UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science Faculty of Engineering, Built Environment & IT University of Pretoria

# CyberAttackText

# Software Requirements Specification

iReportSoftware

September 2022

Sibusiso Mngomezulu, u20441984

A. Introduction:

This document has all the necessary documentation for the iCreateSoftware application.

Sibusiso Mngomezulu

Project:
CyberAttackText

# 1 Introduction

1.1. Purpose of product:

The following documentation will provide the functional and non functional software requirements, acceptance criteria, constraints and an overview of the CyberAttackText system. My role is to build a CyberAttackText system that will be responsible for the following: allow a victim of a cyberattack crime to be able to report the crime anonymously; gather text data by means of social media platforms, web scraping, scam emails; the gathered text-data must be able to be visualised and the diversity of the gathered data must be shown; the system must employ sentiment analysis from the gathered text data to determine weather a conversation can lead to potential cyberattack lures. The sentiment analysis employed in the system must be useable by any digital forensic investigator applications.

1.2 Scope of product:

The goal is to create a system that will allow users to safely report cyberattack crimes anonymously. The reports will come from various means and the system must then show the gathered data and its diversity. The sentiment analysis will then help prevent potential cyberattack lures and will be able to be used by other digital forensic investigative software.

1.3 Acronyms, Abbreviations and Definitions:

• FR – Functional Requirements
• NFR – Non Functional Requirements
• API - Application Programming Interface. • UI - User Interface
• C – Constraint
• ETF – Exchange-traded fund

### 1.4. Overview:

The document follows the following scheme:

- An overall description
- Specific Requirements
- Acceptance Criteria

# 2 Overall Description

### 2.1 Product Function:

- The iReportSoftware user anonymously reports the cyberattack crime
- The iReportSoftware system collects text data of the reported cyberattack crime through various means (scam emails, web scraping and the anonymous cyberattack report)
- The iReportSoftware system allows the collected text data to be visualised and show the diversity of collected data
- The iReportSoftware system sentiment analysis uses gathered text data to determine when conversation can lead to potential cyber attack lures
- The iReportSoftware system sentiment analysis will be allowed to be used by other digital forensic investigator applications.

### 2.2 User Characteristics:

The iReportSoftware system is intended for the use of: Victims who want to report crimes anonymously and to use the data to prevent future potential cyberattacks working with other digital forensic investigative applications.

## 2.4 Constraints:

C1. The system must be maintained and managed by the iReportSoftware engineer.

C2. All system implementations and documentation must be done by the iReportSoftware engineer.

C3. The iReportSoftware system design must be the design the software engineer follows throughout the project.

C4. The frontend will be developed with python framework

C5. The backend will be developed with Python for the completion of this project.

C6. The API of iReportSoftware must use Amazon Comprehend API to complete this project.

C7. The database handling of iReportSoftware must use SQL to complete this project.

## 2.5 Assumption and Dependencies:

- Assumptions:

– The user is reporting a cybercrime. – hence the user has an internet connection.
– The user is using a PC that has a UI.

- Dependencies:

- -Time management - I am dependent on time management and pressure handling to be able to complete this project alone with limited time constraint.

- – Lack of knowledge to create a solution to the requirements - Lack of knowledge to cybercrime software and implementing the features has a big effect on me completing this project.

# • Specific Requirements

3.1 Functional Requirements:

• FR.1. A user should be able to report the cyberattack crime anonymously against which the system should gather and manipulate the text data given. iReportSoftware rules must consist of the following:

- – FR.1.1. The system must allow the user to submit a report of the cybercrime anonymously.
  ∗ FR.1.1.1. The system must collect the text data from social media platforms, web scraping, scam emails, documents, cyberattack reports from the user.
- – FR.1.2. The system must visualise the collected text data
  ∗ FR.1.2.1. The system must show the diversity of the collected text data
- – FR.1.3. The system must employ sentiment analysis to the collected text data
  ∗ FR.1.3.1. The system must use sentiment analysis to constitute conversations that my lead to potential cyber attack lures.
- – FR.1.4. The system sentiment analysis must be useable by any digital forensic investor applications.

3.2 External Interface Requirements:

3.2.1. User Interfaces
• The iReportSoftware system can only be accessed by a PC with a UI.

3.2.2. Hardware Interfaces
• Connection to the internet is required.

3.2.3. Software Interfaces • Web browser.

3.3 Performance Requirements:

All the features of the iReportSoftware system must function as expected just as any cyberattack reporting software application should.

### 3.4 Design Constraints:

The CyberAttackText system is dependent on the design requests from the project owners, hence I cannot go forward without communication from the project owner informing me about the design requirements needed.

### 3.5 Quality Requirements:

1. The basic design of the CyberAttackText system should have a visual look that aligns with the UI of a cyber-attack report application.

2. Data integrity - The CyberAttackText system user must have access to report cyber-attack crimes without their identity being leaked.

3. Security - All information on the users (victims) will be completely anonymous

4. Performance - All the features of the iReportSoftware system must function as expected according to the information and requirements given by the project owner.

## 3.6 Architectural Requirements:

### 3.6.1 Flexibility:

– The CyberAttackText system must function on a device that has a UI.

– The TradeSim system must function using all modern PCs with decent Operating System

### 3.6.2 Maintainability:

- – There must be constant communication between the developer of the iReportSoftware and the project owners in order to identify and fix errors to improve the quality of the features.

- – Clear documentation of requirements must be provided to ensure maintainability.

- 3.6.3 Security
  – All information on the CyberAttackText user needs to remain anonymous.

- 3.6.4 Availability:

- – The CyberAttack user must be able to access the software easily by any device, desktop or mobile with a UI.

- – The CyberAttack software must be available at all times.

- 3.6.5 Reliability:
  – CyberAttack is reliable on internet connection.
  – The CyberAttack system must ensure a stable experience to the user.

- 3.6.6 Usability:
  – CyberAttackText must be mobile friendly.

# 4 Attributes

- Availability

– The CyberAttackText system must be accessible victims at all needed times.

- Security

– Information belonging to the user (victim) cannot be leaked, and must remain anonymous.

# 5 Acceptance criteria

4.1 All information that the user gives to the system must be collected in the database and manipulated to be displayed on the user interface without problems.

- 4.3 The system must allow the user to be able to report cybercrimes anonymously.

# 6 Appendices

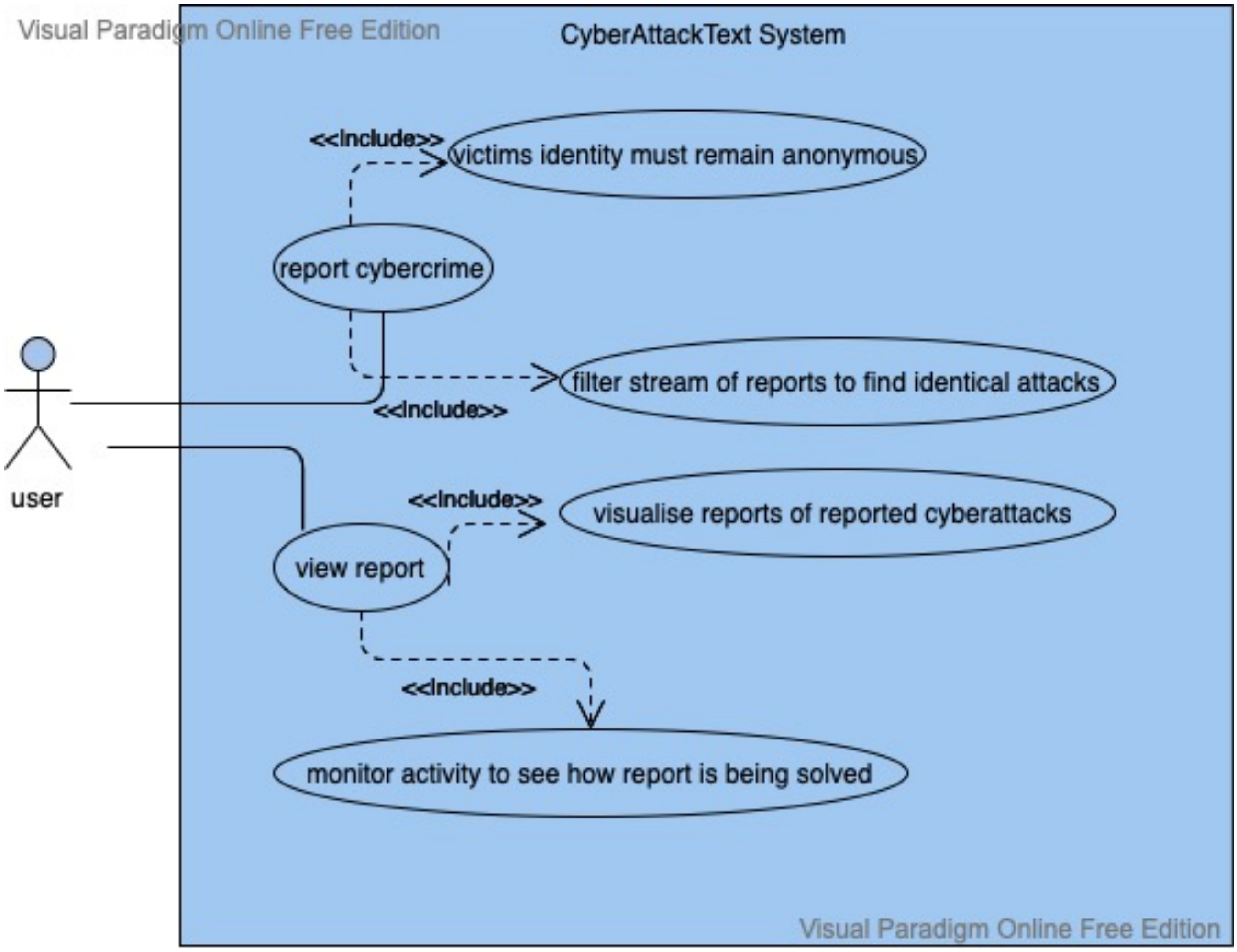
Visual Paradigm Online Free Edition
CyberAttackText System
<<Include>>
victims identity must remain anonymous
report cybercrime
filter stream of reports to find identical attacks
<<Include>>
user
<<Include>>
visualise reports of reported cyberattacks
view report
<<Include>>
monitor activity to see how report is being solved
Visual Paradigm Online Free Edition

Figure 1 is a use-case diagram representing the CyberAttackText system.

# Use Cases:

- The application will be able to receive reports of cyber-attacks and be able to address/solve the situation while the victim's identity is kept anonymous. This will help ensure the victim's safety.
- Users will be able to view their reports as well as monitor the activity which is being taken to solve the dilemma at the time.
- The application will be able to give a solution on how to deal with the set cyber-attack **OR** at least provide a reference to solve the attack if a solution is not generated.
- Streams of reports will be filtered so that similar reports can be identified. This will allow for efficient solution generation as some reports of a cyber-attacks maybe identical in nature.
- Users will also be able to report a cyber-attack directly to the application while keeping their identity anonymous.

CLASS DIAGRAM

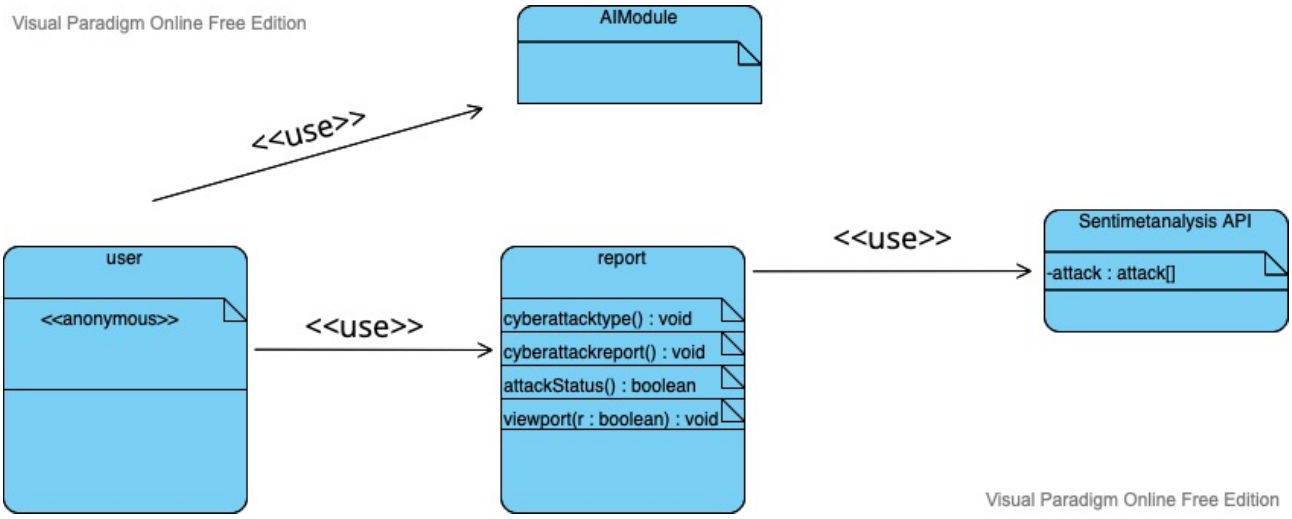

Figure 2 is a class diagram representing the CyberAttackText system.