CHAPTER 1

INTRODUCTION

A grocery shopping is an android application where users can purchase and order groceries online. The system is developed with a user-friendly and attractive GUI. It delivers a wide range of groceries available online. Users have to first login into the system to view the groceries and add them into their cart. They can then order it by making a secure online payment via CREDIT CARD.

1.1 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.

1.2 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.

1.3 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 method

1.4 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.

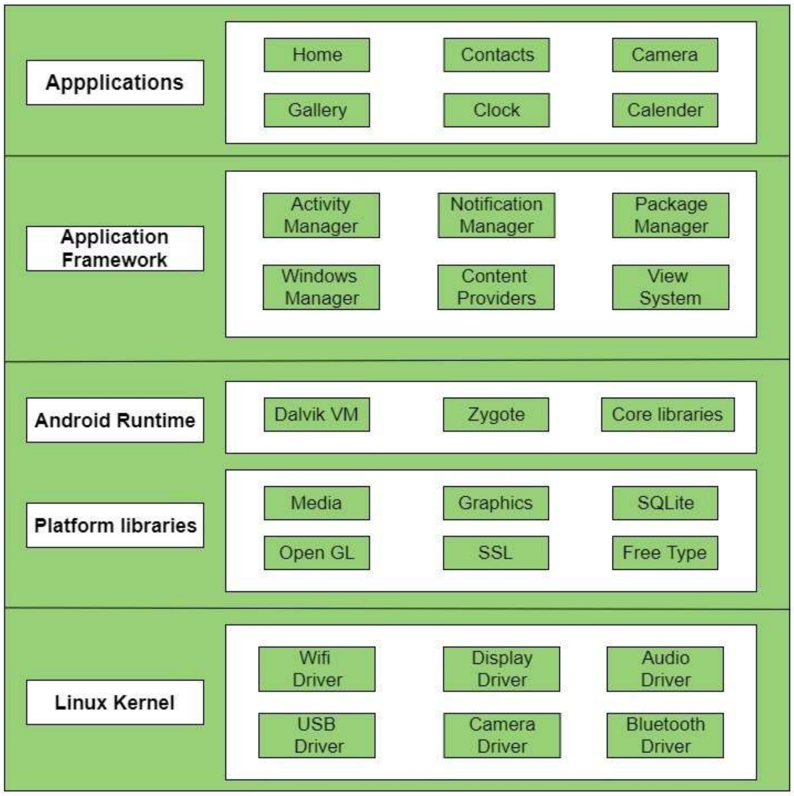


Fig. 1.5.1 Android Architecture

1.6 MOTIVATION

There are several motivations for developing a grocery app for Android. Here are some key reasons:

Convenience: A grocery app provides users with the convenience of ordering groceries anytime and anywhere, directly from their Android devices. They can browse through a wide range of products, add them to their virtual shopping cart, and have them delivered to their doorstep. It eliminates the need to physically visit a grocery store and saves time and effort.

Time-saving: With a grocery app, users can save valuable time that would otherwise be spent traveling to the store, searching for products, and waiting in queues at the checkout counter. The app can provide features like personalized shopping lists, recommended products, and quick reordering of frequently purchased items, making the shopping experience faster and more efficient.

Product variety and availability: A grocery app can offer a vast selection of products, including fresh produce, pantry staples, specialty items, and even niche or hard-to-find products. It allows users to explore a wider range of options compared to a physical store. Moreover, the app can provide real-time information on product availability, helping users know whether an item is in stock before making a purchase.

Personalization and recommendations: By leveraging user data and preferences, a grocery app can offer personalized recommendations and suggestions based on past purchases, dietary restrictions, or favorite brands. This level of personalization enhances the shopping experience and helps users discover new products that align with their preferences.

Deals, discounts, and loyalty programs: Grocery apps often provide exclusive deals, discounts, and loyalty programs that can attract and retain customers. Users can receive notifications about ongoing sales, promotions, or special offers, allowing them to make cost-effective choices while shopping.

Meal planning and recipe integration: Many grocery apps integrate features for meal planning and recipe suggestions. Users can access a database of recipes, select the ones they want to prepare, and automatically generate a shopping list with the required ingredients. This integration simplifies the meal planning process and ensures that users have all the necessary items in their cart.

Order tracking and delivery updates: A grocery app can provide real-time order tracking and delivery updates, allowing users to monitor the status of their orders. They can receive notifications about the estimated delivery time, track the delivery personnel's location, and get notified when the order is out for delivery or has been successfully delivered.

These motivations demonstrate how a grocery app for Android can enhance the overall grocery shopping experience, providing convenience, time-saving, personalization, and a seamless ordering and delivery process.

1.7 PROBLEM STATEMENT

Grocery store is an application which is A nice clean online grocery app based on android application which is user friendly. The app shows up all the menu items by the categories, recently visited and most discounted. On the home page shows the price of the items, add to cart, product details and more.

1.8 OBJECTIVE

The main objective of this application is to provide a user-friendly interface to all the users. We ensure hygienic, best quality products and availability of the products to customers. And provide the products with more discount. Our Metro Mart application helps them by taking the orders from customers and delivering their required groceries to home.

1.9 PROJECT APPLICATION

This application is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus, the customer will get the service of online shopping and home delivery from his favorite shop. Our application has several advantages.

The main objective of this application is to provide a user-friendly interface to all the users. We ensure hygienic, best quality products and availability of the products to customers. And provide the products with more discount. Our Metro Mart application helps them by taking the orders from customers and delivering their required groceries to home.

CHAPTER 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. In the application we can view various news with respect to their category, we can share app, view about the app, contact the helpline, send mail to the developers and provide feedback. Another advantage of the application is that people can news in a particular category and it is very easy to view them.

The project consists of the following components:

1. **Activities**: It deals with the UI and the user interactions to the screen. In other words, it is a User Interface that contains activities.
2. **Services**: Services are the background actions performed by the app; these might be long running operations. A service might need other sub-services so as to perform specific tasks.
3. **Content Provider:** Content Provider is used to transferring the data from one application to the others at the request of the other application.
4. **Broadcast Receivers:** A Broadcast is used to respond to messages from other applications or from the System.

2.2 USER INTERFACE DESIGN

In our project GORCERY STORE, we have used XML with different attributes and a code written in Java language for various activities.

Once the user opens the application it shows the welcome page with the button lets shop, after this new user must register through signup page and already registered users can login to the application by providing their mail id and password. Main activity is the home activity shows different activities. These further can be proceeded to product viewing page which at last can be proceeded to the checkout. Once the order is confirmed the page will show thank you as responding from the admin to the user. All the orders can be viewed in my order activity. The user can logout on clicking” Logout” option provided on the navigation drawer to exit from the app.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFCATIONS

The software specific requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. Software Environment is a technical specification of requirement of software product. This specifies the environment for development, operation and maintenance of the product.

Technology used:

- JAVA
- SQLite

3.1 Functional Requirements

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behaviour between inputs and outputs.

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describe all the cases where the system uses the functional requirements, these are captured in use cases. Functional requirements are supported by non-functional requirements (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do," while non-functional requirements take the form "system shall be ." The plan for implementing functional requirements is detailed in the system design, whereas non-functional requirements are detailed in the system architecture.

The various methods used in this project are as follows: -

- Emulator

  To perform and display the functionality of the project.
- Android studio

  To create, design, test, debug and run the android project
- Mouse

  To navigate through the emulator.
- Keyboard

To give inputs to the project.

3.2 Non-Functional Requirements

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. They are contrasted with functional requirements that define specific behaviour or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

Broadly, functional requirements define what a system is supposed to do and nonfunctional requirements define how a system is supposed to be. Functional requirements are usually in the form of "system shall do ", an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of "system shall be ", an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed.

Non-functional requirements are often called the "quality attributes" of a system. Other terms for non-functional requirements are "qualities", "quality goals", "quality of service requirements", "constraints", "non-behavioural requirements", or "technical requirements".

Informally these are sometimes called the "ileitis", from attributes like stability and portability. Qualities—that is non-functional requirements—can be divided into two main categories:

- Execution qualities, such as safety, security and usability, which are observable during operation (at run time).
- Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

It is important to specify non-functional requirements in a specific and measurable way.

Non-Functional Requirements are as follows: -

1. **Dependability:** The dependability of a computer system is a property of the system that equates to its trustworthiness. Trustworthiness essentially means the degree of user

confidence that the system will operate as they expect and that the system will not fail in normal use.

2. **Availability:** The ability of the system to deliver services when requested. There is no error in the program while executing the program.
3. **Reliability:** The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.
4. **Safety:** The ability of the system to operate without catastrophic failure. This program is user friendly and it will never affect the system.
5. **Security:** The ability of the system to protect itself against accidental or deliberate intrusion.

3.4 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

The hardware requirements for the app is as follows:

- Processor: The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. The processor used for the development of app is 2.0 Ghz processor speed
- Memory: All software, when run, resides in the random-access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. The Memory requirement for the development of app is 2GB RAM.
- Visual Display Unit. The app is tested on a Android Virtual device with the help of a Visual display unit. The Visual Display unit specification is 800*600 colors
- Hard Disk Drive: Data storage device requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space. The secondary memory requirement for the app development is 40 GB HDD

- Mobile Device-The Mobile devise used for testing the app is Nokia 7.2 is an Android smartphone manufactured by HMD Global. It was announced in October 2019 and released in November 2019. The phone has a 6.3-inch IPS LCD display with a resolution of 1080 x 2280 pixels. It is powered by a Qualcomm Snapdragon 660 processor and 4GB of RAM. It has 64GB of internal storage, expandable up to 512GB via a microSD card. The phone has a dual-camera rear setup, with a 48-megapixel main sensor and a 5-megapixel depth sensor. The front-facing camera is 20 megapixels. The phone Android 11.

3.5 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

The software requirements for the app are as follows:

- Operating System: An operating system is system software that manages computer hardware and software resources, and provides common services for computer programs. The operating System used for development of app is Windows 10,11,7 or anyone.
- Android Operating System: Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. The android version used for development and testing of the app is Android 11 SDK 30.
- Frontend Language: t is also referred to as the 'client side of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and a navigation menu. The Frontend Language used is XML and JAVA.
- Backend Language: Back-end languages are what programmers use to create the internal systems that work in the background of a web application. The back end of a program or application refers to the part of the software that users, or clients, don't see, the backend language used for the development of app is SQLite and JAVA.
- IDE-An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. The IDE used for the development of the app is Android Studio Electric Eel (2022.1.1).

CHAPTER 4

SYSTEM ANALYSIS

System Analysis is a process of understanding the system requirements and its environment. It is one of the initial stages of the software development life cycle. System analysis is the process of breaking the system down into its individual components and understanding how each component interacts with the other components to accomplish the system’s overall goal. Here's an overview of the steps involved in the system analysis process:

4.1 Requirement Gathering:

In this phase, we identify the target audience and their needs and determine the core features required, such as buying groceries and doing the payment of it.

4.2 Functional Analysis:

In this phase, we break down the core features into detailed functionalities and create use cases to describe how users will interact with the application and define the flow of activities, screens, and interactions within the app. Potential constraints and dependencies, such as device compatibility and API integrations are also identified in this phase.

4.3 Data Analysis:

Identifying the types of data required for the application and analysing the data storage and retrieval needs. Data privacy and security requirements, such as user authentication are also taken into consideration.

4.4 System Design:

In this phase, we define the app's architecture, including components like activities, fragments, and services. We design the user interface (UI) and user experience (UX) to ensure easy navigation and intuitive interaction. Determining the necessary data models and database structure for storing medication and user information is also carried out in this phase.

4.5 Integrating various features:

In this phase, we try integrating various functionalities and features into our project. We can utilize built-in features of Java, such as alarm manager and notifications, to implement reminder functionalities effectively. Patients and caregivers can receive timely reminders for medication intake, reducing the chances of missed doses.

4.6 Testing and Quality Assurance:

In this phase, we define a testing strategy, including unit testing and integration testing. Test cases are created to verify each functionality and rigorous testing is performed to ensure that the application works as intended and meets the requirements. If any bugs are identified then they are addressed and fixed.

4.7 Monitoring and Maintenance:

In this phase, we plan for ongoing maintenance and future updates, including bug fixes, performance improvements, and feature enhancements. We will try monitoring user feedback and incorporating user suggestions to improve the application.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Introduction to Programming Languages, IDE, Tools, and Technologies used for this Implementation

5.1.1 Java:

As the project is developing an Android Application, the default programming language is Java. All Android applications are built using Java in Android Studio or Eclipse or both. Java is a popular and widely used language throughout the world. Java is one of the powerful programming languages like C, and C++ developed by Sun Microsystems which has many powerful features as described below. After the development of C, C++, and Java has come into evolution by addressing their drawbacks. It is one of the open-source projects that could be easily installed on our machine. The language is also easy to learn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. Many powerful features are supported by Java including various libraries, application services, and graphics libraries for 2D/3D applications.

The language is flexible enough to maintain code complexity, test, implementation, integration, and support. Apart from these, there are other key features that make Java more special. It is an object-oriented programming language, one of the important hierarchies in the programming languages which is used to implement real-time applications, it provides for code reusability, it has a platform independence feature including any virtual machines(Write Once Read Everywhere), as in no need to write the code for different OS as the Java Compilers convert the java source files to bytecode and this could be interpreted by any machine and the actual code is compiled irrespective of any machine, OS. It is more secure as the compilers are designed efficiently to figure out any kind of errors.

5.1.2 IDE’s, Tools, and Technologies:

5.1.2.1 Android Studio:

Android Studio is exclusively designed for developing Android applications. It consists of all Android SDK tools to design, develop, maintain, test, debug, and publish our app. The IDE is designed very efficiently which makes the developer’s job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, and built-in functions or it could be anything else when we press the first letter of it. Say, suppose we declared a few variables or methods that start with an

‘S’, it automatically senses everything that starts with an ‘S’ and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into Git Hub. All Java files, and layout files (for design) are integrated into a single project easily. After the completion of the project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file on any device and use the application. Other main tools include Android SDK, ADB, and Gradle Build.

5.1.2.2 Android Software Development Kit (SDK):

One of the main tools used in developing Android applications is as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing a lot of code and build applications faster.

5.1.2.3 Android Debug Bridge (ADB):

Android SDK uses the ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

5.1.2.4 Gradle Build:

Gradle Scripts is a recent feature that is added to Android Studio. It is basically an automated build system that is used to automate the various phases involved in designing an application that includes design, development, testing, debugging, and publishing. We need to configure the project and modules by mentioning all the supported jar files, SDKs, version name, level, compiled SDK version.

5.1.2.5 Android Device Monitor:

If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app on the device, we need to root the device and run commands in adb shell to get permissions.

5.1.2.6 SDK Manager:

It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with the device or any other compatibility issues and to download any component that is required.

5.1.2.7 AVD Manager:

It is used to create virtual devices of any desired API level to support higher-level SDKs in case our device does not support them. Using emulators to test the application is difficult as it might be a little slower when compared to the real device.

5.1.2.8 SQLite Database:

Android also supports an inbuilt database which is Android SQLite to develop any small applications and perform any CRUD (Create, Update, and Delete) operations. As it is not flexible enough to support a substantial number of data, for complex applications we are using other external databases.

5.2 Security and Permissions in Android:

Security notions in Android are quite high. Whenever a new Android Application is created, a unique user and group ID. This makes the maintenance of the application an easier way to avoid any security or privacy issues. As the application is created uniquely, it becomes private and no one can access other’s applications. Permissions are another important concept that is included in AndroidManifest.XML configuration file. This is required if the application wants to access the external features. For ex, if the application wants to access any feature, it requires permission. It is included within the tags as it is an XML file. Permissions are automatically created for the basic applications at the time when we create the application. If the app uses higher level API or SDK, we must explicitly mention the permissions inside the uses-permissions tag to access the features or components.

CHAPTER 6

INTERPRETATION OF RESULTS

Welcome activity (home screen):

This is the screenshot of our mini-project’s Home screen which will be the first page visible when the application is opened.

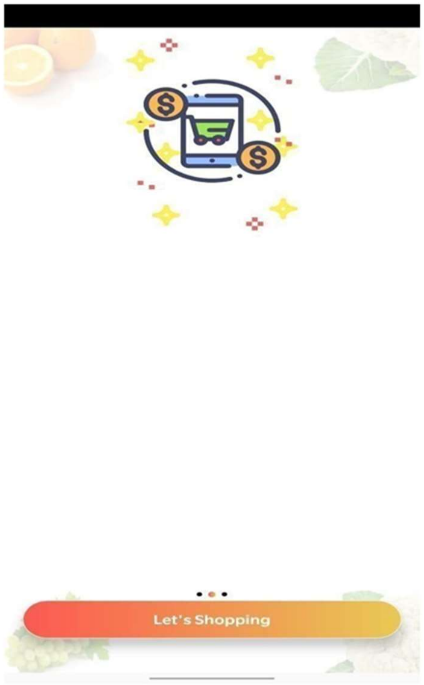
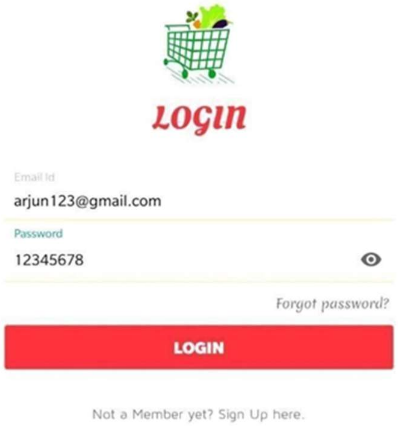


Fig 6.1 Home Screen

The screen after pressing let’s shop page is directed to the Login page where the user of the application can login to book grocery and use the app.

Login Activity:

This is the screenshot project’s login screen where user can login by entering a email id and password validating the credentials to login further are registered to use the app.



Email Id
arjun123@gmail.com

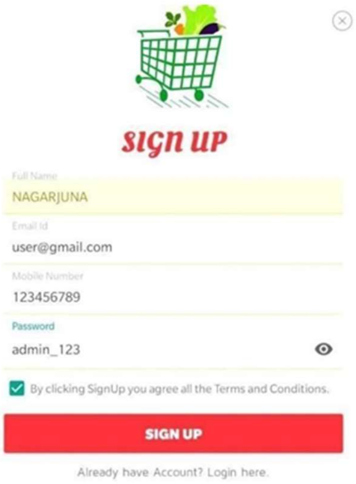
Password
12345678

Fig 6.2: Login Window

The login screen is connected to the Signup Activity where the user can get registered and the Get Started Activity where user can play a video to get themselves aware on how to use the app.

Signup Activity:

This is the project's signup screen where user can get registered by entering a unique username and password validating the credentials to login further using the registered credentials.



Full Name
NAGARJUNA

Email Id
user@gmail.com

Mobile Number
123456789

Password
admin_123

☑ By clicking SignUp you agree all the Terms and Conditions.

Fig 6.3: Signup Window

The signup screen is connected to the Login Activity where the user can login to use the application.

Main Activity:

This is the screenshot of the main activity of the application where all the activities are shown which is the home screen once the app is opened.

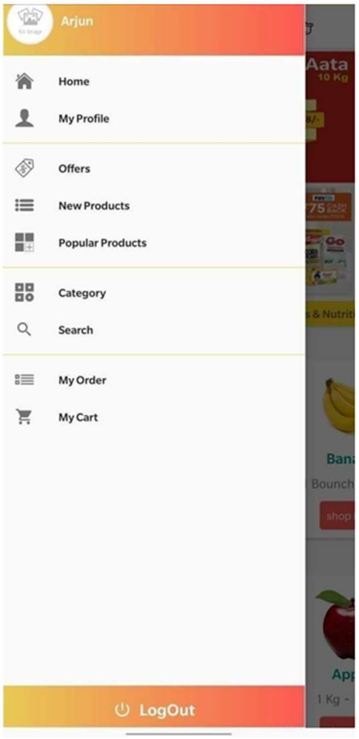
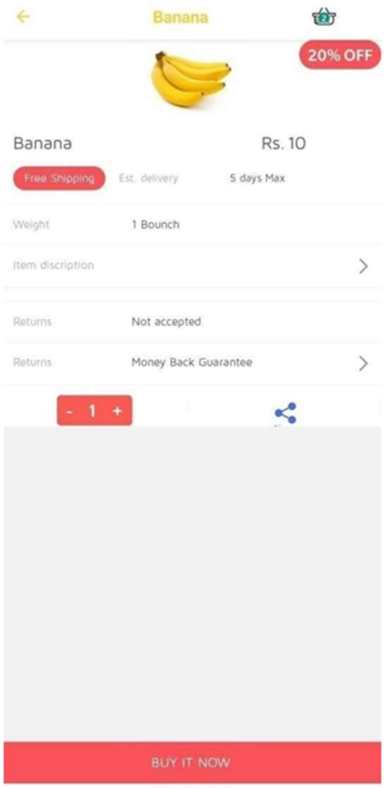


Fig 6.4: Main Screen

Product View Activity:

This cart screen shows the number of products added which can add or delete and prices are shown here where it can be proceeded next for checkout options.



Banana Rs. 10

Est. delivery 5 days Max

Weight 1 Bounch

Item discription

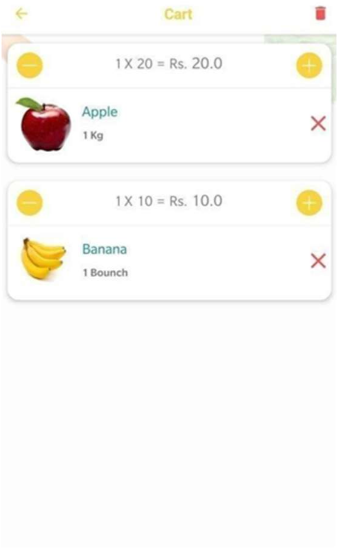
Returns Not accepted

Returns Money Back Guarantee

Fig 6.5: Product View Window

Cart activity:

This cart screen shows the number of products added which can add or delete and prices are shown here where it can be proceeded next for checkout options.



Cart

1 X 20 = Rs. 20.0

Apple

1 Kg

1 X 10 = Rs. 10.0

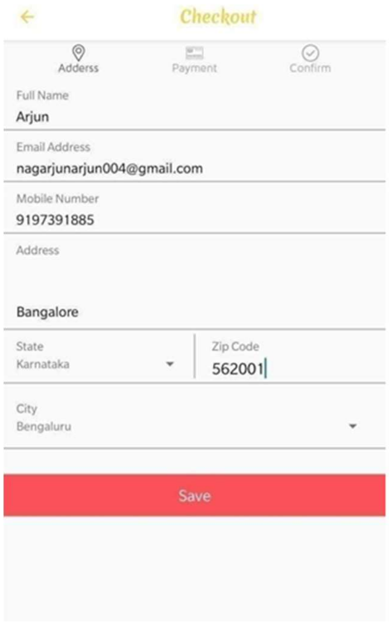
Banana

1 Bounch

Fig 6.6: Cart Window

Address Activity:

After selecting the products in the cart, checking it out address must be uploaded pin code and the shipping address must be updated.



Checkout

Full Name
Arjun

Email Address
nagarjunarjun004@gmail.com

Mobile Number
9197391885

Address
Bangalore

State
Karnataka

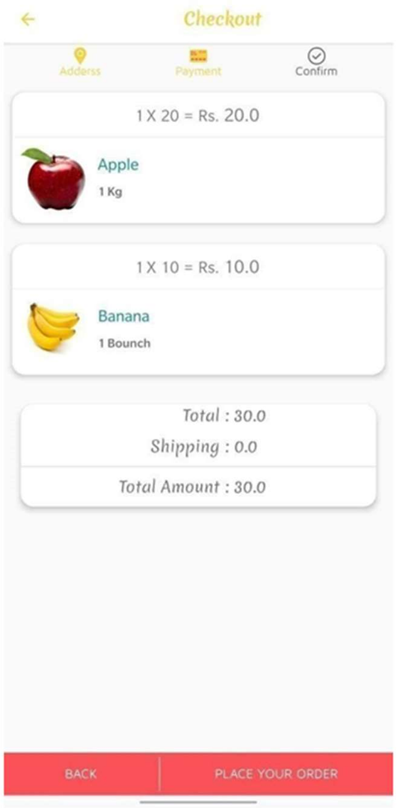
Zip Code
562001

City
Bengaluru

Fig 6.7: Address Activity

Payment Activity:

This screen shows us the payment activity details of the product proceeded for the checkout. It shows two options one is card payment and another is cash on delivery option.



1 X 20 = Rs. 20.0

Apple
1 Kg

1 X 10 = Rs. 10.0

Banana
1 Bounch

Total : 30.0
Shipping : 0.0
Total Amount : 30.0

Fig. 6.8: Payment Window

Success Activity:

After successfully placing the order this page will show the user as “thank you” in form of image which is shown after the order placement.

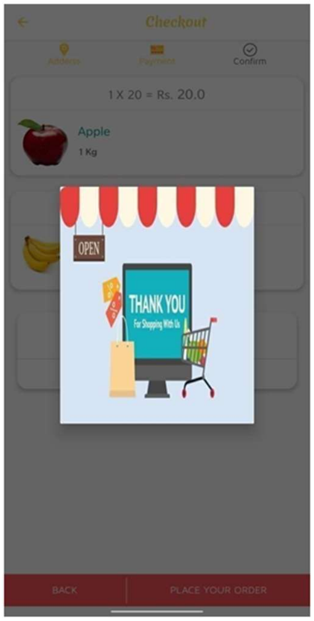


Fig.6.9 Thank you page

CHAPTER 7

CONCLUSION & FUTURE ENHANCEMENTS

7.1 CONCLUSION

Our generation relies mostly on phones to get through the day. Due to this, phones have become more of a personal assistant than a means to just communicate. We ensure hygienic, best quality products and availability of the products to customers, and provide the products with more discount. Henceforth to conclude, we’ve developed a mobile application based on Java using Android Studio. And for the efficient working of the ongoing system we’ve this application to help people to book their grocery in limited time.

7.2 FUTURE ENHANCEMENTS

1. We can use APIs to draw news from web to get displayed in the application.
2. We can even further make it private and secured by enhancing login features.
3. We can allow different users to write comments to news using their account which can be viewed by others.
4. We can make it more space and resource efficient so that this application consumes lesser RAM and ROM.

REFERENCES

1. https://developer.android.com/docs?authuser=1
2. https://www.sqlite.org/docs.html
3. https://en.wikipedia.org/wiki/Android_Studio
4. https://scholarworks.calstate.edu/downloads/np193947n
5. https://www.ijariit.com/manuscripts/v4i2/V4I2-1739.pdf
6. http://www.cs.columbia.edu/~sedwards/classes/2020/4840-spring/reports/tetris.pdf
7. https://projectworlds.in/android-tetris-game-project /
8. https://www.scribd.com/document/147096985/Tetris-Development-for-Android/
9. https://www.youtube.com/watch?v=W9rM8pWOJCQ
10. https://myprojectbazaar.com/product/tetris-game-application-development/