## **SECURE SCAN**

## **INTRODUCTION TO THE PROBLEM STATEMENT:**

## 

## OVERVIEW

One-step way to make payments, know about an item, and much more work in one scan the technology of QR code has facilitated us. A QR code an initialism for quick response code is a type of matrix barcode or two-dimensional barcode that has become a part of our daily life in numerous ways. When we scan a QR code, it merely displays the link, which we can then follow. However, there always exists a question mark on the URL source, whether that link is secured enough.

# EXISTING SYSTEM:

There are numerous applications available for scanning QR Codes. As the QR codes are not inspected or made by a secure authority, anyone can create a QR code in a matter of seconds using simple techniques, yet this piece of code is incomprehensible to the naked eye.

The URL data type is the only scenario in which conventional QR codes can carry executable data. Because a reader would normally send the data to the application associated with the data type utilized by the QR code, these URLs may contain JavaScript code that can be used to exploit vulnerabilities in applications on the host system, such as the reader, web browser, or image viewer.

Linking to dangerous websites with browser exploits, enabling the microphone/camera/GPS and then streaming those feeds to a remote server, analysis of sensitive data (passwords, files, contacts, transactions), sending email/SMS/IM messages or packets for DDoS as part of a botnet, corrupting privacy settings, stealing identity, and even containing malicious logic such as JavaScript or a virus are all risks. These actions could take place in the background, with the user only seeing the reader open a seemingly innocuous web page.

## **OUR SYSTEM:**

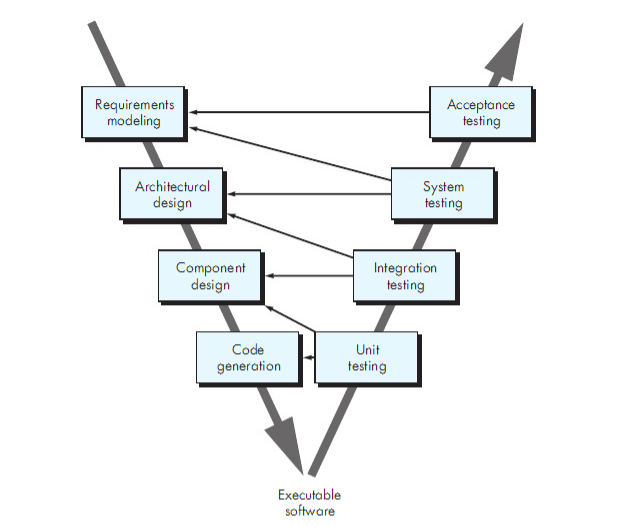
Even though QR code scanner is inseparable from our daily life, many of the users are unaware of the dark sides of using this technology. One way solution is to check each and every QR code and its source and authenticity. Scanning all the QR codes around a large area, maintaining a proper database for it, and going through it each time while scanning a code is a laborious task. An easier way would be to automate this task and make it more efficient and feasible by maintaining a middleware database authenticator software. That software will work as a normal QR code scanner and on scanning verifies the code from the database and informs the user whether the code is secure or not.

## **SOFTWARE PROCESS MODEL:**

Software Processes are a set of actions that are used to specify, develop, implement, and test software systems. A software process model is an abstract representation of a process that provides a description of the process from a certain point of view.

## **V-Model:**

The software model fits for our project is “**V-model”** (a variation of Water-fall Model). V- model provides a way of visualizing how verification and validation actions are applied to early engineering works. As a software team moves down the left side of the V, basic problem requirements are refined into progressively more detailed and technical representations of the problem and its solution. Once the code has been generated, the team moves up the right side of the V, essentially performing a series of tests that validate each of the model created as the team moves down the left side.



### **1. Feasibility Study:**

### The primary goal of this phase is to determine whether developing the application is financially and technically feasible. The feasibility study entails first determining the problem and then determining the various possible solutions. These various solutions are evaluated for their advantages and disadvantages. The best solution is selected, and the remaining phases are carried out in accordance with the solution strategy.

### **2. Requirement analysis and specification:**

The goal of the requirement analysis and specification phase is to fully comprehend the customer's requirements and properly document them. There are two different activities in this phase.

**• Requirement gathering and analysis:**

To begin, all of the software requirements are gathered from the customer, and then the requirements are analyzed. The analysis section's goal is to eliminate gaps and inconsistencies.

**• Requirement specification:**

### A software requirement specification (SRS) document is used to document the analyzed requirements. Customers and the development team sign the SRS document as a contract. The SRS document can be used to resolve any future disputes between customers and developers.

### **3. Design:**

### The design phase's goal is to turn the requirements specified in the SRS document into a structure that can be implemented in a programming language.

### **4. Coding and Unit Testing:**

### During the coding phase, the software design is converted into source code using any programming language that is appropriate. As a result, every designed module is coded. The goal of the unit testing phase is to see if each module is functioning properly.

### **5. Integration and System Testing:**

The integration of various modules occurs shortly after they have been coded and unit tested. The integration of various modules is done in stages over a period of time. Previously planned modules are added to the partially integrated system during each integration step, and the resultant system is tested. Finally, after all of the modules have been successfully integrated and tested, a complete working system is obtained, and system testing is performed.

There are three types of testing activities in system testing, as described below:

**• Alpha testing:** Alpha testing is the system testing performed by the

development team.

**• Beta testing:** Beta testing is the system testing performed by a friendly

set of customers.

**• Acceptance testing:** After the software has been delivered, the customer

performed the acceptance testing to determine whether to accept the

delivered software or to reject it.

### **6. Maintenance:**

The most crucial phase of a software life cycle is maintenance. Maintenance takes up 60% of the total time and effort required to develop a complete software. Maintenance can be divided into three categories:

**• Corrective Maintenance:** This type of maintenance is performed to correct errors that were not discovered during the development phase of the product.

**• Perfective Maintenance:** This type of maintenance is performed in response to a customer's request to improve the system's functionality.

**• Adaptive Maintenance:** When porting software to a new environment, such as working on a new computer platform or with a new operating system, adaptive maintenance is usually required.

# **Why to use V-model?**

* The Secure Scanner Project is very simple and is easy to understand. We are

giving extensions to already existing applications like normal QR Code Scanners.

• V-Model is used where requirements are well understood and changes can be required iff error occurs during the testing phase in our project.

• Process, actions and results are documented but needs verification that can be done only at the time of testing.

• This project reinforces good habits: define-before-design and design-before code. This project will be built using Android-Studio IDE and Java AWT. It can be

further upgraded to IOS.

• The scope of our project precludes a purely linear process. V-Model is

best suitable for linear process.

## **Functional Requirements:**

# **Scanning Module:**

This is the main module of the application that a user views just after opening the app. It provides user a GUI where user is accessible to scan the QR code and view the result of the query.

# **Verified User Registration Module:**

This module will only be for users the users who want them to be verified. Using this module, user can place a request to access the features available for verified users.

# **Log in Module:**

This module will only be accessible by verified users. Using this module, user can enter as a verified user.

# **Request Module:**

This module will only be accessible by verified users. Using this module, user can request a query claiming an URL to be unsafe.

# **Design and Implementation Constraints:**

• This is an Android Based Application using Java which can

be further extended to IOS also.

• This project is built using Android-Studio IDE and Java AWT.

• The application uses MYSQL database server.

## **Software Requirement Specification (SRS):**

**What is SRS?**

A software requirements specification is a document that captures complete description about how the system is expected to perform.

**SRS Overview:**

The SRS document satisfies the following: -

1. It specifies the external system behavior.

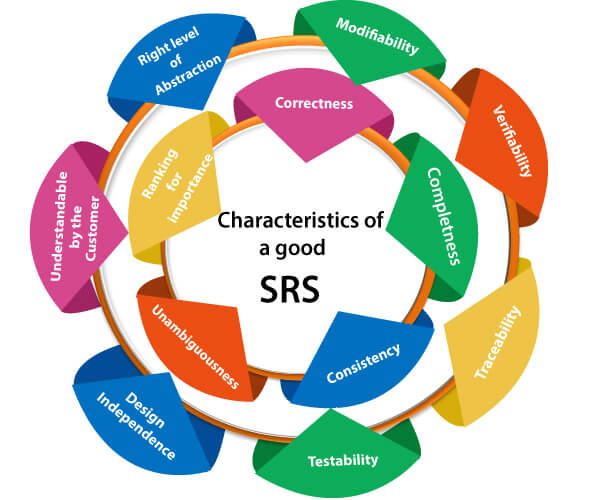
2. It specifies constraints on the implementation.

3. It is easy to change.

4. It serves as reference tool for system maintainers.

5. It record forethought about the lifecycle of the system.

6. It characterizes acceptable response to undesired events.



# **Product Functions:**

* This application aims to provide users security while scanning QR codes so that users can save themselves from the scams.
* User will need to scan a QR code that’ll be redirected to the associated link.
* Using our app users will get a security score of that link.
* Output screen will show a score out of 10 that’ll define the security of the website.
* One can create his/her profile and that will be verified by the moderators. The user has to submit documents to get validated.
* Once the validation is completed, verified users can put up a request to enlist a link as unsafe.

## **User Diagram:**

SCAN QR

USERS

LOG IN

VERIFIED USERS

# **Scanning:**

**Introduction:** This use case describes how a user scans a QR Code.

**Actors:** Users

**Pre-Conditions:** None

**Post-Conditions:** A security score of for the link associated with the QR code will be displayed and after that user can move forward.

**Basic Flow:** This use case initiates its function when actor wants to use the application to scan a QR Code.

* Using normal phone camera, user scans the QR Code s/he wants to.
* According to the link’s authenticity, app displays user a security score for the QR link s/he wants to visit.

**Alternative Flow:** None

**Special Requirements:** None

**Use Case Relationships:**  COMMUNICATES

# **Verified User Registration:**

**Introduction:** This use case describes how a user can get upgraded to a verified user(Moderator).

**Actors:** Users

**Pre-Conditions:** None

**Post-Conditions:** If the use case is successful, the actor is upgraded to a verified user. If not, the system state is unchanged.

**Basic Flow:** This use case initiates its function when the actor wants to put a request for marking an QR code as unsafe.

* Application requests the actor to submit his/her relevant documents.
* Application validates the documents and if those are found to be correct then it allows the actor to be logged in as a verified user.

**Alternative Flows:**

Invalid documents: If due to some reason the basic flow gets interrupted, the actor gets 2 options. First, s/he can go back to the basic page for normal users. Second, can opt for another request to get his/her request verified.

**Special Requirements:** None

**Use Case Relationships:** EXTENDED

# **Log in:**

**Introduction:** This use case describes how a verified user logs into the QR Code scanner app.

**Actors:** Verified Users

**Pre-Conditions:** Relevant documents must be authenticated.

**Post-Conditions:** If the use case is successful, the actor is logged into the system. If not, the system state is unchanged.

**Basic Flow:** This use case initiates its function when the actor wants to login to the application.

* Application requests the actor to enter his/her username and password.
* Actor enters his/her name and password.
* Application validates the inputs and if those are found to be correct then it allows the actor to be logged in.

**Alternative Flows:**

Invalid Username or Password: If due to some reason the basic flow gets interrupted, the actor gets 2 options. First, s/he can go back to the basic page for normal users. Second, can opt for forgot password/username.

**Special Requirements:** None

**Use Case Relationships:**  INCLUDES

# **Request:**

**Introduction:** This use case describes how a verified user requests for a QR Code to be unsafe.

**Actors:** Verified Users

**Pre-Conditions:** None

**Post-Conditions:** If the use case is successful, the request will be approved. If not, the system state is unchanged.

**Basic Flow:** This use case initiates its function when the actor wants to put up a request for a QR code.

* Application requests the actor to input the QR code, reason to mark it as unsafe, proof of the QR to be vulnerable.
* Actor fills the required input fields.
* Application validates the inputs and if those are found to be correct then it approves the request and make the changes in database.

**Alternative Flows:**

Invalid Request: If due to some reason the basic flow gets interrupted, the actor gets 2 options. First, s/he will receive that the request is invalid. Second, s/he can opt for another request.

**Special Requirements:** None

**Use Case Relationships:**  INCLUDES

## **Data Flow Diagram:**

A Data Flow Diagram is a graphical representation of the "flow" of data through an information system, which models the process characteristics. A DFD is frequently used as a basic step in the development of a system overview that can later be refined.

• DFD can also be used for the visualization of data processing.

• A data flow diagram (DFD) depicts the types of data that will be input to and output from the system, as well as where the data will come from and go to, and where it will be kept. It doesn't show information about process timing or whether processes will run in order or in parallel.

