

CHAPTER 1

INTRODUCTION

1.1 ANDROID STUDIO

Android Studio is an integrated development environment (IDE) for the Google Android Operating System. It is built based on JetBrains' IntelliJ IDEA Community Edition, and it is specifically designed for creating applications on Android devices. Some of the key features of Android Studio are as follows:

- **Instant Run** – a feature that pushes code and resource changes to the running app. It allows changes to be made to the app without the need to restart the app, or rebuilding the APK, so that the effects can be seen instantly.
- **An Emulator** – a virtual android device that can simulate a variety of hardware features such as GPS location, network latency, motion sensors, and multi-touch input that can be used to run and install the app. It can then be used for testing purposes.
- **Testing Tools and Frameworks** – extensive testing tools such as JUnit 4 and functional UI test frameworks are included with Android Studio. Espresso Test Recorder can generate UI test code by recording the developer's interactions with the app on a device or emulator. The tests can be run on a device, an emulator, in a Firebase Test Lab, or on a continuous integration environment.

1.1.1 ARCHITECTURE OF ANDROID STUDIO

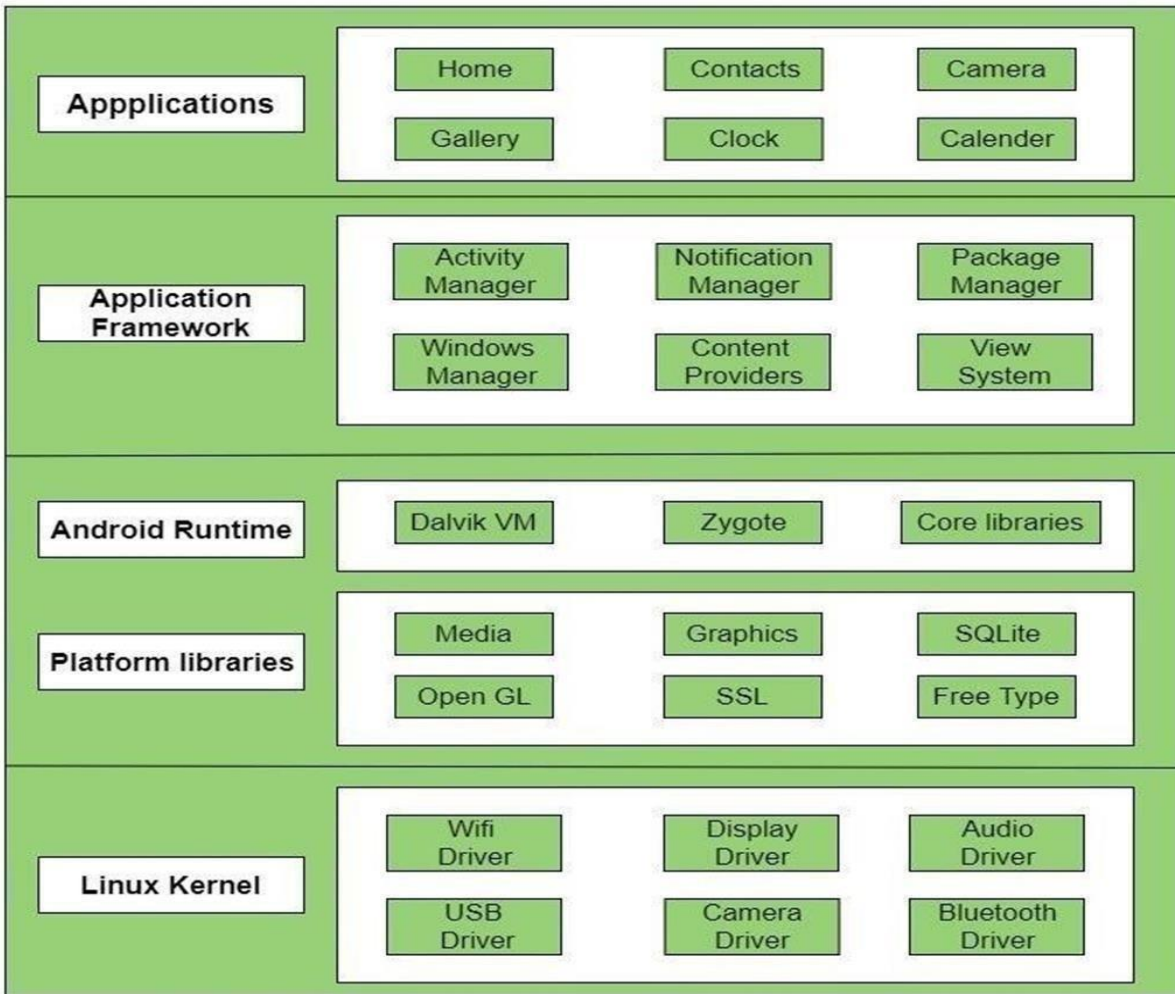


Fig 1.1.1: Architecture of Android studio

1.1.2 INSTALLING AND RUNNING APPLICATIONS ON ANDROID STUDIO

1.1.2.1 SYSTEM REQUIREMENTS

- OS: Windows 8/8.1/10/11 (64-bit)
- CPU: 2nd generation Intel CPU (Sandy Bridge) or newer, AMD CPU with support for a Windows Hypervisor
- Memory: 8 GB RAM
- Free storage: 8 GB
- Screen resolution: 1280 x 800

1.1.2.2 SET UP OF ANDROID STUDIO

To set up Android Studio, you can follow these steps:

Download Android Studio: Go to the official Android Studio website (<https://developer.android.com/studio>) and download the latest version of Android Studio for your operating system (Windows, macOS, or Linux). **Install Android Studio:** Once the download is complete, run the installer and follow the on-screen instructions to install Android Studio on your computer. The installation process may take a few minutes. **Configure Android SDK:** After installation, launch Android Studio. On the welcome screen, select "Configure" and then choose "SDK Manager." The SDK Manager allows you to download the necessary Android SDK components for development.

- a. In the SDK Platforms tab, select the Android versions you want to target with your app. It's recommended to choose the latest stable version as well as the minimum version you want to support.
- b. In the SDK Tools tab, select the components you need, such as the Android Emulator, Android SDK Build-Tools, and others. Again, it's recommended to use the latest stable versions.
- c. Click "Apply" to start downloading and installing the selected SDK components.

1.1.2.3 CREATE ANDRIOD VIRTUAL DEVICE

- Once the SDK installation is complete, click on the "AVD Manager" icon on the toolbar or go to "Tools" > "AVD Manager."
- Click on "Create Virtual Device."
- Choose a device definition that matches your testing requirements and "Next."
- Select a system image for the virtual device. It's recommended to choose the image with the latest API level. Click "Next."
- Configure any additional settings for the virtual device, such as RAM and storage, or leave them as default. Click "Finish" to create the virtual device.

1.2 IMPORTANT FEATURES

1. Incorporating Search Feature

Incorporate the advanced search feature for mobile app users that helps them find and scour for the content they need. This adds a huge impact, especially if the app has a vast database of content; it quickly becomes cumbersome for users to access easily. With the advanced search feature and the search box, top-tier apps allow users to search for relevant content with the help of keywords and key phrases. The search feature makes the app content accessible to all kinds of users.

2. Incorporate Social Media Integration

Social media sharing is not restricted to picture sharing anymore. Most apps now include social media integration so that users can communicate and collaborate with other people, irrespective of whether they have the same app or not. App owners also reap the advantage of having social media on their apps since more users sharing their stuff will, in turn, impact their reach and growth positively. Making social media integration seamless is the key here with social sharing buttons directly at the side of every app page.

3. Use of Responsive App Design

Mobile app design based on every screen size and resolution is quite robust. Responsive app design resolves this problem helping the app acclimatize to any screen size and resolution of small smartphones as well as larger tablets. Some smartphones also feature screens that hover around the size of a tablet. Check out the newest iPhones and its competitors in the market; they are large enough to be called mini-tablets. By incorporating responsive app design during app development, users will experience the app in the best light irrespective of the device they are using.

4. Simplicity in Design with Uncluttered Elements

The mobile app design has now evolved into a smart aesthetic exercise where proper use of white space is quite essential. Irrespective of the logic complexity of the app, the user experience design is expected to be uncluttered and straightforward with the elements placed in proper context, space, and size. Mobile app users are always keen to use apps that are easy to-use and can be navigated without a problem. This is where a simple UI/UX design can help companies describe the navigation structure, workflow, and principles of the app succinctly to its intended audience.

1.3 ADVANTAGES OF ANDROID STUDIO

- **Android is an open-source platform allowing UI customization**

Licensed under Apache, Android is an open-source operating system whose code developers can change to build customized User Interfaces. App developers building applications for this platform can get access to the core codes and are at a liberty to change the theme to get the customized outcomes. This is not possible when it comes to iOS and apps have to strictly adhere to the core code specifications when developing apps for the specific platform.

- **Supports cloud storage enabling sync of devices with G-account**

Since Android is a Google product, users having a Gmail account can have access to cloud storage that is supported by the tech company. This means that users can sync devices using Google accounts. Moreover, Google gives 15GB of free cloud storage to every user, which is good for an average person using it for personal purposes.

- **Continual improvement & removal of old features**

Google Android is supported by a huge community of developers and also users who continue to give feedback about the features, their pros and cons. As a result, there is continuous check on the codes and features, making modifications and alterations, bringing in better upgrades all the time. This is one of the reasons why Android is always adding new features while removing older ones that users do not like.

- **Supports 3rd party widget & information display on screen**

Android gives users the freedom to download 3rd party widgets and also display their content on the home screen. If a user wishes to view time and temperature shown by a specific widget on the home screen, it is possible with devices running on the platform.

- **Supports running multiple apps simultaneously**

With Android running on a device with good hardware specification, as a user you can have multiple apps running simultaneously. You can continue to listen to music as you check your messages or download files that you've received or even upload them from your device or drive. There are a lot of Android app development companies who build applications based on Android that are very useful in our daily lives.

- **Expandable memory & runs on affordable large devices**

One of the biggest advantages of using devices running on the Android platform is that it supports expandable memory. iOS devices on the other hand do not support external memory expansion by adding a memory card to the phone. Users of this platform enjoy the privilege of storing e-books, music, videos and games on their devices.

1.4 ABOUT RECIPE APP

Recipes application is a very useful app for people who love to cook and try out new recipes. Recipe App is a feature-rich recipe app designed to help you discover, organize, and create delicious meals right from your Android device. With its user-friendly interface and powerful functionality, it's the perfect tool for both aspiring chefs and seasoned cooks. Recipe App offers a collection of recipes ranging from quick and easy everyday meals to gourmet delights. Explore various cuisines, dietary preferences, and special occasions, ensuring you always find the perfect recipe to suit your needs. The app's intuitive interface makes it easy to navigate and find what you're looking for. With clean design, accessing recipes and features is a breeze, even for first-time users. Each recipe comes with detailed instructions, including ingredient measurements, step-by-step cooking methods, and helpful tips. Follow along with ease and gain confidence in the kitchen, regardless of your skill level.

1.5 OBJECTIVES

The main aim of our application is to provide recipes to the consumers based on the ingredients already available with them.

- To help the user decide a recipe to cook from the ingredients available with him/her.
- To guide the user to the recipe based on the user needs.
- To help save the user money and time by tediously referencing cook books.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Functional Requirements

- **Recipe Search:** Provide a search functionality that allows users to find recipes.
- **Recipe Display:** Display recipes in a clear and user-friendly format, including ingredients, step-by-step instructions
- **Offline Access:** Support offline access to previously viewed recipes for users to access recipes even without an internet connection.

2.2 Non-functional Requirements

- **Usability:** The app should have a user-friendly and intuitive interface, making it easy for users to access recipes.
- **Performance:** The app should be fast and responsive, with quick loading times for recipe pages.
- **User Interface and Design:** The app should have an intuitive and visually appealing user interface, with clear navigation, readable text, and a consistent design across different devices.
- **Platform Compatibility:** The app should be compatible with popular platforms and devices, such as iOS and Android, and adapt to different screen sizes and orientations.
- **Scalability:** Design the app architecture to handle a growing user base and increasing amounts of recipe data without compromising performance or user experience.

2.3 Software Requirements

SOFTWARE REQUIREMENTS	
Application Required	Android studio Application (Electric Eel)
Windows	Windows 10 and Above
Language Used	Java, XML
Emulator	Pixel 6

2.4 Hardware Requirements

HARDWARE REQUIREMENTS (MINIMUM)	
RAM Memory	8GB or more
Processor	Intel® Core (TM) i3-1005G1 CPU @ 1.20GHz 1.19 GH
System Type	64-bit operating system, x64-based processor
Mobile Device	All devices

CHAPTER 3

SYSTEM DESIGN

3.1 Initialization

To initialize and design the recipe app, we can follow these steps:

- **Step 1:** Identify the main components of the app:
 - MainActivity: This is the entry point of the app and handles the user interface.
 - XML layout files: Design the user interface using XML layout files, such as activity_main.xml for MainActivity
- **Step 2:** Create the necessary classes and XML layout files:
 - Create MainActivity.java class and corresponding activity_main.xml layout file.
 - Create RecipeActivity.java class and RecyclerViewAdapter java file
- **Step 3:** Define the activities and services in the AndroidManifest.xml file:
 - Register the MainActivity and RecipeActivity
- **Step 4:** Design the user interface in the XML layout files:
 - Customize the activity_main.xml files to include the necessary views, images, text fields, and any other components required for your app's design.

- **Step 5:** Implement the logic and functionality in the Java classes:
 - In MainActivity, implement the array list for adding recipes.
 - In RecipeActivity, implement to create intents.

- **Step 6:** Customize the app's design and functionality:
 - Modify the code and XML layouts as per your specific requirements. You can customize the UI design, add additional features, or integrate with external APIs for location tracking or emergency contact notifications.

- **Step 7:** Build and run the app:
 - Build the app and run it on an emulator or a physical device to test the initialization and design.

3.2 Display

To display the recipes for user, a list of recipe images will be displayed. The user can select the image of his choice, then the next page displays the ingredients need to cool and method of cooking.

3.3 Flow Chart

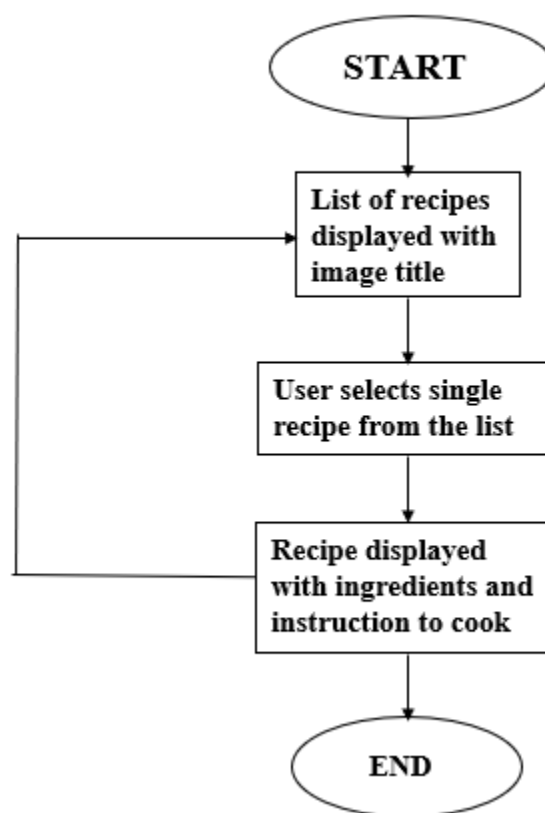


Fig. 3.3.1 Flow chart

CHAPTER 4

IMPLEMENTATION

4.1 Pseudo code

To execute the provided code, you can follow these steps:

Step 1: Gather ingredients Ingredients:

- Ingredient 1
- Ingredient 2
- Ingredient 3.

Step 2: Prepare the ingredients Method:

- Preheat the oven to [temperature].
- Chop the Ingredient 1 into small pieces.
- Mix Ingredient 2 with Ingredient 3 in a bowl.

Step 3: Cook the dish Method:

- Heat a pan over medium heat.
- Add a bit of oil to the pan.
- Place Ingredient 1 in the pan and cook until golden brown.
- Pour the mixture of Ingredient 2 and Ingredient 3 into the pan.

CHAPTER 5

SNAPSHOTS

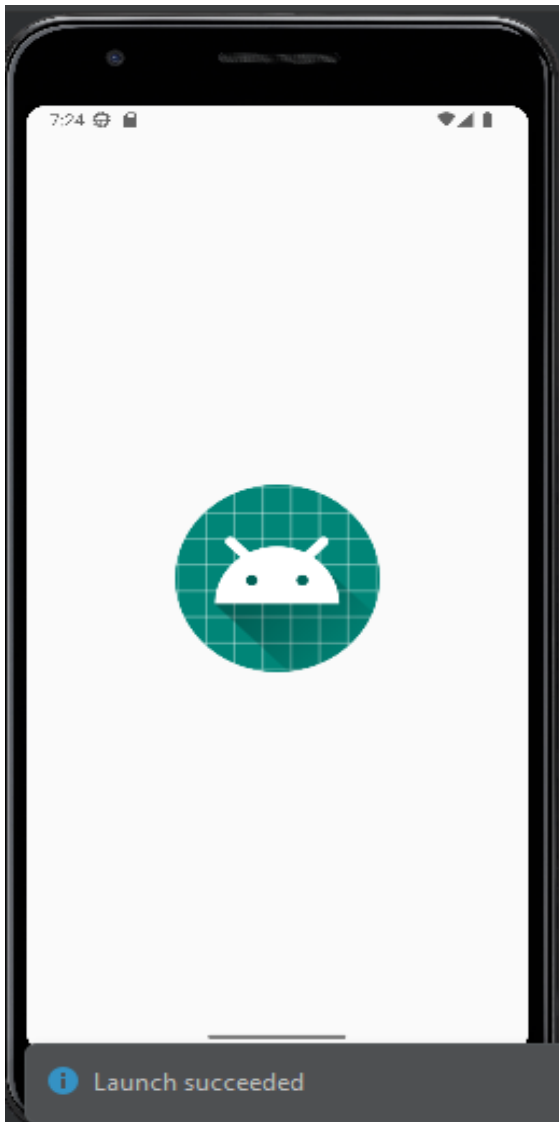


Figure 5.1: Welcome page

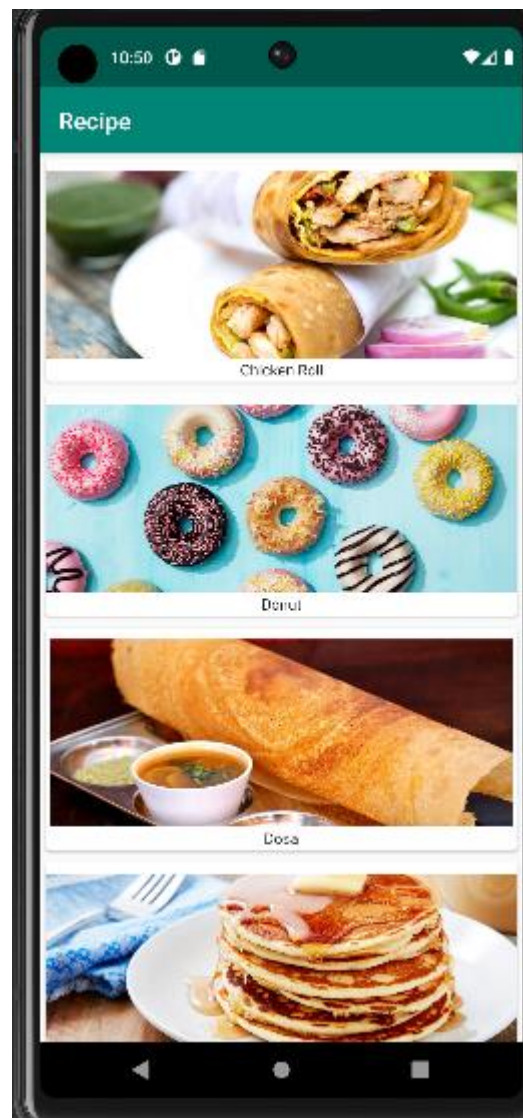


Figure 5.2: Image List of recipes

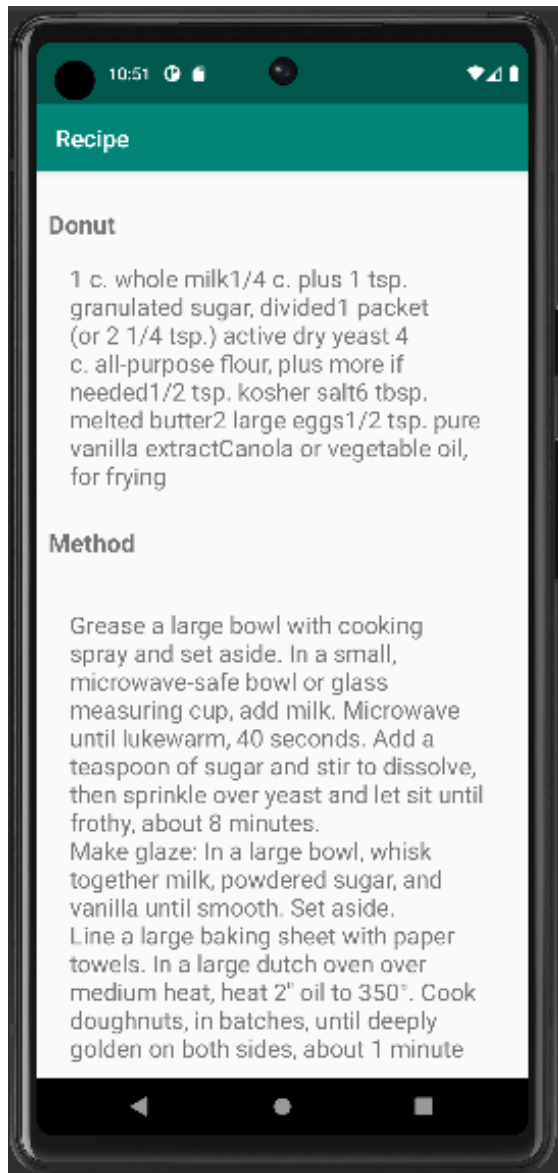


Figure 5.3: Recipe of Donut dish



Figure 5.4: Recipe of Pasta dish

