



**University of
Zurich** ^{UZH}

Development of a Secure Android Application for a Supply Chain Tracking System

*Tanbir Mann
Zurich, Switzerland
Student ID: 16-744-807*

Supervisor: Sina Rafati Niya, Prof. Burkhard Stiller
Date of Submission: August 31, 2018

Abstract

This assignment is based on a Supply Chain Project for maintaining the flow of inbound and outbound stocks in an inventory system and covers the design and implementation of an android application to track the product details in supply chain system. 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 the application has a common database hosted in a third party webhost server. All the data from different users hits the same database as a thread request. The request from the application is made as an HTTP request from a class that extends AsyncTask class for concurrency. The HTTP requests are sent to different php scripts which handles the database connection and data manipulation parts, the application receives a response from the php scripts based on the outcome of the operation. Since all the database operations are handled in the server the application can scale up to thousands of users without impacting the performance of the application. This application is compatible on all the smartphones with basic facilities. This application also encrypts the password before sending it to the server so that midway hacking of the password is not possible. The application is evaluated in terms of deployment, scalability, and security. 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.

Contents

Abstract	i
1 Introduction	1
1.1 Motivation	1
1.2 Description of Work	2
1.3 Thesis Outline	3
2 Related Work	5
3 Introduction to Android and its Architecture	7
3.1 Why Android as OS	7
3.2 Architecture	7
4 Design and Implementation	9
4.1 Database:	9
4.2 SQLite	9
4.3 Room Persistence Library	10
4.4 MYSQL Database	10
4.5 MVC Architecture:	11
4.6 The Model	11
4.7 The View	12
4.8 The Controller	12
4.9 Communication method:	13
4.9.1 Implementation	14

5 Security and Scalability **17**

5.1 Security 17

5.2 Enforce Secure Communication 17

5.3 Show An App Chooser 18

5.4 Use Intents To Defer Permissions 18

5.5 Apply Signature-Based Permissions 19

5.6 A Short description of what have been implemented in our application for
its Security 19

5.7 SQL Injection 19

 5.7.1 Scalability 20

5.8 Conclusion 22

Chapter 1

Introduction

1.1 Motivation

Demand for transparency is increasing, surprisingly we know little 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" [26]. 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 have 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 [27].

1.2 Description of Work

The paper, A readability analysis for QR code application in a traceability system[28] 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 assignment 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.

1.3 Thesis Outline

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

In Section 2, related work has been discussed.

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

In Section 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 our application.

In Section 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 [27]. Dojo Expert, a martial art school uses its mobile application for tracking the attendance of the students by scanning the QR [29].

There are many companies such as Project Provenance Ltd is one of the recognized 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 to power consumer-facing transparency for food and other physical goods [30].

Nowadays, the endorsement of many IoT devices and technologies in the supply chain management sector has attracted a lot of research interest. It is implementation with the blockchain technology in order to have a full traceability of the products from producer to consumer, the work done in this paper [2] represents AgriBlockIoT, a fully decentralized, blockchain-based traceability-ity solution for Agri-Food supply chain management, able to seamless integrate IoT devices producing and consuming digital data along the chain. A food supply chain traceability system for real-time food tracing based on Hazard Analysis and Critical Control Points (HACCP), BC and IoT, which could provide an information platform for all the supply chain members with openness, transparency, neutrality, reliability and security is implemented by this paper [3]. The concept of blockchain with IoT is not very old but the above cited papers concludes that the combination of Blockchain with IoT is pretty much suitable for traceability of products in supply chain.

Whereas, our work is also related to tracking the lifecycle of the product but this assignment 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

3.1 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"[24]. 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 [25].

3.2 Architecture

Android Architecture is shown in figure 1, which consist of number of layers as Applications, Application framework, Libraries, Android runtime and Linux kernel [25]. Application layer is the uppermost layer which provides a set of core applications including an



Figure 3.1: Android Architecture [18]

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. Android has a customized Linux kernel that is specific to Android. It provides power management, memory sharing and management (due to less memory constraints of mobile devices). Inter-process communication is allowed wherein it allows several processes to communicate among themselves in a sophisticated manner. For Android software to flexibly communication between wide range of hardware devices like memory, radio, camera etc, device drivers are required to be properly plugged in. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack [25] [18].

Chapter 4

Design and Implementation

4.1 Database:

For the development of any mobile application, data has always played a key role. An app can only be fairly valuable to its consumers if the data is organized in a well-established fashion, in order to achieve a fully organized data developers need to consider every possible way of storing the substantial amount of data in the database. Android serves the leverage of storing the data with quite different methods, both in active as well as in written files. There are many applications that use remote servers to provide data for the app in question. Like many other platforms, android also supports the data to be stored in its local database whereas the operating system provides a good framework for data hoarding and redeeming. Some of the major databases that can be used are as follows.

4.2 SQLite

SQLite is an in-process library that enforces a self-contained, server-less, zero-configuration, and transactional database engine. In contrast with other databases, SQLite does not have a separate server process, it performs all the reading and writing operations directly to an ordinary disk file [22]. It is a relational database technology that's used most often when the developer requires an embedded database. SQLite is included with the Android system and can be easily used in any Android app [19]. It is written in C and supports more than 30 programming languages. It may be difficult to find a device with no application that uses SQLite. There are certain issues of using SQLite as database in the application. There is much inefficiency due to boilerplate code in the long run with the maintenance of the app especially when database views are involved, the performance of the SQLite is also unreliable as well as there are no compile time checks e.g., SQL queries [23]. Following are the issues with SQLite [14]:

1. Security: The database is stored in one single file which risks losing all the data if an unusual and uncontrolled event occurs and removes the file. In order to avoid

such events strong file level protection should be taken with carefully planning file permissions and the file should be avoided to put into documented root. The other issue regarding the security is that the journal files are not always deleted which provides the opportunity to roll back as the system keeps the journal files for a certain period of time but it may leads to data corruption as the files are not being deleted completely. Other major pitfall behind the journaling is that it cannot be turned off when the app crashes which leads to damaging the database. In order to avoid such problems it is important to strongly encrypt the data before putting it into the database.

2. Flexibility: There are many flexibility issues with the application once it starts getting older as more data piles up in the database. Complicated queries must be written in order to maintain the productivity of the app which creates problem in most cases while using structured query language. Data migration is needed to be done properly which is an issue in SQLite.

4.3 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 [17]. The Room persistence library helps in creating a cache of the app's data on a device that's running on the same application. This cache, which serves as the app's single source of truth, allows the users of the app to view a congruent clone of all the important information within the app, regardless of a working Internet connection in the users device.

It has the following improvements over SQLite [14]:

1. Encryption: It has improved the encryption of data before it is stored in the database whereas SQLite has many lacks in properly organizing encryption.
2. Live Data: Room removes the baffled interrupts and provides easy use of live data.
3. Data Migration: It provides fast and easy data migration unlike SQLite.

4.4 MYSQL Database

MySQL is an open source Relational Database Management System (RDBMS) based on Structured Query Language (SQL). It runs on almost all the platforms, including Linux, Unix and Windows. It is the most modish database management system for every type of web projects. MySQL is most commonly associated with web-based applications and online publishing. It is also an important component of an open source enterprise stack called LAMP [16]. It is used as a database at the webserver for our app and PHP is used to fetch data from the database. 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.

Following are the pros of using MYSQL over above-mentioned databases [15]:

1. Ease of use: It has no complex installation criteria and supports third party tools e.g., GUIs which makes it simple to work with.
2. Secure: It has more inbuilt advanced security features as compared to SQLite.
3. Scalable: It can handle any amount of data, which makes our application scalable to large extend (in terms of number of users).
4. Speed: The response power is extremely high in MySQL, as it has to follow few less standards as compared to SQLite.

From the above databases we have chosen 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 is not stored locally.

4.5 MVC Architecture:

Model View Controller (MVC) is an architecture pattern for software building and is widely used to implement user interfaces. It has been a popular choice among web apps for their architectural design. MVC architecture design works by splitting out the application logic into three parts which helps in advancing the app's modularity, makes it easy for collaboration and reuse, enhancing the flexibility of the app and making it more resistant to iterations [12].

The traditional programming approach used the UI coding, business logic, input logic and applications data domain was written into a single file, which creates certain downfalls in maintenance of the app, testability of the app was also sacrificed along with the scalability. MVC approach has improved the creation of the applications by separating the different aspects of the application (input logic, business logic and UI logic), while providing loose coupling between these elements. The location of each of these logic is specified by the pattern, UI logic belongs in the view, Input logic belongs in the controller and Business logic belongs in the model. This separation helps in building the application efficiently by managing its complexity as it enables the developer to focus on one aspect of implementation at a time. For example, we can focus on view without depending on the model or controller part [13].

4.6 The Model

The model defines the data structure of the application i.e., it defines what data the app should contain. Model objects are the parts of the application that implements the logic for application's data domain and retrieve and store model state in the database. For example, a Customer object might retrieve information from a database, operate on it, and then write updated information back to a Customer table in a SQL Server database.

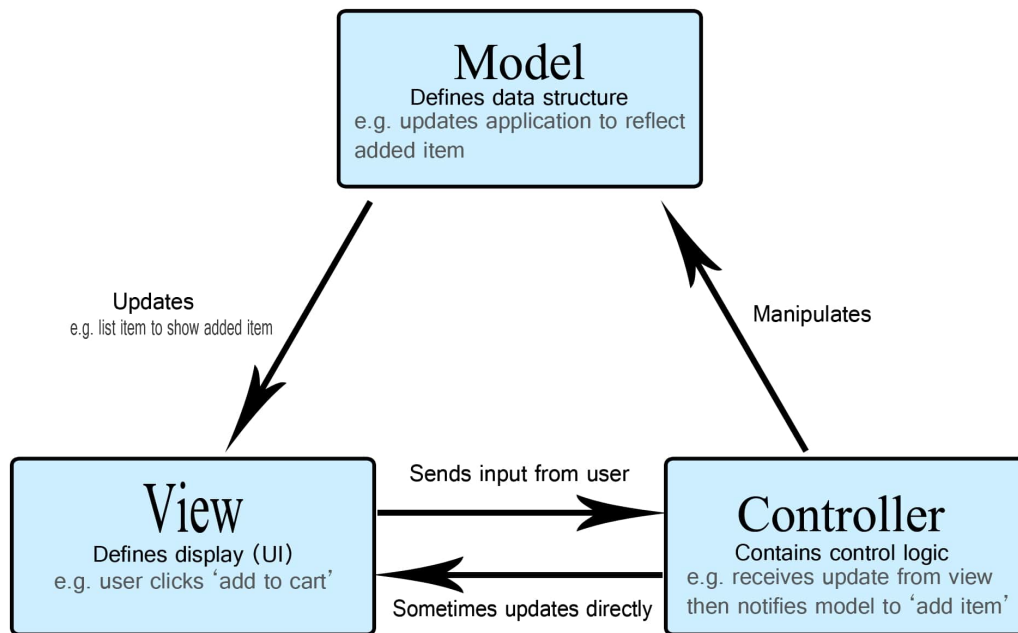


Figure 4.1: MVC Architecture [13].

In small applications, the model is often a conceptual separation instead of a physical one. For example, if the application only reads a dataset and sends it to the view, the application does not have a physical model layer and associated classes. In that case, the dataset takes on the role of a model object [13]. The Model always notifies its neighbors whenever the state of this data changes. It notifies the view about recent data changes so that the display can be changed according to the current state of the data. Sometimes it also notifies the controller if different logics are required to control the updated view [12].

4.7 The View

The view defines how the app's data should be displayed and deals with all the components that display the application's user interface (UI). This user interface is created typically from the model data. An example would be an edit view of the customer table that displays UI Controls based on the current state of the customer object [13].

4.8 The Controller

The controller contains logic that updates the model and/or view in response to input from the users of the app. Controllers are the components that handle user interaction,

work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn might use these values to query the database [13].

Following are the advantages of using MVC architecture [13]:

1. Parallel development: Collaborative developers can work in parallel on the model, controller and views.
2. High Cohesion: MVC enables logical grouping of related actions on a controller together. The views for a specific model are also grouped together.
3. Low coupling: The pattern of the MVC is such that there is very low coupling among different elements i.e., model, controller and view.
4. Ease of modification: As every element in the model has its own separate responsibilities, future development and modification is easier which increases the scalability of the product.

In our application, this architecture provides encapsulation of data since the attributes are declared private and are accessed by means of an object of that model. In the code User and Product model classes are created to store the user entered data as an object. These classes are POJO classes which holds the values for different instances of the object. Whenever a new User or Product is created the data from the application is created as an object which in turn is used for the database operations. These classes are created under the Model package. The View module comprises of the classes that have the GUI components mapped, this class is responsible for getting the data from the Graphical interface and creates them as an Object using the classes from model, this comprises the view package. The last module is the controller package. This application has a class named BackgroundWorker class which gets the object from the user entered data and creates a HTTP connection to the webserver to do the database manipulations. This controller package is completely responsible for all the database operations.

POJO is known as Plain Old Java Object, commonly used to describe a class that doesn't need to be a subclass of any other class, or bound to implement specific interfaces, or follow a specific pattern. It has properties, getters and setters to manage POJO objects.

4.9 Communication method:

Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It is one of the primary protocols for exchanging information across the Internet. Traditionally, it is designed for communication between web browsers and web servers. The information between clients and servers is exchanged in the form of Hyperlink documents. HTTP follows a classical client-server model with a client opening a connection to make a request and the server issues a response that

includes not only the requested content, but also relevant status information about the request. The self-contained design allows for the distributed nature of the Internet, where a request or response might pass through many intermediate routers and proxy servers. It also allows intermediary servers to perform value-added functions such as load balancing, caching, encryption, and compression [8]. 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 [9].

The extensible nature of HTTP, granted more functionality and control of the web. Following are the features controllable with HTTP [10].

1. Cache : Clients and the proxies are used to cache the data, the server instructs the proxies, and clients, which data to cache and for how long.
2. Relaxing the origin constraint: To prevent snooping and other privacy invasions, Web browsers enforce strict separation between Web sites. Only pages from the same origin can access all the information of a Web page. Though such constraint is a burden to the server, HTTP headers can relax this strict separation server-side, allowing a document to become a patchwork of information sourced from different domains (there could even be security-related reasons to do so) [10].
3. Authentication: There are some pages that are needed to be protected in order to give access to only specific users. Basic authentication is provided by the HTTP by using WWW-Authenticate and similar heads or by creating specific session using HTTP cookies [10].
4. Session : HTTP cookies are responsible for linking the requests with the servers state which in turn creates sessions, despite HTTP being a stateless protocol. These sessions are useful for e-commerce shopping baskets as well as for any site that allows user configuration of the output.
5. Proxy and tunneling : Servers and/or clients are often located on intranets and hide their true IP address to others. HTTP requests then go through proxies to cross this network barrier. Not all proxies are HTTP proxies. The SOCKS protocol, for example, operates at a lower level. Others, like FTP, can be handled by these proxies [10].

In our application, HTTP communication is used to create connection between the application and the web host server by means of POST request. These connections are made to connect to remote mysql database and perform the actions related to the database. In our code BackgroundWorker class creates these HTTP requests, passing string parameter to the database as a thread.

4.9.1 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. Some glimpse of the App are shown in the figure 4.2, 4.3, 4.4 and 4.5.

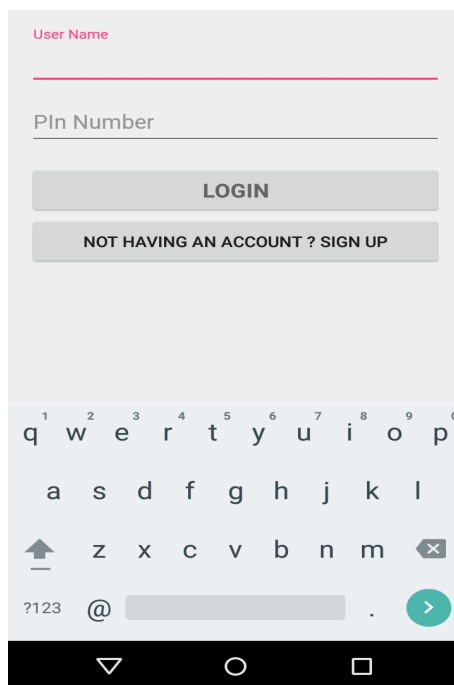


Figure 4.2: Startup Screen

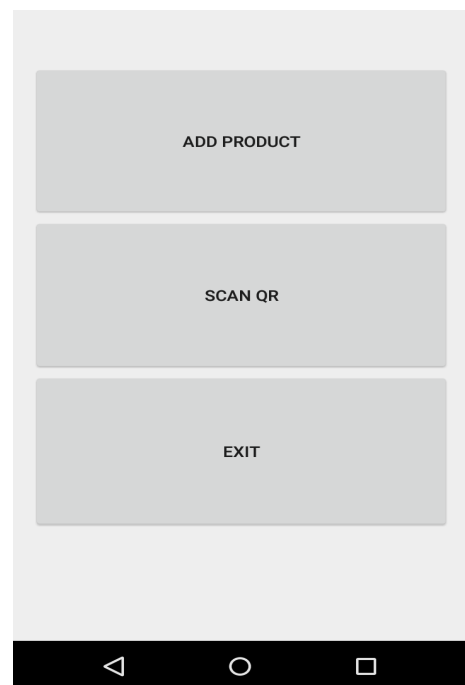


Figure 4.3: Login Screen

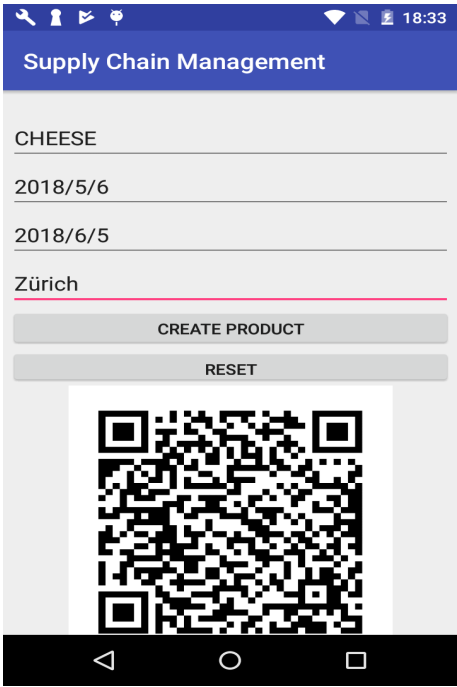


Figure 4.4: Startup Screen



Figure 4.5: Login Screen

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. Android has built-in security features such as Robust security at the OS level through the Linux kernel, Mandatory application sandbox for all applications, Secure interprocess communication, Application signing, Application-defined and user-granted permissions that significantly reduce the frequency and impact of application security issues [7]. The system is designed in such way that it allows user to build the app with the default system and file permissions without being worried about the security decisions and issues. There are multiple practices that ought to be followed to create a secure mobile application. Some of the important practices are as follows [6]:

5.2 Enforce Secure Communication

The best procedure for this practice is to protecting the data which has been exchanged between the developed app and other apps or websites. This improves the stability of the app and guard the data which has been sent or received from any unusual activity. To enforce secure communication, our application creates HTTP connection between the app and the web host server and transfer the data by means of a post request. These connections are made to connect to the remote mysql database and perform the actions related to the mysql database through a php script saved in the server.

PHP stands for Hypertext Pre-Process. It is a server side language, meaning it is not understand by the browser. It comes with many functionalities such as it can translate information, do math, connect to databases and more. In our case it is used to connect to the database. It has the ability to loop over sections of content based on conditions, ask the computer questions and similar programming functions, an example of a PHP script can be seen in figure 5.1.

/public_html/Android/login.php

```
1 <?php
2 require "connect.php";
3 $user_name = $_POST["user_name"];
4 $user_pass = $_POST["password"];
5 $mysql_qry = "select * from users where user_name like ? and password like ?;";
6 $stmt=$conn->prepare($mysql_qry);
7 $stmt->bind_param("ss",$user_name,$user_pass);
8 $stmt->execute();
9 if($stmt->fetch()) {
10 echo "login success !!!!! Welcome user";
11 }
12 else {
13 echo "login not success";
14 }
15
16 ?>
```

Figure 5.1: PHP Script

5.3 Show An App Chooser

This practice comes into play when an indirect intent is able to initiate more than one possible apps on a users device, the user must be provided to select an app chooser. This interaction approach helps the users to move sensitive data to an app or website they trust. Our application can share the QR code generated for the product using other third party apps like Whatsapp, Gmail, Facebook etc. To accomplish this practice our app shows an app chooser window which will allow the user to send the QR code to their trusted third party.

5.4 Use Intents To Defer Permissions

The best way to perform this practice is by disallowing the permissions within the app to complete certain actions which could be completed with the help of other apps. Instead, an intent must be used to postpone the request to a different app that already exhibit the necessary permission. Since our app runs on API 23, it requires runtime user permission for few of the actions. For example, when the app is opened for the first time the app asks for user permission to write on the storage location of the phone. These permissions requests are created by means of a intent requests in a sense that when a user write its credentials in the Login page, at that point it checks if the App have some permissions to write on the device. We used intent for permissions only in login (figure 3a) and Product creation part (figure 4a), where it checks if the App has permissions to write on the device.

5.5 Apply Signature-Based Permissions

Signature-based permissions are required when we need to share data between two apps that we own or control. These permissions are user independent, e.g., they donot require user confirmation, on the other hand, they check if the data accessed by the app is signed using the same signing key. Therefore, these permissions offer a more streamlined, secure user experience. These permissions are not implemented by our app as we have not used the content provider for data storage within the device

5.6 A Short description of what have been implemented in our application for its Security

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, all 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.

5.7 SQL Injection

SQL injection is a code infection method of a SQL query, it usually occurs when client insert data in the application through an untrusted source or the data which is used to build a SQL query dynamically. If this attack become successful, it is capable of reading sensitive information from the database, can temper the data in the database such as Insert/Update/Delete, can have access to control operations on the database such as shutting down the database management system and can issue commands to the operating system [4]. It allows the attacker to trick with its identity and manipulate the existing data causing various issues such as invalid transactions, destroy data and make it inaccessible and become legitimate owner of the database server. This attack is very usual in PHP and ASP applications due to the universality of older functional interfaces. The main effects of the attack are [4]:

1. Confidentiality: Dissipation of confidentiality is very common problem with SQL injection as SQL database carry sensitive data.
2. Authentication: It is possible to get into the system as a fake user with no background knowledge of the password if poor SQL commands are used to check user names and passwords.

3. Authorization: If authorization information is kept in a SQL database, it may be possible to alter this information through the successful exploitation of a SQL Injection vulnerability.
4. Integrity: With the possibility to read sensitive information, it is also possible to alter or delete this information with a SQL Injection attack.

Prevention from this attack is done by making the use of prepared statements in the PHP script on our remote server, a snapshot of the PHP script can be seen in figure 5.1. Prepared statements validates the user identity whenever a user send its data e.g., user Id and password to the application. By making the use of named parameters we can specify a variable name and can give its value during the run time which helps in preventing unauthorized access of the data [5].

5.7.1 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 handles 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. Section 4.4 and 4.5 helps in covering the most of the scalability part of our application as we have used Mysql database and MVC architecture which helps the application to meet the scalability requirements.

The application uses minimum Sdk Version 19 which will make all the devices having sdk 19 or more to be compatible for the application . The application screens are build using Constraintlayout. It allows to drag new views to the layout, attach their constraints to the parent view and other sibling views, and edit the view's properties, all without editing any XML by hand. To ensure that our lay out is flexible and adapts to the different screen sizes, we used wrap-content and match-parent for the width and height of most of the view components, instead of hard-coded sizes. Wrap-content tells the view to set its size to whatever is necessary to fit the content within that view. Match-parent makes the view expand to as much as possible within the parent view.

1. MVC Architecture: The MVC architecture is one of the most relevant architecture for developing the applications considering scalability and maintainability requirements. In this architecture design the coupling among the modules is loose which helps in parallel proceeding by separating the transaction logic from presentation. By using the Mysql Database, maintainability and scalability of the whole system can be improved greatly [1].
2. MySQL: Mysql database is used to handle the data in our application as it allows 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 addresses the scalability in terms of number of

users who can store data without any lag, hence backend of the application supports large number of users.

No further tests or scalability practices are performed, assuming that the application will scale to thousands of user with its current design choices.

5.8 Conclusion

The Implemented Android application satisfies all the requirements for this assignment. It allows the user of the app to efficiently and securely add its user details in the dataset. The user can have its personal account in the app and can access the app whenever required. The main aim of the application was to successfully add product details and to generate a QR of the product which can be mapped later by the scanning part considering security and scalability challenges of building such a system. The scanning QR module of the app gives us the location of the product where it was produced as well as the producer details along with the product details (e.g., manufacturing/ expiring date). The application is successfully performing all the above tasks. This assignment is a prerequisite work for the Master Project which is based on the same concept i.e., to develop an android application for supply chain products and to map the life cycle of the product. Our future work is to develop such an application on large scale whose back-end will work using BC to trace the life cycle of the product which will allow us to get the full details of the product in secure manner. We will use the supply chain application for tracking the product details in every phase (e.g., production, processing or transporting) and from producer to its end user, the main focus of the work will be to address more security and scalability challenges.

Bibliography

- [1] Luo GuangChun, WangYanhua Lu, Xianliang Hanhong : A Novel Web Application Frame Developed by MVC *MVC Architecture*,2003,[online] <https://dl.acm.org/citation.cfm?id=638779>
- [2] Miguel Pincheira Caro, Muhammad Salek Ali,Massimo Vecchio : Blockchain-based Traceability in Agri-Food Supply Chain Management *A Practical Implementation*,2018,[online] <https://ieeexplore.ieee.org/document/8373021/>
- [3] Feng Tian : A Supply Chain Traceability System for Food Safety Based on HACCP, Blockchain and Internet of Things ,2017,[online] <https://ieeexplore.ieee.org/document/7996119/>
- [4] SQL Injection : Threat Modelling and Risk Factors,[online] https://www.owasp.org/index.php/SQL_Injection
- [5] SQL Injection : Prepared Statements to prevent SQL Injection,[online] <https://websitebeaver.com/php-pdo-prepared-statements-to-prevent-sql-injection>
- [6] Security :Android Security and its Best Practices ,[online] <https://developer.android.com/topic/security/best-practices>
- [7] Security :Android In-Build Security Features,[online] <https://source.android.com/security/>
- [8] NGINX :What is Hypertext Transfer Protocol ,[online] <https://www.nginx.com/resources/glossary/http/>
- [9] HTTP :Asymmetric request-response client-server Protocol ,[online] http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http_basics.html
- [10] MDN web docs : An Overview of Http ,[online] <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- [11] Android : Security tips and Requesting Permissions ,[online] <https://developer.android.com/training/articles/security-tips#RequestingPermissions>
- [12] Medium : Why MVC Architecture ,[online] <https://medium.com/@socraticsol/why-mvc-architecture-e833e28e0c76>

- [13] MDN we docs : MVC Architecture ,[online] https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture
- [14] Hackernoon : SQLite or Realm: What is the Top Choice for Android App? ,[online] <https://hackernoon.com/sqlite-or-realm-what-is-the-top-choice-for-android-app-2ee07dfcb82f>
- [15] DigitalOcean : SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems,[online] <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-system>
- [16] TechTarget: MySQL ,[online] <https://searchoracle.techtarget.com/definition/MySQL>
- [17] Android Developers : Room Persistence Library? ,[online] <https://developer.android.com/topic/libraries/architecture/room>
- [18] Introduction to Android Platform and Android Development Environment [online] <http://gowrishankarnath.com/introduction-to-android-platform-and-android-development-environment/>
- [19] SitePoint : Mastering Database Storage and Retrieval in Android ,[online] <https://www.sitepoint.com/using-database-in-android/>
- [20] Android Developers : Room Persistence Library Architecture ,[online] <https://developer.android.com/topic/libraries/architecture/room>
- [21] MDC web docs : Modern Web App Architecture : MVC Architecture ,[online] https://developer.mozilla.org/enUS/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture
- [22] SQLite : About SQLite ,[online] <https://www.sqlite.org/about.html>
- [23] Greenboot : Mobile databases: SQLite and SQLite alternatives for Android and iOS, [online] <http://greenrobot.org/news/mobile-databases-sqlite-alternatives-and-nosql-for-android-and-ios/>
- [24] Statista : Global mobile OS market share in sales to end users from 1st quarter 2009 to 1st quarter 2018 ,[online] <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>
- [25] Abhishek Dubey, Anmol Mishra : Android Security *Attacks and Defenses* ,[online] <https://doc.lagout.org/programmation/Android/Android%20Security%20Attacks%20and%20Defenses%20%5BMisra%20%26%20Dubey%202013-04-08%5D.pdf>
- [26] CISION: Ninety-four Percent of Consumers say Food Product Transparency From Brands and Manufacturers is Important, Impacts Purchase,[online] <https://www.prnewswire.com/news-releases/>

study-ninety-four-percent-of-consumers-say-food-product-transparency-from-brand.html

- [27] GreenBiz : 9 supply chain tech companies you should know ,[online] <https://www.greenbiz.com/article/9-supply-chain-tech-companies-you-should-know>
- [28] ScienceDirect : A readability analysis for QR code application in a traceability system,[online] <https://www.sciencedirect.com/science/article/pii/S0168169914002142>
- [29] DojoExpert : A software for martial arts school ,[online] <http://www.dojoexpert.com/new-version-of-qr-scanning-app/>
- [30] Provenance: Introductio to Blockchains for supply chain transparency,[online] <https://www.provenance.org/tracking-tuna-on-the-blockchain>