**Restaurant Management System (RMS)**

Submitted in partial fulfilment of the requirements

of the syllabus of

Android Apps Development Lab

in

Information Technology

by

Prashanth Venkatesan 118A3040

Rohit Viramani 118A3045

Under the Guidance of:

Ms. Bushra Shaikh



Department of Information Technology

SIES Graduate School of Technology

2021-22

**CERTIFICATE**

This is to certify that the project entitled **“Restaurant Management App (RMS)”** is a bonafide work of the following students, submitted to the University of Mumbai in partial fulfillment of the requirement of the syllabus of **Android Apps Development Lab** in **Information Technology.**

**Prashanth Venkatesan 118A3040**

**Rohit Viramani 118A3045**

Ms. Bushra Shaikh Dr. Lakshmi Sudha Dr. Atul N Kemkar

Internal Guide Head of Department Principal

**PROJECT REPORT APPROVAL**

This project report entitled **Restaurant Management App (RMS)** by following students is approved for the requirement of the syllabus of ***Android Apps Development Lab*** in ***Information Technology.***

**Prashanth Venkatesan 118A3040**

**Rohit Viramani 118A3045**

**Name of External Examiner: --------------------------------**

**Signature:--------------------------------**

**Name of Internal Examiner: --------------------------------**

**Signature:--------------------------------**

**Date:**

**Place:**

**DECLARATION**

I declare that this written submission represents my ideas in my own words and where others’ ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

PRASHANTH VENKATESAN 118A3040 \_\_\_\_\_\_\_\_\_\_\_\_\_

ROHIT VIRAMANI 118A3045 \_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

Date:

**ACKNOWLEDGEMENT**

It gives us immense pleasure to thank Dr. Atul N Kemkar, our Principal for extending his support to carry out and develop the project. We also thank our Head of Department Dr. Lakshmi Sudha for her support in completing the project. We wish to express our deep sense of gratitude and thank to our Internal Guide, Ms. Bushra Shaikh for her guidance, help and useful suggestions, which helped in completing our project work in time.

We would like to thank the entire faculty of Information Technology Department for their valuable ideas and timely assistance in this project, last but not least, we would like to thank our non-teaching staff members of our college for their support, in facilitating timely completion of this project.

**Project Team**

Prashanth Venkatesan

Rohit Viramani

**ABSTRACT**

Restaurant Management System (RMS) is good to be a good solution for people who seek convenience in running a busy restaurant with multiple branches.

Users of RMS are the employees of the restaurant chain and will be registered in the app and connected to their restaurant branch through the reliable RMS system.

Restaurant Management System (RMS) is very user friendly, especially to new chefs as they can check out and learn new recipes, both veg and non veg through the app and easily change, update and innovate the menu, on popular demand or as they wish!

**Contents**

|  |  |  |
| --- | --- | --- |
|  |  | **Page No.** |
| **Chapter 1** | **Introduction** | **8** |
| **Chapter 2** | **Survey of Existing Apps** | **9** |
| **Chapter 3** | **Report on Present Investigation** | **10** |
|  | 3.1 Problem Statement | **10** |
|  | 3.2 Source of Problem Statement | **10** |
| **Chapter 4** | **Design and Implementation of Android Apps Components** | **11-16** |
|  | 4.1 Layouts | **11** |
|  | 4.2 Intents | **11-12** |
|  | 4.3 Activity | **12-13** |
|  | 4.4 SQLite | **13** |
|  | 4.5 Camera | **13-14** |
|  | 4.6 Location API | **14-15** |
|  | 4.7 WebView | **15-16** |
|  | 4.8 Generate APK file | **17** |
|  |  |  |
| **Chapter 5** | **Report on Proposed System and its Implementation** | **18-20** |
|  | 5.1 Block Diagram | **18** |
|  | 5.2 Flowchart | **19** |
|  | 5.3 Hardware | **20** |
| **Chapter 6** | **Results and Discussions** | 21-29 |
|  | 6.1 Summary of Screenshots with Navigational Flow | 29-29 |
| **Chapter 7** | **Conclusions** | **30** |
| **References** |  | **31** |

**Introduction**

Over the years, technology has tremendously revolutionized the restaurant industry. Much of the innovation has been with point-of-sale (POS) operations. There is a famous saying that “People eat with their eyes”.

The e-Menu provides additional information about menu items and drinks than a traditional paper menu. The simplicity and ease of access of a menu are the main things that facilitate ordering food in a restaurant. The service goes quicker. Restaurants can build their e-reputation and customer community in live.

The restaurant menu has evolved from its humble beginnings on carte chalkboards and imageless print to today’s detailed, colorful displays. With the emergence of digital tablets and user-friendly touch screen technology menus can move to a whole new surface. With this electronic menu, orders can be taken correctly the first time. There is no need to run back and forth.

**Survey On Existing Apps**

**1. Avero**

**Introduction**

Avero gives restaurateurs invaluable insights into business operations and sales. It delivers a powerful solution that gives you the answers you need about your restaurant's performance so you can get out of the back office and out onto the floor to connect with guests. When integrated with your restaurant POS, you can access a full suite of solutions like server performance tracking, inventory management, labor and food cost tracking, revenue management, and more.

**2. Bonusly**

**Introduction**

Buffer is an app that helps you manage all your restaurant social media accounts in one place. You can plan and schedule posts ahead of time, measure social media performance, and more right from your phone or laptop.

**3. Chowly**

**Introduction.**

Chowly seamlessly combines your third-party online ordering platforms – like GrubHub, UberEats, DoorDash, and others – with your restaurant's point of sale. Reduce errors and the time spent manually entering orders so your team can focus on doing what they love.

**4. RestaurantHub**

**Introduction.**

Calculating food cost percentage can be time-consuming and it’s often guesswork, but it doesn’t have to either. Toast’s free Food Cost Calculator helps you calculate your food cost percentage based on actual data, not a hunch.

**Report on Present Investigation**

**3.1) Problem Statement:**

The restaurant waiter information also by manual system kept use paper

and this is difficult for restaurant administrator to find waiter information, probability

missing the paper and difficult to arrange the schedule. Sometimes, waiter information

and customer information is important to restaurant administrator for reference in the

future.

Furthermore, restaurant side needs management in the section menu. This is the

important to restaurant waiter to manage the menu. Besides this section is for customer

viewer the menu that restaurant prepared and make their ordering.

**3.2) Source of Problem Statement:**

Restaurant Management Apps automates the whole process of creating and displaying menu, making the system completely autopilot and with the added camera feature it’s a cheery on top. Customers can easily scroll through the menu without any hassle, and also without touching the physical menu which is totally essential in this pandemic era

**Design and Implementation of Android Apps Components**

**4.1) Layouts**

Layout basically refers to the arrangement of elements on a page these elements are likely to be images, texts or styles. These are a part of **Android Jetpack**. They define the structure of [android user interface](http://web.cs.wpi.edu/~emmanuel/courses/cs4518/C17/slides/lecture03.pdf) in the app, like in an activity. All elements in the layout are built with the help of Views and ViewGroups. These layouts can have various widgets like buttons, labels, textboxes, and many others.

Some of the Layouts in Android are

* Linear Layout
* Relative Layout
* Constraint Layout
* Table Layout
* Frame Layout
* Absolute Layout

You can declare a layout in two ways:

* **Declare UI elements in XML**. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.

You can also use Android Studio's [Layout Editor](https://developer.android.com/studio/write/layout-editor) to build your XML layout using a drag-and-drop interface.

* **Instantiate layout elements at runtime**. Your app can create View and ViewGroup objects (and manipulate their properties) programmatically.

**We have used Constraint Layout for the login and registration page.**

**We have used Linear Layout for the profile page.**

**We have used Relative Layout and Card view in the menu viewing page**.

**4.2) Intents**

**Android Intent** is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.

It is generally used with startActivity() method to invoke activity, broadcast receivers etc.

There are two types of intents:

* **Explicit intents** specify which application will satisfy the intent, by supplying either the target app's package name or a fully-qualified component class name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start. For example, you might start a new activity within your app in response to a user action, or start a service to download a file in the background.

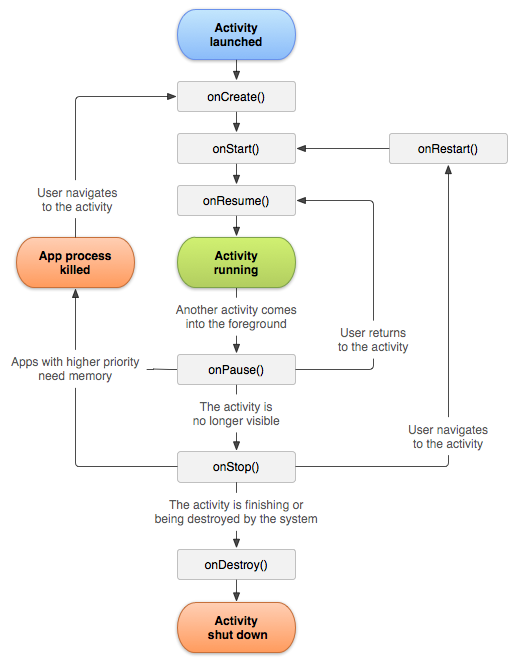
**We have used Explicit intent to connect various activities like going to Login page from Register Page.**

*startActivity(new Intent(RegisterActivity.this,LoginActivity.class));*

* **Implicit intents** do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. For example, if you want to show the user a location on a map, you can use an implicit intent to request that another capable app show a specified location on a map.

**4.3) Activity**

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.



**Activities in our application are:**

**Login Page**

**Employee Registration Page**

**Menu**

**Add Menu Items**

**Branch**

**Veg Recipes**

**Non-Veg Recipes**

**4.4) SQLite**

**SQLite** is an **open-source relational database** i.e. used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database.

It is embedded in android by default. So, there is no need to perform any database setup or administration task.

Here, we are going to see the example of sqlite to store and fetch the data. Data is displayed in the logcat. For displaying data on the spinner or listview, move to the next page.

**SQLiteOpenHelper** class provides the functionality to use the SQLite database

**We have used SQLite to store the employee details as well as store the menu items with their details like image of the dish, cost and dish type.**

**4.5) Camera**

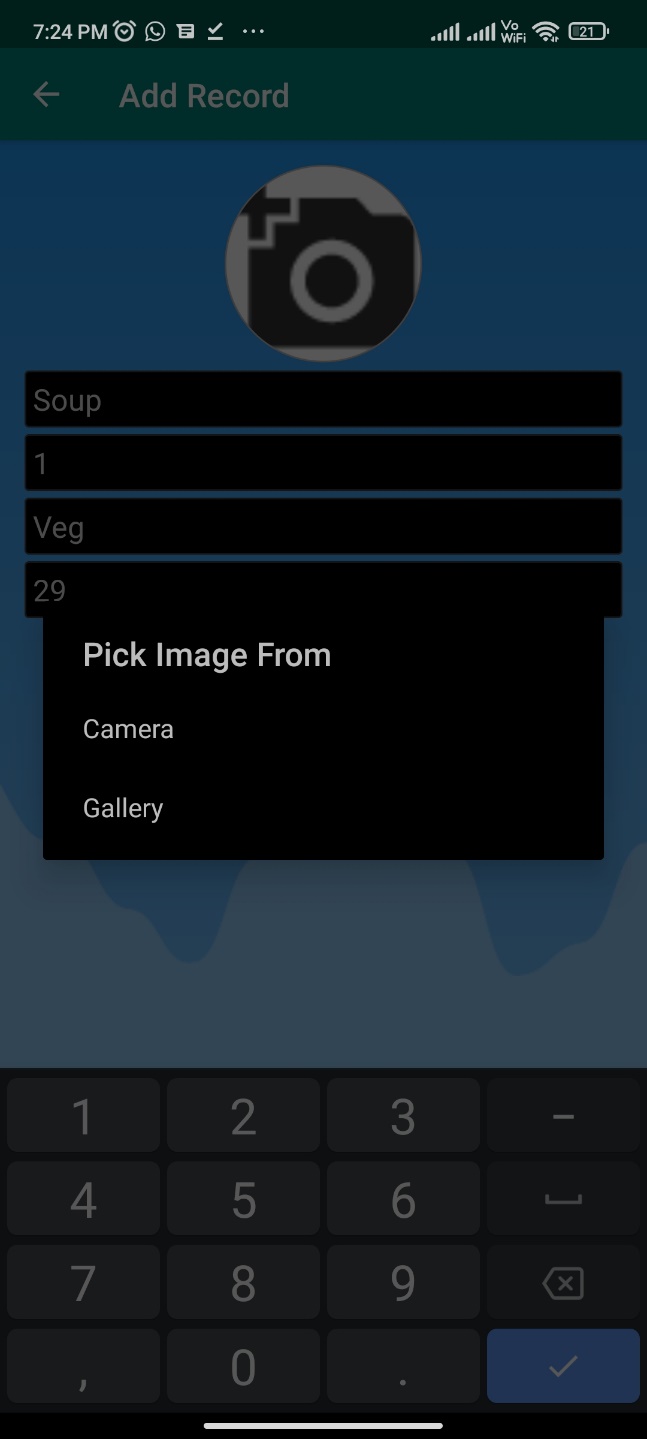
**Camera** is mainly used to capture picture of the house and we also added the functionality to crop the image of the house.

Android provides the facility to work on camera by 2 ways:

1. By Camera Intent
2. By Camera API

**We have used camera to allow employee take a image of the dish for viewing. We used the Camera Intent**

Intent cameraIntent=new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

****

**4.6) Location API**

The location APIs available in Google Play services facilitate **adding location awareness to** your app with automated location tracking.

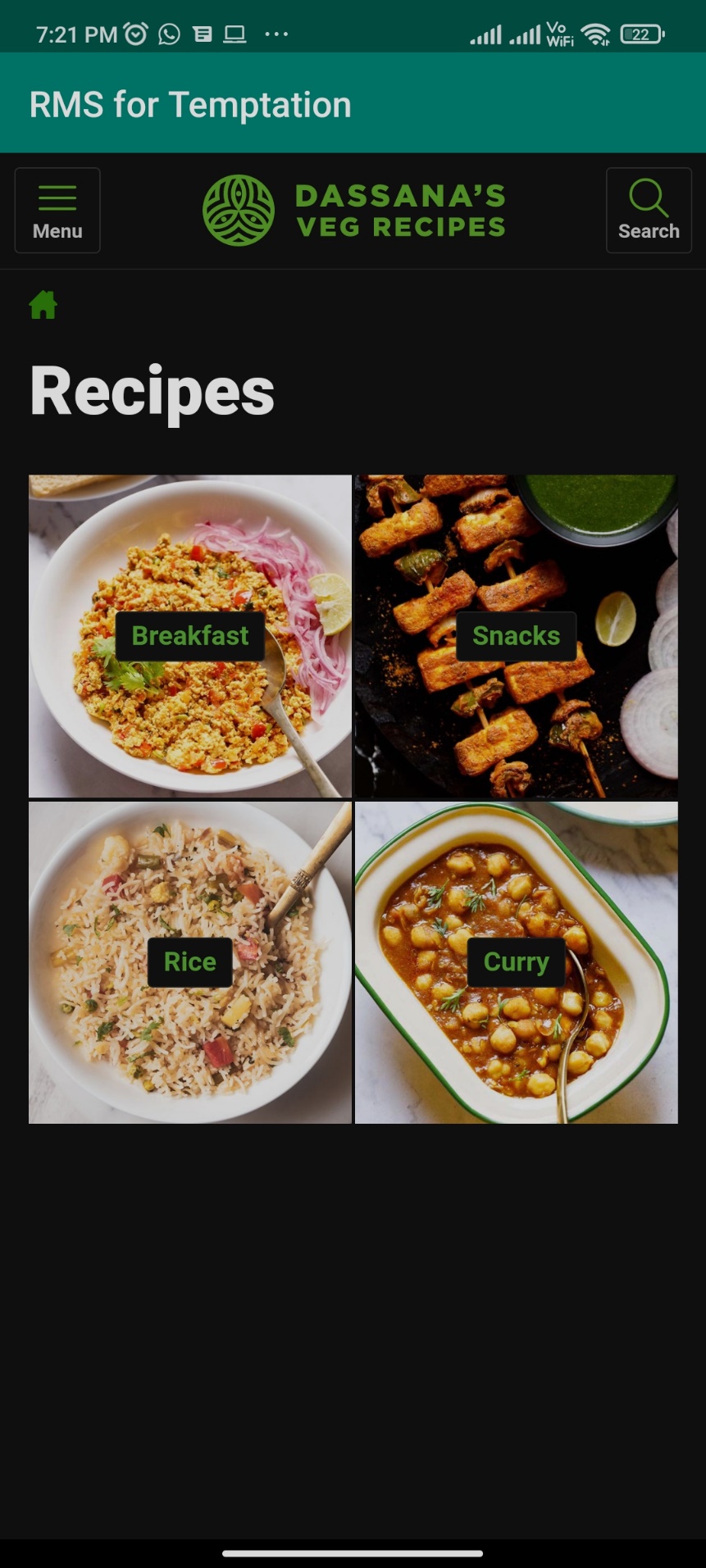
**We have used Location API to locate our nearest restaurant branches including the current location.**

****

**4.7) Webview**

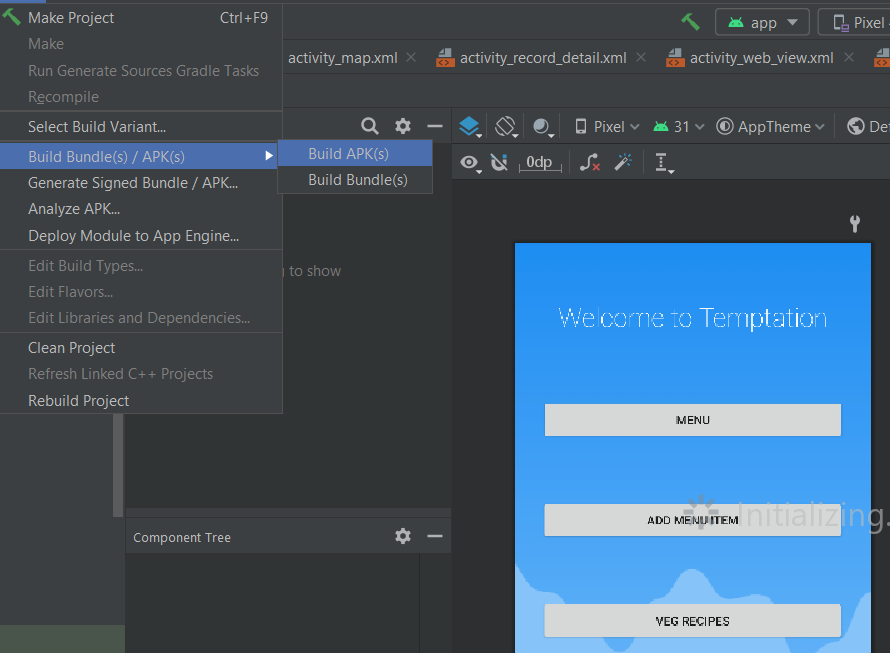
WebView is a view that display web pages inside your application. You can also specify HTML string and can show it inside your application using WebView. WebView makes turns your application to a web application.

**We have used WebView to connect our app with the recipe’s websites (veg/non-veg) which will be really essential for newly hired employees.**

****

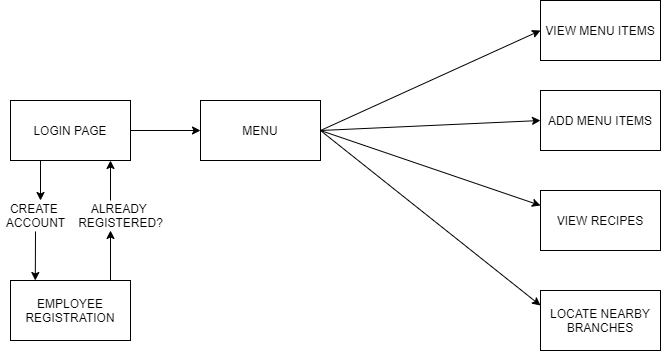
**4.8) Generate APK**

1. In the Android menu, go to Build > Build Bundle(s) / APK (s) > Build APK(s).
2. Android Studio will start building the APK for you. Once done, a pop-up on the bottom right will notify you of its completion. Click the ‘locate’ button in this dialog.
3. The ‘locate’ button should open File Explorer with the debug folder open that contains a file called “app-debug.apk”.
4. That’s it. Rename this file and share!

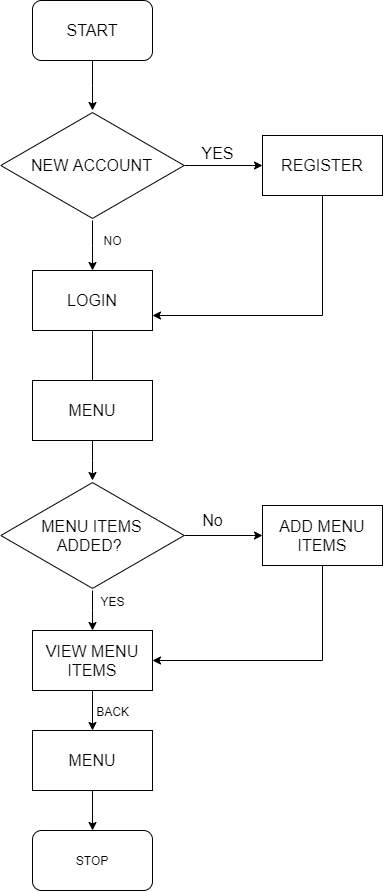


**Report on Proposed System and its Implementation**

**Block Diagram:**

****

**Flowchart:**

****

**Hardware –**

* Android Device
* GPS
* Internet
* Camera

**Software / External Libraries used with description –**

* Android Studio

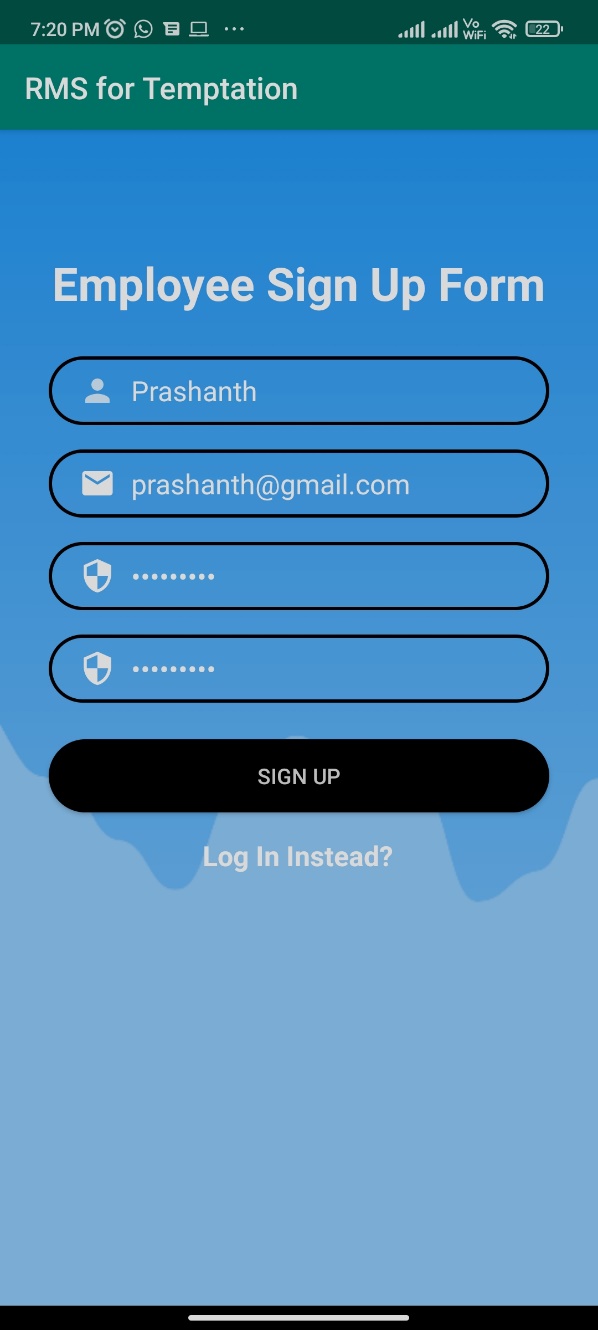
Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

**Results and Discussions:**

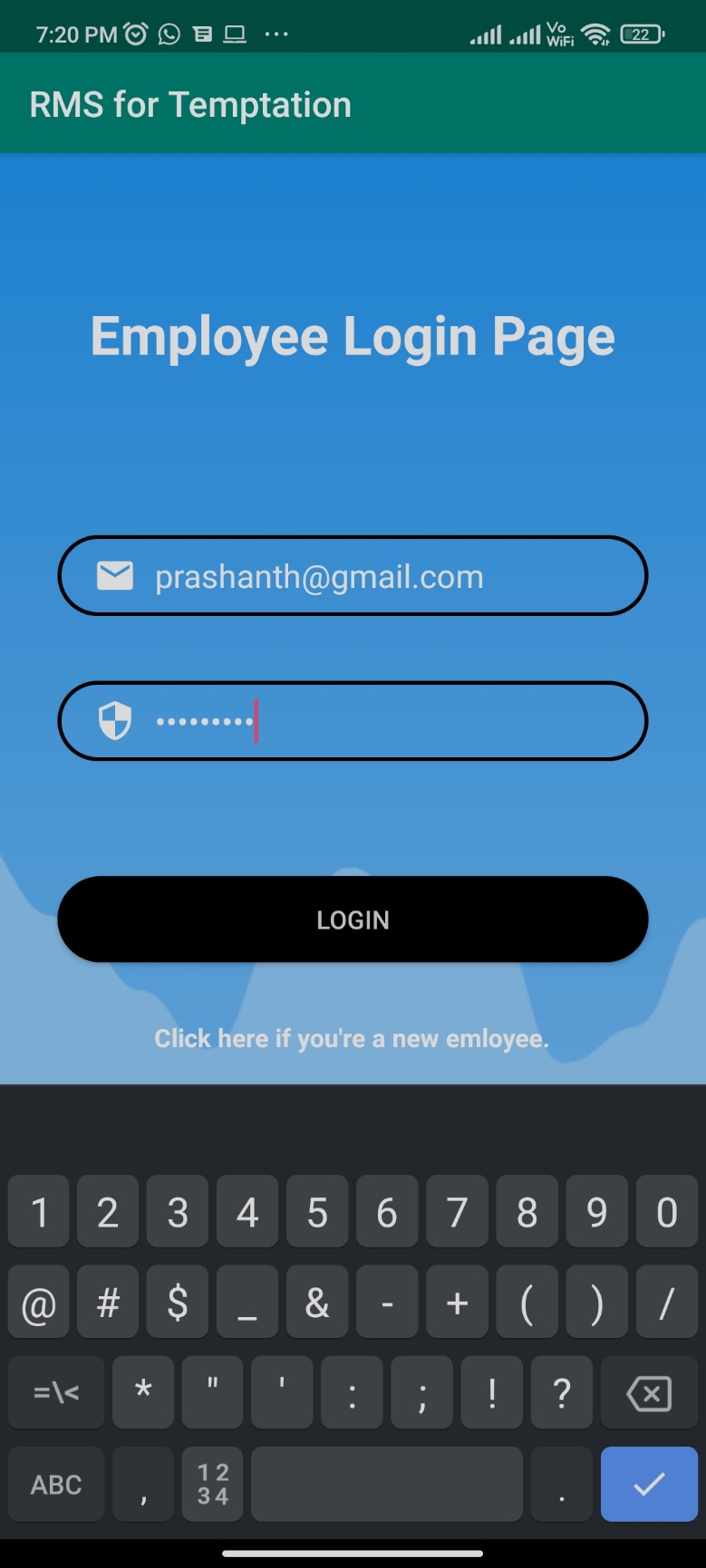
**Module A:**

**Employee Sign Up & Login In Page:**

For new users, they have to go through the registration process and basic details like name, email id, and have to make sure to create a strong password containing a special character, uppercase character, lowercase character and a number as well and the fields are 100% mandatory to be filled.



Once registered, the employee can login using the registered email id and password to access the content.



**Module B:**

**Main Restaurant Menu**

The employees can view the menu and create and edit new dishes as well.

****

**Module C:**

**Add Menu Item**

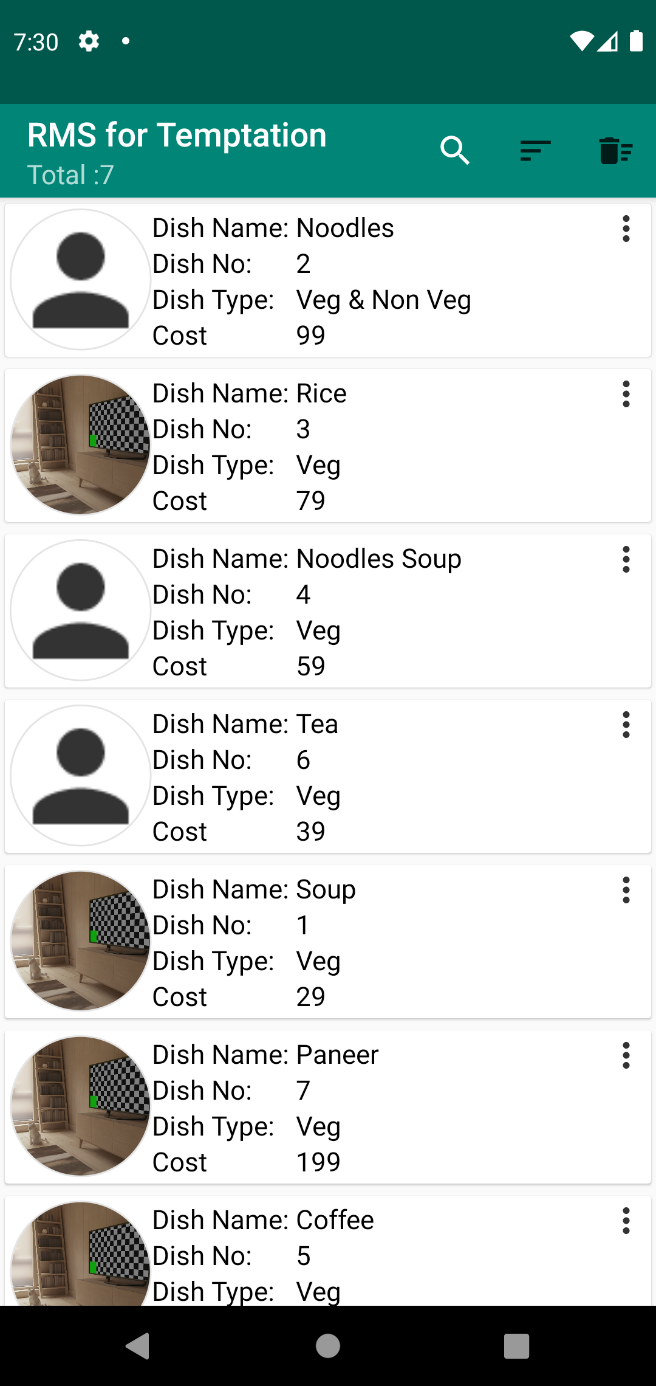
Employees can add new dishes which will include the name, serial no, dish type, dish image as well as the cost of the dish.

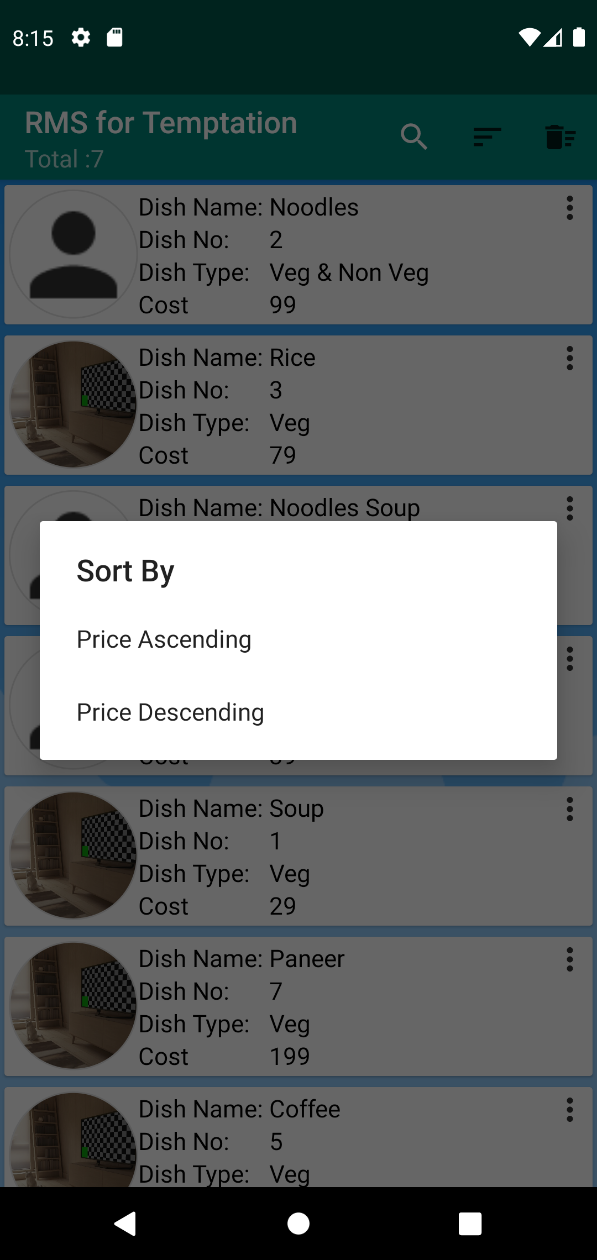
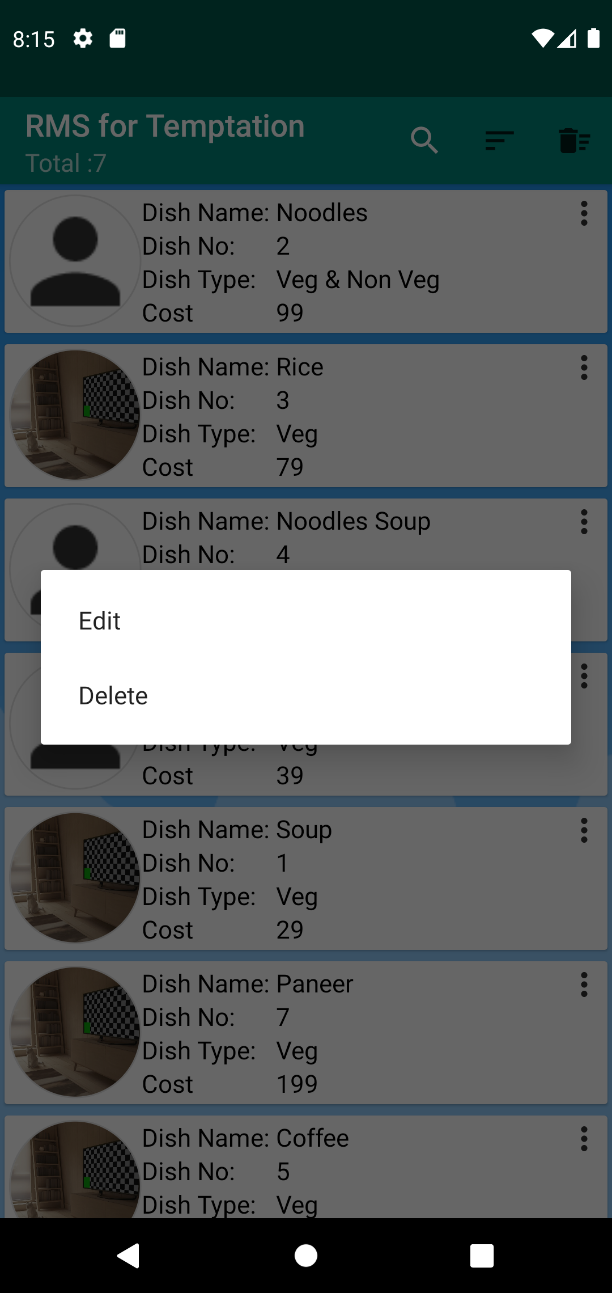
****

**Module D:**

**Menu**

Employees can view and edit/sort the dishes according to their preference and final menu will include the pricing as well as the dish type.

****

** **

**Module E:**

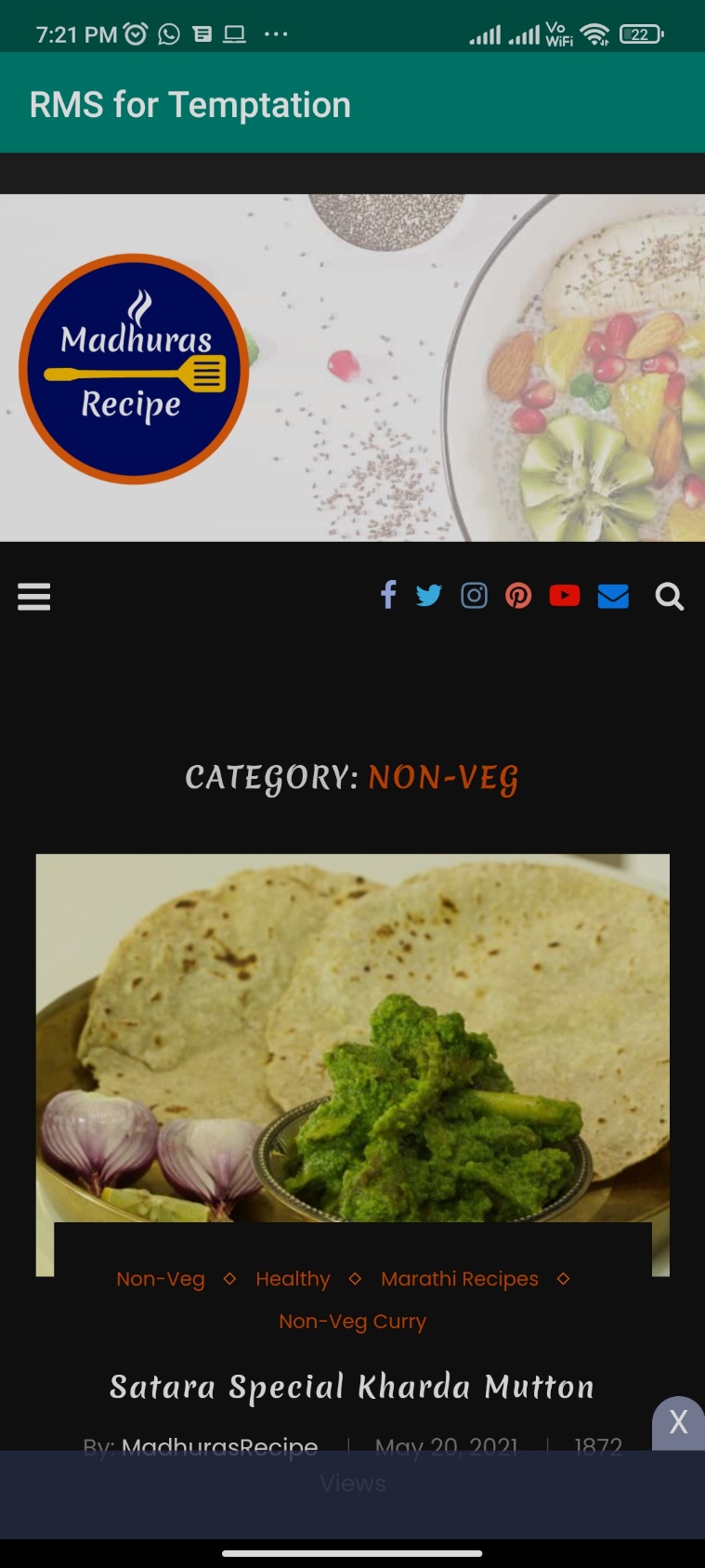
**Veg Recipes**

A webview site to view vegetarian recipes, which could be really useful for newly hired employees to view the ingredients and making process.

****

**Non Veg Recipes**

A webview site to view non-vegetarian recipes, which could be really useful for newly hired employees to view the ingredients and making process.

****

**Module F:**

**Branches**

Employees can view the nearby restaurant branches which could be useful to locate the branch while there is an interchange between the work of different branches of restaurant.

****

**Conclusion**

So therefore, through Restaurant Management System (RMS), the employees can conveniently increase their productivity while enjoying their work, especially the chefs. The management can also sort and keep track of their ingredients inventory in the menu module super simply. Happy mind, Happy food!

Future Scope:

In the life of software development, problem analysis provides a base for the design and development phase. The problem is analysed so that sufficient matter is provided to design a new system.

So, in the future, we would be integrating cross communication between the branches, i.e an Inter-Branch module. Which can cover the broad spectrum of the horizon, from communication related to inventory/menu, etc. up till HR/ Man-Power management between the branches.

**References**

1. https://developer.android.com/training/data-storage/sqlite
2. https://developers.google.com/maps/documentation/android-sdk/start

[3] https://developer.android.com/guide/topics/media/camera