

Documentation for Smishing Detection

A Project by Hardhat Enterprises

Software Requirements Specification (SRS)	6
I. Introduction	6
Purpose	6
Product Scope	6
Product Value	6
Intended Audience	6
Intended Use	6
II. Overall Description	7
Product Perspective	7
1. Integration with Mobile Ecosystem	7
2. Continuous Improvement	7
3. Community Participation	7
4. User-Centric Features	7
5. Business and Organisational Use	7
6. Technical Dependencies	7
Product Features	8
• Real-Time Smishing Detection:	8
• User-Friendly Interface:	8
• Alerts:	8
• Customisable Alerts:	8
• App Updates:	8
User Stories	8
User Characteristics	9
Assumptions and Dependencies	9
III. Functional Requirements	10
User Story 1:	10
Functional Requirements:	10
User Story 2:	10
Functional Requirements:	10
User Story 3:	10
Functional Requirements:	10
User Story 4:	10
Functional Requirements:	11
User Story 5:	11
Functional Requirements:	11
User Story 6:	11
Functional Requirements:	11
User Story 7:	11
Functional Requirements:	11
User Story 8:	11
Functional Requirements:	11

User Story 9:	12
Functional Requirements:	12
User Story 10:	12
Functional Requirements:	12
IV. Interface Requirements	13
User Interfaces	13
Hardware Interfaces	13
Software Interfaces	13
Communication Interfaces	13
System Features	13
V. Non-Functional Requirements	14
Security	14
Reliability	14
Maintainability	14
Availability	14
Recoverability	14
Performance	14
Capacity	14
Usability	15
VI. Appendices	15
Definitions, Acronyms and Abbreviations	15
Research	16
I. Introduction	16
Purpose	16
Audience	16
II. Research Deliverables	17
Research 1: Comprehensive Research on Strategies and Tactics of Defenders and Attackers Based on Proper Threat Models in Smishing Detection	17
Overview	17
Major Resources Utilised	17
Special References for the Case Studies	17
Hours Spent Researching and Working on the Report	17
GitHub Repository	18
Research 2: Comprehensive Research on State-of-the-Art Progress in Smishing Detection Methods and Techniques	18
Overview	18
Major Resources Utilised	18
Hours Spent Researching and Working on the Report	18
Future Directions	18
GitHub Repository	19
III. How was Research used for the technical launch of the project?	19

AI Team	19
UX & UI	20
I. Introduction	20
Purpose	20
Audience	20
II. Deliverables	20
Overview	20
Major Resources Used	20
Hours Spent on UX/UI	21
GitHub	21
III. Designs	22
Mock-ups	22
IV. Wireframes & Prototypes	24
Wireframes	24
Prototypes	26
V. User Research	28
Desired Features	28
Security & Privacy Expectations	28
Scam Type & Frequency	28
User Challenges	28
User Personas	28
Persona 1	28
Persona 2	29
Persona 3	29
VI. Testing	30
UI Testing	30
Usability	30
Visual Consistency	30
Accessibility	30
User Feedback	30
VII. Conclusion	31
Algorithm	32
I. Introduction	32
Purpose	32
Audience	32
II. Deliverables	32
Overview	32
Major Resources Used	32
Hours Spent	33
GitHub	33
III. Application Prototype	33

Overview	33
Features and Components	33
Future Enhancements	33
IV. Machine Learning Model	34
Overview	34
Components	34
Key Features	34
Highlights	35
V. Spam Detection	35
• Data Preprocessing:	35
• Feature Extraction:	36
• Machine Learning Model:	36
• Model Evaluation:	36
• Real-Time Classification:	36
• Thresholding:	36
VI. Conclusion	36

Software Requirements Specification (SRS)

I. Introduction

Purpose

The purpose of this document is to present a detailed description of the Smishing Detection Application. It will explain the purpose and features of the application, the user, the interfaces of the application, what the application will do and the requirements of the application. This document is intended for both the developers of the application and the stakeholders.

Product Scope

The Smishing Detection App will encompass the development of a mobile application compatible with major operating systems. The app will focus solely on detecting smishing attempts within SMS messages. It will not cover other forms of phishing, email-based attacks, or non-SMS communication channels.

Product Value

The app's value comes from its ability to provide users with an effective defence against smishing attacks. By automating the SMS message analysis, the app saves users time and reduces the likelihood of victims falling to fraud. This enhances users' trust in digital communication, contributes to overall security, and minimises any potential damage that could be caused by the scam attempt, including financial and personal losses.

Intended Audience

The Smishing Detection App is designed to cater to a diverse audience. Any individual who owns a smartphone and receives smishing messages are encouraged to download this app. It is particularly beneficial for those who engage in online financial transactions, share sensitive information, or rely on SMS for critical updates. The app is accessible to and suited for both cybersecurity-savvy users and those with limited technical expertise.

Intended Use

The app is intended for use on mobile devices, such as smartphones and tablets, running popular operating systems. Users will install the app to actively scan incoming SMS messages. Upon detecting a potential smishing attempt, the app will generate alerts, empowering users to take appropriate actions. The app aims to seamlessly integrate into users' messaging routines, ensuring minimal disruption.

II. Overall Description

Product Perspective

The Smishing Detection Application is designed with a clear focus on enhancing user security by specifically targeting and mitigating smishing attacks within SMS messages. While it is important to acknowledge that the app's core functionality centres around real-time smishing detection and user-friendly alerting, it's equally crucial to emphasise how these features fit into the broader product perspective.

1. Integration with Mobile Ecosystem

- The app aligns with the broader mobile ecosystem, integrating seamlessly into users' daily routines. It acknowledges that users rely heavily on SMS messages for various purposes and aims to enhance this communication channel's security without disrupting the overall user experience.

2. Continuous Improvement

- The inclusion of app updates highlights its commitment to evolving security threats. This aspect underscores the product's adaptability and its intention to remain effective against emerging smishing techniques.

3. Community Participation

- The reporting mechanism is a testament to the app's community-driven approach. Users are encouraged to actively contribute to the app's threat database, fostering a sense of collective security. This feature strengthens the product's defence capabilities and promotes user engagement.

4. User-Centric Features

- The product is crafted with a clear understanding of diverse user needs and preferences. Customizable alerts, discreet notifications, and clear communication regarding permissions cater to users with varying levels of technical proficiency and privacy concerns.

5. Business and Organisational Use

- While the primary focus is on individual users, the product also recognizes its relevance for businesses and organisations concerned about SMS security. This acknowledgment positions the app as a versatile tool for safeguarding both personal and professional communications.

6. Technical Dependencies

- The product acknowledges its reliance on mobile network access, stable internet connectivity for updates and reporting, and the effectiveness of machine learning algorithms. These dependencies are essential for ensuring the app's functionality.

In conclusion, the Smishing Detection Application's product perspective emphasises its integration into users' daily mobile interactions, its commitment to ongoing improvement, community participation, user-centric design, and its potential applicability in business contexts.

This perspective helps stakeholders understand how the product fits within the larger landscape of mobile security and communication.

Product Features

- **Real-Time Smishing Detection:**
 - The app will continue to monitor all incoming SMS messages, employing methods to identify potential smishing attempts to protect the user's data and privacy.
- **User-Friendly Interface:**
 - A user-friendly and intuitive interface along with a small tutorial will allow for a stress-free set up, and ease of use within the app.
- **Alerts:**
 - When a smishing attempt is detected, the app will generate real-time alerts, notifying users of the potential threat.
- **Customisable Alerts:**
 - Users can personalise alert preferences, choosing between notifications, sound alerts, or silent alerts.
- **App Updates:**
 - Updates will be provided to upgrade the capabilities of our smishing detection systems and address any emerging threats.

User Stories

1. As a tech-savvy student who uses Android, I want the app to automatically identify and delete scam messages, ensuring that my inbox remains free from unwanted and potentially harmful content.
2. As a user who frequently interacts with promotions and deals, I want the app to detect and notify me about suspicious messages containing promotions I never signed up for, helping me differentiate legitimate offers from scams.
3. As a user who values minimalist notifications, I want the app to provide discreet scam alerts, like a notification badge or subtle indicator, without taking up excessive screen space in my messaging app.
4. As a user who relies on visual cues for trustworthiness, I want the app to subtly highlight or differentiate scam messages using non-red colours or unobtrusive marks, ensuring that I can quickly identify potentially harmful messages without overwhelming visual warnings.
5. As an Android user concerned about app permissions, I want the app to ensure that clickable links within flagged scam messages are non-functional, enhancing security and preventing accidental engagement with malicious content.
6. As a student who frequently uses Instagram and TikTok, I want the app to be vigilant about scam messages related to the apps I commonly use, enhancing my awareness of potential threats within my preferred platforms.

7. As a privacy-conscious user, I want the app to clearly communicate the level of access it requires and provides, allowing me to make informed decisions about my data and ensuring my privacy is respected.
8. As a user who has not used scam identification apps before, I want the app to have a user-friendly interface and clear instructions, making it easy for me to understand how to use and benefit from the app's features.
9. As a user who receives a variety of scam messages, I want the app to offer information about each flagged message's potential risks and indicators of scam, enabling me to make informed decisions about further actions.
10. As a user who values seamless integration, I want the app to integrate with my messaging app's spam folder, keeping me informed about the number of flagged scam messages while maintaining a clutter-free inbox.

User Characteristics

Users of the Smishing Detection App vary widely in terms of technical ability and backgrounds. They can be individuals of all age groups, including tech-savvy consumers and can be those with limited technology understanding. Businesses and organisations concerned about their employee's SMS security are also potential users. The app is designed to be accessible to a diverse user base and provide a seamless experience regardless of the user's level of technical expertise.

Assumptions and Dependencies

- The app assumes that users have mobile devices running compatible operating systems (e.g., iOS, Android).
- The app depends on users have access to a mobile network and stable internet connection, especially for updates and reporting*.
- Detection accuracy depends on the effectiveness of the machine learning algorithms, this can be improved periodically.
- Users are depended on to configure their in-app settings in accordance with their preferences.

III. Functional Requirements

User Story 1:

As a tech-savvy student who uses Android, I want the app to automatically identify and delete scam messages, ensuring that my inbox remains free from unwanted and potentially harmful content.

Functional Requirements:

- The app will automatically identify the scam message and send a notification to the user.
- The app will have an option feature with the notification to allow users to either immediately delete, send to 'junk' folder or provide a warning.

User Story 2:

As a user who frequently interacts with promotions and deals, I want the app to detect and notify me about suspicious messages containing promotions I never signed up for, helping me differentiate legitimate offers from scams.

Functional Requirements:

- During the sign-up process, the app will ask the user to either collect all promotional content and only notify once a day/week with a summary of all promotions or skip this step.
- The app will have the option to leave the promotional content the user has signed up for and remove those they did not.

User Story 3:

As a user who values minimalist notifications, I want the app to provide discreet scam alerts, like a notification badge or subtle indicator, without taking up excessive screen space in my messaging app.

Functional Requirements:

- The app will employ minimalist notification techniques whilst alerting the user of any suspicious messages.
- A subtle indicator will appear on the user's screen, indicating the presence of a suspicious message.
- The notifications will not disrupt the user's messaging experience or cause intrusive pop-ups.

User Story 4:

As a user who relies on visual cues for trustworthiness, I want the app to subtly highlight or differentiate scam messages using non-red colours or unobtrusive marks, ensuring that I can quickly identify potentially harmful messages without overwhelming visual warnings.

Functional Requirements:

- A visual differentiation between regular and suspicious messages will occur.
- Avoid the use of red and intrusive colours, by implementing non-intrusive colours and subtle visual cues to indicate potential scams.
- A clean and user-friendly design appearance.

User Story 5:

As an Android user concerned about app permissions, I want the app to ensure that clickable links within flagged scam messages are non-functional, enhancing security and preventing accidental engagement with malicious content.

Functional Requirements:

- Disable clickable links within suspicious scam messages.

User Story 6:

As a student who frequently uses Instagram and TikTok, I want the app to be vigilant about scam messages related to the apps I commonly use, enhancing my awareness of potential threats within my preferred platforms.

Functional Requirements:

- The app will notify the user about suspicious messages with the number and contents of the message, including the malicious link.

User Story 7:

As a privacy-conscious user, I want the app to clearly communicate the level of access it requires and provides, allowing me to make informed decisions about my data and ensuring my privacy is respected.

Functional Requirements:

- During the sign-up process, the app will explain the level of access required and provided via the terms and conditions.

User Story 8:

As a user who has not used scam identification apps before, I want the app to have a user-friendly interface and clear instructions, making it easy for me to understand how to use and benefit from the app's features.

Functional Requirements:

- The app will have a clean and user-friendly design appearance.
- The app will display a scam warning flag with the scam message that is recognisable.
- The app will have clear instructions during sign-up.

User Story 9:

As a user who receives a variety of scam messages, I want the app to offer information about each flagged message's potential risks and indicators of scam, enabling me to make informed decisions about further actions.

Functional Requirements:

- Along with each scam message, the app will provide a list of potential risks and indicators.
- The app will offer an option feature with the notification to allow users to either immediately delete, send to 'junk' folder or provide a warning.

User Story 10:

As a user who values seamless integration, I want the app to integrate with my messaging app's spam folder, keeping me informed about the number of flagged scam messages while maintaining a clutter-free inbox.

Functional Requirements:

- The app will notify users about the number of flagged scam messages.
- The app will allow users to create a clutter-free inbox by having the option to categorise scam messages in groups.

IV. Interface Requirements

User Interfaces

The Smishing app's user interface is designed for simplicity and usability. It will feature a clean, intuitive design to ensure that users, including those with limited technical expertise, can navigate it easily. The main screen will display smishing alerts prominently, allowing users to quickly identify potential threats. The sign-up process is simple, requiring a username, email, and password. The app will also allow for settings and filter customisation, including keyword blocking, region blocking and other advanced filtering options. There is room to include a reporting interface that will include a user-friendly form for reporting suspicious messages with ease. An uncomplicated login screen will ensure secure access for users. This user-focused design prioritises an intuitive experience, enhancing the app's accessibility and effectiveness in combating smishing.

Hardware Interfaces

The SMS Warden app will require internet or cellular data to run and display notifications.

Software Interfaces

The operation systems for the app will be Android and IOS (Apple). There is an effective machine learning model for identifying smishing attempts in text messages. The programming languages, JSON and JS, are used to identify the scam message and display a notification to the user and build the app main pages. The programming language, Python was used to build the machine learning model.

Communication Interfaces

Communication interfaces for the Smishing app will prioritise security and accessibility. The app will establish secure connections and safeguard user data and privacy. Notifications of suspicious messages will be delivered to the user based on their personal settings to allow for a more individualised communication experience. These communication interfaces will ensure seamless and secure interactions with the app, instilling confidence in users about the confidentiality and integrity of their data.

System Features

The system features within the app will be made certain to revolve around effectiveness and practicality. It will employ basic smishing message detection techniques, making it a suitable choice for our project's scale. Users will be able to register securely, and a straightforward login mechanism will provide secure access. User's will be able to customise their settings for a more personalised experience and the scope for new features will only further enhance the app in future. These features align with the project's objectives, ensuring a functional and user-friendly app for combating smishing threats.

V. Non-Functional Requirements

Security

The Smishing App will prioritise security. User data protection is a priority and thus sufficient means will be taken to achieve it. A secure sign-up and login processes will be completed to ensure security and privacy needs are met. Regular updates and further smishing methods will be introduced to ensure security from suspicious entities. It is of paramount importance that security measures are put in place, ensuring user security and data privacy for all consumers.

Reliability

With the amount of testing and prototyping, it will ensure that the app will perform with a very low chance of any critical failures occurring. Users can access the app 95% of the time without failure. If the app experiences a failure, the administrator will be prepared to resolve the issue.

Maintainability

If the app experiences a failure, the administrator and team will gather and find the cause of the failure immediately. Depending on the severity of critical failure, ideally it should take an hour, a few hours or worse case a day to resolve.

Availability

There will be routine software upgrades once every three months and the app will encourage the user to update their app when a new software upgrade is available. The user will not be able to use the app until the upgrade is complete. The upgrade should not take longer than 5 minutes to install.

Recoverability

If a critical issue occurs, the administrator must take measures to ensure the app is operational within three days.

Performance

The load time for each page in the app should not be more than two seconds for users.

Capacity

The user can store up to 15G of stored scam messages.

Usability

Usability will be a core focus. User testing and feedback will guide interface design and interaction flows. The app will adhere to accessibility standards, making it inclusive for users with any ailments or lack of technological experience. A user-friendly interface will be designed

with clear navigation and intuitive features, supported by comprehensive user guides and onboarding tutorials. Ensuring users are satisfied with the app will be a regular and on-going process, users can offer feedback in this area to help our team improve its usability.

VI. Appendices

Definitions, Acronyms and Abbreviations

App	Application, a software program designed to perform specific tasks on mobile devices.
Cybersecurity	The practice of protecting computer systems, networks, and data from security breaches and unauthorised access.
JS	JavaScript, a programming language of the Web.
JSON	JavaScript Object Notation, leading data interchange format for applications.
Phishing	A cyber-attack where malicious actors attempt to trick individuals into exposing confidential information.
Python	A general-purpose programming language used in machine learning and applications.
Smishing	SMS phishing, a fraudulent attempt to deceive individuals into revealing sensitive information via SMS messages.
SMS	Short Message Service, commonly referred to as text messages, a means of sending short text-based communications between mobile devices.
SRS	Software Requirements Specification, a comprehensive document detailing the functional and non-functional requirements of a software project.

Research

I. Introduction

Purpose

The purpose of this document is to provide a comprehensive overview of our research efforts in the development of the Smishing App, a critical tool in the fight against SMS Scamming. Within this research document's pages, we examine two key research initiatives that underpin the app's foundation. Research 1 explores the strategies of both attackers and defenders in Smishing detection, based on robust threat models. Research 2 delves into state-of-the-art advancements in Smishing detection methods. Additionally, we highlight how research has shaped the technical launch of our project, offering the insight from our brilliant teams in AI and App Development and the valuable perspectives collected through user surveys and stories.

Audience

This document is designed to accommodate a diverse audience, including our project stakeholders, technical teams, and anyone with a vested interest in the development of the Smishing App. It especially serves as a valuable resource for:

- **Project Team:** Our project team members, including AI specialists, app developers, cybersecurity experts, and researchers. They will find detailed insights into the research outcomes and technical specifics that drive the Smishing App's development.
- **External Partners:** External partners that are engaged in the project's welfare can align their efforts with our research outcomes and project objective, this fosters effective communication.

By addressing the diverse needs of these audiences, we ensure that the Smishing App's research and development efforts are transparent, accessible, and aligned with the expectations of all involved parties.

Our research and development efforts for the Smishing App have been tailored to meet the needs and expectations of two primary audiences:

- **End Users:** The core audience of the Smishing App comprising individuals who seek protection against SMS-based phishing attacks. This includes smartphone users across various demographics who rely on text messages for personal and professional communication. Understanding their preferences, concerns, and usage patterns has been central to our design and development processes.
- **Technical Teams:** Our secondary audience includes the technical teams responsible for the app's development, maintenance, and continuous improvement. This includes our AI experts, app developers, and cybersecurity professionals who need in-depth technical details and insights concerning the app's functionalities.

Our research and development strategies are aligned with catering to the specific needs and expectations of both these audiences, ensuring the Smishing App is user-friendly, effective, and technically sound.

II. Research Deliverables

Research 1: Comprehensive Research on Strategies and Tactics of Defenders and Attackers Based on Proper Threat Models in Smishing Detection

Overview

In this section, we will provide an overview of our extensive research on the strategies and tactics employed by both defenders and attackers in the realm of Smishing detection. Our research is grounded in vigorous threat models, enabling us to gain a profound understanding of the developing landscape of Smishing attacks.

Major Resources Utilised

There were many different reliable resources used by team members to construct the report. The top six organisations used to research and find information are:

- [Kaspersky](#)
- [Alkahalil et al. 2021](#)
- [Njuguna et al. 2022](#)
- [Martens 2023](#)
- [Australian Cyber Security Centre \(ACSC\)](#)
- [Mishra and Soni 2019](#)

Special References for the Case Studies

Uber and Twilio were case studies examined for real-world smishing attacks. The references used to find out the analysis of the notable smishing incidents, the strategies employed by attackers and the defence tactics and responses in each case study are:

- [Kost 2023](#)
- [Storm 2022](#)
- [Uber Team 2022](#)
- [Security 2022](#)

Hours Spent Researching and Working on the Report

- Naghma: 35 hours
- Holly: 22 hours
- Rhonda: 20 hours
- Nicolai: 20 hours

TOTAL: 97 hours

GitHub Repository

Please access the detailed report of the Research 1 Document by the following link to our GitHub Repository:

<https://github.com/Cuzza312/SmishingDetection/blob/naghma242-research-1/Research%201.pdf>

Research 2: Comprehensive Research on State-of-the-Art Progress in Smishing Detection Methods and Techniques

Overview

This section delves into our exploration of the profound advancements in Smishing detection methods and techniques. Our research goes beyond traditional approaches, seeking to identify state-of-the-art methods and innovative technologies that can bolster the effectiveness of our Smishing App.

Major Resources Utilised

There were many different reliable resources used by team members to construct the report. The top four organisations used to research and find information are:

- [Kaspersky](#)
- [Alkahalil et al. 2021](#)
- [Martens 2023](#)
- [Mishra and Soni 2021](#)

Hours Spent Researching and Working on the Report

- Naghma: 35 hours
- Holly: 21 hours
- Nicolai: 15 hours
- Dylan: 3 hours

TOTAL: 74 hours

Future Directions

Due to the nature of the project, the research team were unable to finish certain sections in this report. This is a list for the next team to research these important sections for further development of the Smishing Detection app.

State-of-the-Art Smishing Detection Systems

- Overview of Prominent Smishing Detection Solution
- Comparative Analysis of Leading Smishing Detection Tools
- Performance Metrics Evaluation

Data Collection and Preprocessing

- Data Sources and Types
- Data Preprocessing Techniques
- Data Augmentation Strategies

Experimental Setup

- Description of the Dataset

- Feature Selection and Engineering
- Model Training and Evaluation

Results and Discussion

- Performance Comparison of Smishing Detection Techniques
- Analysis of False Positives and False Negatives
- Discussion of Key Findings and Observations

Challenges and Limitations

- Inherent Challenges in Smishing Detection
- Limitations of Existing Techniques
- Ethical and Privacy Considerations

Future Directions and Research Opportunities

- Emerging Trends in Smishing Attacks
- Potential Enhancements to Detection Techniques
- Integration of AI and Advanced Technologies

Conclusion

- Summary of Research Findings
- Importance of Continuous Smishing Detection Improvement
- Implications for Future Cyber Security Strategies

GitHub Repository

Please access the detailed report of the Research 2 Document by the following link to our GitHub Repository:

LINK:

<https://github.com/Cuzza312/SmishingDetection/blob/naghma242-research/Research%202.pdf>

III. How was Research used for the technical launch of the project?

AI Team

The comprehensive research conducted by the research team played a pivotal role for the technical launch of the project. The AI Team reviewed all the research reports when developing the models and gained an understanding of what smishing is and its inherent nature as a threat. The information was necessary for the AI team to develop a model capable of distinguishing between regular messages and smishing attempts. The AI team used the research to create the application prototype and machine learning model guide.

UX & UI

I. Introduction

Purpose

The purpose of this UX/UI Document is to provide clear guidance and specifications for the user experience (UX) and user interface (UI) designs of the Smishing App. This document serves as a comprehensive reference that outlines visual and interactive elements seen throughout the app, ensuring consistency and alignment with user needs and project objectives. It aims to facilitate a user-focused design approach and enable the development team to create an intuitive and engaging user interface. This document sets the foundation for a secure, user-friendly, and effective tool for combating smishing threats.

Audience

The primary audience for this UX/UI Document includes members of the development team, including UX/UI designers, front-end developers, and project leaders. It is also relevant to the users of the app who have a vested interest in the app's design and functionality. Additionally, this document can serve as a valuable resource for conducting user testing and gathering feedback from its users, ensuring that the app's design aligns with the needs and preferences of its target audience.

II. Deliverables

Overview

Our team made significant progress based on user research and feedback, ensuring that the Smishing App aligns seamlessly with user expectations and requirements. Here's a breakdown of the key updates.

Major Resources Used

Design Tools: We employed industry-standard design software such as Figma to refine the app's user interface, ensuring an intuitive and visually appealing design.

User Research: Extensive user research and feedback collection helped us understand user challenges and expectations, driving the decisions we made within our designs.

Colour Scheme: We introduced a vibrant blue and green colour scheme to create a visually engaging and distinctive look.

Hours Spent on UX/UI

Our team was very dedicated to the UX/UI side of the project and collectively invested upwards of a few hundred hours in brainstorming, designs, usability testing, to implement all that is seen within the app's prototype.

- Tom: 60 hours
- Nik: 40 hours
- Naghma: 30 hours
- Rhonda: 25 hours
- Nicolai: 20 hours
- Holly: 16 hours

TOTAL: 191 hours

GitHub

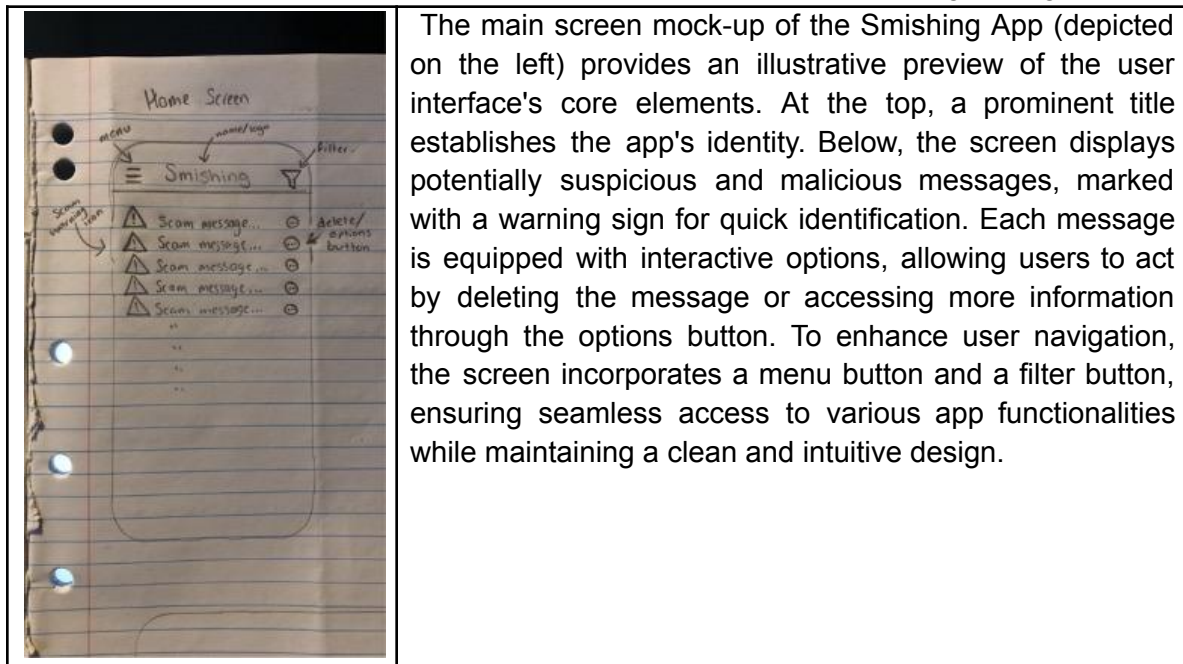
Our team has updated our GitHub Repository with all our UX/UI resources including mock-ups, final design prototypes, wireframes, as well as the surveys conducted. This is for accessibility and transparency reasonings, and all our resources can be found here: <https://github.com/Cuzza312/SmishingDetection>

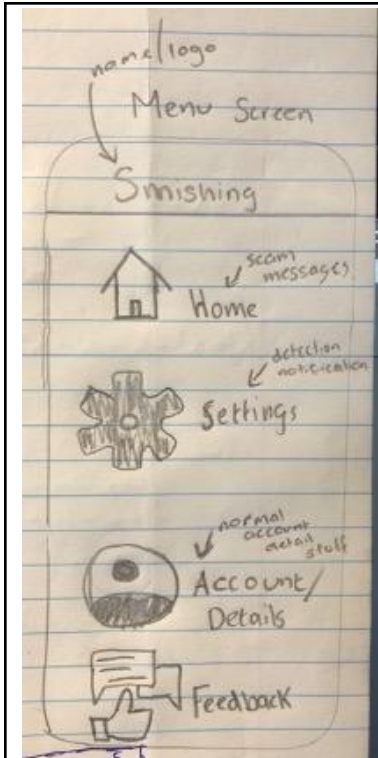
III. Designs

Mock-ups

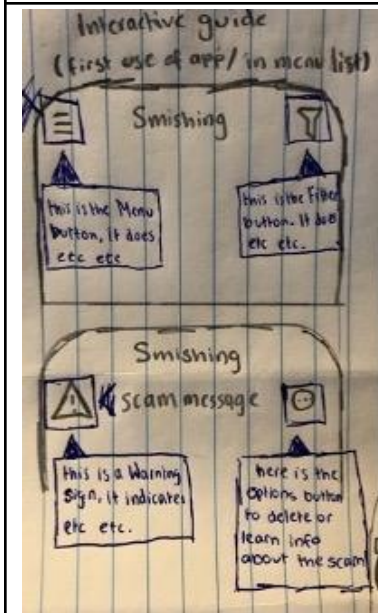
Mock-ups play a vital role in the design process by enabling the stakeholders, including designers, developers, and potential users, to visualise the app's look and feel. They help identify potential design challenges and ensure alignment with the project's goals and user requirements. Throughout the development cycle, these mock-ups will be refined and evolve into wireframes and prototypes that closely resemble the final product, helping to create a user-friendly and visually appealing Smishing App.

Below are some designs that visually represent the proposed user interface (UI) elements and layout for the Smishing app. It served as a visual blueprint, offering a glimpse into how the app will appear and how users will interact with it. Based on user feedback and design iterations, further refinement can and will take place to ensure we implement a design of high value.





The menu screen mock-up (featured on the left) presents a clear and structured user interface. At the top, the app's title reaffirms its identity, ensuring users know their location within the app. Dominating the centre of the screen are four large logos, each representing a key function. The home button offers a pathway to return to the main screen. The settings icon leads users to configure app preferences. The account/details icon facilitates access to user profiles and personal information management. Lastly, the feedback button allows users to provide valuable input, promoting user engagement and improvement of the app's features. Together, these elements make for an efficient and user-friendly menu experience.

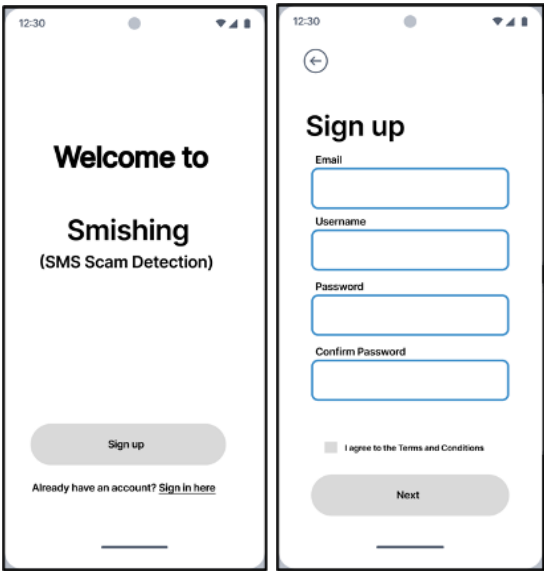


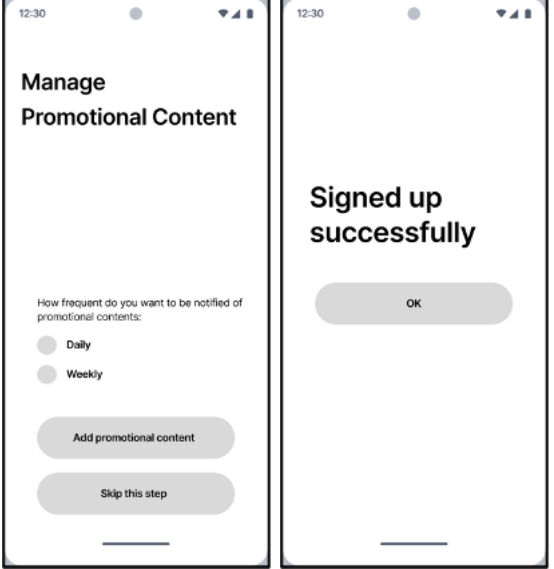

The interactive guide mock-up (displayed on the left) serves as an insightful tutorial for first-time users of the Smishing app, introducing them to its functionality step by step. The app's logo then appears, accompanied by informative text explaining the app's purpose and mission. As users progress, the tutorial navigates them through the main screen, highlighting each section's features. For instance, it explains that the warning logo signifies potential risks, and the scam messages are prominently displayed. This informative approach extends to all app features, including the menu, options, and filter buttons, ensuring a comprehensive and user-friendly experience.


IV. Wireframes & Prototypes

Wireframes

Wireframes are valuable in the early stages of design and help to define the app's interface and user interaction. They also allow for collaboration among the designers, developers, and other stakeholders, helping to refine the app's design before progressing to higher-quality prototypes. Included below are wireframes that serve as foundational blueprints for the Smishing app's user interface (UI) design. These wireframes provide a visual representation of the app's layout and key interactive elements. These wireframes showcase visual design elements and just as importantly the app's structural layout.


	<p>The first wireframe introduces users to the Smishing App with a "Welcome to Smishing (SMS Scam Detection)" page. A central, prominent button invites users to sign up, making the onboarding process instinctive. Below, is an "Already have an account? Sign in here..." button which provides a quick path for returning users. This wireframe rationalises the initial user experience, ensuring users can easily access the app while maintaining a friendly and inviting design. The second wireframe in this section illustrates the sign-up process. Users are prompted to input their email, username, password. These clear instructions guide users through the process, and opting for users to accept the terms and conditions agreement will ensure transparency and user consent. This facilitates a smooth and secure sign-up process whilst showcasing to users our concern for their protection and consent.</p>
--	--



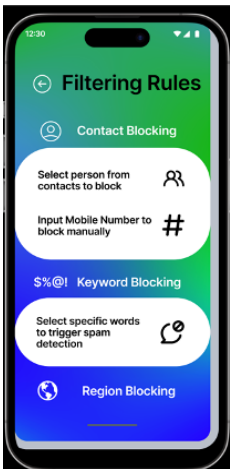
	<p>The first wireframe presents the "Manage Promotional Content" page. Users are given the ability to control their promotional preferences, choosing between Daily, Weekly, or skipping the step entirely. It simplifies user interaction, allowing customisation of the app's promotional notifications based on individual preferences. The second wireframe in this section is a simple confirmation of a successful registration. Offering a positive confirmation to the consumer allows our app to set the tone for a user-friendly and encouraging experience.</p>
	<p>This wireframe offers a view of the Smishing Menu page. This wireframe closely resembles the mock-up mentioned earlier, with intuitive navigation elements like a home button, settings, account/details, and feedback options. The clean and structured layout ensures users can easily access various app functionalities. The second wireframe within this section provides a look at the Menu Page's design, showcasing the app's core features. It also closely resembles the earlier mock-up, allowing users to see the main functions that they will operate with. Both these wireframes allow our team to present the app's user-friendly interface, intuitive navigation, and slick design.</p>

 A wireframe of a mobile app's login page. It features a status bar at the top with the time 12:30 and signal icons. Below is a back arrow icon. The title 'Log In' is centered. There are two input fields labeled 'Username' and 'Password'. At the bottom is a 'Log In' button.	<p>The final wireframe in this document is the Login Page, where users input their username and password to gain secure access into our application. This straightforward design priorities user authentication and data security. It ensures a seamless login experience, ensuring user data is protected.</p>
---	---

Prototypes

In this section, we present the prototypes that offer a glimpse into the user interface and functionality of the Smishing App. These prototypes serve as interactive models, providing a great representation of the app's design and core features. They offer a hands-on experience of how the app will appear and how it will function, facilitating user feedback and refining of the final product. The team has also incorporated a fresh and dynamic colour scheme of vibrant blues and greens to our prototypes which will provide an enhanced visual experience for users. These prototypes offer a new revamp of the Smishing App, ensuring that the user interface is both visually appealing and functional.

 Two mobile app prototypes side-by-side. The left one is the 'Welcome to SMS WARDEN' screen with a green and blue gradient background, a 'Sign up' button, and a link 'Already have an account? Sign in here'. The right one is the 'Sign up' page with a similar gradient, a back arrow, and input fields for 'Email', 'Username', 'Password', and 'Confirm Password'. It also has a checkbox for 'Agree to T&Cs' and a 'Next' button.	<p>Displayed on the left are two snapshots of our prototype for the app. The first image showcases our welcome screen, which now boldly displays 'Welcome to SMS Warden' along with our slogan, 'Guarding Your Mobile World.' The page also prominently features a 'Sign-Up' button and provides a quick 'Already have an account? Sign in here' option for convenience. The second image presents our sign-up page, prompting users to enter their email, username, password, and agree to the Terms and Conditions.</p>
--	---

	<p>This section presents two prototype images on the left. The first image illustrates the content management screen, allowing users to select their preference for receiving promotional content—daily, weekly, or the option to skip this step if they prefer not to see promotions. The second image is a simplified 'Welcome to SMS Warden' screen, inviting users to proceed into the app with a straightforward 'OK' button, where they will then be moved into the belly of our app, the messages section.</p>
	<p>Two prototype images appear in this section, with the first portraying the Smishing App's home screen. The title 'Smishing' at the top clearly indicates the current section the users are in. Users can now differentiate potential scam messages with an intuitive colour system, where red signifies a definite scam and green represents a likely trustworthy message. Two buttons, located at the top on either side of the title, offer easy access to the menu and filter functions. The second image depicts the user-friendly menu screen, enabling seamless navigation between Messages, Filtering Rules, Settings, Account Details, and Feedback.</p>
	<p>This final section showcases one prototype image on the left, the filtering rules screen presents users with three distinct categories: 'Contact Blocking,' 'Keyword Blocking,' and 'Region Blocking.' These categories empower users to customise their filtering rules. Users can choose to block specific contacts, manually input phone numbers for blocking, set keywords to trigger scam detection, and even block messages from specific regions, tailoring their smishing protection to their individual preferences.</p>

V. User Research

From the user research that was completed, we have assembled a compilation of the most common responses to our questionnaire, including challenges faced, desired app features, etc. It is worth noting that the questionnaire returned answers from a variety of individuals including android users and apple users, none of which had ever used a scam identification app previously.

Desired Features

- Simple and easy-to-use functionality, with a one-button on/off switch.
- Automatic deletion of scam messages.
- Notification blocking for scam alerts.
- A classification system (e.g., this is a scam, this could be a scam, this is not a scam) with corresponding colour-coding.
- In-app alerts were preferred over push notifications.
- Security measures to prevent accidental clicks on scam messages.
- Minimalistic design and non-intrusive warnings.
- Privacy features, including encryption and non-sharing of user information.

Security & Privacy Expectations

- Expectation of strong security measures, including protection against accidental interactions with scam messages and encryption of personal data.
- Expectation of no sharing of user information with third parties.

Scam Type & Frequency

- Scam messages are received daily or weekly, with variations in frequency.
- Common scam types include parcel delivery notifications, government reimbursements, and prize winnings.
- Scam messages often contain links.

User Challenges

- Difficulty determining the legitimacy of scam messages, as they are becoming more convincing.
- Annoyance and distraction caused by frequent scam alerts. Privacy features, including encryption and non-sharing of user information.

User Personas

Persona 1

- **Demographic:** Male, early 30s, Android user.

- **Scam Message Frequency:** Receives scam messages occasionally, about once a month.
- **Common Scam Types:** Parcel delivery notifications and government reimbursements.
- **Challenges:** Finds scam messages annoying and occasionally clicks on links by mistake.
- **Desired App Features:** Prefers an on/off switch for app functionality, automatic deletion of scam messages, and minimalistic design.
- **Alert Preferences:** Favours push notifications to be alerted immediately.
- **Visual Indicators:** Supports a colour-coded classification system for easy identification.
- **Security and Privacy Expectations:** Expects security measures for personal data, like email and passwords.
- **Scam Identification App Experience:** None.

Persona 2

- **Demographic:** Female, late 20s, Apple user.
- **Scam Message Frequency:** Receives scam messages daily, sometimes multiple times a day.
- **Common Scam Types:** Parcel delivery notifications and prize winnings.
- **Challenges:** Often struggles to determine the legitimacy of scam messages as they become more convincing.
- **Desired App Features:** Prioritises a colour-coded classification system for quick identification and minimalistic design. Wants to avoid frequent notifications.
- **Alert Preferences:** Prefers in-app alerts to reduce distractions.
- **Visual Indicators:** Supports colour-coding for scam identification.
- **Security and Privacy Expectations:** Expects safeguard of personal information and never to receive another scam again.
- **Scam Identification App Experience:** None.

Persona 3

- **Demographic:** Female, late 60s, Apple user.
- **Scam Message Frequency:** Receives scam messages sporadically, around once a month.
- **Common Scam Types:** Parcel delivery notifications, government reimbursements.
- **Challenges:** Finds scam messages confusing and occasionally clicks on links unintentionally.
- **Desired App Features:** Prefers a straightforward, user-friendly interface. Values automatic deletion of scam messages and a classification system.
- **Alert Preferences:** Likes in-app alerts as they are less intrusive.
- **Visual Indicators:** Appreciates clear visual cues.
- **Security and Privacy Expectations:** Expects strong security measures to prevent accidental interactions with scam messages.
- **Scam Identification App Experience:** None.

VI. Testing

UI Testing

Throughout the course of the project, testing was conducted in the realm of UX/UI to evaluate the effectiveness of the design changes. The following areas were assessed:

Usability

Usability tests were conducted to ensure that users can easily navigate the app, identify smishing messages, and take appropriate actions.

Visual Consistency

The UI initially was very plain with white backgrounds and black writing, so adjustments were made, with attention paid to the new colour scheme creating a more cohesive and visually appealing design.

Accessibility

We ensured that the app's UI adheres to accessibility standards, making it inclusive for all users, including those with no technological ability or skill.

User Feedback

Feedback from users will be invaluable in refining the UI design to meet user expectations, users can do so within the app itself as we are very open and welcome feedback.

VII. Conclusion

In conclusion, the updates made to the Smishing App's UX/UI have been guided by thorough research, testing, and feedback. The introduction of a vibrant colour scheme, made for a more appealing app, and clear visual indicators enhances the user experience, making it easier for users to identify and respond to smishing threats. Our commitment to security, privacy, and accessibility ensures that the app not only looks great but also prioritises user safety and satisfaction. As we move forward, we remain dedicated to delivering a user-friendly and effective tool for combating smishing threats.

Algorithm

I. Introduction

Purpose

The purpose of this document is to provide a comprehensive overview of the AI teams efforts in the development of machine learning, and natural language processing models for detecting SMS spams / scams on the Smishing App. This document explores the production of a machine learning model that will be able to input text and output whether it is a scam or not. It examines the way the AI team gathered data, created the model architecture and fine-tuned/trained the model. The deliverables of the application prototype, machine learning model, spam detection and application development are examined.

Audience

This document aims to target a wide audience, including technical teams, project stakeholders and future project leaders and members that are interested in developing and extending the project of the Smishing Detection App. The project team members include AI specialists, AI experts, app developers and cyber security experts. The technical teams responsible for the app's development, maintenance and continuous improvement includes back-end developers, app developers and cyber security professionals.

II. Deliverables

Overview

In this section, we will provide an overview of the significant progress that the AI team made on machine learning, and natural language processing models for detecting SMS spams/scams.

Major Resources Used

GitHub: A platform used to help developers store, manage, track and control their code. Allows developers to collaborate and store their code in one place.

Android Studio: Allows app developers with an integrated development environment to build a cross-platform mobile application for Android apps.

React Native: Allows developers to create an application by having a JavaScript framework for writing mobile applications.

Hours Spent

Both Jake and Halim spent around 100 hours each to gather data, create the model architecture and fine-tune/train the model to create the final product ready for the app development team to integrate.

GitHub

The AI team has updated their GitHub Repository with all the algorithm resources including the work on the application prototype, machine learning model guide, spam detection and communication between the app and the detection algorithm.

https://github.com/Cuzza312/SmishingDetection/tree/Prototype_Halim

https://github.com/Cuzza312/SmishingDetection/tree/Prototype_Jake

III. Application Prototype

Overview

This section of the project involves the development of the smishing application utilising Android Studio and constructed using React Native. The primary objective is to seamlessly embed machine learning functionalities into the app, thereby augmenting its capabilities and providing users with smart and advanced features. There are a number of files stored on GitHub with files used to develop Java applications (iml), JS files and JSON files for the development of the model.

Features and Components

Button: A button component is available for user interaction, facilitating actions or data submission.

Textbox: Users have the ability to input text using the textbox element, empowering them to provide input or engage with the application.

Text view: The application currently includes a text view component, enabling users to observe and exhibit text-based information.

Future Enhancements

There are some potential enhancements that may be considered:

- **Machine Learning Algorithms:** Implement machine learning algorithms to execute functions such as text analysis, image recognition, or predictive modelling. These algorithms have the potential to offer valuable insights or automation within the app.
- **Predictive Analytics:** Develop machine learning models for predictive analysis, such as recommending content, foreseeing user actions or making personalised suggestions.

- Natural Language Processing (NLP): Integrate NLP methods to enable the app to comprehend and process human language. This capability can be used for chatbots, sentiment evaluation, or language translation.
- Training and Model Management: Consider implementing a system for training and administering machine learning models within the application, involving model updates, retraining and version control.
- User Profiling: Create user profiling using machine learning techniques to deliver a personalised user experience, including personalised recommendations, content filtering or targeted notifications.
- Data Privacy and Security: Guarantee the secure handling of user data and establish appropriate measures to safeguard privacy, ensuring compliance with data protection regulations.

IV. Machine Learning Model

Overview

This section of the project involves the development of the Smishing App with the integration of machine learning capabilities. The primary objective is to create a robust and intelligent mobile application that influences the advantages of machine learning to improve user experiences and introduce innovative features. The AI Team conducted a series of experiments with each one contributing to the gradual refinement of the model. With each iteration, the team took steps to optimise it, one at a time. After experimenting for several weeks, the AI Team created a working model capable of detecting smishing in messages. There are a number of files stored on GitHub, including testing files, datasets and model creations.

Components

The key components of the machine learning model include:

- Machine Learning Integration: Separately with the app development process, the project integrates machine learning components. These components allow the app to execute tasks that transcend traditional mobile apps, thus making it more intelligent and responsive.
- Mobile Application Development: The project's foundation lies in the development of the Smishing App, designed to be compatible with multiple platforms, thereby ensuring accessibility to a broad user audience.

Key Features

- User-Friendly Interface: The Smishing App possesses an intuitive and user-friendly interface, guaranteeing a seamless experience for users with varying skill levels.
- Machine Learning Algorithms: Numerous machine learning algorithms are implemented to enable intelligent functionalities such as recommendation systems, predictive analytics, image recognition, NLP, and more.

- Data Collection: Machine learning models require data for training and inference. The app is designed to efficiently gather and process relevant data.
- Real-Time Inference: The Smishing App influences the power of machine learning to provide real-time insights and predictions, recommending users valuable information and assistance.
- Scalability: The app is architected with scalability as a consideration, enabling the seamless integration of supplementary machine learning models and features as the project progresses.
- Customisation: Users have the option to customise the machine learning components to align the application to their distinct requirements and preferences.

Highlights

- Machine Learning Model Development: The development and optimisation of machine learning models with the ability to detect smishing attempts in text messages.
- Step-By-Step Guide: A comprehensive step-by-step guide is available in the form of Jupyter Notebook (.ipynb) files, making it easier for others to replicate our efforts and gain a comprehensive understanding of the methodology behind the model development process.
- Model Evaluation: The team conducts thorough testing and evaluation of a range of machine learning models and methodologies. Also, document the performance metrics of each approach, helping users make well-informed decisions.
- Best Model Selection: After detailed testing and analysis, the top-performing machine learning models are chosen and documented, offering a well-defined path for their implementation in real world applications.

V. Spam Detection

In the context of the Smishing Detection App, the implementation of spam detection mechanisms is a pivotal component to ensure user safety and the app's effectiveness. The primary objective of spam detection is to classify incoming SMS messages as either legitimate or potentially malicious smishing attempts. This is accomplished through the integration of machine learning and natural language processing (NLP) techniques.

- **Data Preprocessing:**
 - The first step in spam detection is data preprocessing. Incoming SMS messages are tokenized, cleaned of irrelevant characters, and transformed into a format suitable for analysis. This ensures that the text data is in a standardised and usable form.
- **Feature Extraction:**
 - Next, features are extracted from the preprocessed text data. These features could include word frequencies, n-grams, and other linguistic characteristics.

Feature extraction is a crucial step as it provides the model with relevant information for classification.

- **Machine Learning Model:**
 - A machine learning model is trained on a labelled dataset containing examples of both legitimate and smishing messages. Popular algorithms such as Support Vector Machines (SVMs), Random Forests, or neural networks can be employed for this classification task. The model learns to recognize patterns and characteristics that distinguish spam from legitimate messages.
- **Model Evaluation:**
 - To ensure the accuracy and reliability of the spam detection mechanism, rigorous evaluation is performed. This involves testing the model on a separate dataset that it has never seen before. Performance metrics such as precision, recall, and F1-score are calculated to gauge the model's effectiveness.
- **Real-Time Classification:**
 - Once trained and evaluated, the machine learning model is integrated into the Smishing Detection App, enabling real-time classification of incoming SMS messages. When a message is received, the model quickly analyses its content and assigns a probability score indicating the likelihood of it being a smishing attempt.
- **Thresholding:**
 - A threshold is set to determine when a message should be classified as spam. Messages with probability scores exceeding this threshold are flagged as potential smishing attempts and presented to the user with appropriate warnings.

VI. Conclusion

In conclusion, the development of machine learning and NLP-based spam detection capabilities within the Smishing Detection App is a significant stride towards enhancing user security. By leveraging these technologies, the app can intelligently identify and alert users to potential smishing attempts, thereby safeguarding their personal and financial information.

The AI team has dedicated extensive effort to gather data, design robust machine learning models, and integrate them seamlessly into the app's framework. Continuous evaluation and improvement are essential to ensure the model's accuracy and adaptability to evolving smishing tactics.

With the completion of this algorithm documentation, the Smishing Detection App is well-equipped to provide users with a comprehensive defence against SMS spams and scams, reflecting the team's commitment to technical excellence and user safety.