**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI, KARNATAKA**



A Mini Project Report

**(Sixth Semester)**

On

**“Parking Management App for Android”**

Submitted in the partial fulfillment for the requirements for

the conferment of degreeof

**BACHELOR OF ENGINEERING**

In

**INFORMATION SCIENCE AND ENGINEERING**

By

|  |  |
| --- | --- |
| **Mr. PRAJWAL R** | **1BY18IS084** |
| **Mr. PRASHANTH R** | **1BY18IS089** |
| **Mr. RAGHAVENDRA K M** | **1BY18IS093** |
| **Ms. SUGANDHA SINHA** | **1BY18IS119** |
|  |  |

Under the guidance of

**Mrs. Dr Surekha K.B. Asst. Professor**



**2020-2021**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI, KARNATAKA**

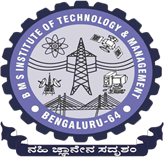
**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI – 590 018, KARNATAKA**

**BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT**

**YELAHANKA, BENGALURU-560064**

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

****

**CERTIFICATE**

This is to certify that the Project work entitled **“PARKING MANAGEMENT APP”** is a bonafide work carried out by **Mr. Prajwal R (1BY18IS084)**, **Mr. Prashanth R (1BY18IS089), Mr. Raghavendra K M (1BY18IS093), Ms. Sugandha Sinha (1BY18IS119)** in partial fulfillment of Mobile Application Development Laboratory with Mini Project (18CSMP68) for the award of **Bachelor of Engineering Degree in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work for the B.EDegree.

**Signature of the Guide Signature of the HOD**

Dr Surekha K.B. Dr. Pushpa S.K.

Associate Professor Professor and Head

Department of ISE Department of ISE

**EXTERNAL EXAMINERS**

Name of the Examiners Signature with Date

1.

2.

**ABSTRACT**

It is very common waiting outside a parking lot and looking if any space available in there, this overtime will be irritating and time consuming especially in traffic rich cities like Bangalore, Delhi etc.

Everyone would have frequently experienced that they park their vehicle somewhere beside the road and by the time they return the vehicle is gone, this would have been the worst day for a person having plans for the day. Thus, knowing the availability of parking space in advance would help people to plan for the same and save the time which would have been spent in looking for a spot in the lot. The parking fee is also a major concern of people, wherein many of them would not know the cost of parking per hour basis and which would cause them to pay more than the actual fee for it.

This Parking Management App is a parking solution which to solve the above problems by providing the user with the available space in the lot and calculate the amount accurately with the parameters of entry time and exit time. It will also keep history of all the vehicles parked at any parking spot with specific date and time.

**ACKNOWLEDGEMENT**

We heartily thank our Principal **Dr. Mohan Babu G N**, BMS Institute of technology and Management for his constant encouragement and Inspiration in doing the project.

We heartily thank our Head of the Department **Dr. Pushpa S K**, Dept of Information Science and Engineering, BMS Institute of technology and Management for her constant encouragement and Inspiration in doing the project.

We gracefully thank our project guide **Prof. Surekha K B**, Assistant Professor, Dept. of Information Science and Engineering for her constant support, advice, guidance and inspiration for the project.

Nevertheless, we express our gratitude towards our family and friends for the encouragement and support which helped us to finish this project successfully.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TOPIC** | **Pg. No.** |
| 1 | Introduction   * 1. Introduction   2. Motivation   3. Objectives | 🛑 |
| 2 | Background   * 1. Introduction   2. Algorithm   3. Design/Methods   4. Results/Discussion   5. Applications | 4-11 |
| 3 | Conclusion  References | 12-13 |

List of Screenshots:

Fig. 4.1.1 Home Activity

Fig. 4.1.2 Admin Home Activity

Fig. 4.1.3 Login Activity

Fig. 4.1.4 Register Activity

Fig. 4.1.5 Available Slots Activity

Fig. 4.1.6 Parked Vehicles Activity

Fig. 4.1.7 Parked History Activity

Fig. 4.1.8 Add Vehicle Activity

Fig. 4.1.9 Remove Vehicle Alert Dialog

Fig. 4.1.10 Vehicle Exit Activity

Fig. 4.1.11 Add Vehicle Alert Dialog

Fig. 4.1.12 Forgot Password Alert Dialog

**List all the figures/tables/images used**

Fig 1.5 Android architecture

Fig 1.6 Android components

**List of Figures/Tables**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Name of Figures** | **Page No.** |
| 3.1 | Overall System Design | 05 |
| 3.2 | Display Process Flow Diagram | 10 |
| 3.3 | Update Process Flow Diagram | 10 |
| 3.4 | Deletion Process Flow Diagram | 11 |
| 4.1 | Entity Relationship Diagram | 12 |
| 4.2 | Schema Diagram | 12 |
| 7.1 | Login Page | 19 |
| 7.2 | Registration Page | 19 |
| 7.3 | View book page | 20 |
| 7.4 | Borrow book page (Student login) | 20 |
| 7.5 | Return book page (Student login) | 21 |
| 7.6 | Dashboard page (Student login) | 21 |
| 7.7 | Authors tab in Manage page (Staff login) | 22 |
| 7.8 | Publishers tab in Manage page (Staff login) | 22 |
| 7.9 | Books tab in Manage page (Staff login) | 23 |
| 7.10 | Search Results Page | 23 |
| 7.11 | Add book page | 24 |
| 7.12 | Dashboard page (Staff login) | 24 |
| 7.13 | Updated Author Page (Staff Login) | 25 |
| 7.14 | After deletion of a book | 25 |
| 7.15 | Admin Panel | 26 |
| 7.16 | Records in BOOK Table | 26 |

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name of Tables** | **Page No.** |
| 6.1 | User Login Module Test | 16 |
| 6.2 | Student Registration Module Test | 16 |
| 6.3 | Borrow book Module Test | 16 |
| 6.4 | Return book Module Test | 17 |
| 6.5 | Search Module Test | 17 |
| 6.6 | System Test | 18 |

**INTRODUCTION**

With the increase in use of private vehicle in recent years, the problem of car parking has raised in busy and big cities of the world. In crowded cities of the world, mostly a person must spend a lot of time in finding the vacant parking lot.

As an important component of traffic system, parking management system is playing an important role and affecting people’s daily life.

By detecting and processing the information from parking lots, smart parking systems allows driver to obtain real-time parking information and alleviates parking contentions.

A Smart Parking Management system is a parking solution which is embedded into parking spots to detect whether parking bays are free or occupied through real-time data collection. It will not only help in generating bill for parking of car but will also keep record of all the vehicles parked at any parking spot with specific date and time.

**INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT**

Mobile application development is the process to making software for Smartphone and digital assistants, most commonly for Android and iOS.

The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used for this kind of software development include Java, Swift, C# and HTML5.

Mobile app development is rapidly growing. From retail, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transaction and access information. Today, mobile devices- and the mobile applications that unlock their value-are the most popular way for people and business to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications that their customers, partners and employees demand.

**WHAT IS MOBILE APP?**

A mobile application or mobile app is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases, so that there are now millions of apps available.

Apps are generally downloaded from application distribution platforms which are operated by the owner of the mobile operating system, such as the App Store (iOS) or Google Play Store.

Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device.

Mobile App has many advantages like within a short app we can communicate a lot of information to the client/customers and even it is an ease of access to client/customer for services update or sale/purchase activity.

**WHAT IS MOBILE OS?**

A mobile operating system is an operating system for mobile phones, tablets, smart watches, 2-in-1 PCs, smart speakers, or other mobile devices. While computers such as typical laptops are ‘mobile’, the operating systems used on them are generally not considered mobile ones, as they were originally designed for desktop computers that historically did not have or need specific mobile features. This distinction is becoming blurred in some newer operating systems that are hybrid made for both uses.

A mobile OS is responsible for identifying and defining mobile device features and functions, including keypads, application synchronization, email, thumbwheel and text messaging. A mobile OS is similar to a standard OS (like Windows, Linux, and Mac) but is relatively simple and light and primarily manages the wireless variations of local and broadband connections, mobile multimedia and various input methods.

**INTRODUCTION TO ANDROID STUDIO**

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of Intellij’s powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

A flexible Gradle-based build system

A fast and feature-rich emulator

A unified environment where you can develop for all Android devices.

Apply Changes to push code and resource changes to your running app without restarting your app.

Code templates and GitHub integration to help you build common app features and import sample code.

Extensive testing tools and frameworks.

Lint tools to catch performance, usability, version compatibility, and other problems.

C++ and NDK support.

Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

Android Studio provides a unified environment where we can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto.

**1.5 Android Architecture**

Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services. Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

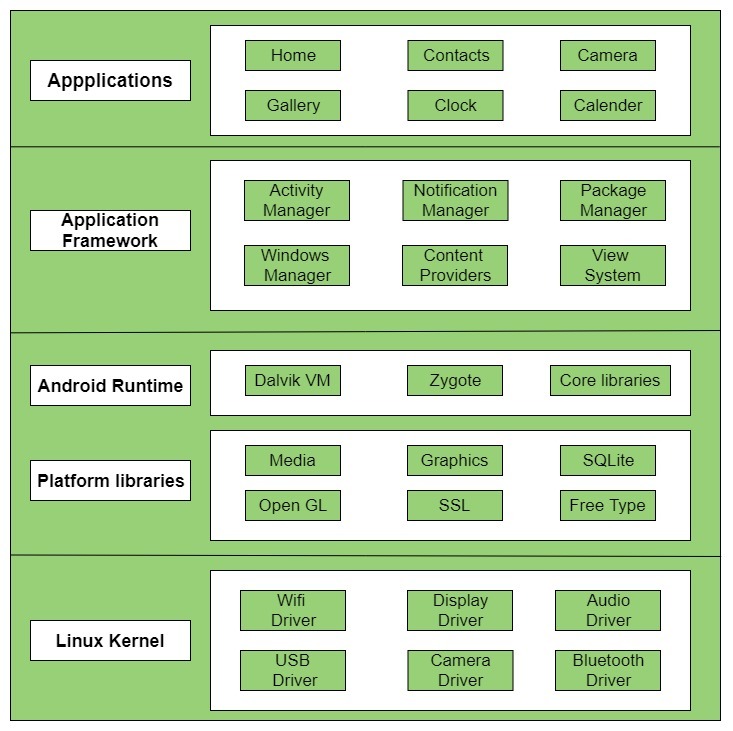
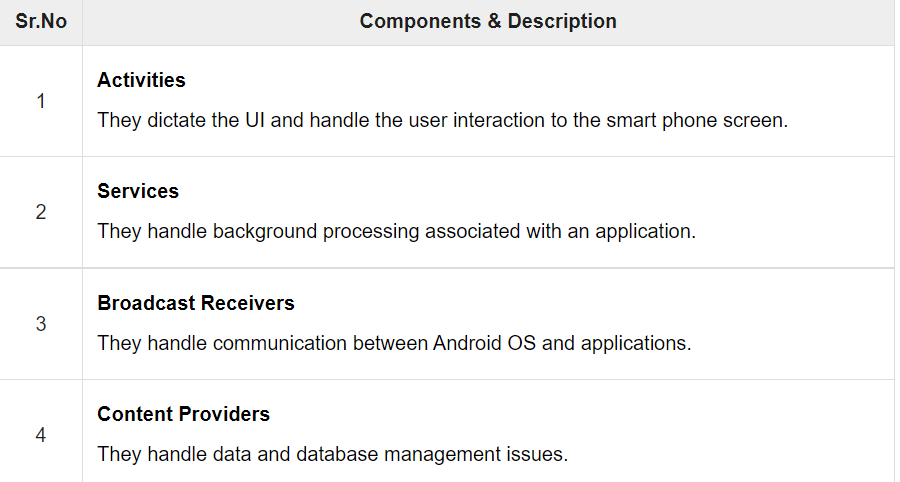


fig 1.5: Android Architecture

**1.6 Android Application Components**

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file AndroidManifest.xml that describes each component of the application and how they interact.

****

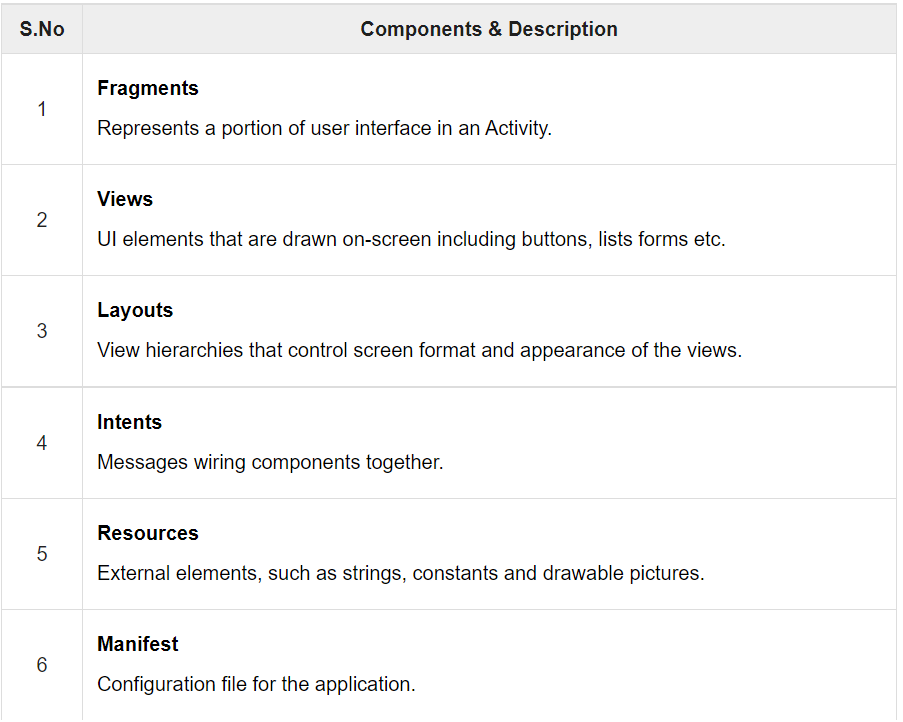
****

Fig 1.6 Android components

**1.7 Problem Statement**

With the increasing number of vehicles and the decreasing efficiency of modern busy parking lots, major problems which we people are facing is:

1. Valuable time wasted from inconvenient and inefficient parking lots.
2. More fuel consumed while driving around parking lots, leading to CO2 emissions.
3. Potential accidents caused by abundance of moving vehicles in disorganized parking lots.

Therefore, there is a need to develop an affordable system which solve the problem and obtain the information about the parking lot on real time.

**1.8 Objectives**

* To develop an intelligent, user friendly automated car parking system which reduces the manpower and traffic congestion.
* To improve the performance and satisfy the need of free parking lot.
* To reduces the time wastage in finding the vacant parking lot.
  1. **Project Applications**
* This project can be implemented in shopping malls, public parking areas to monitor parking.
* Automation and increased efficiency and transparency of parking charges.
* General public can check the slots available for parking.

**2. SYSTEM DESIGN**

**2.1 APPLICATION COMPONENTS OF PROJECT**

Application components are Core Building Blocks of an Android Application. It is an entry Point for System or Users from which they can enter in App.



The project consists of the following components:

1. **Activities:**

An activity represents a single screen with a user interface, in-short Activity performs actions on the screen. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched, which is done in the manifest file.

The project has 9 activities, starting from “HomeActivity”, which is the first screen visible to the users once they open the app, The clicks on the views will lead to trigger of respective activities, the flow is represented below:

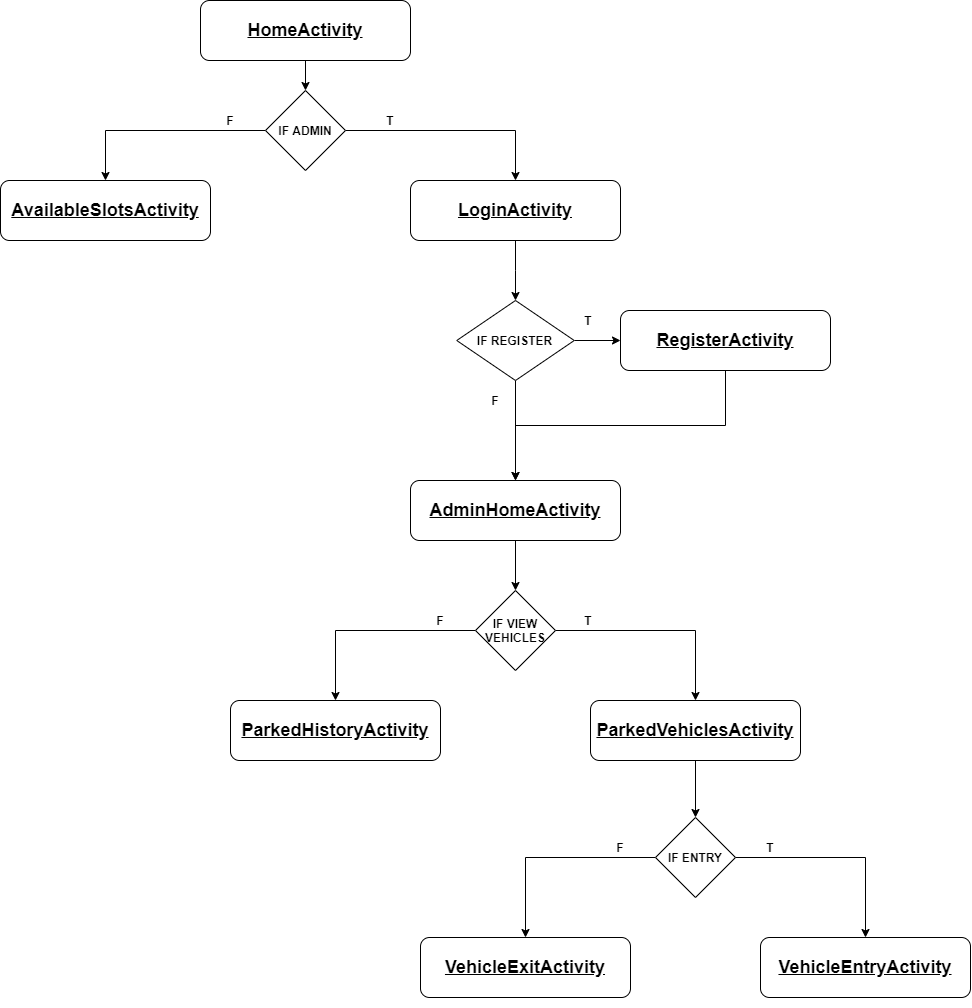


Figure 1: Flow of the Activities in Application

1. **Views:**

View is the basic building block of UI (User Interface) in android. View refers to the android.view.View class, which is the super class for all the GUI components.

The app contains Text View, Image View, Button, Edit Text, List View, Progress Bar, Card View which helps in achieving the flow of Activities and design the layout responsively.

Text Views are used to display the field labels and the details of the Admin. Edit Texts are used to take the details during login, register and vehicle entry. Image Views are used to display the icon of the app and List Views are used to display the slots for the users with details of number of available slots, area, name and display the admin list of parked vehicles and list of vehicles visited.

1. **Layouts:**

Android Layout is used to define the user interface that holds the UI controls or widgets that will appear on the screen of an android application or activity screen.

The app is designed using Constraint Layout, Linear Layout, Swipe Refresh layout.

Swipe Refresh layout is used in AvailableSlotsActivity for users, ParkedVehicleActivity and ParkedHistoryActivity for admin to retrieve the latest information from the database.

1. **Intents, Resources:**

The app communicates or transfers the control and information from one Activity to other Activity with the help of intents.

The ID’s, String values, Colors, Styles, XML file for every Activity, Drawable such as icons are stored in resource folder.

**2.2 User Interface Design**

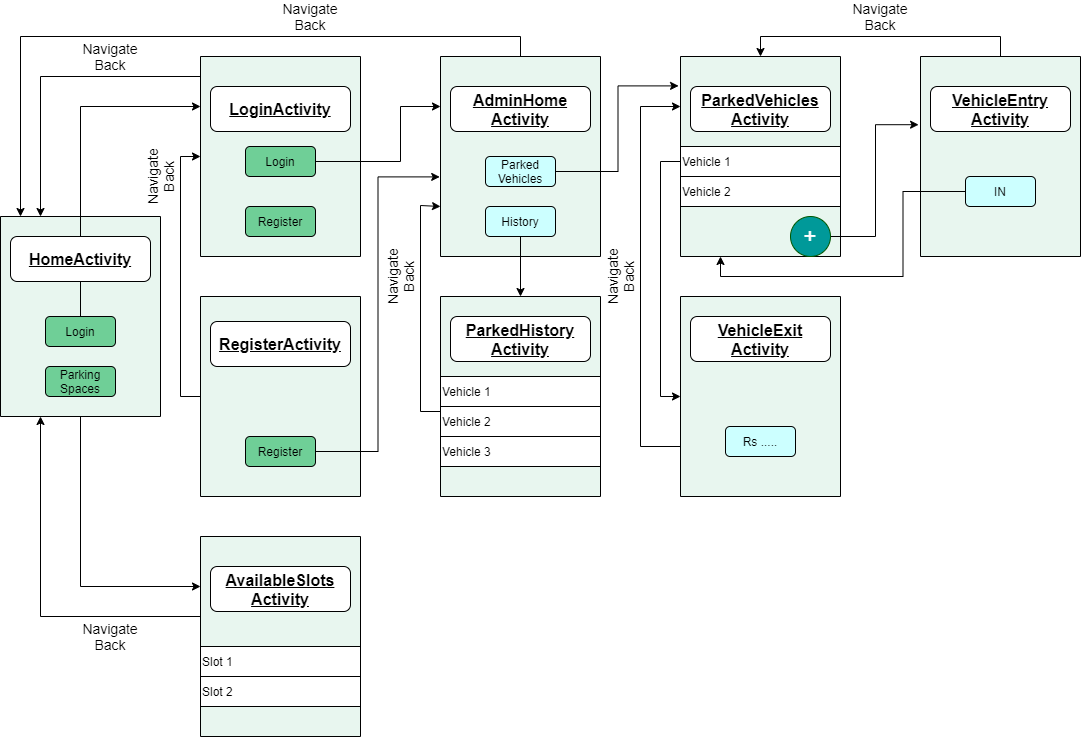
****

Figure 2:The design of Application

The user interface (UI) for an Android app is built as a hierarchy of layouts and widgets. The layouts are View Group objects, containers that control how their child views are positioned on the screen. Widgets are View objects, UI components such as buttons and text boxes.

The Application is designed using the components of android such as, Layouts, Buttons, Text Views, Edit Texts, Card Layouts etc.

The above figure gives a brief design of the application and its flow. The app is designed using Constraint Layout which is a View Group containing many Views such as Edit Texts, Buttons etc.

The Constraint Layout gives the option of constraining the views with respect to the parent i.e., View Group or other Views in the layout.

The List View is designed using a Recycler View, which is been loaded with data with the help of adapters which acts as a bridge between the View and the underlying data.

These are used in AvailableSlotsActivity, ParkedVehicleActivity, ParkedHistoryActivity.

**3. IMPLEMENTATION**

**3.1 Explanation of the modules with Java Code and XML code**

1. **Adding data to database:**

The main operation carried out throughout the app is adding the data to firebase. Starts by adding the admin details to the database after they register in RegisterActivity and then adding the vehicle number, entry time and exit time to the database for the respective slots in the VehicleEntryActivity and VehicleExitActivity, respectively.



1. **Retrieving the data from database:**

The data to be displayed in AdminHomeActivity, ParkedVehiclesActivity, ParkedHistoryActivity, VehicleExitActivity and AvailableSlotsActivty is carried out by this function, it fetches the data from the firebase, which would be added before by using the previous method.



1. **Validating the Vehicle Number:**

The validation of vehicle number which leads to insertion of it to database forms an important part of the app. The vehicle number is first read through a masked Edit Text, which gives the feature of formatting the text when entered and give the restriction on the length etc.

After reading the vehicle number, it is validated using regular expression methods available in java. We have considered most recommended and common format of a vehicle i.e., 2 letters, 2 digits followed by 2 letters and 4 digits. Each of this has its meaning where the first 2 letters represent the state at which the vehicle has registered (Ex : KA – KARNATAKA), the next 2 digits indicate sequential number of a district(Ex : 03 – BANGALORE EAST), the next 2 letters shows the ongoing series of an RTO, and the last 4 numbers are unique to each plate.

After the validation using regex, the data will be stored in the firebase for the respective building.

**XML**

****

**JAVA**

The regex used is (**"^[A-Z]{2}[-][0-9]{2}[-][A-Z]{2}[-][0-9]{4}$"**) which is matched with the vehicle number using matches() method, which returns true if the number matches with the regex else returns false.

After this the available space value will be reduced by 1 and updated to the firebase for that building, and the entry time and vehicle number will get stored in a new collection named “parked-vehicles” and during exit of vehicle the exit time will get stored, the available space will be increased by 1 and updated,

The bill will be calculated based on the minute difference between exit and entry time, respectively.

****

**Results**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Fig. 4.1.1 Home Activity |  | Fig. 4.1.2 Admin Home Activity |
|  |  |  |
|  |  |  |
| Fig. 4.1.3 Login Activity |  | Fig. 4.1.4 Register Activity |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Fig. 4.1.5 Available Slots Activity |  | Fig. 4.1.6 Parked Vehicles Activity |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
| Fig. 4.1.7 Parked History Activity |  | | Fig. 4.1.8 Add Vehicle Activity |
|  |  |  | |
| Fig. 4.1.9 Remove Vehicle Alert Dialog |  | Fig. 4.1.10 Vehicle Exit Activity | |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Fig. 4.1.11 Add Vehicle Alert Dialog |  | Fig. 4.1.12 Forgot Password Alert Dialog |

**5 Application Testing**

**Table 5.0 Application Test**

|  |  |  |
| --- | --- | --- |
| **System Unit** | **Test Case** | **Result** |
| Login Activity | Click on Login button | Authenticate user based on the input user credentials |
| Login Activity | Click on Register TextView | Navigates to Register Activity |
| Admin Login Activity | Click on the Parked Vehicles button | Navigates to Parked Vehicles Activity |
| Admin Login Activity | Click on the History button | Navigates to Parked History Activity |
| Admin Login Activity | Click on the Logout button | Signs out the currently authenticated user from the app |
| Home Activity | click on the Login Page button. | Navigates to Login Activity |
| Home Activity | click on the Parking Spaces button. | Navigates to Login Activity |
| Available Slots Activity | On opening this Activity | Displays a list of Parking Spaces that are available |
| Parking Vehicles Activity | On opening this Activity | View the currently parked vehicles in the Parking Space |
| Parking Vehicles Activity | Click on Floating Action Button | Navigates to Vehicles Entry Activity |
| Parked History Activity | On opening this Activity | View the previously parked vehicles in the Parking Space |
| Register Activity | click on Register button | A user account will be created if the entered data is valid |
| Vehicles Entry Activity | click on IN button | Adds a vehicle into the database if the entered data is valid |
| Parking Vehicles Activity | Click on added Vehicle Card | Alert Dialog pops up asking to whether to remove vehicle from database.  On successful removal of the vehicle, app navigates to Vehicles Exit Activity |
| Vehicles Exit Activity | On opening this Activity | View the auto generated bill for the calculated parking duration |
| Forgot Password Alert Dialog | Click on Yes button | Sends a password reset email to the entered mail id |

Conclusion

1. The parking management system app is effective in detecting the available parking slots through real-time data collection.
2. In addition to that the app can be a part of a bigger revolution by reducing the traffic in a city by informing the user about the parking spaces.
3. The manual process of entering the vehicles in the vehicle register book is eliminated.
4. Further enhancement is required to make the user view his or her nearby parking areas.
5. Overall system can handle all the requests from the User to give a fresh digital look for current parking management system

Future enhancements

1. Online booking of parking slots in a parking space/area so that users can reserve the parking slot for his/her vehicle.
2. Redesigning the user interface of the app.
3. Creating a profile page for the admin of the parking space so that he/she can update not just his/her profile but also details regarding the parking space/area.
4. Integrating the app with Google maps to easily locate or navigate to the parking area.
5. A feature to view the nearby parking areas

7. References

[1]. M. Owayjan, B. Sleem, E. Saad and A. Maroun, "Parking management system using mobile application," 2017 Sensors Networks Smart and Emerging Technologies (SENSET), 2017, pp. 1-4, doi: 10.1109/SENSET.2017.8125048.

[2]. C. Ng, S. Cheong, E. Hajimohammadhosseinmemar and W. Yap, "Mobile outdoor parking space detection application," 2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC), 2017, pp. 81-86, doi: 10.1109/ICSGRC.2017.8070573.

[3]. B. K. Patil, A. Deshpande, S. Suryavanshi, R. Magdum and B. Manjunath, "Smart Parking System for Cars," 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), 2018, pp. 1118-1121, doi: 10.1109/ICRIEECE44171.2018.9008662.