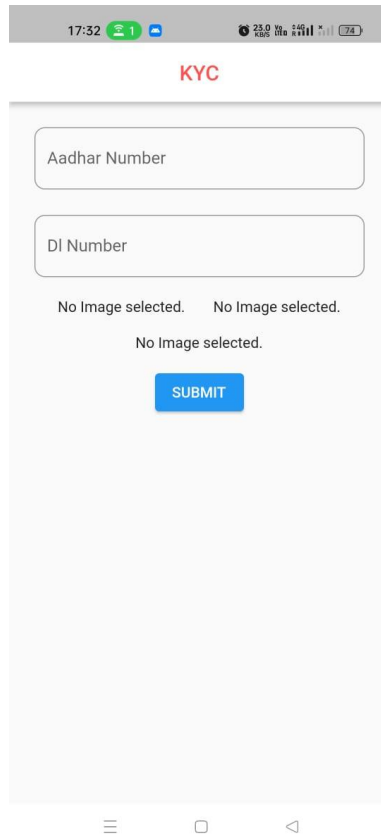


Category Selection

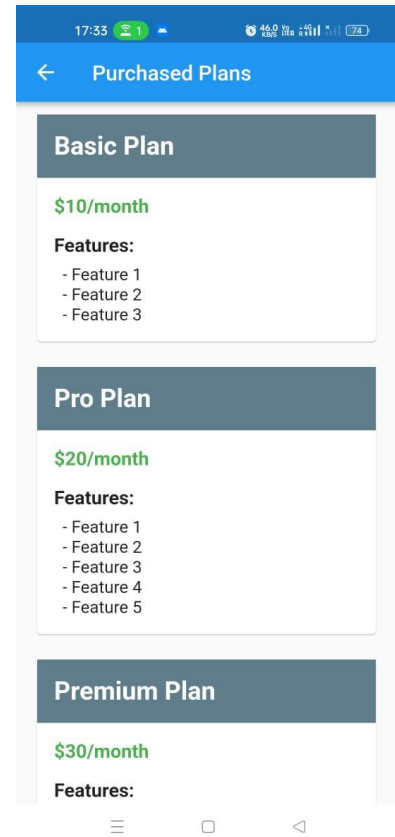




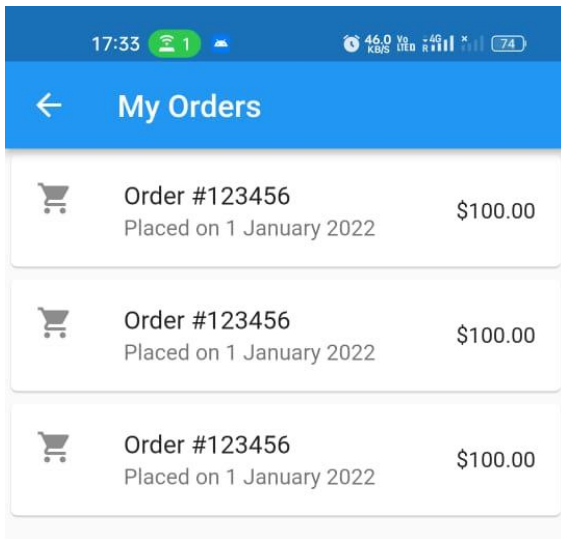
Profile Page



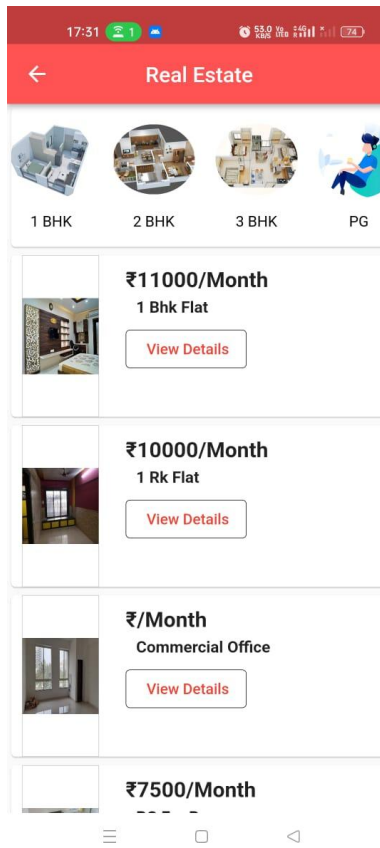
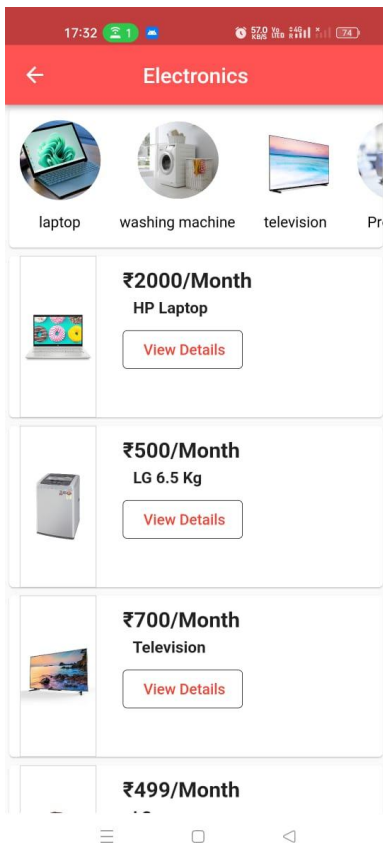
KYC Document Uploading Screen



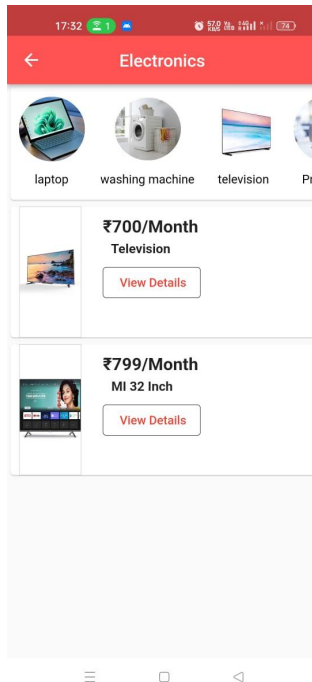
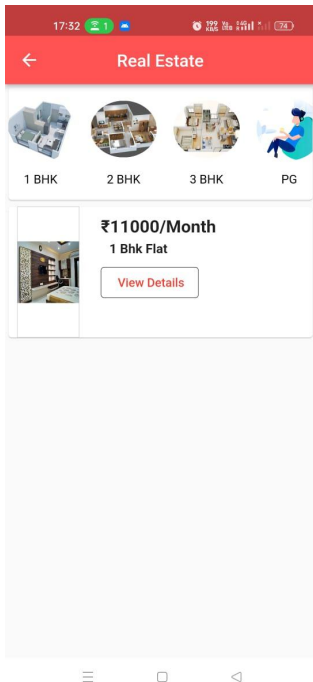
Post Subscription Screen



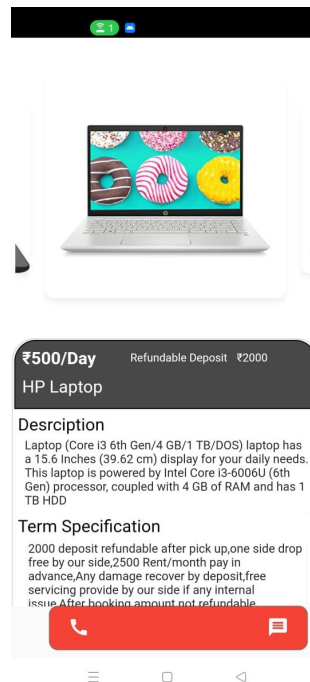
Purchased Order history Screen



Subcategory Screen without filter



Subcategory Screen after applying filters



Description/Detail Screen



## Post Rent Module:

17:32 6.00 Ultra 4G+ 74%

← Post Your Product

Choose Category

Real estate

Choose SubCategory

Select Tenure

Per Day

Enter Price

Based on Selected Tenure

Product Name

e.g: HP/Asus/Iphone/Samsung

Add Description

Based on Selected Tenure

Next Page

Adding Product Description Screen

17:32 29.0 Ultra 4G+ 74%

Address

Get Current Location

# pincode

city name

Enter Your Locality

Upload Your Product Images

Next Page

Product Address & Image Screen

17:32 6.00 Ultra 4G+ 74%

← Subscribe Now

Choose a Plan

Number of days : 30  
Price : 499  
Number of post : 5

Number of days : 45  
Price : 999  
Number of post : 10

Number of days : 90  
Price : 2000  
Number of post : 30

Number of days : 20  
Price : ...  
Number of post : ...

Buy Subscription Screen/Page

Acme Corp  
order\_CtxBaaUHU15z  
₹ 10000

+91 9085626859 | ashok.kumar@g... Edit

PREFERRED PAYMENT METHODS

HDFC Credit Card - 2745 CVV

ashok.kumar@okicici

ALL PAYMENT METHODS

Card  
Mastercard, Visa, Rupay, Maestro, Amex

EMI  
Card, EarlySalary and more

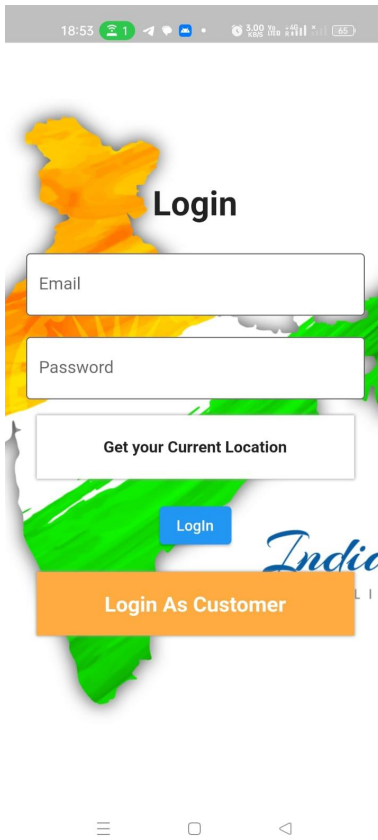
3 Offers Available Select Offer

Razorpay Payment Gateway

## Authentic Guards outputs/screens:

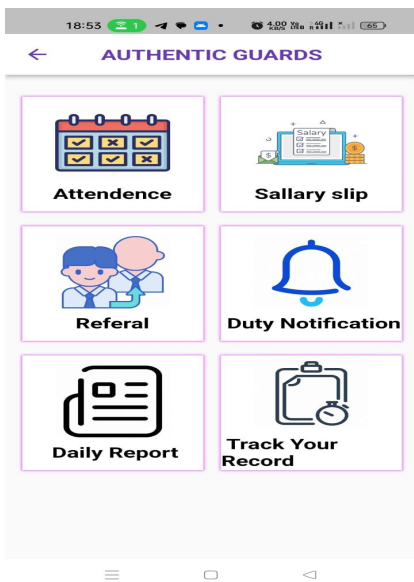


Splash Screen with timer 3 sec

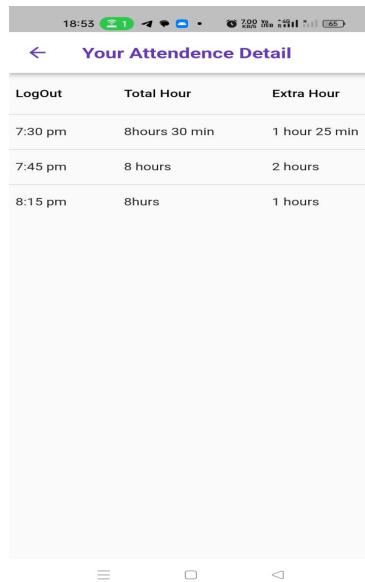


Login Screen for both guards and clients, guards will be login only if they will be at particular location i.e fetching live location

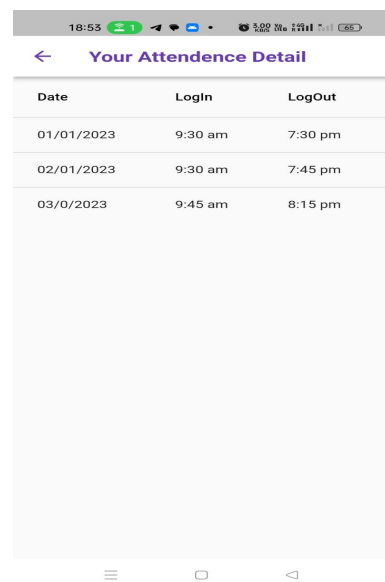
## LOGIN AS GUARD:

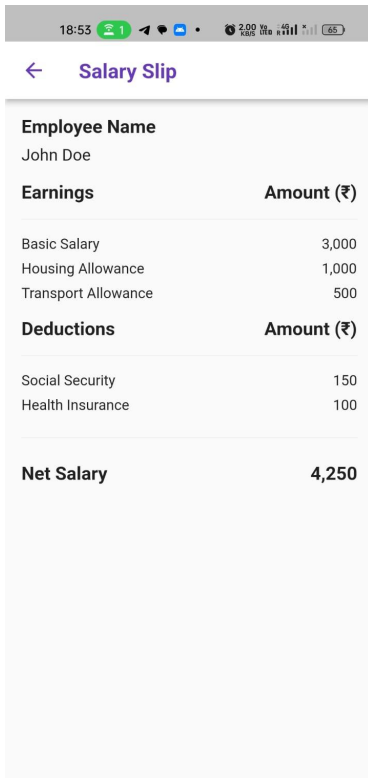


Home Screen

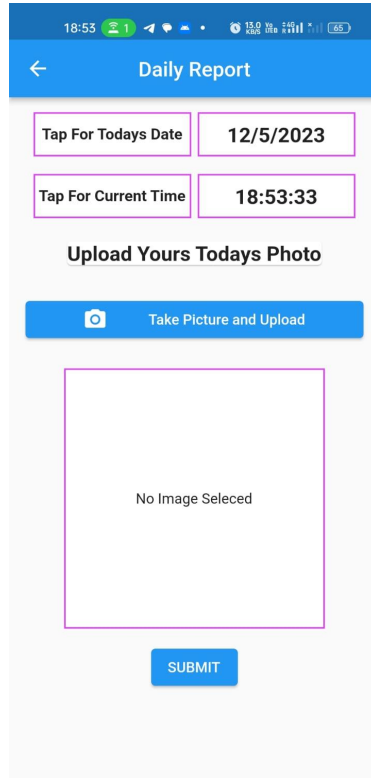


Attendance Detail

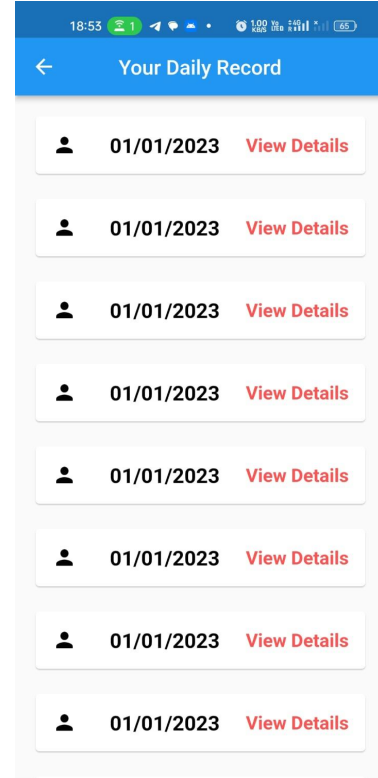




Salary Slip of Guard

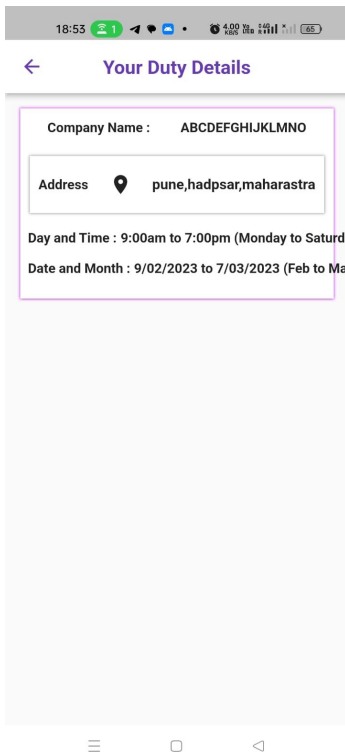


Screen For Guards to post their daily report including image at site

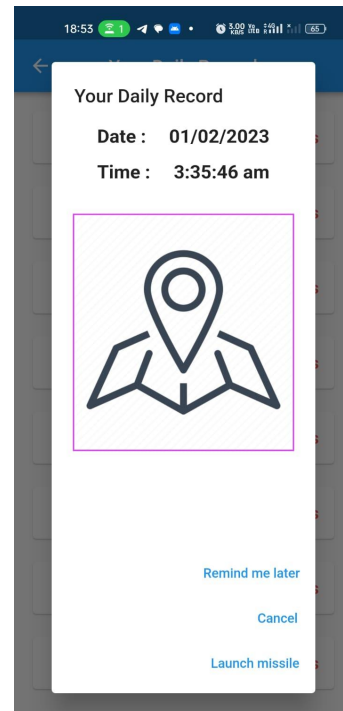


Screen for guards to trace their daily record anytime

↓ After clicking on view details



Duty Notification for Guards



# Login As Customer/Client



Screen for guard to refer other guards

Clients can enquire if they need guards and also check details of guards who are present for their service.

After Clicking on Check Employee Details at work



Guards Name	View Details
Guards Name	View Details
Guards Name	View Details
Guards Name	View Details
Guards Name	View Details
Guards Name	View Details
Guards Name	View Details
Guards Name	View Details

Age Criteria(if any):  
Age Criteria

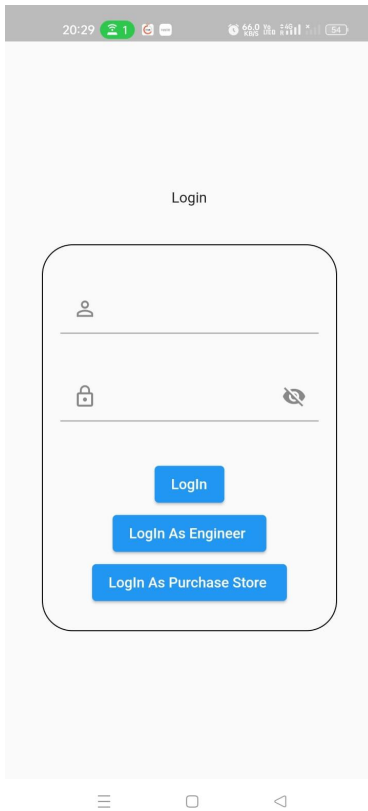
Starting From(Select Date):  
12-05-2023

Till Date(Service Ending Date):  
12-05-2023

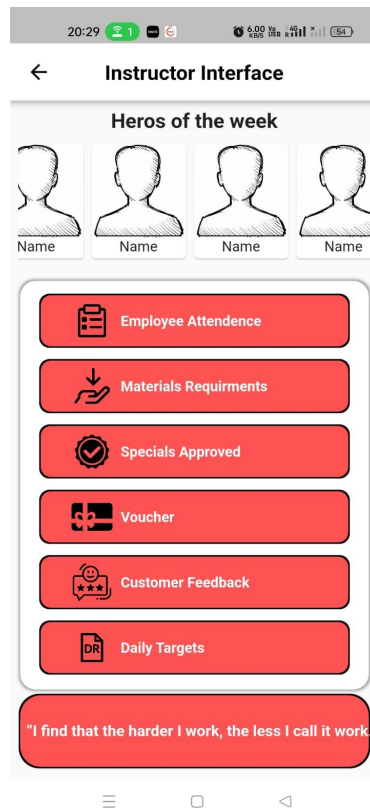
Address(Site)  
Note: Select Location when you are on working site  
Tap for Site Location

Send Enquiry

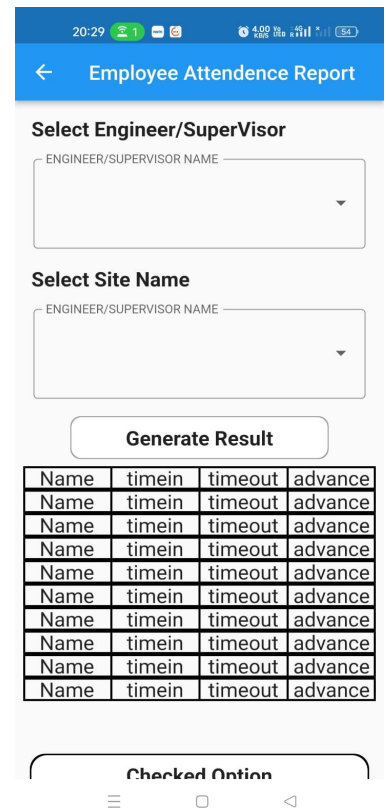
# RK Construction:



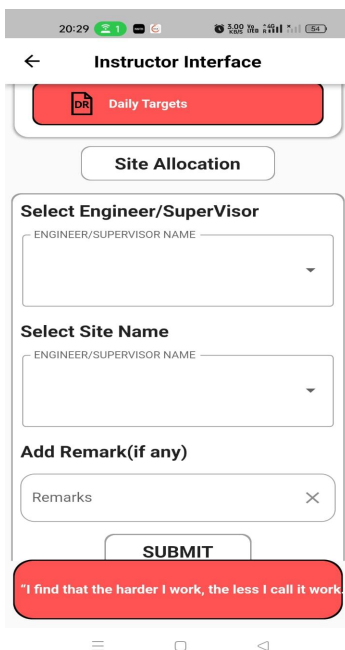
Same Login Interface for multiple type of user



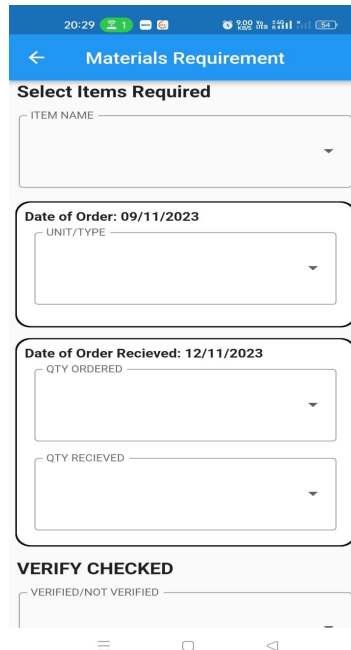
Similar dashboard for all type of users with different mathematical calculations and credential



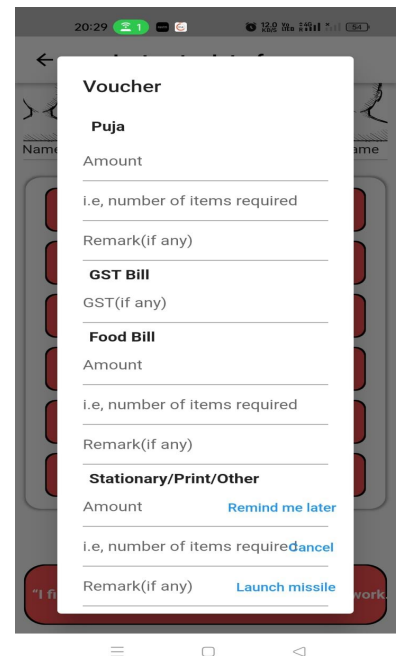
Attendance Report of Engineers approved by supervisors



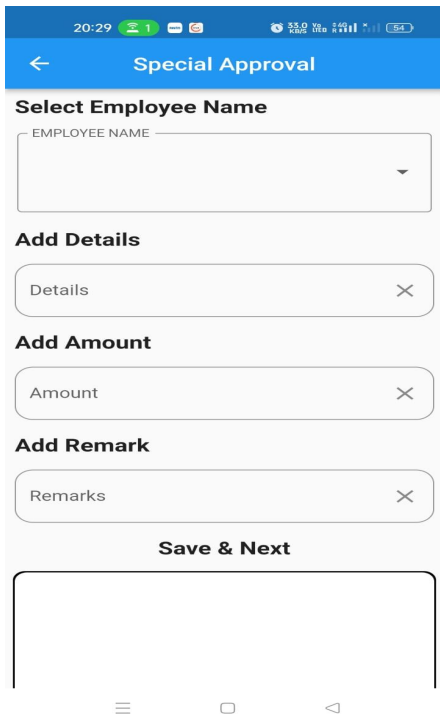
Site allocation to engineers by supervisors



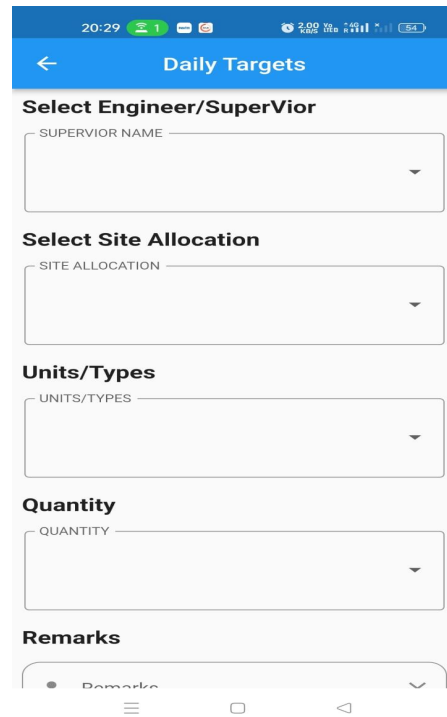
Materials requirement requested to supervisors and verification credential to supervisors



Special Voucher give to supervisor/ Engineers/Instructor on special occasions



Special Approval Feature given to supervisors



Daily Targets assigned to engineers by supervisors

For all the different type of users ,Dashboard , Attendance, Materials Requirements,Materials Approval, Special Approval, Voucher, Site Allocations all the different features are almost similar backend logic and database connection is different , Different types of Restful Apis are integrated , therefore mathematical calculations and different types of results are available in different cases. I was assigned to create the user interface and frontend design of the mobile application , which I have completely presented in the report.

## **Test Cases:**

For all the above mentioned Mobile Application development projects majorly manual testing were performed. As organizations were following the agile methodology , major testing parts were done successfully manually after completion of each module.

Some of the manual test case performed in different projects:

### **Renstvale:**

- . login/ signup properly.
- . kyc documents proper.i.e aadhar number and dl number must be of 16 and 12 digits.
- . images and files are uploaded properly.
- . dependent dropdown working properly.
- . updates by admin panels are properly visible in the frontend.
- .payment gateway working properly.
- . Category and subcategory are properly mapped.
- .list in the subcategory section is working properly after applying filters.
- .make sure the user is logged in and have done kyc before calling and chatting to the vendors.
- . The media query is working properly i.e, app screens are working perfectly in all the screen sizes.

### **Authentic Guards:**

- . login/ signup working properly.
- . GeoLocator/ Geocity working properly.
- . File Picker/ Image Picker working properly.
- . Text form field and controller working properly.
- . Contact numbers and email entered are correct.

### **RK Construction:**

- .Multiple Login is working properly.
- . Material Request and Materials Approved are working properly.
- . Attendance system is working smoothly.
- .Site Allocation and labor allocation is working without lag.
- .Image Picker and File picker is working properly.

## Chapter-5

### **CONCLUSIONS:**

In Conclusions we can conclude that all the above mentioned projects are developed and structured ,keeping in mind the business requirements . Frontend, Backend ,Database , server used are scalable, efficient and can handle a large number of user base.

As a junior flutter developer working on such types of projects was a great exposure. All three different projects are from different industries/sectors, working on these projects I learnt how tech is involved and contributing to the different sectors . web based applications,integrated ERP, mobile applications are helping businesses to grow efficiently, reducing their problems,and efficient cost management.

The Flutter framework's amazing architecture makes it feasible to build mobile applications that are completely platform independent. By streamlining the development process, guaranteeing high performance in the finished mobile application, and providing a rich and pertinent user interface for both the Android and iOS platforms, the Flutter framework will undoubtedly assist many new developers in the near future in creating high-performing and feature-rich mobile applications.

### **Future Work:**

In all above described three mobile applications projects there is always a future possibility and all these applications can be made more scalable and more business oriented.

#### **Renstvale:**

- . Currently call function is out of the app i.e, direct calling , in future in app calling feature can be given.
- . Lead management ,CRM-like features can also be added .
- . make applications more efficient by contact between nearby vendors and customers.i.,e fetching the list of vendors in a particular zone and making them available to the customers available there.

#### **Authentic Guards:**

- . Currently fetching the location only at login time i.e once login done for the day, in future it can be done that if guards device goes out of particular radius it will logout automatically.
- .various features can be given to the clients .

#### **RK Construction:**

- . Currently little innovative i.e, trying to bring all the web based construction ERP features into mobile applications.
- In the future whole web based construction ERP features can be customized into mobile applications according to the business needs.



## REFERENCES:

- Mascot, B. (2016) [1]. Job title: Bit Mascot. Bit Mascot on [website]. Easily accessed at: <https://www.bitmascot.com/top-10-challenges-faced-mobile-app-developers/>>. As of: September 27, 2020
- 2019; Technologies, T. Why Android App Developers Should Think About Flutter. [Blog] Consider upcoming technologies. Easily accessed at: <https://www.tftus.com/blog/why-mostly-android-developer-consider-flutter-app-development/>. retrieved on: September 29, 2020
- "Flutter" To Build iOS & Android Apps. [3] Kumar, D. [Blog] Medium. Easily accessed at: <https://levelup.gitconnected.com/flutter-to-build-ios-android-apps-f8786d6fe987>>. Retrieved on: September 26, 2020
- The Dart Programming Language, Dart dev., n.d. [website] Easily accessible at: <https://dart.dev/> Retrieved on: September 26, 2020
- React Native vs Flutter, Cross-Platform Mobile Application Framework, Thesis March 2018- Wenhao Wu. A clean approach to Flutter Development through the Flutter Clean architecture package, IEEE 2019, Shady Boukhari, Eduardo Colemenares
- Exploring end user's perception of Flutter mobile apps, Malmo University Nov 2019- Dahl, Ola.
- Flutter for Cross-Platform App and SDK Development, Metropolia University Thesis May 2019- Lucas Dagne.
- Cross-Platform Framework comparison- Flutter vs React Native.
- Flutter Native Performance and Expressive UX/UI, paper 2019- Tran Thanh.

