RPADML: REALTIME PHISHING ATTACK DETECTION USING MACHINE LEARNING

*An industrial oriented major project report*

Submitted By

**ABDUL GHANI (Regd. No: 15W91A0501)**

**B. SINDHUJA (Regd. No: 15W91A0509)**

**K.SAI AMULYA (Regd. No: 15W91A0523)**

Under the Esteemed Guidance of

**DR. P. KIRAN KUMAR REDDY**

Professor, CSE

*to*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

In partial fulfillment of the requirements for award of degree of

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE & ENGINEERING**

2020



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY (MRET - W9)**

(Sponsored by Malla Reddy Educational society)

(Accredited by NBA, Permanent Affiliated to JNTU, Hyderabad)

Maisammaguda, Dhulapally post, Secunderabad-500014.

**DECLARATION**

We hereby declare that the project entitled “**RPADML: REALTIME PHISHING ATTACK DETECTION USING MACHINE LEARNING**” submitted to Malla Reddy Institute of Engineering and Technology (MRET-W9), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in Computer Science & Engineering is a result of original industrial oriented project done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

1. **ABDUL GHANI - (15W91A0501)**
2. **B. SINDHUJA - (15W91A0509)**
3. **K.SAI AMULYA - (15W91A0523)**

MALLA REDDY INSTITUTE OF ENGINEERING & TECHNOLOGY

(Sponsored by Malla Reddy Educational Society)

Accredited by NBA, Affiliated to JNTU, Hyderabad

Maisammaguda, Dhulapally (post via Hakimpet), Sec’Bad-500 014.

Phone: 040-65969674, Cell: 9348161223

Department of Computer Science and Engineering

PROJECT CERTIFICATE

This is to certify that this is the bonafide record of the project titled **“RPADML: REALTIME PHISHING ATTACK DETECTION USING MACHINE LEARNING”** is submitted by **ABDUL GHANI (15W91A0501), B. SINDHUJA (15W91A0509), K.SAI AMULYA (15W91A0523)** of B.Tech in the partial fulfillment of the requirements for the degree of **Bachelor of Technology in Computer Science and** **Engineering**, Dept. of Computer Science & Engineering and this has not been submitted for the award of any other degree of this institution.

**HOD**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

First and foremost, I am grateful to the Principal **Dr. M. ASHOK**, for providing me with all the resources in the college to make my project a success. I thank him for his valuable suggestions at the time of seminars which encouraged me to give my best in the project.

I would like to express my gratitude to **Dr. ANANTHA RAMAN,** Head of the Department, Department of Computer Science and Engineering for his support and valuable suggestions during the dissertation work

I offer my sincere gratitude to my project -coordinator **Mr. B. SUNIL** and internal guide **DR. P. KIRAN KUMAR REDDY** Assistant Professor of Computer Science and Engineering department who has supported me throughout this project with their patience and valuable suggestions.

I would also like to thank all the supporting staff of the Dept. of CSE and all other departments who have been helpful directly or indirectly in making the project a success.

I am extremely grateful to my parents for their blessings and prayers for my completion of project that gave me strength to do my project.

**ABDUL GHANI - (15W91A0501)**

**B. SINDHUJA - (15W91A0509)**

**K.SAI AMULYA - (15W91A0523)**

**INDEX**

**Abstract………………… I**

**List of Figures………….. II**

**List of Tables……………… III**

**List of Screens…………… IV**

**Symbols & Abbreviations…. V**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **CONTENT** | **PAGENO** |
| **1** | **1 Introduction**   * 1. **Motivation**   2. **Problem definition**   3. **Objective of Project**   4. **Limitations of Project**   5. **Organization of Documentation** |  |
| **2** | **2 LITERATURE SURVEY**   * 1. **Introduction**   2. **Existing System**   3. **Disadvantages of Existing system**   4. **Proposed System**   5. **Conclusion** |  |
| **3** | 1. **Analysis**     1. **Introduction**    2. **Software Requirement Specification**       1. **User requirement**       2. **Software requirement**       3. **Hardware requirement**    3. **Content diagram of Project**    4. **Algorithms ad Flowcharts**    5. **Conclusion** |  |
|  | 1. **DESIGN**     1. **Introduction**    2. **DFD / ER / UML diagram (any other project diagrams)**    3. **Module design and organization**    4. **Conclusion** |  |
|  | 1. **IMPLEMENTATION & RESULTS**     1. **Introduction**    2. **Explanation of Key functions**    3. **Method of Implementation**   **5.2.1 Forms**  **5.2.2 Output Screens**  **5.2.3 Result Analysis**   * 1. **Conclusion** |  |
|  | 1. **TESTING & VALIDATION**     1. **Introduction**    2. **Design of test cases and scenarios**    3. **Validation**    4. **Conclusion** |  |
|  | 1. **CONCLUSION**     1. **Project Conclusion**    2. **Future enhancement** |  |
|  | 1. **REFERENCES**    1. **Referred Books**    2. **Referred sites** |  |

**INTRODUCTION**

**1. INTRODUCTION**

* 1. **MOTIVATION:**

Social Engineering based attack leverages psychological manipulation of people, tricked into performing actions or disclosing confidential information. Phishing is one of the more known social engineering attack and aims at exploiting weaknesses in system processes caused by users’ behavior. Indeed, a system can be secure enough against password theft(e.g. the client-server communication channel is encrypted),but nothing can be done against a naive user threatening the security of the system by revealing her/his password to a fake Web site reached, for example, via an email-embedded HTTP link.

* 1. **PROBLEM DEFINITION**

The evolution of phishing techniques, the switch over malware-based phishing campaigns together with the massive spread of mobile devices for business or personal use, has clearly expanded the threat posed by phishing, especially towards the mobile world, along with people underestimating the mobile cyber-threats and neglecting even basic security measures. 17% of all Android apps (nearly one million total) were actually malware in disguise and 36% of all mobile apps, even not malicious by design, did inadvertently harmful things like tracking user behavior. Phishing attack detection in desktop environment has been widely investigated in literature. On the contrary, despite the severity of the phishing problem in the mobile ecosystem, we noticed a lack of literature on the phishing attack detection for mobile device and, even extending the scope into the mobile malware detection, very few works have converged towards solutions running on mobile devices and related to malware-based phishing.

**1.3 OBJECTIVE OF THE PROJECT:**

**PURPOSE:**

 Many of the approaches proposed in literature, regardless of their effectiveness, still have a strong verticality and focus on specific aspects such as: attack techniques; existing security context; systems and protocols used to capture data; methodological approaches used for phishing detection (black lists, heuristics, machine learning, etc.); devices on which deploy the developed solution. To the best of our knowledge, none of the solutions proposed so far, shown a unified approach across different environments (such as mobile and desktop) and across subsets of the above mentioned aspects. This is mainly due to the fact that, each solution often needs a number of prerequisites that are: i) difficult to remove, in case some detection technique is covered by other third party software (e.g. antivirus); ii) hard to adapt to other contexts, enabling the exploitation of the same detection technique for other threats (e.g. botnets); iii) and burdensome to merge in a single tool that can exploit different targets, sources or approaches. Therefore, in this project, we propose a unified reference model and present the Real-time Phishing Attack Detection using Machine Learning (RPAD-ML) framework that implements it for the Android platform. Our unified reference model has three main pillars:

* Context and platform independence. The approach must realize a proxy-based architecture, natively interposed to any type of traffic to and from the Internet, able to intercept every network pattern and deployable both on mobile and desktop devices;
* Adaptability and multimodality/multi functionality. Each detection technique implemented (blacklists, heuristics, machine learning, etc.) must be enabled/disabled as needed (e.g. when the same feature is already implemented by third party software such as the antivirus client) and must be effectively extended to other malware detection contexts (e.g. botnets);
* Scalability. The approach must be able to manage different rates and types of connections, required by user applications (e.g. the proposed solution must not represent a bottleneck in case of high speed and low latency network connections) and must be able to move the computational intensive client tasks remotely in a cloud platform, in order to reduce workloads.

**1.4 LIMITATIONS OF THE PROJECT:**

The Model is pre-compiled. Transferring the data back to the neural net is not possible with this approach. As it would increase the runtime of the application and also requires a server maintenance.

**1.5 ORGANIZATION OF DOCUMENTATION:**

In this project documentation we have initially put the definition and objective of the project as well as the design of the project which is followed by the implementation and testing phases. Finally the project has been concluded successfully and also the future enhancements of the project were given in this documentation. All the screen of the system designed with the view to provide the user with easy operations in simple and efficient way, minimum key stroke possible instructions and important information is emphasized on the screen. Almost every screen is provided with no error and important message and selection facilitates. Each screen assigned to make it as much user friendly as possible by using interaction procedure.

**2. LITERATURE SURVEY**

**2. LITERATURE SURVEY**

**2.1 INTRODUCTION**

Literature survey is the most important step in the software development process. The huge literature on phishing detection techniques is almost completely oriented toward methodologies tailored for desktop/laptop environments (see [4] for an extensive survey). Considering the literature related to the mobile environment that, as mentioned before, is not exhaustive, we can distinguish between work on "traditional" phishing [8-13] (resulting in no particular research trend) and work on Malware-based phishing detection [14]. The lack of literature on this specific subject is a consequence of a variety of reasons spanning from (a) the inevitable convergence of the same threat to any Internet enabled device (sometimes addressed as the Internet of Things), (b) the difficulty of implementing a performing security solution on a mobile device, (c) the absence of basic security settings on mobile devices (e.g. antivirus and firewall) and (d) the increased variability of attack vectors directed towards mobile devices.

**EXISTING SYSTEM:**

A. "Traditional" phishing detection concerning the specific subject of phishing detection on mobile devices, in [6] is proposed a phishing detection taxonomy for mobile environment, trying to depict all possible scenario of phishing attacks and related countermeasures. Leaving aside all possible attacks related to very specific vectors (e.g. SMS, Bluetooth and Vishing), the paper stresses the lack of solutions dedicated to mobile devices other than black/white lists. A risk assessment on mobile platforms has been proposed in [9], where a study conducted on 85 web sites and 100 mobile applications discovered that web sites and applications regularly ask users to type their passwords into contexts that are vulnerable to spoofing. The implementation of sample phishing attacks on the Android and iOS platforms demonstrated that attackers can spoof legitimate applications with high accuracy, suggesting that the risk of phishing attacks on mobile platforms is greater than it has previously been appreciated. In [10] the authors proposed a proof-of-concept defense against spoofing and phishing attacks, exploiting the full-screen touch-based user interface. The developed framework logs user keystrokes and warns the user when potentially sensitive information is about to be entered while running an untrusted application. A phishing scheme for mobile phones is presented in [11], trying to exploit OCR text extraction tools, in order to verify the legitimacy of a website, comparing the text extracted from a login form with the corresponding second-level domain name (SLD). This stems from the assumption that most well-known enterprises use brand name as the SLD of their official websites which is also used, as an image, within their login forms. However, the presented workflow uses also white lists and heuristics to reduce false positives or to increase the efficiency of the overall system (e.g. legitimate websites always use domain names as verification of their identities while phishers are likely to list IP address in URL to disguise their fake identities). In [12] is proposed a study on the threat posed by the notification systems, typical of smartphones, suggesting also a design principle for a OS-controlled framework for the management of the notification systems used by all the Apps installed on the mobile devices. A further research was aimed at identifying anti-phishing techniques, using the visual similarity analysis, in order to protect mobile users using the Internet via public Wi-Fi hotspots [13]. B. Malware-based phishing detection With regards to phishing as a mean for spreading malware (malware-based phishing), it was possible to recover some additional scientific material related to malware detection for mobile environments through network analysis techniques [14- 17]. In this area, however, regardless of the selected method among the most common phishing detection strategies (e.g black/white lists, heuristics, visual similarity matching and machine learning algorithms), it is important to highlight that none of the proposed solutions is implemented in a software module suitable for mobile devices. Further, only in few cases the anomaly detection techniques used were deployed directly on the device. Most of the systems use a server-side component to send the observed data or to perform the learning process offline and plant the learned models back to the devices for the detection process. The main difficulties seem to lie in capturing the traffic generated by the mobile devices directly from the device itself that, in the solutions analyzed, is emulated with the tools usually available on desktop environments, such as Tcpdump and Wireshark. The few examples in which it was made a smart engine on board the device [18], continue to have, however, a strong verticality on the phishing attack being investigated (e.g. application update attack) or on the type of technology used. Hence, it's clear that modern mobile devices, although highlighting high performance levels of interaction with the Internet, still pose great obstacles for the implementation of algorithms and technologies available in traditional detection systems, due to their reduced processing capabilities.

**DRAWBACKS OF EXISTING SYSTEM:**

* **Although, it protects user from entering into phishing sites on Chrome Browser, it fails to do so on other browsers.**
* **It cannot detect newly created phishing sites (Someone has to report it)**
* **Doesn’t work in other mobile browsers**
* **Doesn’t scan all the links opened by the device(Mobile)**
* **Cannot intercept all network traffic like GET requests.**

**PROPOSED SYSTEM:**

To overcome the drawbacks of existing system, we propose a framework RPAD-ML (REALTIME PHISHING ATTACK DETECTION USING MACHINE LEARNING).

It is based on Machine learning classification model and can detect phishing attacks in mobile. The machine learning model is kept on server and the client device is provided with an app which can communicate with this ML server and intercept the web pages and apps.

**FEATURES OF PROPOSED SYSTEM:**

1. **FUNCTIONAL CAPABILITIES:**    The ultimate aim of this project is to detect phishing attacks in real-time. The app in the client device keeps on checking with machine learning server for any maliciousness in the accessed site/app.
2. **PERFORMANCE LEVEL:**   The current Machine learning server is setup on 1GB RAM Virtual Private Server. It gives the response within 500-1000ms. At the client side, it takes 1-2 seconds to detect whether a site is phishing or not.
3. **DATA STRUCTURES:**  The data in this project are maintained in the CSV form. It provides easy access to the user.
4. **SAFETY:**No data loss occurs in this system.
5. **RELIABILITY:**          We assure that the project is completely authenticated in order to enhance security and corruptions of database as well as the software. The person is given access only if he/she has a valid username and password.
6. **QUALITY:**     The project is developed with the help of Android Studio, Weka, PyCharm software which meets the requirement of the user, the project is checked whether the phases individually have a served its purpose.

**CONCLUSION**

**From the literature survey we can conclude that we have overcome the drawbacks of existing system and we introduce new technology which would be helpful to use. In proposed system to overcome that problem we create our project by providing sharing through e mail. This helps user to reduce time as well as work.**

**3. ANALYSIS**

**3. ANALYSIS**

**3.1 INTRODUCTION**

The purpose of this SRS document is to mark and secure the required Uniform Resource locators (URL) and its functionalities for Intelligent Network Backup Tool. The SRS will define how our team and the client conceive the final product and the characteristics or functionality it must have. This document also makes a note of the optional requirements like sharing, renaming which we wish to send URL path but are not mandatory for the functioning of the project.

This phase appraises the needed requirements for the Bookmarks for a systematic way of evaluating the requirements several processes are involved. The first step involved in analyzing the requirements of the system is having a protection to the URL’s without accessing by others within the system for a reliable investigation and all the case are formulated to better understand the analysis of the Bookmarks (dataset).

**Document Conventions:**

The convention used in the size of fonts remains the same as for other documents in the project. The section headings have the largest font of 14, subheadings have a font size of 12(bold), and the text is on font 12. The priorities of the requirements are specified with the requirement statements.

**Intended Audience and Reading Suggestions:**

This document is intended for project developers, managers, users, testers and documentation writers. This document aims at discussing design and implementation constraints, dependencies, system features, external interface requirements and other non-functional requirements.

**Identification of Needs:**

The foremost and important necessity for a business firm or an organization is to know how they are performing in the market and parallely they need to know how to overcome their competitors in the market.

To do so we need to analysis our data based on all the available factors. The system requirements for the project to be accomplished are:

**3.2 SOFTWARE REQUIREMENT SPECIFICATION:**

**3.2.1 FUNCTIONAL REQUIREMENTS:**

**User:** User can register on the web application and access the resources available in the application. He can view, add, modify and send bookmark, and also update the profile..

**3.2.2 SOFTWARE REQUIREMENTS:**

**Development & Usage**

Server Operating System : Linux 16.04 LTS

Web Server : Python Flask 1.0.2

Backend Language : Python 3.7

Database : MongoDB 4.0

Modeling Tool : Weka 3.8.3

Client Operating System : Android 5.0+

**3.2.3 HARDWARE REQUIREMENTS:**

**Development & Usage:**

Processor : Intel I3 or greater

RAM : 8GB

Hard Disk : 50GB

**3.3 CONTENT DIAGRAM OF THE PROJECT**

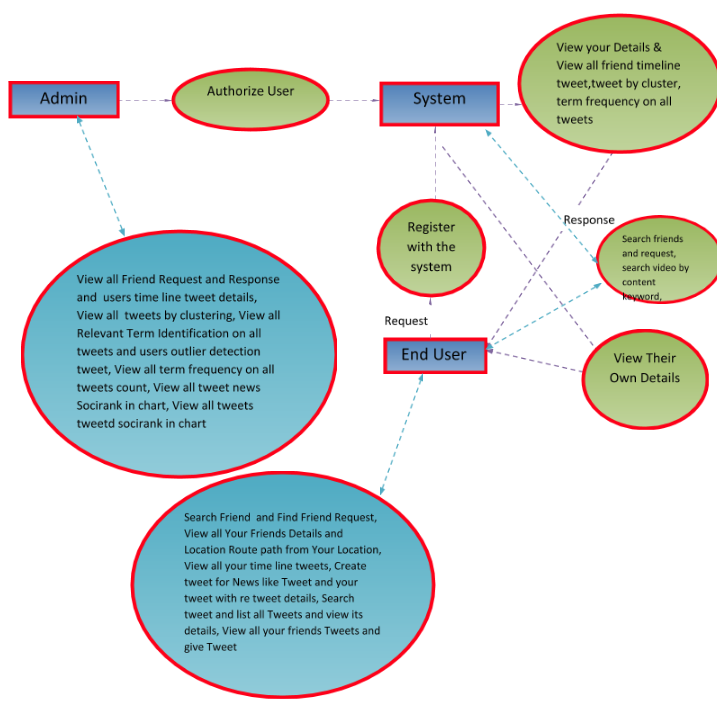
****

Fig No.3.3.1Content Diagram

**3.4 ALGORITHMS AD FLOWCHARTS**

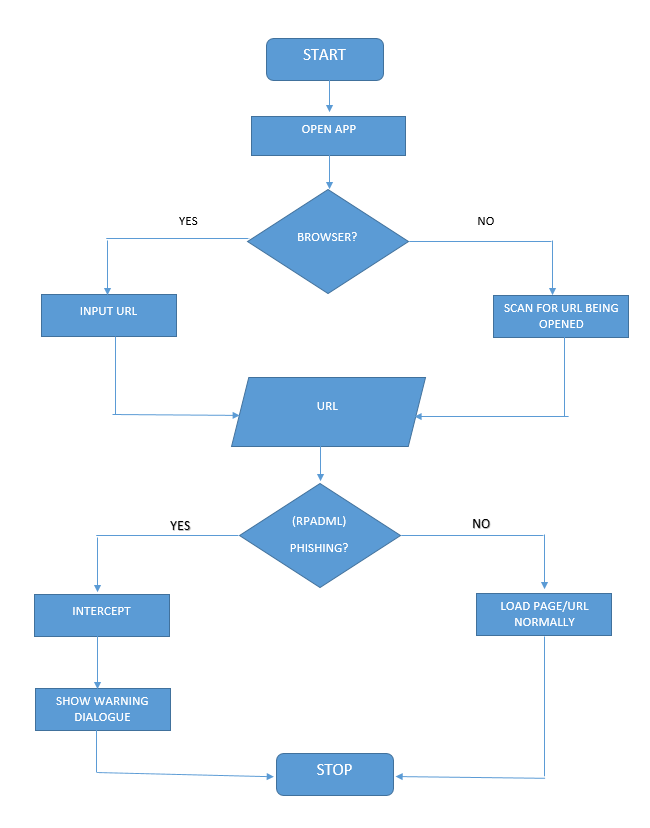


Fig No.3.4.1 Flow Chart - USER

**3.5 CONCLUSION:**

The analysis tells as the requirement specifications of the project and the required algorithms, flowcharts of the project. The functional requirements specify the functionality and functional requirements were as the software requirements tell the required software and supporting files to process the data. The hardware requirements tell about the hardware components required to run the software. Flowcharts describe the flow of the total process.

**DESIGN**

**DESIGN**

**4.1 INTRODUCTION**

System Design is the process or art of defining the architecture components, modules, interfaces and data for a system to satisfy specified requirements. One should see as the applications of the systems theory to product development.

**Architecture Flow**

The architecture represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tiers separately using 3 layers called presentation layer, business logic layer and data link layer.

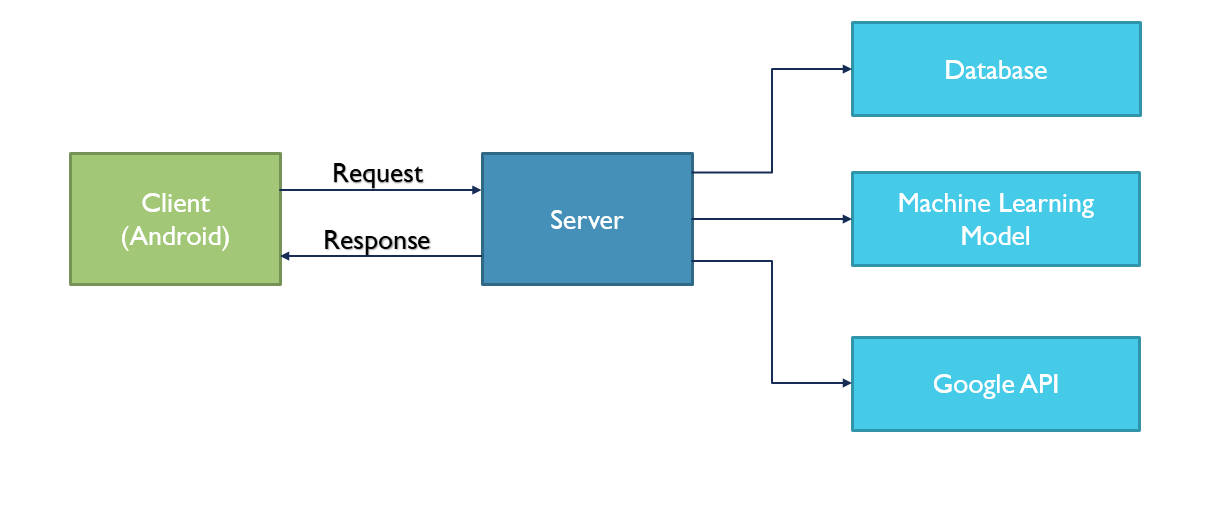


Fig No.4.1.1 Architecture

**4.2 UML DIAGRAMS**:

**Unified Modelling Language:**

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

∙ **User Model View**

i. This view represents the system from the user’s perspective.

ii. The analysis representation describes a usage scenario from the end-users perspective.

∙ **Structural model view**

i. In this model the data and functionality are arrived from inside the system.

ii. This model view models the static structures.

∙ **Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

∙ **Implementation Model View**

In this the structural and behavioural as parts of the system are represented as they are

to be built.

∙ **Environmental Model View**

In this the structural and behavioural aspects of the environment in which the system

is to be implemented are UML is specifically constructed through two different domains they are:

∙ UML Analysis modelling, this focuses on the user model and structural model views of the system represented.

UML is specifically constructed through two different domains they are:

∙ UML Analysis modelling, this focuses on the user model and structural model views of the system.

∙UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

UML is specifically constructed through two different domains they are:

∙ UML Analysis modelling, this focuses on the user model and structural model views of the system.

∙UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behaviour of the system from external point of view. Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

**4.2.1 Use Case Diagram**

A use case diagram at its simplest is a representation of a user’s interaction with the system and depicting the specifications of the use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

They provide the simplified and graphical representation of what the system must actually do. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

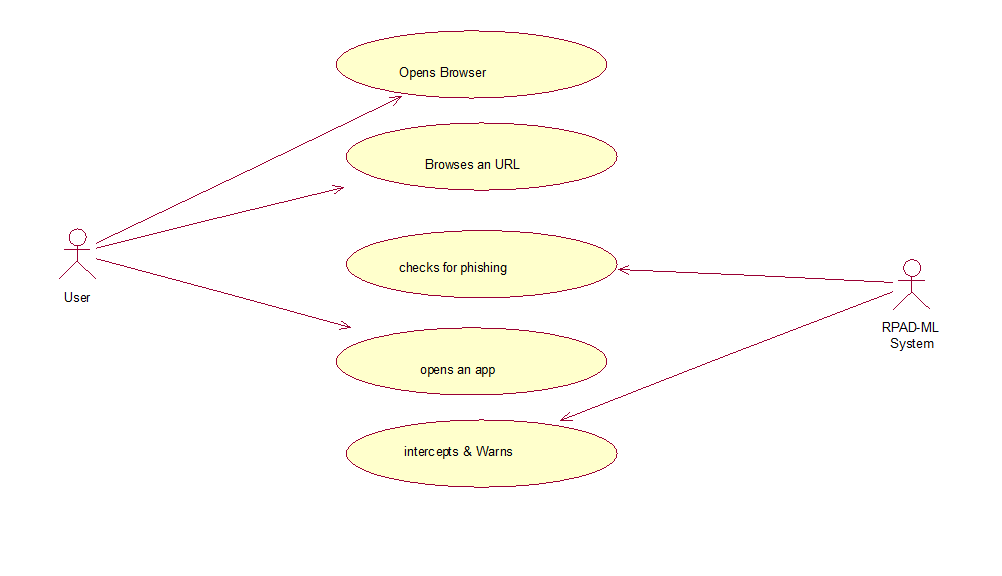
****

Fig No.4.2.1.1 Use Case Diagram

**4.2.2 Sequence Diagram:**

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

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 shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

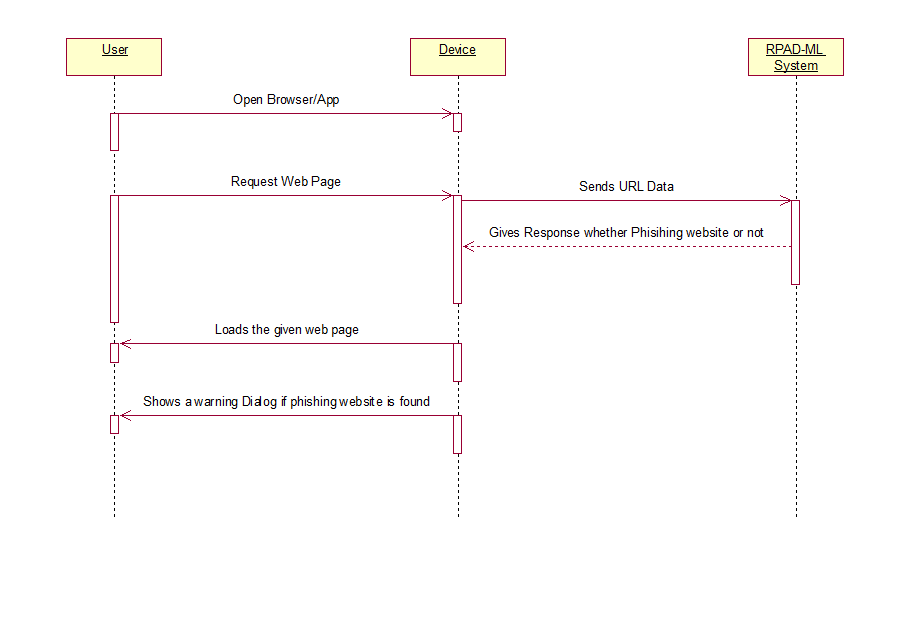


Fig No.4.2.2.1 Sequence Diagrams

**4.2.3 Class Diagram**

In software Engineering, a class diagram  in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects translating the models into programming code. Class diagrams can also be used for data modeling the classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling

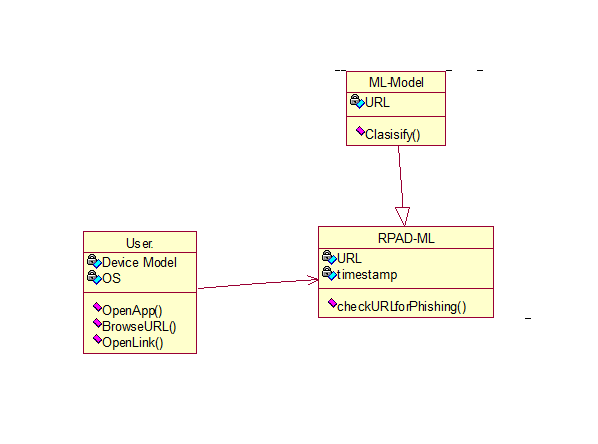


Fig No.4.2.3.1Class Diagram

**4.3MODULES**

Here we are using two modules. The name and description of each module are given below

**USER**

In this module, the user can open an app or browser and then input an URL if the opened app is a browser. The user can load web pages and open web links in apps.

**RPAD-ML**

In this module, the web page/link to be loaded is sent as data to the server. Where this server having machine learning model at its backend checks whether the requested URL is phishing or not. And gives back the result to the user.

**Android App**

It acts as an interface for the user and machine learning server. All the website URLs/Links to be opened are first scanned by the app and then sent to the machine learning server. The server then classifies and gives a result. Based on the response received from the server the app intercepts the web page/app which is loading a possible phishing site/URL.

**4.4 CONCLUSION:**

By design content we can describe the required modules and different diagrams. Using diagrams what are the communications present and we can also understand the project easily. Modules help as in designing the project to fulfill the user requirements

**5. IMPLEMENTATION & RESULTS**

**5. IMPLEMENTATION & RESULTS**

**5.1 INTRODUCTION:**

Implementation is the carrying out, execution, or practice of a plan, a method, or any design, idea, model, specifications, standard or policy for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to actually happen.

For an implementation process to be successful, many tasks between different departments need to be accomplished in sequence. Companies strive to use proven methodologies and enlist professional help to guide them through the implementation of a system but the failure of many implementation processes obtain stems from the lack of accurate planning in the beginning stage of project.

**Java Technology**

Initially the language was called as “oak” but it was renamed as “Java” in 1995. 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.

 Finally, Java is to Internet programming where C was to system programming.

**JVM**

**Java byte Code**

**Java Source**

# 

Fig No.5.1.1 Java Compiler

**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.

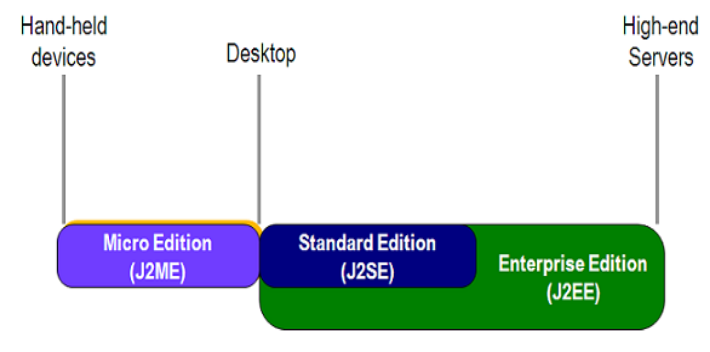


Fig No.5.1.2Java Architecture

**Java Database Connectivity**

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

Simply put, JDBC makes it possible to do three things:

* Establish a connection with a database
* Send SQL statements
* Process the results

**Two-tier and Three-tier Models**

The JDBC API supports both two-tier and three-tier models for database access.

In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the database as the server. The network can be an Intranet, which, for example, connects employees within a corporation, or it can be the Internet.

|  |  |  |
| --- | --- | --- |
| **JAVA** |  |  |
| **Application** | **Client machine** |  |
|  |  |  |
|  | **DBMS-proprietary protocol** |  |
| **JDBC** |  |
|  |  |
|  |  |  |

**DBMS**

**Database server**

Fig No.5.1.3 JDBC API

**Eclipse IDE**

Eclipse is an open-source software framework written primarily in Java. In its default form it is an Integrated Development Environment (IDE) for Java developers, consisting of the Java Development Tools (JDT) and the Eclipse Compiler for Java (ECJ). Users can extend its capabilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Language packs are available for over a dozen languages.

**Architecture**

The basis for Eclipse is the Rich Client Platform (RCP). The following components constitute the rich client platform:

OSGI - a standard bundling framework

* Core platform - boot Eclipse, run plug-ins
* The Standard Widget Toolkit (SWT) - a portable widget toolkit
* Face - viewer classes to bring model view controller programming to SWT, file buffers, text handling, text editors

Eclipse's widgets are implemented by a widget toolkit for Java called SWT, unlike most Java applications, which use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also leverages an intermediate GUI layer called Face, which simplifies the construction of applications based on SWT.

The Eclipse SDK includes the Eclipse Java Development Tools, offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards. The Visual Editor project allows interfaces to be created interactively, hence allowing Eclipse to be used as a RAD tool.

**Python Technology**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.[26] Van Rossum led the language community until stepping down as leader in July 2018.[27][28]

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. It also has a comprehensive standard library.[29]

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software[30] and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

**5.2 METHOD OF IMPLEMENTATION**

**Activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:background="#FFF"

tools:context=".MainActivity">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="300dp"

android:background="@drawable/bg1" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center">

<TextView

android:id="@+id/ProtectionStatus"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!"

android:textAlignment="center"

android:textColor="@color/black"

android:textSize="30dp"

/>

<Button

android:id="@+id/isEnabled"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Enable"

android:background="@drawable/gradient1"

android:textColor="#FFFFFF"

android:layout\_marginTop="10dp"

android:onClick="checkEnabled"/>

<!-- <Button

android:id="@+id/buttonCreateWidget"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="Create Floating Widget"

android:onClick="requestPermission"/>-->

</LinearLayout>

<!-- <Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Disable"

android:background="@color/colorAccent"/>-->

</LinearLayout>

**layout\_floating\_widget.xml**

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<RelativeLayout

android:id="@+id/relativeLayoutParent"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="UselessParent">

<!-- this is the collapsed layout -->

<RelativeLayout

android:id="@+id/layoutCollapsed"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:visibility="visible">

<ImageView

android:id="@+id/collapsed\_iv"

android:layout\_width="70dp"

android:layout\_height="70dp"

android:layout\_marginTop="8dp"

android:src="@drawable/ic\_launcher\_background" />

<ImageView

android:id="@+id/buttonClose"

android:layout\_width="25dp"

android:layout\_height="25dp"

android:layout\_marginLeft="50dp"

android:src="@drawable/ic\_close\_btn" />

</RelativeLayout>

<!-- this is the expanded layout -->

<LinearLayout

android:id="@+id/layoutExpanded"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/redgradient"

android:orientation="horizontal"

android:padding="8dp"

android:visibility="gone">

<ImageView

android:id="@+id/buttonSimplifiedCodingExpanded"

android:layout\_width="80dp"

android:layout\_height="80dp"

android:src="@drawable/ic\_launcher\_background"

tools:ignore="ContentDescription" />

<LinearLayout

android:id="@+id/buttonSimplifiedCoding"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="15dp"

android:paddingRight="15dp"

android:paddingTop="8dp"

android:text="This site is Unsafe!"

android:textAlignment="center"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"

android:textColor="#ffffff"

android:textStyle="bold" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Probable Phishing Site!"

android:textAlignment="center"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"

android:textColor="#ffffff"

android:textStyle="bold" />

<TextView

android:id="@+id/url"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="http://www.google.com/shdshgkshk/sgjhsgh.php"

android:textAlignment="center"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Small"

android:textColor="#ffffff"

/>

</LinearLayout>

</LinearLayout>

</RelativeLayout>

</FrameLayout>

**MainActivity.java**

package com.techglows.rpadml;

import android.accessibilityservice.AccessibilityService;

import android.accessibilityservice.AccessibilityServiceInfo;

import android.content.Context;

import android.content.Intent;

import android.content.pm.ServiceInfo;

import android.net.Uri;

import android.os.Build;

import android.provider.Settings;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.accessibility.AccessibilityManager;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import java.util.List;

public class MainActivity extends AppCompatActivity {

public boolean enabled;

private static final String TAG = "RPAD\_MainActivity";

public Button btn;

public TextView status;

public final static int REQUEST\_CODE = 676;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btn = findViewById(R.id.isEnabled);

status = findViewById(R.id.ProtectionStatus);

accessibilityEnabledStatusCheck();

//findViewById(R.id.buttonCreateWidget).setOnClickListener(this);

}

public void requestPermission(View v) {

if (Build.VERSION.SDK\_INT < Build.VERSION\_CODES.M) {

startService(new Intent(MainActivity.this, FloatingViewService.class));

finish();

} else if (Settings.canDrawOverlays(this)) {

startService(new Intent(MainActivity.this, FloatingViewService.class));

finish();

} else {

askPermission();

Toast.makeText(this, "You need System Alert Window Permission to do this", Toast.LENGTH\_SHORT).show();

}

}

public void accessibilityEnabledStatusCheck(){

if(enabled = isAccessibilityServiceEnabled(getApplicationContext(), MyAccessibilityService.class)){

//Enabled

btn.setText("Disable");

btn.setBackground(getResources().getDrawable(R.drawable.redgradient));

status.setText("You're Protected!");

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M && !Settings.canDrawOverlays(this)) {

askPermission();

}

}else{

//disabled

//show Enable btn

btn.setText("Enable");

btn.setBackground(getResources().getDrawable(R.drawable.gradient1));

status.setText("You're NOT Protected!");

}

}

public void checkEnabled(View view){

if(enabled){

//permission given

Log.d(TAG,"accessibility permission given");

getAccessibilityPermissions();

}else{

//not given show UI to give permission

Log.d(TAG,"accessibility permission NOT given");

getAccessibilityPermissions();

}

}

public void getAccessibilityPermissions(){

Log.d(TAG,"getting accessibility permissions!");

Intent intent = new Intent(android.provider.Settings.ACTION\_ACCESSIBILITY\_SETTINGS);

startActivityForResult(intent, 0);

}

public static boolean isAccessibilityServiceEnabled(Context context, Class<? extends AccessibilityService> service) {

AccessibilityManager am = (AccessibilityManager) context.getSystemService(Context.ACCESSIBILITY\_SERVICE);

List<AccessibilityServiceInfo> enabledServices = am.getEnabledAccessibilityServiceList(AccessibilityServiceInfo.FEEDBACK\_ALL\_MASK);

for (AccessibilityServiceInfo enabledService : enabledServices) {

ServiceInfo enabledServiceInfo = enabledService.getResolveInfo().serviceInfo;

if (enabledServiceInfo.packageName.equals(context.getPackageName()) && enabledServiceInfo.name.equals(service.getName()))

return true;

}

return false;

}

private void askPermission() {

Intent intent = new Intent(Settings.ACTION\_MANAGE\_OVERLAY\_PERMISSION,

Uri.parse("package:" + getPackageName()));

startActivityForResult(intent, REQUEST\_CODE);

}

@Override

public void onRestart()

{

super.onRestart();

// after resuming the activity

accessibilityEnabledStatusCheck();

}

}

**Server Side:**

**Index.py**

# -\*- coding: utf-8 -\*-

#importing libraries

from sklearn.externals import joblib

import inputScript

#load the pickle file

classifier = joblib.load('final\_models/rf\_final.pkl')

#input url

print("enter url")

url = input()

#checking and predicting

checkprediction = inputScript.main(url)

prediction = classifier.predict(checkprediction)

# print(prediction)

# x = prediction.tolist()

#print(type(prediction))

print(prediction)

**InputScript.py**

# -\*- coding: utf-8 -\*-

import regex

from tldextract import extract

import ssl

import socket

from bs4 import BeautifulSoup

import urllib.request

import whois

import datetime

def url\_having\_ip(url):

#using regular function

# symbol = regex.findall(r'(http((s)?)://)((((\d)+).)\*)((\w)+)(/((\w)+))?',url)

# if(len(symbol)!=0):

# having\_ip = 1 #phishing

# else:

# having\_ip = -1 #legitimate

#return(having\_ip)

return 0

def url\_length(url):

length=len(url)

if(length<54):

return -1

elif(54<=length<=75):

return 0

else:

return 1

def url\_short(url):

#ongoing

return 0

def having\_at\_symbol(url):

symbol=regex.findall(r'@',url)

if(len(symbol)==0):

return -1

else:

return 1

def doubleSlash(url):

#ongoing

return 0

def prefix\_suffix(url):

subDomain, domain, suffix = extract(url)

if(domain.count('-')):

return 1

else:

return -1

def sub\_domain(url):

subDomain, domain, suffix = extract(url)

if(subDomain.count('.')==0):

return -1

elif(subDomain.count('.')==1):

return 0

else:

return 1

def SSLfinal\_State(url):

try:

#check wheather contains https

if(regex.search('^https',url)):

usehttps = 1

else:

usehttps = 0

#getting the certificate issuer to later compare with trusted issuer

#getting host name

subDomain, domain, suffix = extract(url)

host\_name = domain + "." + suffix

context = ssl.create\_default\_context()

sct = context.wrap\_socket(socket.socket(), server\_hostname = host\_name)

sct.connect((host\_name, 443))

certificate = sct.getpeercert()

issuer = dict(x[0] for x in certificate['issuer'])

certificate\_Auth = str(issuer['commonName'])

certificate\_Auth = certificate\_Auth.split()

if(certificate\_Auth[0] == "Network" or certificate\_Auth == "Deutsche"):

certificate\_Auth = certificate\_Auth[0] + " " + certificate\_Auth[1]

else:

certificate\_Auth = certificate\_Auth[0]

trusted\_Auth = ['Comodo','Symantec','GoDaddy','GlobalSign','DigiCert','StartCom','Entrust','Verizon','Trustwave','Unizeto','Buypass','QuoVadis','Deutsche Telekom','Network Solutions','SwissSign','IdenTrust','Secom','TWCA','GeoTrust','Thawte','Doster','VeriSign']

#getting age of certificate

startingDate = str(certificate['notBefore'])

endingDate = str(certificate['notAfter'])

startingYear = int(startingDate.split()[3])

endingYear = int(endingDate.split()[3])

Age\_of\_certificate = endingYear-startingYear

#checking final conditions

if((usehttps==1) and (certificate\_Auth in trusted\_Auth) and (Age\_of\_certificate>=1) ):

return -1 #legitimate

elif((usehttps==1) and (certificate\_Auth not in trusted\_Auth)):

return 0 #suspicious

else:

return 1 #phishing

except Exception as e:

print(e)

return 1

def domain\_registration(url):

try:

w = whois.whois(url)

updated = w.updated\_date

exp = w.expiration\_date

length = (exp[0]-updated[0]).days

if(length<=365):

return 1

else:

return -1

except:

return 0

def favicon(url):

#ongoing

return 0

def port(url):

#ongoing

return 0

def https\_token(url):

subDomain, domain, suffix = extract(url)

host =subDomain +'.' + domain + '.' + suffix

if(host.count('https')): #attacker can trick by putting https in domain part

return 1

else:

return -1

def request\_url(url):

try:

subDomain, domain, suffix = extract(url)

websiteDomain = domain

opener = urllib.request.urlopen(url).read()

soup = BeautifulSoup(opener, 'lxml')

imgs = soup.findAll('img', src=True)

total = len(imgs)

linked\_to\_same = 0

avg =0

for image in imgs:

subDomain, domain, suffix = extract(image['src'])

imageDomain = domain

if(websiteDomain==imageDomain or imageDomain==''):

linked\_to\_same = linked\_to\_same + 1

vids = soup.findAll('video', src=True)

total = total + len(vids)

for video in vids:

subDomain, domain, suffix = extract(video['src'])

vidDomain = domain

if(websiteDomain==vidDomain or vidDomain==''):

linked\_to\_same = linked\_to\_same + 1

linked\_outside = total-linked\_to\_same

if(total!=0):

avg = linked\_outside/total

if(avg<0.22):

return -1

elif(0.22<=avg<=0.61):

return 0

else:

return 1

except:

return 0

def url\_of\_anchor(url):

try:

subDomain, domain, suffix = extract(url)

websiteDomain = domain

opener = urllib.request.urlopen(url).read()

soup = BeautifulSoup(opener, 'lxml')

anchors = soup.findAll('a', href=True)

total = len(anchors)

linked\_to\_same = 0

avg = 0

for anchor in anchors:

subDomain, domain, suffix = extract(anchor['href'])

anchorDomain = domain

if(websiteDomain==anchorDomain or anchorDomain==''):

linked\_to\_same = linked\_to\_same + 1

linked\_outside = total-linked\_to\_same

if(total!=0):

avg = linked\_outside/total

if(avg<0.31):

return -1

elif(0.31<=avg<=0.67):

return 0

else:

return 1

except:

return 0

def Links\_in\_tags(url):

try:

opener = urllib.request.urlopen(url).read()

soup = BeautifulSoup(opener, 'lxml')

no\_of\_meta =0

no\_of\_link =0

no\_of\_script =0

anchors=0

avg =0

for meta in soup.find\_all('meta'):

no\_of\_meta = no\_of\_meta+1

for link in soup.find\_all('link'):

no\_of\_link = no\_of\_link +1

for script in soup.find\_all('script'):

no\_of\_script = no\_of\_script+1

for anchor in soup.find\_all('a'):

anchors = anchors+1

total = no\_of\_meta + no\_of\_link + no\_of\_script+anchors

tags = no\_of\_meta + no\_of\_link + no\_of\_script

if(total!=0):

avg = tags/total

if(avg<0.25):

return -1

elif(0.25<=avg<=0.81):

return 0

else:

return 1

except:

return 0

def sfh(url):

#ongoing

return 0

def email\_submit(url):

try:

opener = urllib.request.urlopen(url).read()

soup = BeautifulSoup(opener, 'lxml')

if(soup.find('mailto:')):

return 1

else:

return -1

except:

return 0

def abnormal\_url(url):

#ongoing

return 0

def redirect(url):

#ongoing

return 0

def on\_mouseover(url):

#ongoing

return 0

def rightClick(url):

#ongoing

return 0

def popup(url):

#ongoing

return 0

def iframe(url):

#ongoing

return 0

def age\_of\_domain(url):

try:

w = whois.whois(url)

start\_date = w.creation\_date

current\_date = datetime.datetime.now()

age =(current\_date-start\_date[0]).days

if(age>=180):

return -1

else:

return 1

except Exception as e:

#print("Here we go")

print(e)

return 0

def dns(url):

#ongoing

return 0

def web\_traffic(url):

#ongoing

return 0

def page\_rank(url):

#ongoing

return 0

def google\_index(url):

#ongoing

return 0

def links\_pointing(url):

#ongoing

return 0

def statistical(url):

#ongoing

return 0

def main(url):

check = [[url\_having\_ip(url),url\_length(url),url\_short(url),having\_at\_symbol(url),

doubleSlash(url),prefix\_suffix(url),sub\_domain(url),SSLfinal\_State(url),

domain\_registration(url),favicon(url),port(url),https\_token(url),request\_url(url),

url\_of\_anchor(url),Links\_in\_tags(url),sfh(url),email\_submit(url),abnormal\_url(url),

redirect(url),on\_mouseover(url),rightClick(url),popup(url),iframe(url),

age\_of\_domain(url),dns(url),web\_traffic(url),page\_rank(url),google\_index(url),

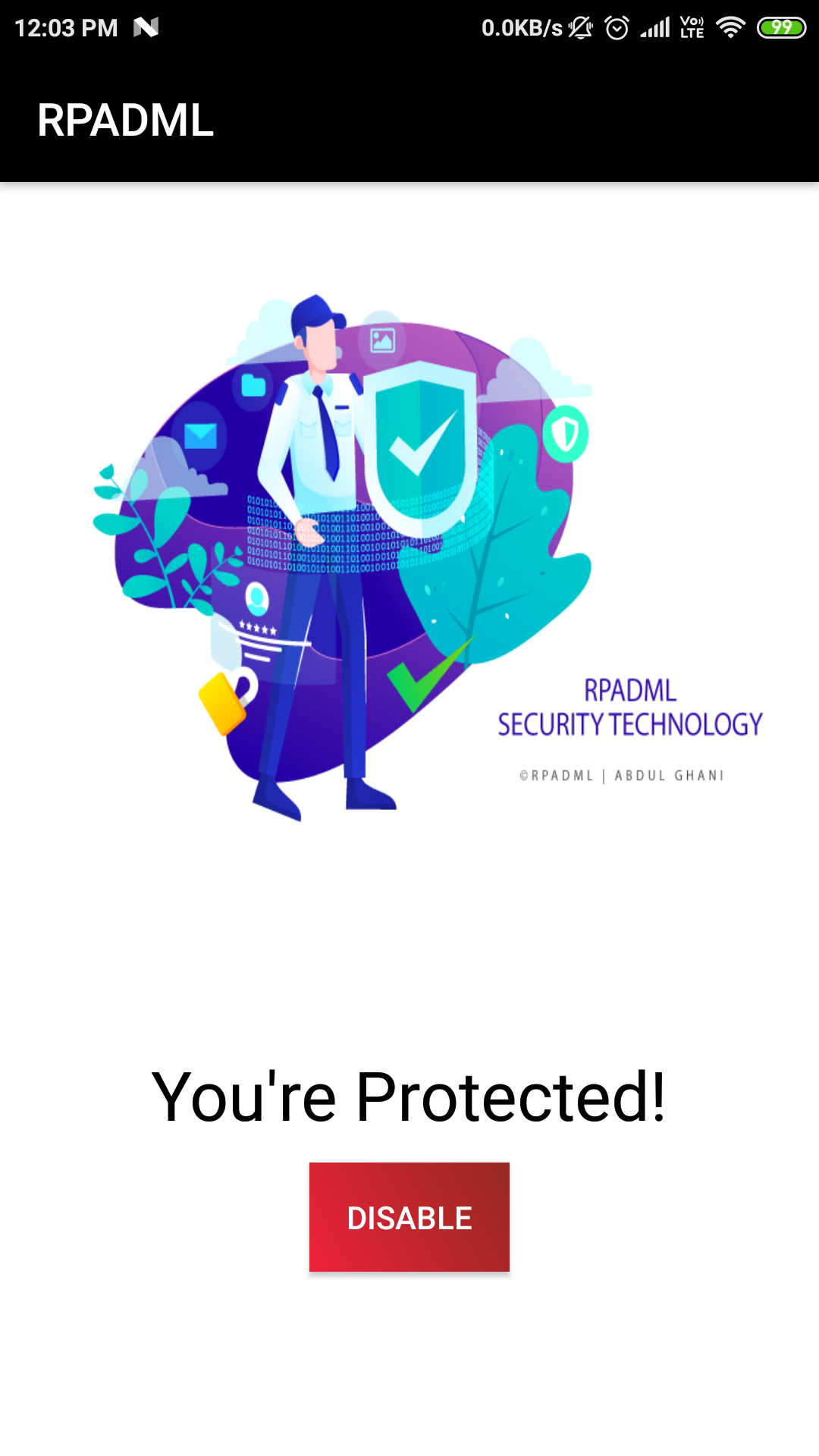
links\_pointing(url),statistical(url)]]

print(check)

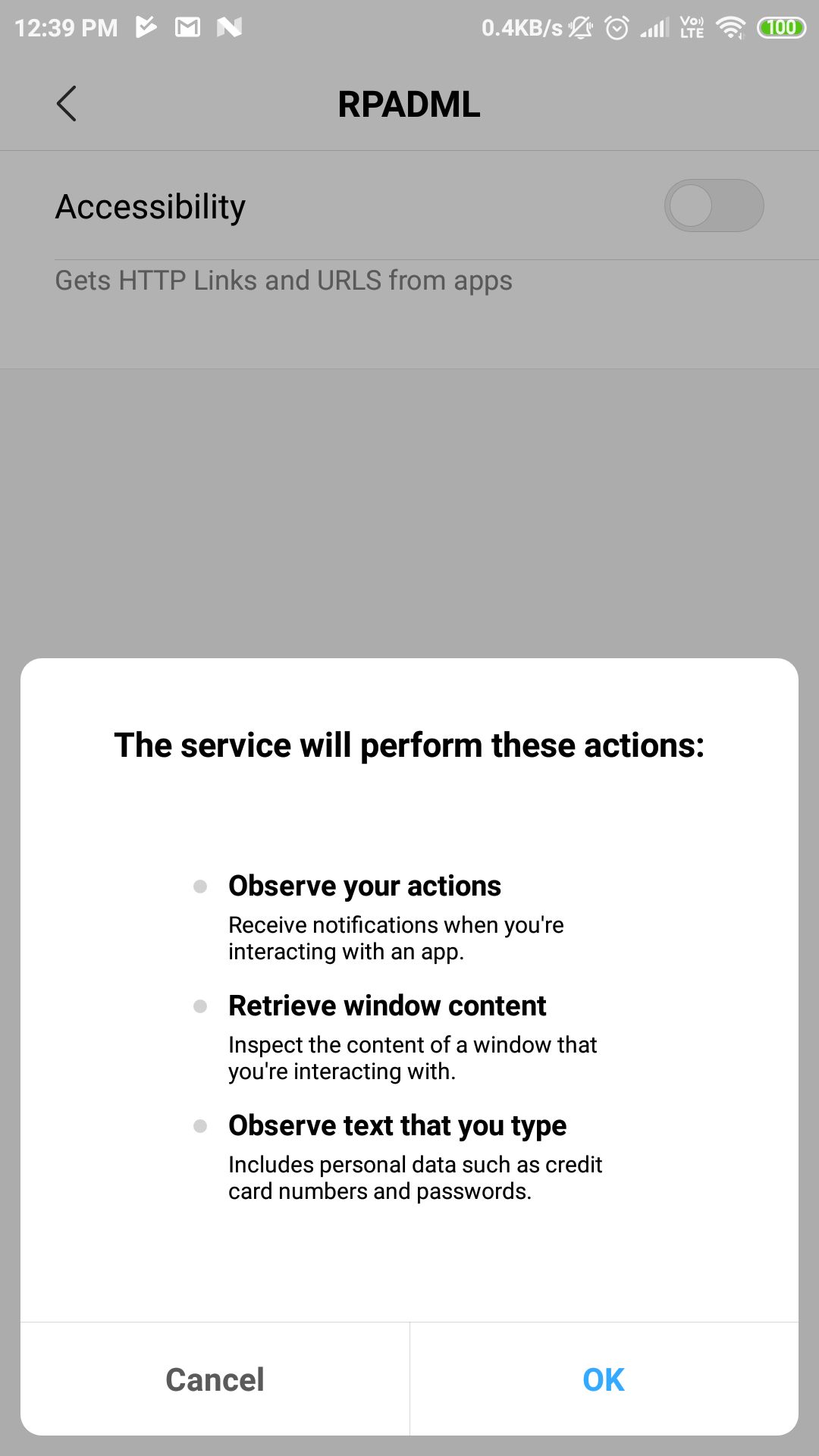
return check

#main("http://wwww.techglows.com")

**5.2.1 RESULT ANALYSIS**

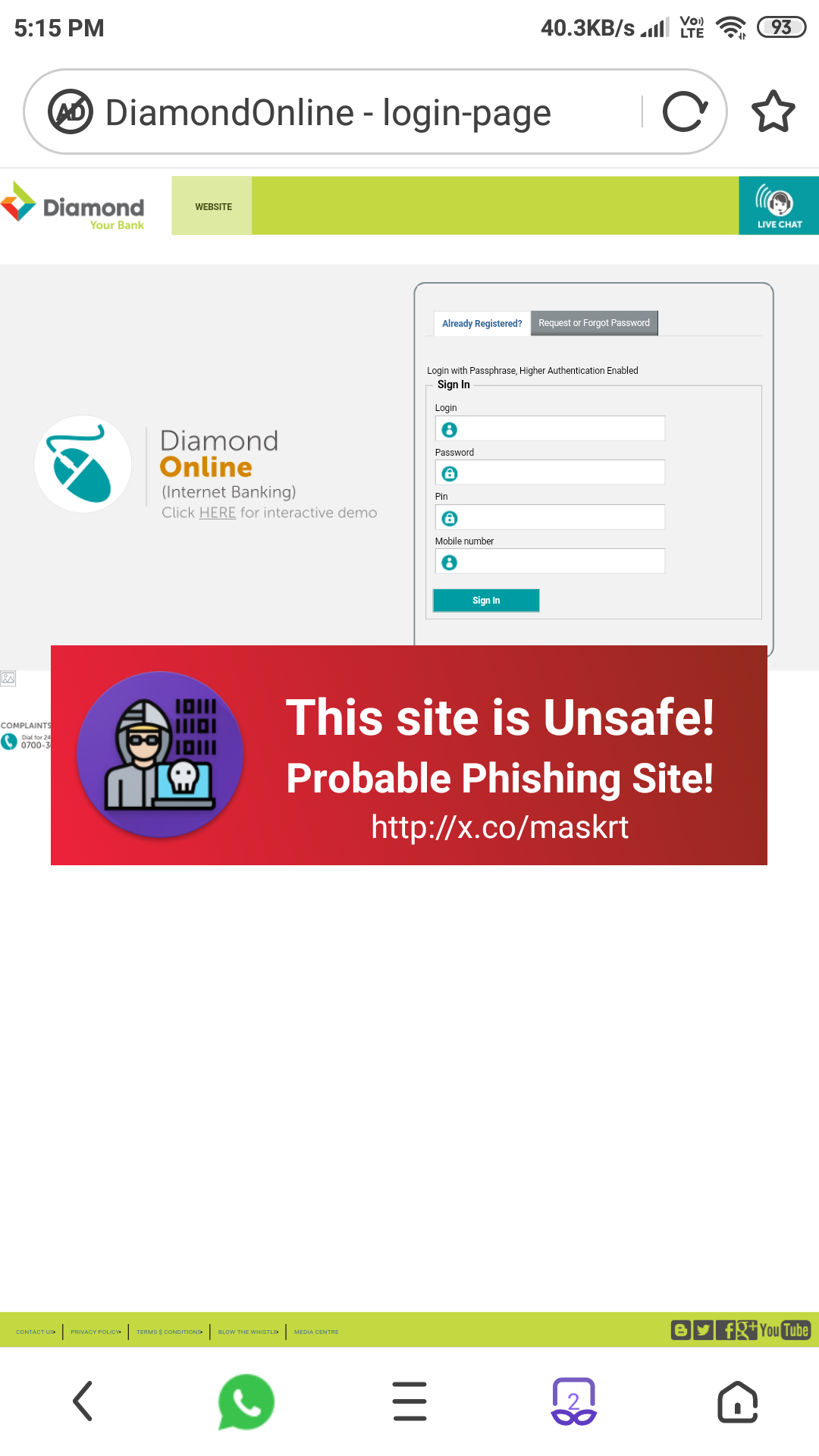


Screen No.5.2.1.1 Welcome/Main page

****

Screen No.5.2.1.2 Accessibility Permission page

**5.2.2 OUTPUT SCREENS:**

****

Screen No.5.2.2.1 Phishing Website detected Page

**5.3 CONCLUSION**

At the time of the beginning of the development of this project we had kept certain goals in the mind, and it is great pleasure that the system is meeting most of its requirements. In this project report we have mentioned all the details of the system, which includes all the stages of the system.

The goals that are expected to achieve by the software are:

* It can be used anywhere any time as it is a web based application.
* Avoiding the errors by minimizing human interaction through user friendly screens to enter data and retrieve the information from the application.
* Portable and flexible for further enhancement.

The system is also tested in Windows xp and found to be running smoothly. It is recommended to run the system with Windows 2000 and later version.

**6. TESTING & VALIDATION**

**6. TESTING & VALIDATION**

**6.1 INTRODUCTION:**

Application Testing is the process used to help identify the correctness, completeness, security, and quality of developed user application. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behavior of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

**Test levels**

Unit testing tests the minimal software component and sub-component or modules by the programmers.

* Integration testing exposes defects in the interfaces and interaction between integrated components (modules).
* Functional testing tests the product according to programmable work. System testing tests an integrated system to verify/validate that it meets its requirements.
* Acceptance testing can be conducted by the client. It allows the end-user or customer or client to decide whether or not to accept the product. Acceptance testing may be performed after the testing and before the implementation phase. See also Development stage
* Beta testing comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the company. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

**Test cases, suites, scripts and scenarios:**

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined (IEEE 829-1998) a test case is an input and an expected result. This can be as pragmatic as 'for condition x your derived result is y', whereas other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps (but often steps are contained in a separate test procedure that can be exercised against multiple test cases, as a matter of economy) but with one expected result or expected outcome. The optional fields are a test case ID, test step or order of execution number, related requirement(s), depth, test category, author, and check boxes for whether the test is automatable and has been automated. A test case should also contain a place for the actual result. These steps can be stored in a word processor document, spreadsheet, database or other common repository. In a database system, you may also be able to see past test results and who generated the results and the system configuration used to generate those results. These past results would usually be stored in a separate table.

The term test script is the combination of a test case, test procedure and test data. Initially the term was derived from the byproduct of work created by automated regression test tools. Today, test scripts can be manual, automated or a combination of both. The most common term for a collection of test cases is a test suite. The test suite often also contains more detailed instructions or goals for each collection of test cases. It definitely contains a section where the tester identifies the system configuration used during testing. A group of test cases may also contain prerequisite states or steps, and descriptions of the following tests.

**6.2 TEST CASES:**

**GUIDELINES FOR TEST CASES:**

**GUI Test Cases**

* Total no of features that need to be check Look and Feel
* Look for Default values if at all any (date & Time, if at all any require) Look for spell check.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case name** | **Testcase description** | **Expected value** | **Actual value** | **Result** |
| GUI test | Check for all the features in the screen. | The screen must contain features | The result we get on checking | True/False |
| GUI test | Check for the alignment of the objects as per the validations. | The alignment should be in proper way. | The result we get on checking | True/False |

Table 6.2.1. Sample for GUI Test Case

**6.3 CONCLUSION:**

By testing the software we can know the errors in the software and we can modify the error occurring area. By doing testing process more than once we can know the rarely occurring errors.

**7. CONCLUSION**

**7.1 PROJECT CONCLUSION**

By using RPADML system, we solve the problem of detecting phishing sites on mobile devices in real-time. Now, the users are able to identify phishing sites/links without performing any activity. RPADML system itself shows floating warning sign before entering such websites.

**7.2 FUTURE ENHANCEMENTS**

As future work, we tend to Adding compiled machine learning model in local devices, Increasing the efficiency of API i.e., Response Time by leveraging Server Resources, Ability to report false-positive results, Better garbage management in client device.

**8. REFERENCES**

**8.1 REFERENCE BOOKS**

1. Unified Modeling Language - James Rumbaugh , Grady Booch

2. Java 2 The Complete Reference - Herbert Schildt

3. JAVA Handbook - TATA InfoTech

**8.2 REFERRED SITES**

www.w3schools.com

https://www.tutorialspoint.com

https://www.quora.com

https://en.wikipedia.org/wiki/Unified\_Modeling\_Language

https://en.wikipedia.org/wiki/Des

MRIET

sMAJOR PROJECT REPORT

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Father Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Contact No:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Company Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Contact No:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Final Records Submission Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Soft Copy Hard Copy HOD PRINCIPAL**

**Verified By Verified By**