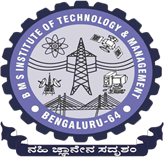
**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**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.

**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:

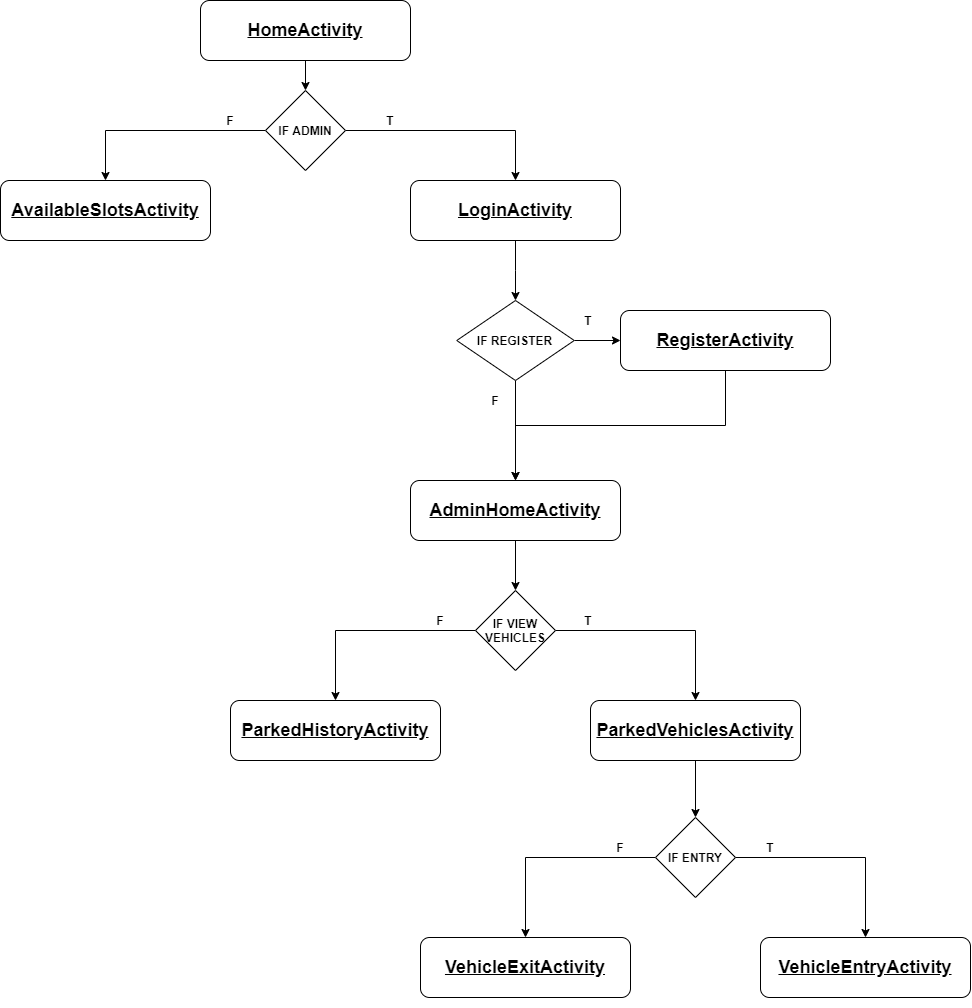


Figure 1: Flow of the Activities in Application

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**

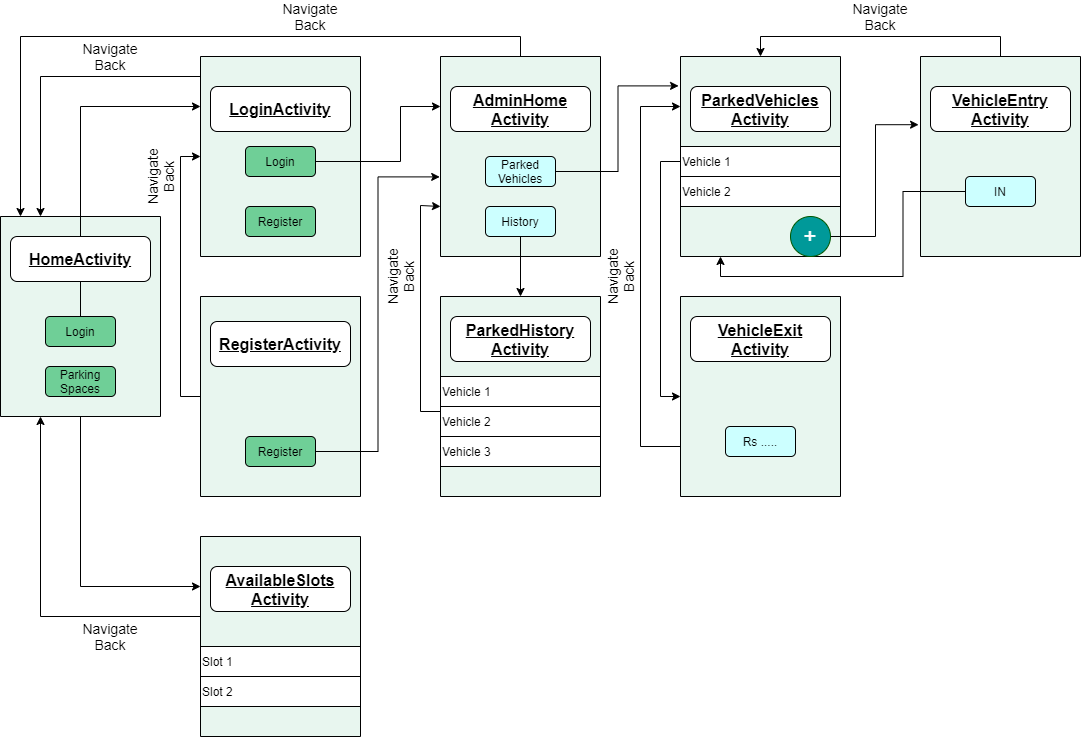
****

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.

****