A report in a soft cover binding and in 3 copies (in English, preferred): It covers the problem to be solved, the discussion of the design choices, a set of arguments on the final design choice, a list of solved and open issues, a table of content and figures (including tables), a valid list of biblio- graphic references, and optional appendices as required. A critical consideration of the task, the assignment, and the result will conclude the report. The official acknowledgement section is man- datory, a personal one optional, however recommended, as usually a number of people took part in the process of finalizing the thesis.

Summary :

**Table of content :**

**Summary / Abstract** :introduction, aims and goals, results, and further work.

1. **Introduction**

1.1 Motivation

1.2 Description of work

1.3 Thesis Outline

1. **Related Work**
2. **Introduction to Android and its Architecture**
   1. Why android as OS
   2. Architecture

1. **Design and Implementation**

4.1Design Choices : Discussion and final design

4.2 Implementation

1. **Security and Scalability**

5.1Security

5.2 Scalability

1. **Summary and Conclusion**

**Bibliography**

**Chapter 1 : Introduction**

1. **Motivation**

Demand for transparency is increasing, surprisingly we know minute about most of the products we use in our day to day life. Products travel frequently through huge network of stakeholders before reaching the end consumer which includes retailers, distributors, transporters, storage facilities, and suppliers that participate in design, production, delivery, and sales etc., still in almost all the cases these journeys remain an unobserved attribute of our possession. Nowadays, the consumers as well the government's demands for more transparency from brands, manufacturers, and producers throughout the supply chain is increasing. Governments tend to provide more transparency in goods and their manufacturing details. ‘The 2016 Food Revolution Study, which surveyed 1,522 consumers to discover how they make food choices, shop and what they expect from brands in terms of product information, reveals that brands that meet customer expectations for product information and deliver that information instantly develop a new dynamic of convenience, trust, and long-term value’{1}. This leads to an increase demand to producers in order to provide complete information about the product and its origin. Retailers who provide more information about the product may get good chances to increase their long term with the evolving trend. In the past decades many supply chain management systems has been developed and used. For human beings the food and its life cycle has always been important, humans with the passing time are becoming more conscious about what they feed to their bodies. By increasing the demand in the twenty first century, many production process of foods have encountered enormous changes and food health has become an imperative requirement.

The supply chain monitoring and tracking systems are becoming the key means for providing the detailed description of a good and its life cycle to the producers as well as to the consumers who may select the end product based on the food chain details. Such tracking system enable consumers to track product’s life cycle from the first producer via processing through different stakeholders involved to the very end shelf, where the end consumer finally purchased the product. Many companies are using supply chain to trace back to the source of products, increasing transparency into day-to-day supplier operations and reduce the irreverent journey of products to their end users. Infor, Ecovadis, Elementum, Globe Ranger, LaborVoices, Flexe, Cargomatic are the names which are involved in the supply chain technology to provide detailed information about the products to their end users. {2}

1. **Description of Work**

The paper “A readability analysis for QR code application in a traceability system”{3} proposed to use QR for tracking the food products from its initial stage which involves the raw products to the final product, this two-dimensional bar code (QR code) was read by smartphone running the android platform.

The goal of this thesis is to design and implement an android application for tracking the product details e.g., location , production date , expiration date etc., and the producer details to have the full knowledge of the products and its producer. The application allows a user to flexibly switch between different modules of the application to access the user as well as the product information. The main activity of the app consists of three modules Add Product, Scan QR and Exit. The first module add product creates the new product and stores the data in the database. When the product is created, a QR is generated on the same page that contains the products as well as the producer details. This QR code can be shared from the app once created.The second module is scanning the QR, when a user scans the QR, the application leads to the map activity where the location of the product is marked in the map and details of the product are displayed as a toast message. The third module is the exit, by clicking the exit button the user logs out and moves to the login page. The application uses MVC architecture for better manipulation of data and can be extended easily to next versions of the app. Mysql database is used for the backend module and application has a common database hosted in a third party web host server.

The Application also has to satisfy the following non-functional requirements :

1. Security : the application consider key security challenges such as a secure authentication method, secure user and product data storage with secure communication between the back and front end and DataBase.
2. Scalability : Scalability of the user interfaces is applied in the system to provide the same user interface for different hardwares in respect to font size, figures, elements position, and structure of the pages. Scalability of the application is considered to support various Android operating system (OS) versions. Finally the Backend of this application is to be implemented to support large number of users.
3. Usability : The application provides a user- friendly interface.
4. **Thesis Outline**

In chapter 1, the basic introduction to the application and its functioning is discussed based on the motivation behind the work.

In chapter 2, related work has been discussed.

In Chapter 3, introduction to android as an Os is defined along with its architecture.

In Chapter 4, all the design choices for the application and their implementation are benchmarked.The chapter describes the reasons behind certain design choices and how they are being implemented.

In Chapter 5, various security and scalability issues are addressed in order to come up with a secured and scalable application.

**Chapter 2 : Related Work**

There are wide range of articles addressing the android application development but the topic related to the supply chain management is not very old. There are many companies like Infor, Ecovadis, Elementum, Globe Ranger, LaborVoices, Flexe, Cargomatic are the names which are involved in the supply chain technology to provide detailed information about the products to their end users{2}. Dojo Expert, a martial art school uses its mobile application for tracking the attendance of the students by scanning the QR.{4}

The company Project Provenance Ltd is one of the recognised names that are into the concept of the providing transparency regarding the information about the goods we buy everyday, inspiring us to be an active citizens through the things we choose to buy. Mobile, blockchain technology and smart tagging were used to track the products with verified social sustainability claims. The goal was to explore how these new technologies could form the basis for an open system for traceability powering consumer-facing transparency for food and other physical goods. {5}

This thesis aims to explore the supply chain management system to track the product details via android platform. The main idea behind the work is the usage of QR code for tracking the product details by keeping the security and scalability measures in mind which includes secure authentication, data storage and scalable user interface and backend.

**Chapter 3 : Introduction to Android and its Architecture**

1. **Why Android as OS**

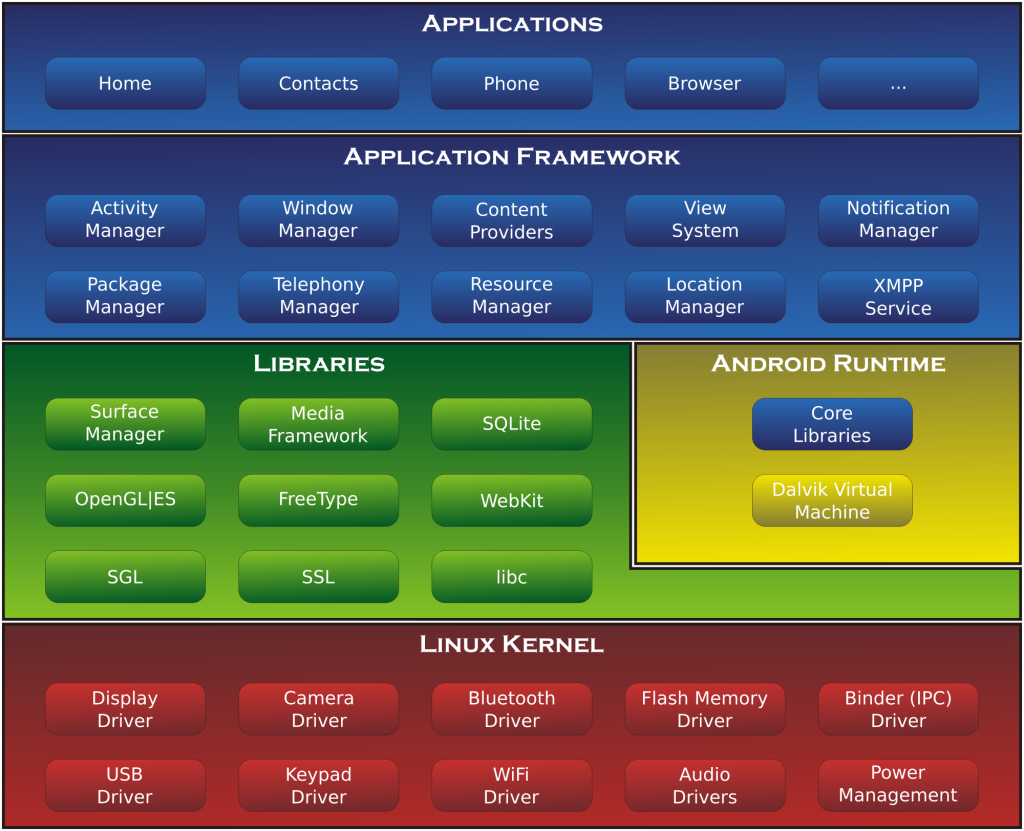
With the changing time, the number of mobile and internet users has been growing with huge pace and the adoption of mobile devices is emerging and is slated for huge growth in the next decades. According to data available in the statista, the android is one of the leading OS providers in all the smartphone devices, “In the first quarter of 2018, 85.9 percent of all smartphones sold to end users were phones with the Android operating system”{6}.

Android is gaining strength both in the mobile industry and in other industries with different hardware architectures. The increasing interest from the industry arises from two core aspects: its open-source nature and its architectural model. Being an open- source project, Android allows us to fully analyze and understand it, which enables feature comprehension, bug fixing, further improvements regarding new functionalities and finally, porting to new hardware. On the other hand, its Linux kernel-based architecture model also adds the use of Linux to the mobile industry, allowing to take advantage of the knowledge and features offered by Linux. The Android platform consists of several layers which provide a complete software stack. Android applications are Java-based and this factor entails the use of a virtual machine VM environment, with its advantages. Android uses its own VM called Dalvik, which interprets and executes portable Java-style bytecode after transforming it, which is optimized to operate on the mobile platform. All of these aspects make Android an appealing target to be used in other type of environments. {7}

1. **Architecture**

Android Architecture is shown in fig1, which consist of number of layers as Applications, Application framework, Libraries, Android runtime & Linux kernel {7}. Application layer is the uppermost layer which provides a set of core applications including an email, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language. It should be mentioned that applications can be run simultaneously; it is possible to hear music and read an email at the same time. The Application Framework is a software framework that is used to implement a standard structure of an application for a specific operating system. With the help of managers, content providers and other services programmers it can reassemble functions used by other existing applications. Layer which is present below Application framework consist of two parts as Libraries which are all written in C/C++. They will be called through a Java interface. This includes the Surface Manager, 2D and 3D graphics, Media Codecs like MPEG4 and MP3, the SQL database SQLite and the web browser engine WebKit. Second part is Android Runtime which includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The lowest layer is

Linux Kernel, Android basically relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. {7} {8}



**Fig 1 : Android Architecture**

**{8}**<http://gowrishankarnath.com/introduction-to-android-platform-and-android-development-environment/>

**Chapter 4: Design and Implementation**

**4.1 Design Choices: Discussion and final design**

The main aim of this project was to make an application that is scalable and secure. To achieve them few architectural considerations were made to achieve security and scalability for the application.

**Database:**

Data has always been the most important part of any mobile application. Developers need to store substantial amounts of data in an organized fashion in order for their apps to be truly valuable to their users. In Android, you can store your data using quite a few different methods, both in active memory and in written files. Many apps also have a remote web service that provide the data for the app in question. Android also supports data storage in a local database, and the operating system provides a good infrastructure for storing and retrieving data. Some of the major databases that can used are as follows:

### **SQLite**

SQLite is a relational database technology that’s used most often when the developer requires an embedded database in a small system{9}. SQLite is included with the Android system and can be easily used in your Android app.

# **Room Persistence Library:**

The Room persistence library provides an abstraction layer over SQLite to allow for more robust database access while harnessing the full power of SQLite {10}.

The library helps you create a cache of your app's data on a device that's running your app. This cache, which serves as your app's single source of truth, allows users to view a consistent copy of key information within your app, regardless of whether users have an internet connection.

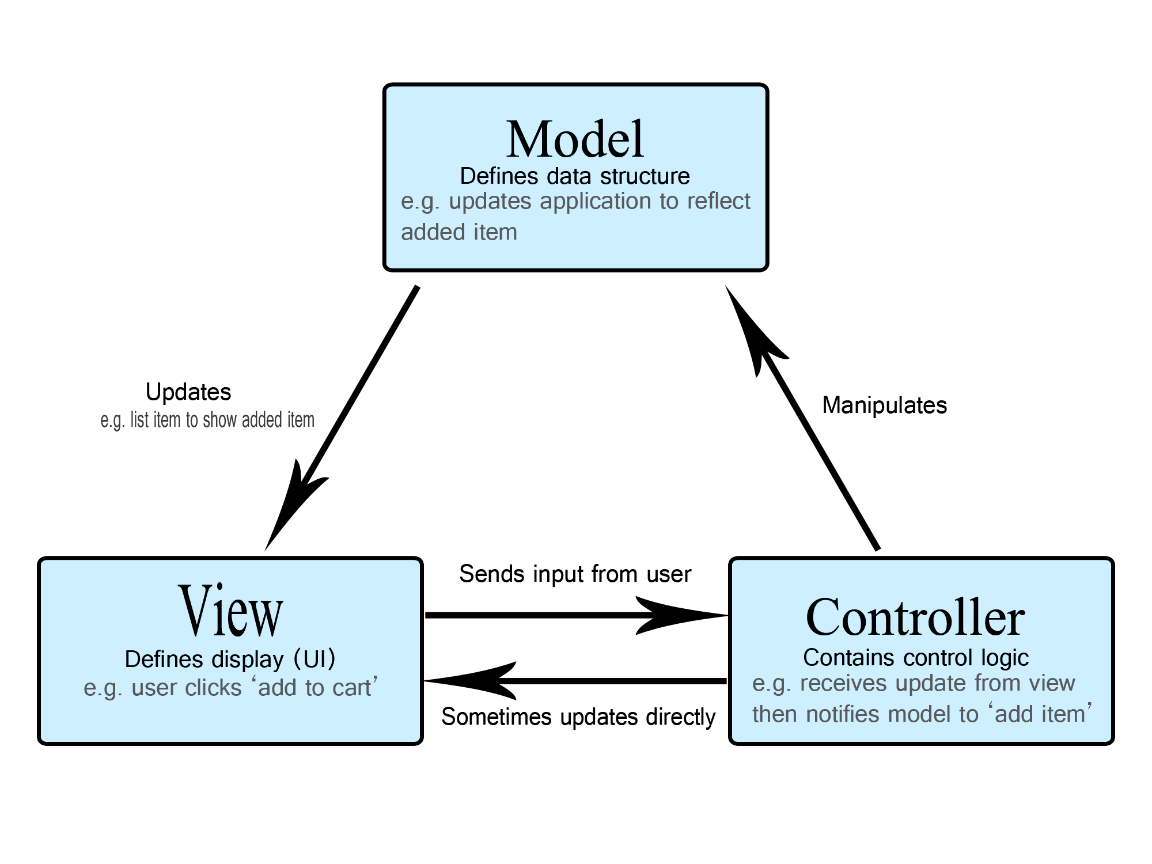
**MYSQL Database:**

MYSQL is used as a database at the webserver and PHP is used to fetch data from the database{11}. Our application will communicate with the PHP page with necessary parameters and PHP will contact MYSQL database and will fetch the result and return the results to us.

From the above databases we have chosen to use MySql database to allow multiple user to access the same dataset and also to keep the data consistent. This approach also mitigates the load on the application since the data storage and manipulations are done on a third party server and does not occupy much space in the device for the storage part. This also enables the application to be more secure since sensitive information are not stored locally.

**MVC Architecture:**

Model View Controller (MVC) is a software architecture pattern, commonly used to implement user interfaces: it is therefore a popular choice for architecting web apps{12}. In general, it separates out the application logic into three separate parts, promoting modularity and ease of collaboration and reuse. It also makes applications more flexible and welcoming to iterations.



**The Model:**

The model defines what data the app should contain. If the state of this data changes, then the model will usually notify the view (so the display can change as needed) and sometimes the controller (if different logic is needed to control the updated view).

**The View**

The view defines how the app's data should be displayed.

**The Controller:**

The controller contains logic that updates the model and/or view in response to input from the users of the app.

This architecture provides encapsulation of data since the attributes are declared private and are accessed by means of an object of that model.

**Communication method:**

**Hypertext Transfer Protocol (HTTP)** is an application-layer protocol for transmitting hypermedia documents, such as HTML {13}. It was designed for communication between web browsers and web servers, but it can also be used for other purposes. HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response. HTTP is a stateless protocol, meaning that the server does not keep any data (state) between two requests. Though often based on a TCP/IP layer, it can be used on any reliable transport layer; that is, a protocol that doesn't lose messages silently, such as UDP.

**4.2 Implementation**

The app starts with a Login module where the user validates the credential and enter into the main page. If a new user wants to sign up, a sign up options is given to create a user account. This page creates an auto generated User Id which will be used as the producer Id. The pin number created by the user is hashed using Base64Encoder and the hashed password is stored in the system. The Main Activity of the app consists of three button Add Product, Scan QR and Exit Button. The First Module Add product creates the new products and stores the details of the products in the database. Also when the product is created, a QR code is generated on the same page that contains the product as well as the producer details. The Created QR code is stored in the phone storage which can be used in future for sharing. The producer/user can also share the QR code from the app when creating the product. The Second module is the Scanning module where the user can scan the QR code. Once the QR is scanned properly the application moves to a map activity where the location of the product is marked in the map and the details of the product is displayed using the toast message. The Third module in the Exit module clicking which the application logs out and moves to the Log in page.

**Chapter 5**: **Security and Scalability**

**5.1 Security**

Security is an integral part of every application to empower users to have reliance over the system and also to guard delicate data securely. There multiple practices that ought to be followed to create a secure mobile application{14}. Some of the important practices are as follows:

**Enforce secure communication:**

When you safeguard the data that you exchange between your app and other apps, or between your app and a website, you improve your app's stability and protect the data that you send and receive.

#### **Show an app chooser:**

If an implicit intent can launch at least two possible apps on a user's device, explicitly show an app chooser. This interaction strategy allows users to transfer sensitive information to an app that they trust.

**Apply signature-based permissions:**

When sharing data between two apps that you control or own, use *signature-based* permissions. These permissions don't require user confirmation and instead check that the apps accessing the data are signed using the same signing key. Therefore, these permissions offer a more streamlined, secure user experience.

**Disallow access to your app's content providers:**

Unless you intend to send data from your app to a different app that you don't own, you should explicitly disallow other developers' apps from accessing the ContentProvider objects that your app contains. This setting is particularly important if your app can be installed on devices running Android 4.1.1 (API level 16) or lower, as the android:exported attribute of the <provider> element is true by default on those versions of Android.

**Use intents to defer permissions:**

Whenever possible, don't add a permission to your app to complete an action that could be completed in another app. Instead, use an intent to defer the request to a different app that already has the necessary permission.

**What have been implemented in our application?**

The Application is made using MVC architecture using which all the inputs and manipulation of data are done as objects and they are well encapsulated. The password typed by the user at the time of sign up and login are hashed using Base64Encoder before sending it to the server for storage and validation respectively. The application also requires a user authentication before going into the system. The communications with the database are done using Http POST request and the parameters and the values are transferred securely. The database of the application is created in a secured webhost server and does not have any sensitive data about the user in the device. The user is also prompted to give a run time permission to write the files in the phones storage for security.

**5.2 Scalability**

Since the application stores data in a remote server it does not have any load in inserting or retrieving the data of the user. Based on the capacity of the server and strength of the network our application can perform seamlessly without any lag. The application will have to handle the http requests which are created as threads to enable multiple access at a time. The performance of the application is completely independent of the number of users in the system.

{1}<https://www.prnewswire.com/news-releases/study-ninety-four-percent-of-consumers-say-food-product-transparency-from-brands-and-manufacturers-is-important-impacts-purchase-300287311.html>

{2} <https://www.greenbiz.com/article/9-supply-chain-tech-companies-you-should-know>

{3} <https://www.sciencedirect.com/science/article/pii/S0168169914002142>

{4} <http://www.dojoexpert.com/new-version-of-qr-scanning-app/>

{5} <https://www.provenance.org/tracking-tuna-on-the-blockchain>

{6}<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>

{7}<https://doc.lagout.org/programmation/Android/Android%20Security_%20Attacks%20and%20Defenses%20%5BMisra%20%26%20Dubey%202013-04-08%5D.pdf>

{8}<http://gowrishankarnath.com/introduction-to-android-platform-and-android-development-environment/>

{9} <https://www.sitepoint.com/using-database-in-android/>

{10}https://developer.android.com/topic/libraries/architecture/room

{11}https://www.tutorialspoint.com/android/android\_php\_mysql.htm

{12}https://developer.mozilla.org/enUS/docs/Web/Apps/Fundamentals/Modern\_web\_app\_architecture/MVC\_architecture

{13}https://developer.mozilla.org/en-US/docs/Web/HTTP

{14}https://developer.android.com/topic/security/

https://docs.google.com/document/d/1lvvnqeUyV4rsmPgSg403xJFLnVOQH8qL1Gapor8aiPA/edit