# PRIVACY PROTECTION AND INTRUSION AVOIDENCE FOR CLOUDLET-BASED MEDICAL DATA SHARING

**A dissertation submitted in partial fulfillment of the requirements for the award of degree of**

**BACHELOR OF COMPUTER APPLICATIONS**

of

# University of Mysore



**By**

**Lokesh Rao LK : BC201145**

**Sandeep V : BC201196**

**Bhaskar M : BC201111**

**Anirudh TS : BC201104**

**Under the Guidance of:**

**M N GURUNAGARAJA**

**Assistant Professor, Dept of BCA**

**Mysuru**



**DEPARTMENT OF BACHELOR OF COMPUTER APPLICATIONS**

**BGS FIRST GRADE COLLEGE MYSURU-23**

**2022-2023**

# DECLARATION

We, student of final semester BCA of BGS First Grade College, Mysuru, hereby declare that the dissertation entitled “**AUTOMATED AND TIME SAVING PURCHASE AND CHECKOUT SOLUTIONS USING QR CODEFOR SUPERMARKET**” has been independently carried at ―**BGS First Grade College**‖, Mysuru and submitted in partial fulfillment of the requirement for the award of **Bachelor of Computer Applications** affiliated to the **University of Mysore**, during the academic year 2022 – 2023. Further the matter embodied in the report is an original and bonafide work done by us.

**To our knowledge this dissertation has not been submitted to any other college or university or published at any time prior to this.**

Place: Mysuru **LOKESH RAO LK BC201145**

## Date: SANDEEP V BC201196

**BHASKAR M BC201111**

**ANIRUDH TS BC201104**



**CERTIFICATE**

This is to Certify that **LOKESH RAO LK** bearing ―**BC201145**‖ has completed his final semester project work entitled ― **AUTOMATED AND TIME SAVING PURCHASE AND CHECKOUT SOLUTIONS USING QR CODEFOR SUPERMARKET**‖ as a partial fulfillment for the award of Bachelor of Computer Applications of University of Mysore during the year 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

**Internal Guide:**

**M N GURUNAGARAJA**

Asst. Professor, Dept of

BCA, BGS FGC, Mysuru.

## Mr. Hareesha.C Dr. R.S. Narasegowda

Assistant Professor & HOD Principal,

Dept of BCA, BGS FGC , Mysuru. BGS FGC , Mysuru

## Examiner’s Name & Signature

**1. 2.**



**CERTIFICATE**

This is to Certify that ―**SANDEEP V**‖ bearing ―**BC201196**‖ has completed his final semester project work entitled ― **AUTOMATED AND TIME SAVING PURCHASE AND CHECKOUT SOLUTIONS USING QR CODE FOR SUPERMARKET** ‖ as a partial fulfillment for the award of Bachelor of Computer Applications of University of Mysore during the year 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

**Internal Guide:**

**M N GURUNAGARAJA**

Asst. Professor, Dept of

BCA, BGS FGC, Mysuru.

## Mr. Hareesha.C Dr. R.S. Narasegowda

Assistant Professor & HOD Principal,

Dept of BCA, BGS FGC , Mysuru. BGS FGC , Mysuru

## Examiner’s Name & Signature

**1. 2.**



**CERTIFICATE**

This is to Certify that ―**BHASKAR M**‖ bearing ―**BC201111**‖ has completed his final semester project work entitled ― **AUTOMATED AND TIME SAVING PURCHASE AND CHECKOUT SOLUTIONS USING QR CODE FOR SUPERMARKET**‖ as a partial fulfillment for the award of Bachelor of Computer Applications of University of Mysore during the year 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

**Internal Guide:**

**M N GURUNAGARAJA**

Asst. Professor, Dept of

BCA, BGS FGC, Mysuru.

## Mr. Hareesha.C Dr. R.S. Narasegowda

Assistant Professor & HOD Principal,

Dept of BCA, BGS FGC , Mysuru. BGS FGC , Mysuru

## Examiner’s Name & Signature

**1. 2.**



**CERTIFICATE**

This is to Certify that ―**ANIRUDH TS** ‖ bearing ―**BC201104**‖ has completed his final semester project work entitled ― **AUTOMATED AND TIME SAVING PURCHASE AND CHECKOUT SOLUTIONS USING QR CODE FOR SUPERMARKET**‖ as a partial fulfillment for the award of Bachelor of Computer Applications of University of Mysore during the year 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

**Internal Guide:**

**M N GURUNAGARAJA**

Asst. Professor, Dept of

BCA, BGS FGC, Mysuru.

## Mr. Hareesha.C Dr. R.S. Narasegowda

Assistant Professor & HOD Principal,

Dept of BCA, BGS FGC , Mysuru. BGS FGC , Mysuru

## Examiner’s Name & Signature

**1. 2.**

# ABSTRACT

Smart and hassle free checkout in supermarket is a mobile application. The customer can scan the product and pay using a QR code. This makes the work simple for customers so that they need not wait for long hours in a queue.One advantage of this mobile application is that there is no need for lengthy queue and in this busy schedule of everyone this mobile application makes the work easy and it can save time

# ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

We consider it as a privilege to express our sincere gratitude and respect to all those who guided me during the completion of the project.

We would like to extend our sincere **thanks Dr.R.S.Narasegowda, Principal, BGS FGC, Kuvempunagar, Mysuru** for providing the facilities to complete this project.

We express our deep-felt sincere gratitude to **Mr.Hareesha.C, HOD Department of BCA,** for his encouragement and immense help.

We thank our internal guide ―**MN GURUNAGARAJA SIR**‖, Assistant Professor, Dept. of BCA, for his inspiration, encouragement, timely guidance and valid suggestion throughout our project.

Finally, we wish to thank all the staff members and non-teaching staff of our department for their help and guidance.

Above all, my sincere gratitude to our parents and friends, also those who have directly or indirectly helped me to complete this project successfully.

**LOKESH RAO L K BC201172**

**SANDEEP V BC201196**

**BHASKAR M BC201111**

**ANIRUDH TS BC201104**

**TABLE OF CONTENT**

**Chapter 1: Introduction**

1:1 Objectiveof Project

# Chapter 2: Literature Survey

2:1 Literature Review

2:2 Problem Statement

# Chapter 3: System Analysis

3:1 Existing System

3:2 Proposed System

3:3 Functional Requirement

3:4 Non-Functional Requirements

3:5 System Requirements

3:6 Feasibility Study

# Chapter 4: System Design

4:1 System Architecture

4:2 Data Flow Diagram

4:3 Flow Chart

4:4 UML Diagrams

4:4:1 Goals

4:5 Use Case Diagram

4:6 Class Diagram

4:7 Sequence Diagram

4:8 Database Table

**Chapter 5: Implementation**

# Chapter 6: Snapshots

# Chapter 7: Software Testing

7:1 Black-Box Testing

7:2 White-Box Testing

7:3 Grey-Box Testing

7:4 A Comparison of Testing Methods

7:5 Levels of Testing

7:6 Unit Testing

7:7 Integration Testing

7:8 System Testing

7:9 Regression Testing

7:10 Acceptance Testing

**INTRODUCTION**

**Basic Introduction of Project**

Two-dimensional matrix codes also known as the QR Code (Quick Response Code) , on other side, is advancement over the Barcode Systems and is used for various commercial purposes. The QR Code stores information in two dimensions that can be scanned and read by an application known as the QR-Code Reader. The QR code can be used to store various form of information like images, videos, price and all. The objective of this project is to propose a real time capturing system for consumer supplies using Quick Response (QR) code in an Android smart phone. Using multiplexing and demultiplexing process encode and decode the information from single QR code with special symbols and split the data back to their QR Code pattern where these QR Code pattern can be read by Android smart phones.The customer will have to put the product in front of QR Code scanner; it scans the Code and saves the information of the product in the cart.

**Objectives & Scope**

The inspiration for doing this project was essentially an enthusiasm for undertaking a difficult venture in a fascinating territory of research. This project has given me the opportunity to learn and develop a android app from scratch. This app can be used by admin and the customer, floor manager to add products available in that particular floor.

**Project Category**

Mobile APPLICATION – It is a Mob App developed for Android Mobiles using Java ,XML Language and SQLite database as the data source.

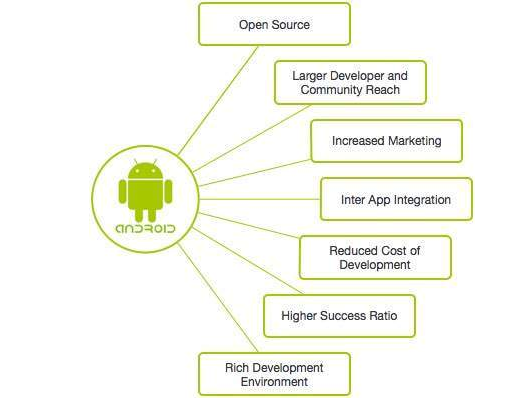
**Tools and Technologies Used**

**ANDROID OPERATING SYSTEM**

When Android first arrived in 2008, it was almost seen as a poor relation to the much more stylish iOS on Apple iPhone. But, quite quickly, through diverse handset offers that struck a chord with both the practical price-conscious as well as the fashion-conscious and tech hungry consumers, Android user numbers exploded. Now, after eight major releases, the annual sales of Android devices is increasing almost every year.

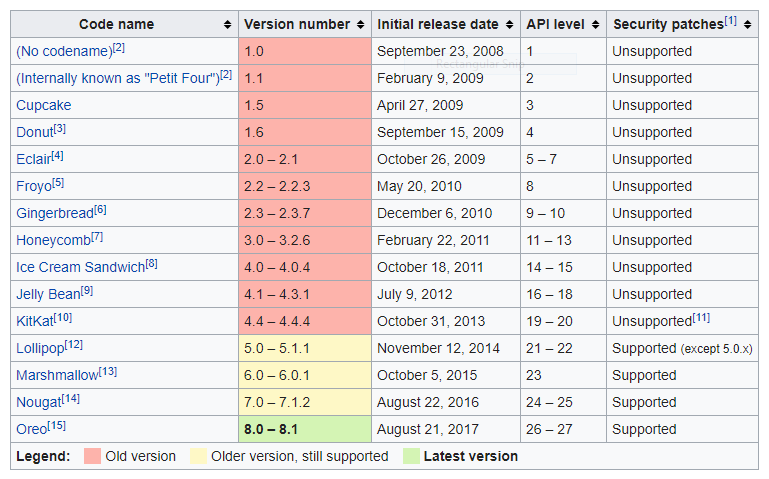
The reason developing for Android gives such a buzz is the nature of the devices. They are deeply personal. We can create apps that actually interact with people's lives. We can educate, entertain, organize and so on. Everyone uses them, from infants to seniors.

Android is an open source and Linux-based Operating Systemfor mobile devices such as Smartphone and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need to develop only for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007, whereas the first commercial version, Android 1.0, was released in September 2008. The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2



**Android Logo**

**Android Versions**

****

**2.1.3. Architecture**

Android operating system is a stack of software components as shown below in the architecture diagram.



* **System Apps** - The Android Open Source Project contains several default applications like the Browser, Camera, Gallery, Music, Phone and more.
* **Java API framework -**. The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications
* **Android runtime** - This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called **Dalvik Virtual** **Machine** which is a kind of Java Virtual Machine specially designed and optimized for Android. The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language
* **Libraries** – On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.
* **Hardware Abstraction Layer (HAL)** - The hardware abstraction layer (HAL) provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware component, such as the camera or bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component
* **Linux kernel** - Communication layer for the underlying hardware. A kernel is the first layer of software that interacts with the device hardware. The kernel is responsible for providing basic architectural model for process scheduling, resource handling, memory management, networking and isolation etc. Android uses the Linux kernel under the hood. Because Linux is open-source, Google’s Android developers could modify the Linux kernel to fit their needs. Linux gives the Android developers a pre-built, already maintained operating system kernel to start with so they don’t have to write their own kernel.

**Android Applications**

An Android app is a software application running on the Android OS. Most Android apps are uploaded and published on the Google Play Store, an online store dedicated to these applications. The Play Store features both free and priced apps. The extension of Android application is .apk APK stands for Application package file. APK files contain the code files, resource files and the AndroidManifest.xml file. AndroidManifest.xml contains info about the app.

**Android Activity Life Cycle**

The [Activity](https://developer.android.com/reference/android/app/Activity.html) class is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms in which apps are launched with a main() method, the Android system initiates code in an [Activity](https://developer.android.com/reference/android/app/Activity.html) instance by invoking specific callback methods that correspond to specific stages of its lifecycle.

Activities in the system are managed as an activity stack. When a new activity is started, it is placed on the top of the stack and becomes the running activity -- the previous activity always remains below it in the stack, and will not come to the foreground again until the new activity exits.

**An activity has essentially four states:**

1. If an activity is in the foreground of the screen (at the top of the stack), it is active or running.
2. If an activity has lost focus but is still visible (that is, a new non-full-sized or transparent activity has focus on top of your activity), it is paused. A paused activity is completely alive (it maintains all state and member information and remains attached to the window manager), but can be killed by the system in extreme low memory situations.
3. If an activity is completely obscured by another activity, it is stopped. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere.
4. If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish, or simply killing its process. When it is displayed again to the user, it must be completely restarted and restored to its previous state.

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.

The square rectangles represent callback methods you can implement to perform operations when the Activity moves between states. The colored ovals are major states the Activity can be in.



There are three key loops you may be interested in monitoring within your activity:

The entire lifetime of an activity happens between the first call to onCreate(Bundle) through to a single final call to onDestroy(). An activity will do all setup of "global" state in onCreate(), and release all remaining resources in onDestroy().

The visible lifetime of an activity happens between a call to onStart() until a corresponding call to onStop(). During this time the user can see the activity on-screen, though it may not be in the foreground and interacting with the user. Between these two methods you can maintain resources that are needed to show the activity to the user.

The foreground lifetime of an activity happens between a call to onResume() until a corresponding call to onPause(). During this time the activity is in front of all other activities and interacting with the user. An activity can frequently go between the resumed and paused states.

**Android Studio**

****

Android Studio is the official IDE (Integrated Development Environment) for the Android platform, developed by Google and used to make the majority of the apps.

Android Studio was first announced at a Google I/O conference in 2013 and was released to the general public in 2014 after various beta versions. The current stable version is 3.0 released in October 2017.

As an IDE, Android Studio’s job is to provide the interface for developers to create their apps and to handle much of the complicated file-management behind the scenes. Android Studio is simply where developers will write, edit and save their projects and the files that comprise said projects. At the same time, Android Studio gives access to the Android SDK or ‘Software Development Kit’. Android Studio also enables to run the code, either through an emulator or through a piece of hardware connected to the machine.

**SQLite Database Service**

SQLite is a relational database management system, similar to Oracle, MySQL, PostgreSQL and SQL Server. It implements most of the SQL standard, but unlike the four database engines mentioned above, it is not a client-server database engine. Rather, it is embedded into the end program. What this means is that you can bundle a SQLite database with your application, and get access to all the power of a relational database within your application. SQLite is native to both Android and iOS, and every app can create and use an SQLite database if they so desire. In fact, in Android, device contacts, and media are stored and referenced using SQLite databases.

**JAVA**

Java is a language that allows us to write code once that can be used over and over again. This is very useful because it saves us time and allows us to use other people's code to perform tasks we might otherwise not have the time or knowledge to write for ourselves. Most of the time, we do not even need to see this code or even know how it does its work!

Java is a programming language that has been around a lot longer than Android. It is an object-oriented language.

The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language
* Java is cohesive and consistent
* Except for those constraint imposed by the Internet environment.
* Java is a known language, developers know it and don't have to learn it
* It is harder to shoot yourself with Java than with C/C++ code since it has no pointer arithmetic
* It runs in a VM, so no need to recompile it for every phone out there and easy to secure
* There are a large number of development tools for Java
* Several mobile phones already used Java ME, so Java was known in the industry
* There are a large number of developers already proficient in Java.
* Java has huge open source support, with many libraries and tools available to make developers life easier.
* Java protects you from many of the problems inherent in native code, like memory leaks, bad pointer usage, etc.
* Java allows them to create sandbox applications, and create a better security model so that one bad App can't take down your entire OS.

**2.2.1 Java Architecture**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

**Compilation of code** - When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is created for overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

**XML**

A markup language is a mechanism to identify structures in a document. XML (Extensible Markup language) is a markup language for documents containing structured information. As a meta-language, XML is about to describe data and its structure; while HTML is about to display the data.

XML is not to replace HTML, but compliment it by allowing writers to create and format their own document markups. For HTML, The tags used to mark up HTML documents and the structure of HTML documents are predefined. For XML, there is no predefined tag set, so there is no any preconceived semantics. Writers are provided a facility to define tags and the structural relationships. That is the extensible feature of XML.

To be valid, XML documents should be well-formed. That means XML documents follow some strict rules. For example, every opening tag in XML documents must have a matching closing tag. While in HTML, pairing is optional for some of the tags, and only the opening tag is required.

**Some rules for composing XML documents**:

**1. Closing Tags**

An element in XML must have a closing tag.

For example:

<firstname>James</firstname>

**2. Overlapping Elements**

An element must be nested properly by closing all child elements before closing the parent elements.

For example:

<student>

<firstname>James</firstname>

<lastname>Smith</lastname>

</student>

It would be incorrectly nested if the student element was closed before forename element.

**3. Single Root Element**

An XML document only can have a single root element. All other elements should be within this root element.

For example:

<mec>

<student>

<firstname>James</firstname>

<lastname>Smith</lastname>

</student>

</mec>

As above, mec is the root element, and student is its descendant.

**4. Case Sensitive**

XML is case sensitive, and care should be taken to ensure that Opening and closing tags are in the same case. An element name of "MEC" is not the same as "mec".

**5. Quoted Attribute Values**

In XML, attributes must be specified a value, and the values must be in quotes, regardless of the data type. Single and double quotes are both acceptable.

For example:

Correct: <input type="checkbox" checked="checked"/>

Wrong: <input type="checkbox" checked>

**6. Naming Conventions**

In XML, since there is no any pre-defined tag, there are no reserved words needed to be avoided when naming elements. The following are some simple rules when naming an element.

a. Elements must start with a character or an underscore but not a number or punctuation. After the first character, numbers, hyphens and periods are allowed.

b. Names must not contain spaces.

c. A colon is reserved for namespaces

**SYSTEM ANALYSIS**

**Preliminary analysis & information gathering**

The existing system is manual in nature. The customer should stand in the queue manually for billing. It is more time consuming. The customer should stand in the queue to pay bill even if she or he purchases only one or two products. Existing system is not computerised it’s manually the customer have to stand in the queue manually.Existing system is time consuming some time the customer has to spend a lot of time for billing.

**PROPOSED SYSTEM**

The aim of the proposed system is to get over the problems or disadvantages in existing manual system and to develop an Android app with improved version.This app can be user friendly and avoid many hardships faced by the existing system. The proposed system can overcome the disadvantages of existing system.

**Advantages of proposed system:**

* No formal knowledge is required by the user to use the app hence it’s user friendly.
* This app is more secured
* It reduces the queue and save time.
* It’s helpful for customer.
* Instead of standing in the Queue the customer can scan, pay and checkout it avoids even the rush in the supermarket.
* So by using this app work load decreases for the workers in supermarket.
* And even the customer can save time and avoid wastage time, Now the customer can scan pay and checkout in just few clicks in supermarket.

**Input/Outputs**

**INPUT**

The input is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

1. Input is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**OUTPUT**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.

**Feasibility Study**

Feasibility study is a procedure that identifies, describes and evaluates candidate android application and selects the best application feature for the job. An estimate is made whether the identified users need may be satisfied using the current networking facilities and hardware technologies. The study will decide whether the proposed android application will be cost effective from a business point of view and if it can be developed using the given existing budgetary constraints.

The key considerations involved in the feasibility analysis of our proposed application are the following:

* Operational Feasibility
* Technical Feasibility
* Economic Feasibility
* Schedule Feasibility

**Operational Feasibility**

Operational feasibility is necessary as it ensures that the project developed is a successful one.

The operational feasibility of this project is high since it is user friendly and the application provides all the expected outputs to the user.

**Technical Feasibility**

Technical feasibility analysis makes a comparison between the level of technology available and that is needed for the development of the project. The level of technology consists of the factors like software tools, and platform developed and so on. Since, the resources for the development of the project is available, the project is technically feasible

**Economic Feasibility**

This is the most important part of the project because the terms and conditions for implementing the project have to be economically feasible. The risk of finance does not exist as the existing hardware that is smart phone is sufficient and the software is free of cost. So, it is believed that the system is economically feasible

**Schedule Feasibility**

Schedule feasibility is defined as the likelihood of a project being completed within its desired timeframe. Since this project has a high likelihood of completion by the desired due date, schedule feasibility is considered to be high.

**Software Requirement Specification**

**SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| **Operating System** | Windows 7 or higher |
| **Languages Used** | Java, XML |
| **Development Environment (IDE)** | Android Studio |
| **Database** | SQLite |

**HARDWARE REQUIREMENTS**

|  |  |
| --- | --- |
| **Hard disk** | Minimum of 40GB |
| **RAM** | 4 GB or higher |
| **Device** | Android mobile |

**Software Engineering Model Used**

The Waterfall Model was first Process Model to be introduced. It is very simple to understand and use. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Waterfall model is the earliest SDLC approach that was used for software development.

In “The Waterfall” approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. As the Waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a Linear-Sequential Life Cycle Model.

**Sequential Phases in Waterfall Model**

* Requirements: The first phase involves understanding what need to be design and what is its function, purpose etc. Here, the specifications of the input and output or the final product are studied and marked.
* System Design: The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and helps in defining overall system architecture. The software code to be written in the next stage is created now.
* Implementation: With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
* Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaw or errors. Testing is done so that the client does not face any problem during the installation of the software.
* Deployment of System: Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
* Maintenance: This step occurs after installation and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Client is provided with regular maintenance and support for the developed software.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name “Waterfall Model “.

**Project Scheduling**



**SYSTEM DESIGN**

**Modules**

The system comprises of 3 major modules:

1.**Admin**

2.**Floor Manager**

**3. Staff**

**4.Customer**

**Admin:** This module contains all the information about the admin. Once the admin is logged in to the app he or she can approve the registered customer, and even delete the registered customer.

**Login:**

Allow the admin to login to the app with valid credentials.

**Approve customer/Floor Manager:**

Allow the admin to approve the customer and verify their details.

**Delete customer/Floor Manager:**

Allow the admin to delete the customer and floor manager.

**View customers:**

Allow the admin to see the list of the approved customers.

**Logout:**

Allow the admin to logout from the app

**Floor manager:** This module consists all the information about the floor manager. Once the floor manager is logged in to the app he or she can add Staff details and view feedback and complaints.

**Register:**

Allow the floor manager to register with the app with all his details like Name, Username, password, mobile number etc.

**Login:**

Allow the floor manager to login to the app with valid credentials once he is approved by the admin.

**Add Staff:**

Floor Manager can add details of staff along with login credentials.

**Delete Staff:**

Floor Manager can delete a staff.

**View feedback**:

Allow floor manager to view feedback given by the customers.

**View complaints**:

Allow floor manager to view complaints posted by the customers.

**Logout:**

Allow floor manager to logout of the app.

**Staff:**

**Login:**

The staff can login to the app with the credentials provided by Floor Manager.

**Add product details:**

Staff can to add the information about products.

**Generate QR code:**

Staff can generate QR code after adding product information.

**View Profile:**

Staff can view profile details.

**Edit Profile:**

Staff can edit profile details.

**Logout:**

Allow staff to logout of the app.

**Customer:** This module contains all the information about the customers. Once the customers are logged in he or she can scan, pay and use the app.

**Register:**

Allow the customer to register with the app with all his details like Name, Username, Password, Mail, Mobile number.

**Login:**

Allow the customer to login to the app with valid credentials. Once he is approved by the admin.

**Scan the products:**

Allow the customer to scan the product.

**Pay:**

Allow the customer to pay after he or she scan the product.

**Give Feedback**:

Allow the customer to give feedback.

**Post Complaint:**

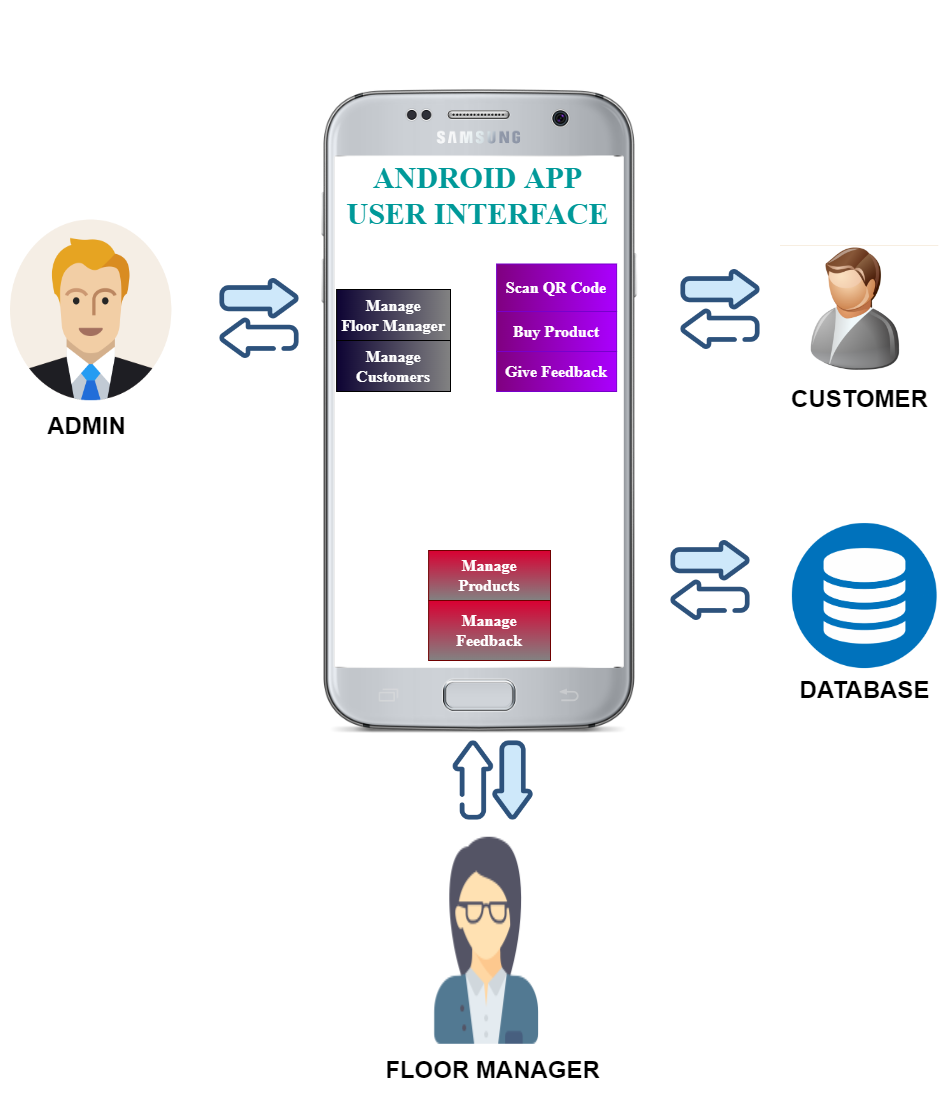
Customer can post complaint about any issues that they face.

**Logout:**

Allow the customer to logout of the app.

**System Architecture**

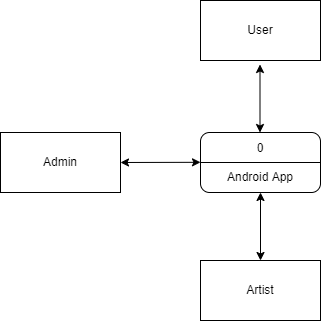
A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



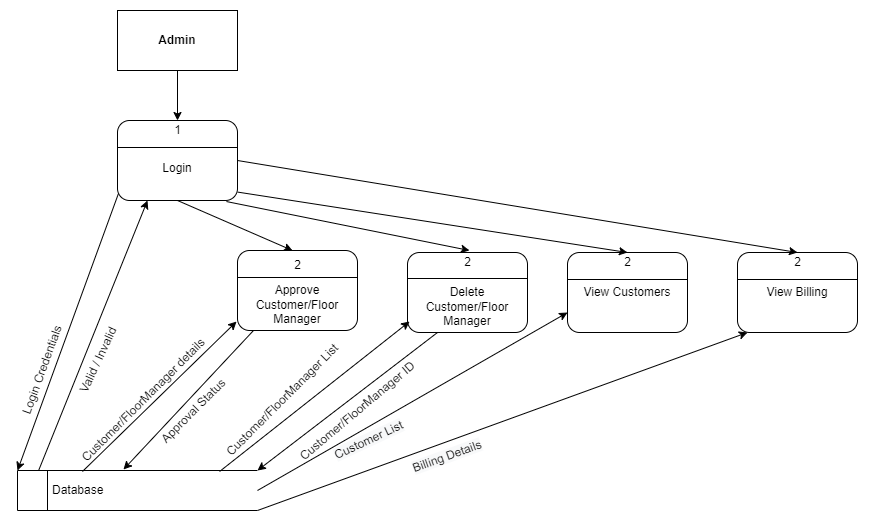
**Architecture Design**

**Dataflow Diagram**

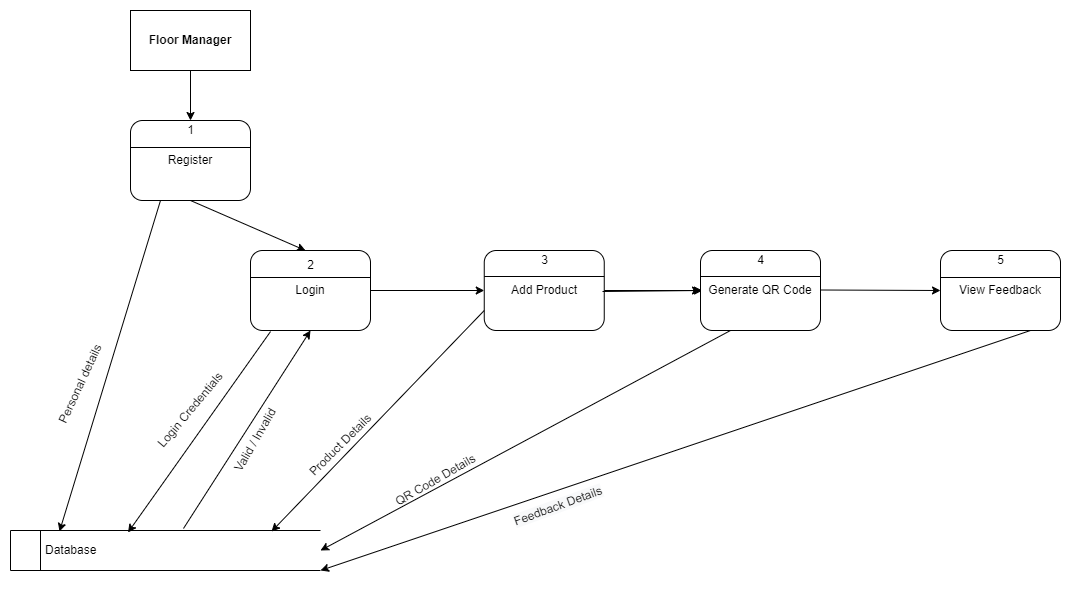
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO.



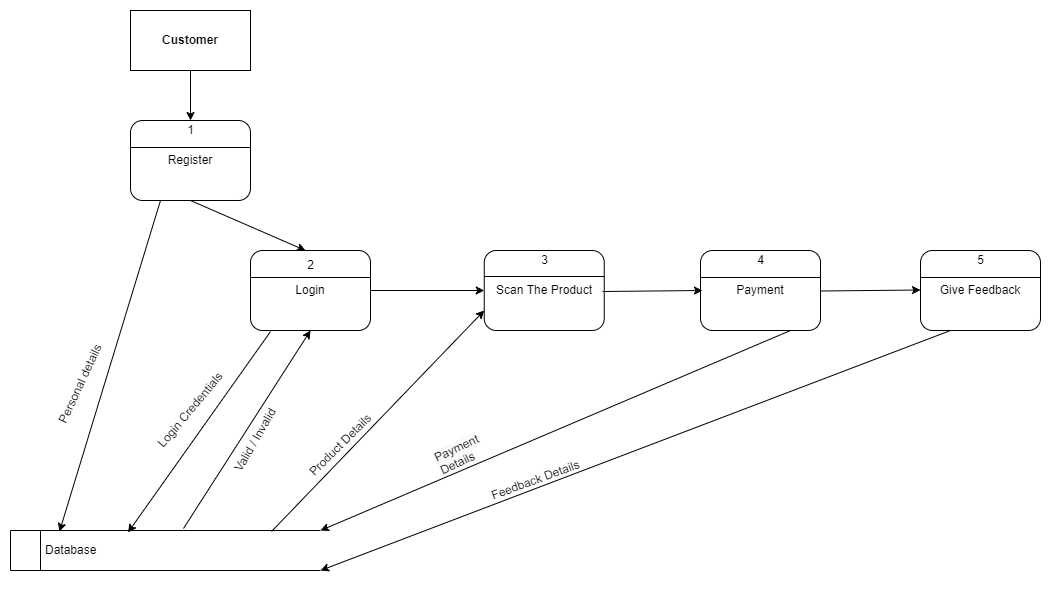
**Level 0 DFD**



**Level 1 DFD Admin**



**Level 1 DFD Floor Manager**

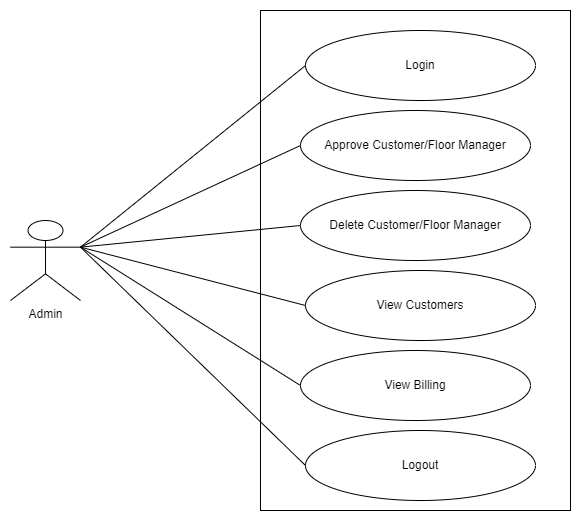


**Use case Diagram**

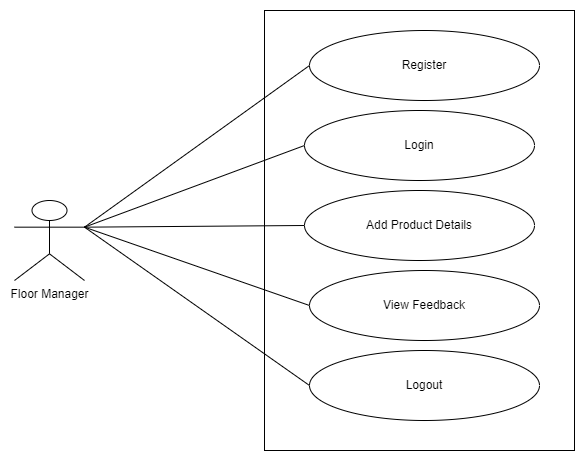
A use case diagram is a graphical representation of the user’s interaction with the system. It can portray the different types of users of a system and the various ways they interact with the system. Use cases are diagrammed to be easily understood, no matter who is looking at the diagram.

A use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. An effective use case diagram helps to represent:

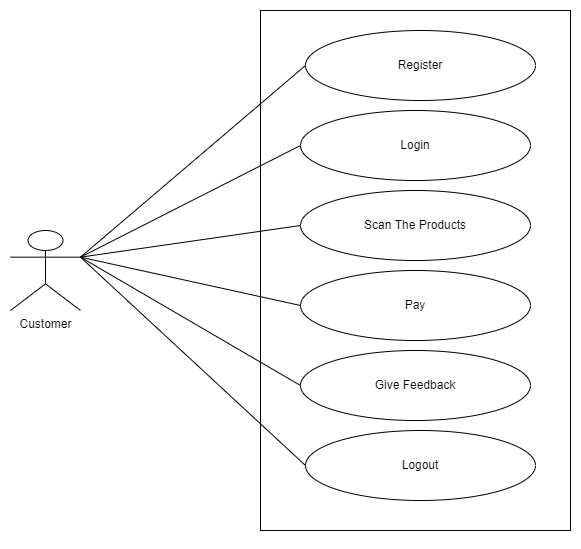
1. Scenarios in which your system or application interacts with people, organizations, or external systems
2. Goals that it helps those entities (known as actors) achieve
3. The scope of your system



**Usecase – Admin**



**Usecase –Floor Manager**



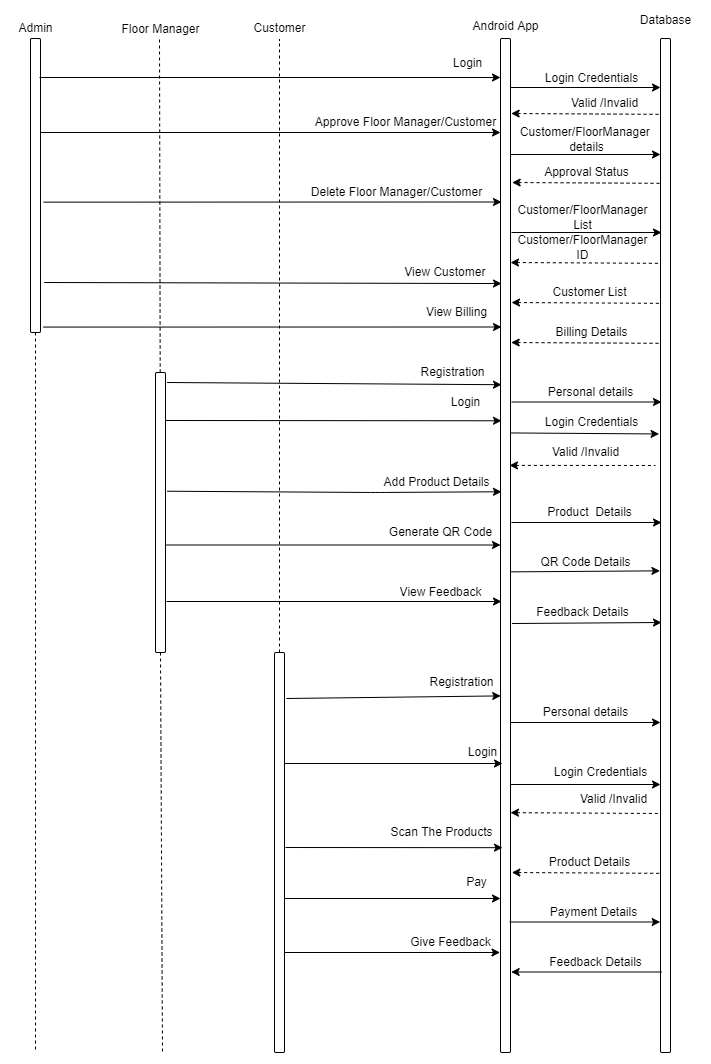
**Usecase - Customer**

**Sequence Diagram**

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order.

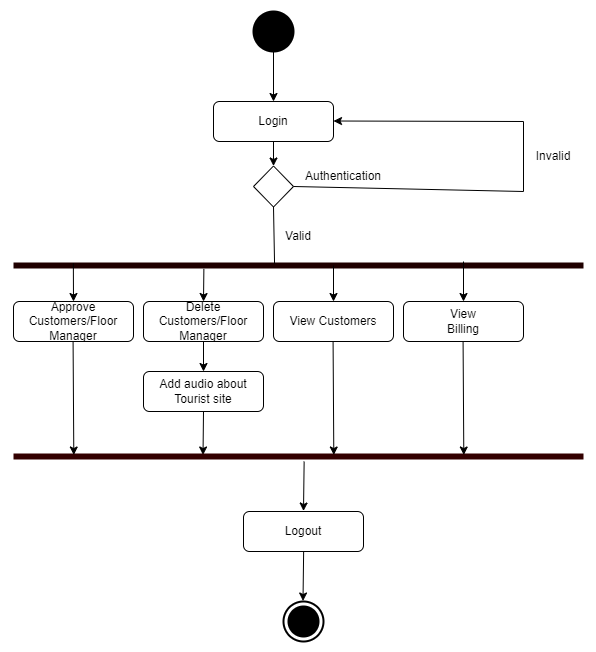
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

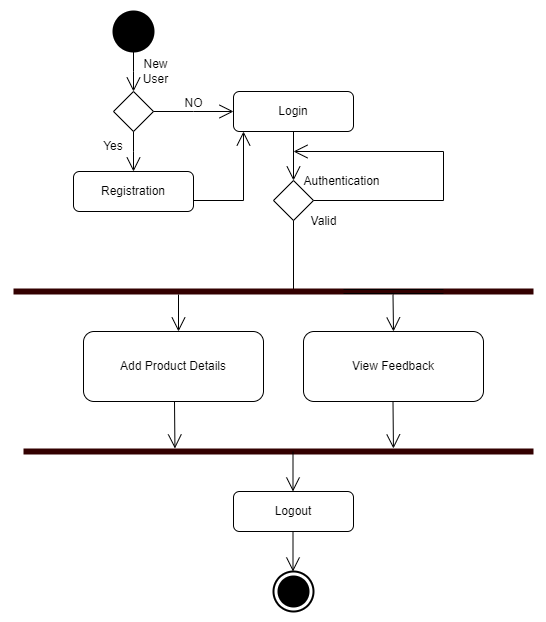


**Activity Diagram**

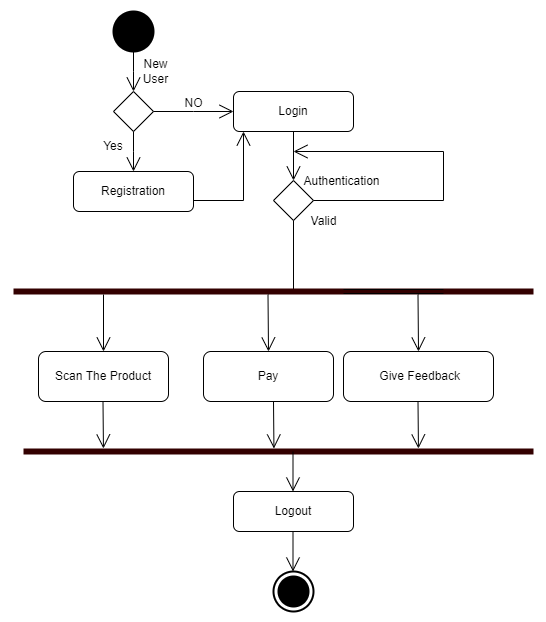
An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



**Activity Diagram - Admin**



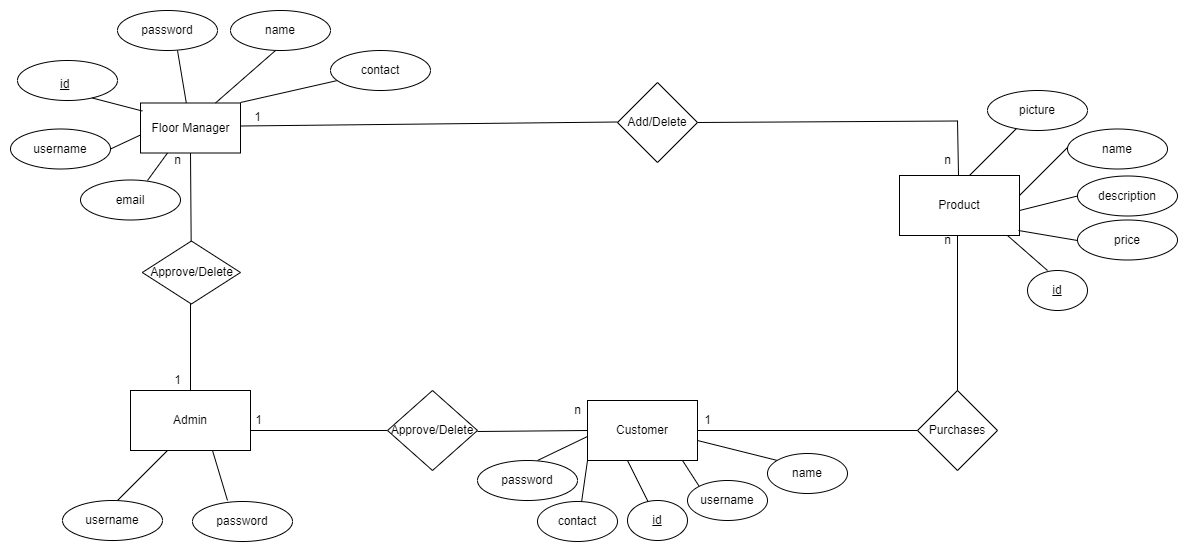
**Activity Diagram – Floor Manager**



**Activity Diagram - Customer**

**ER Diagram**

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.



**User Interface Designs Screenshots**

**CODING & TESTING**

**CODE SNIPPETS**

**Testing**

Software testing is a process of executing a program or application with the intent of finding the software bugs.

Software testing is a critical element of software quality assurance and represents the ultimate process to ensure the correctness of the product. The quality product always enhances the customer confidence in using the product thereby increasing the business economics. In other words, a good quality product means zero defects, which is derived from a better quality process in testing.

Testing the product means adding value to it by raising the quality or reliability of the product. Raising the reliability of the product means finding and removing errors. Hence one should not test a product to show that it works; rather, one should start with the assumption that the program contains errors and then test the program to find as many of the errors as possible.

The main objective of testing is to find defects in requirements, design, documentation, and code as early as possible. The test process should be such that the software product that will be delivered to the customer is defect less. All Tests should be traceable to customer requirements. Test cases must be written for invalid and unexpected, as well as for valid and expected input conditions.

A necessary part of a test case is a definition of the expected output or result. A good test case is one that has high probability of detecting an as-yet undiscovered error.

**Manual Testing**

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

**Different stages for manual testing:**

**Unit Testing**

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

**Tests that are performed during the unit testing in the app are explained below:**

**1) Module Interface test:** In module interface test, it is checked whether the information is properly flowing in to the program unit (or module) and properly happen out of it or not.

E.g. The user registration details should be available from the layout to the corresponding controller and from the controller it should flow to the model.

**2) Boundary conditions:** It is observed that much software often fails at boundary related conditions. That’s why boundary related conditions are always tested to make safe that the program is properly working at its boundary condition’s.

E.g. In case of if...else if... else... construct all the conditions are checked in the app.In case of loops, it is checked to see that the loops are not infinite and terminate once the condition becomes false.

**3) Error handling paths:** These are tested to review if errors are handled properly.

E.g. 1. Validation during login (Checking for wrong credentials).

2. Validation of password during registration (Password should adhere to password policy - minimum 8 characters with a number and special character).

3. In case the user tries to send SMS without configuring contacts, he should be redirected to Add Contacts screen.

4. It should be possible to send SMS and play alarm sound even if the user has not configured these settings in Preferences. The app should have default SMS template message and Alarm Sound.

5. In case the user tries to contact emergency contacts without configuring contacts, he should be redirected to Add Contacts screen.

**Integration Testing**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing. In this project we have followed the Bottom-up integration method.

Here testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

Once all the different modules were integrated in the app, the app was tested for the following:

1. Transition from one screen to another
2. Data from the layouts is getting saved properly in the database.
3. Data is retrieved properly from the database and displayed in the layouts.
4. SMS is sent properly to configured contacts during Send SMS actions.
5. Menu items are working properly.

**System Testing**

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards.

The app was installed on an Android mobile and all the features were tested rigorously for all possible inputs. Different test cases were executed to see if the app behaved as executed and there were no crashes and unexpected behavior.

**TEST CASES**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Description** | **Test Steps** | | | | **Test Status P/F** |
| **Steps** | **I/P Given** | **Expected O/P** | **Actual O/P** |
| TC01 | Admin Login | To verify that the admin has entered valid username and password | Login with valid admin name and password | Valid user name and password | Login successful | Login successful | Pass |
| Admin Login | To verify that the admin has entered valid username and password | Login with invalid admin name and password | Invalid admin name and password | Login unsuccessful | Login unsuccessful | Pass |
| TC02 | Approve customers | Admin will approve customers | Admin will check valid details of customer then give approval | Valid customers | Approved successfully | Approved successfully | Pass |
| TC03 | Add floor manager | Admin will add floor manager | Admin will check valid details of floor manager then he will add | Valid floor manager | Added successfully | Added successfully | Pass |
| TC04 | View customers | Admin can view customers Details | Admin can view the list of customers Details | List of **customers Details** | Viewed Successfully | Viewed Successfully | Pass |
| TC05 | View floor manager | Admin can view floor manager Details | Admin can view the floor manager Details | List of floor manager Details | Viewed Successfully | Viewed Successfully | Pass |
| TC06 | Delete customers | Admin can Delete customers | Admin can Delete customers | Delete customers Details | Deleted successfully | Deleted successfully | Pass |
| TC07 | Add Product | Floor Manager can add product details | Select Add Product option | Add Product option is selected | Product details are added successfully | Product details are added successfully | Pass |
| TC08 | Delete Product | Floor Manager can delete an added product | Select Delete Product option | Delete Product option is selected | Product is deleted successfully | Product is deleted successfully | Pass |
| TC09 | View Feedback | Floor Manager can view Feedback Details | Floor Manager can view the list of Feedback Details | List of Feedback Details | Viewed Successfully | Viewed Successfully | Pass |
| TC10 | Scan Product | Customer can scan product QR code to view its details | Select Scan Product option | After Scan Product option is selected, customer has to click on a product and then scan its corresponding QR code | Product details displayed successfully | Product details displayed successfully | Pass |
| TC11 | View Cart | Customer can view cart to see the list of products that he is interested in buying | Select View Cart option | View Cart option is selected | Cart products viewed successfully | Cart products viewed successfully | Pass |

**IMPLEMENTATION & MAINTENANCE**

The goals of the implementation phase is to translate the design of the system produce during the phase ,into coded form in a given programming language, which can then be executed by a computer performing the computation specified by the design the coding phase affects both testing and maintenance profoundly. Well written code can reduce the testing and maintains cost.

A crucial phase in the system lifecycle is the successful implementation of the system design. Implementation simply means converting the system designs into operation. Implementation is the process of bringing the developed system into operational use and providing it to the user.

This stage is considered to be most crucial stage in the development of a successful system since a new system is developed and the users are get information in effective manner

Implementation is a stage in which the design is converted into working system that is it is the stage of the project where theoretical design is turned into a working system .The implementation involves careful planning, investing of the current system and its constraint on implementation, design of methods to achieve the changeover.

The Project is implemented in different phases as follows

* First phase includes table design for database module.
* Second phase includes coding for modules.
* Third phase includes the integration of modules.
* Fourth phase includes connection establishment between the front end and back end.
* Fifth phase includes error handling and message generator.

The coding was done with the following characteristics in mind

* Code efficiency
* Memory efficiency
* Response time
* Security
* Maintainability
* Efficient and consistent logic

**CONSLUSION**

The proposed project is feasible as it is built on technology that already exists. We have improved it by eliminating the long waiting time of queue at the time of billing, secured it by locking after completion of shopping and made the app available to all customers. Smart and hassle free checkout in supermarket is reliable and easy to use app. This app helps customer to save time in his busy schedule. This is the app which can be used by the customers and admin and floor manager which is user friendly for everyone. The development of this project helps everyone.

**FUTURE ENHANCEMNET**

To improve the application we plan to implement the following enhancements:

1. Additional feature to store favorite items/ recently added items so that user can quickly add frequently bought items
2. Membership Module: admin can add membership plans and user can buy membership to get same day delivery, instant delivery, 2x, 3x rewards and free delivery.

**REFERENCES**

https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

<https://developer.android.com/reference/android/telephony/SmsManager.html>

<https://developer.android.com>

https://www.materialpalette.com/

https://developer.android.com/guide/topics/ui/declaring-layout

https://developer.android.com/guide/topics/ui/menus

https://www.w3schools.com/xml/

https://www.javatpoint.com/java-tutorial

https://developer.android.com/training/basics/firstapp