Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2024

Team Honeycomb
HoneyTrap

Jair Rea
Raed Ali
Hugo Mendoza
Praisy Daniel
Joshua Catalan

Revision History

Revision	Date	Author(s)	Description
0.1	1.01.2016	GH	document creation
0.2	1.05.2016	AT, GH	complete draft
0.3	1.12.2016	AT, GH	release candidate 1
1.0	1.20.2016	AT, GH, CB	official release
1.1	1.31.2016	AL	added design review requests

CONTENTS

List of Figures

List of Tables

1 Introduction

This product is a honeypot method detection type of software meant to detect scammers by creating a simulated persona using AI. The persona is designed to mimic and feel like an elderly human, specifically a widow, to attract potential scammers. The AI-driven system engages with other people and when their actions or communication patterns are deemed similar to tactics used by scammers, the system will flag/report them. This report will then place the scamming profile into a database, letting the system know they have already interacted with the person and they are already deemed a scammer and no further interaction will take place. More details about the system can be found within the Architectural Design and Specification document.

2 System Overview

The Account Profile is a multi step process in which the team formulates a page that simulates a real-world person. This page will contain the necessary requirements such as posts and discussions to convince and lure any potential scamming threat. The Chatbot Control is the main component of the software as it interacts and communicates. Any type of fault in the chatbot might reveal its intentions to scammers and potentially ruin any chances of luring future threats. The Reporting System is a by product of the chatbots interactions with scammers. At any point the system deems an interaction suspicious of scamming tactics, the chatbot will report the person and save their profile into a database. This database will be a collection of scammers and will be made public so that people check and see who is a scammer. The LLM, Web Scraping code and Communication layer are created by the team to help the chatbot understand, speak, and function properly. They all work together, feeding each other information so that no different iteration remains the same.

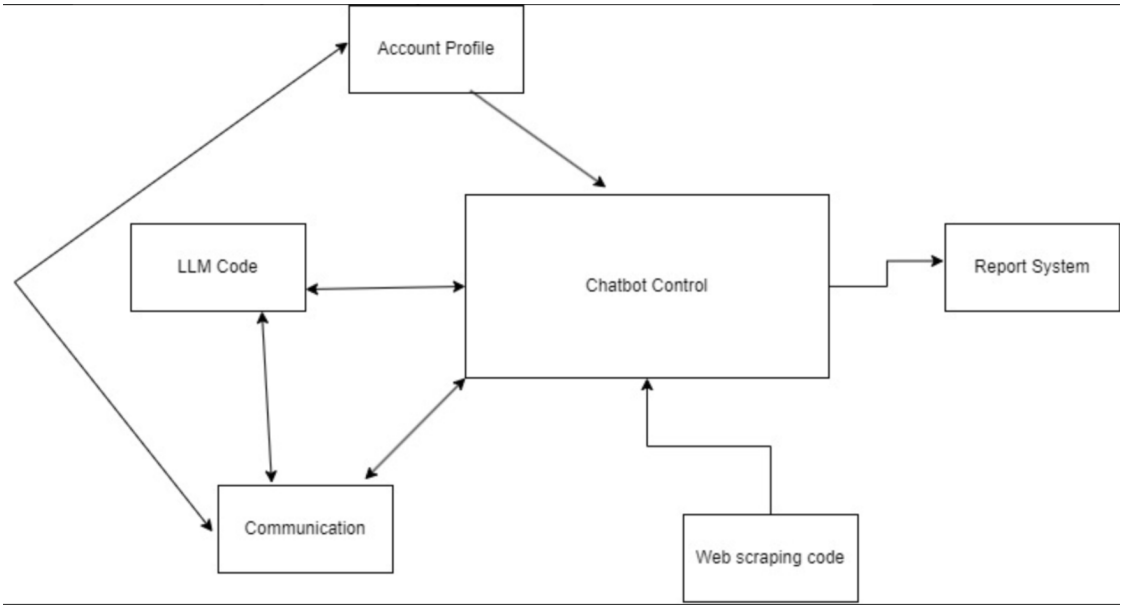

Figure 1: System architecture

3 X Layer Subsystems - Account Profile

3.1 Layer Hardware

The layer relies on dedicated hardware components, this includes a database server for user information storage, an authentication server for login processes, and a web server for HTTP handling.

3.2 Layer Operating System

The OS used is Windows. We are not limiting to solely this OS but do not have any plans currently to expand to others. The reason for this choice is the compatibility and utilization of the tools Windows provides.

3.3 Layer Software Dependencies

The software will be depending on some encryption libraries, communication with Facebook API, web framework, and some database management system. There may be more that follow as this project continues on.

3.4 Subsystem 1

The Account Posts subsystem is a step within the "account profile" layer, designed to handle the creation, retrieval, and deletion of user-generated posts. This subsystem operates as a set of classes and functions, simulating an active user profile. It ensures organized storage of posts, enabling users to engage with the platform through the creation and interaction with various posts.

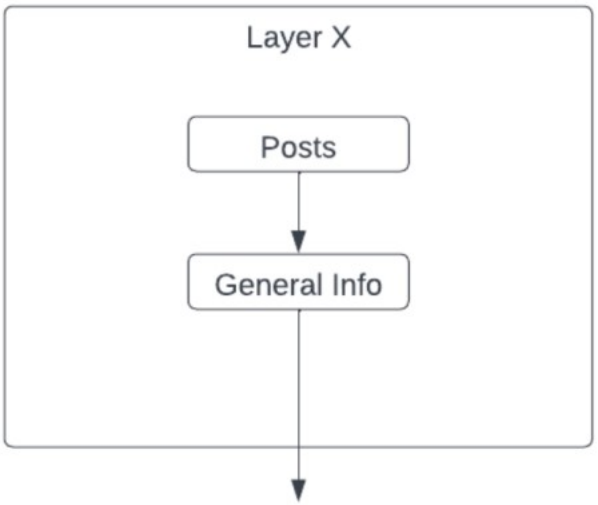

Figure 2: Example subsystem description diagram

3.4.1 Subsystem Hardware

As the system is entirely online and via software, there is none.

3.4.2 Subsystem Operating System

Windows is required.

3.4.3 Subsystem Software Dependencies

All stated above in the main system.

3.4.4 Subsystem Programming Languages

The language used is Python.

3.4.5 Subsystem Data Structures

None.

3.4.6 Subsystem Data Processing

The system will automatically create posts of recent event that are made to look real. The posts will be based on the profile's attributes and characteristics to make it more believable.

3.5 Subsystem 2

The Account General Information subsystem provides information about the comprehensive user details, preferences, and account settings. This subsystem operates within the broader "account profile" layer. The subsystem facilitates the updating of non-post-related user data, providing an overall look at the user's profile and attributes.

3.5.1 Subsystem Hardware

The subsystem is online and does not contain hardware, so none.

3.5.2 Subsystem Operating System

Windows is required

3.5.3 Subsystem Software Dependencies

All stated above in the main system.

3.5.4 Subsystem Programming Languages

The language used is Python.

3.5.5 Subsystem Data Structures

None.

3.5.6 Subsystem Data Processing

None.

4 Y Layer Subsystems

4.1 Layer Hardware

4.2 Layer Operating System

4.3 Layer Software Dependencies

Python: The primary programming language for developing the Chatbot and its subsystems due to its versatility and extensive library support. Selenium: Web automation framework to web scrape and allow the Chatbot to interact and extract information from Facebook profiles. OpenAI: Used to generate responses to scammers and analyze any incoming messages for potential threats/scams.

4.4 Profile Connection

The Profile Connection subsystem acts as the intermediary between the Chatbot and the user's Facebook profile, enabling seamless interaction. It functions as a backend system, facilitating authentication and authorization processes to gain access to the user's profile and retrieve relevant information. Additionally, it monitors the user's profile for updates and changes, potentially gathering insights for more personalized interactions.

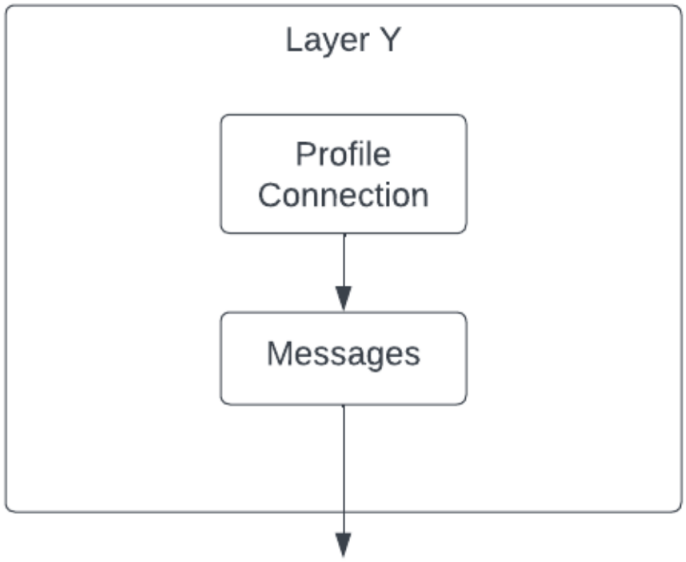

Figure 3: Example subsystem description diagram

4.4.1 Subsystem Hardware

4.4.2 Subsystem Operating System

4.4.3 Subsystem Software Dependencies

We used Selenium (Python Library) to make the connection and gather the info we need.

4.4.4 Subsystem Programming Languages

Python is the programming language used.

4.4.5 Subsystem Data Structures

4.4.6 Subsystem Data Processing

Collecting the information necessary and sending them to the message subsystem is how this Profile Connection works. It is basically pipeline to establish the connection while Messages does the bulk of the work

4.5 Messages

The Messages subsystem handles the exchange of messages between the Chatbot and the Facebook user. It comprises several key components. The Incoming Message Handling component interprets user messages, understanding their meaning. Following this, the Outgoing Message Composition formulates suitable responses, which may include various content types. Lastly, the Message Delivery component ensures that the Chatbot's responses are sent back to the user on Facebook, ensuring smooth conversation flow.

4.6 Subsystem Dependencies

OpenAI is used here to reveive the message and send back an appropriate message. After the bot is trained, this subsystem comes into play since it should have to knowledge and details required to give a proper message back to the sender.

4.7 Subsystem Programming Languages

Python is the programming language used

4.8 Subsystem Date Processing

The processed data goes to OpenAI which generates the response needed along with any findings the Chatbot received to determine a scam.

5 Z Layer Subsystems

In this section, the layer is described in terms of the hardware and software design. Specific implementation details, such as hardware components, programming languages, software dependencies, operating systems, etc. should be discussed. Any unnecessary items can be ommitted (for example, a pure software module without any specific hardware should not include a hardware subsection). The organization, titles, and content of the sections below can be modified as necessary for the project.

5.1 Layer Hardware

The LLM code will be on a python file on a Raspberry Pi 3.

5.2 Layer Operating System

The raspberry pi is running a 32 bit version of Raspbian OS.

5.3 Layer Software Dependencies

The layer is dependent on the openai python library in order to function. Also the os python library is used to manage the backend of the messages log which works with the LLM code in order for the LLM code operate as intended.

5.4 CONNECTING TO OUR LLM

This section connects our code to our large language model. The other sublayers will be performing actions with the content imported here, thus information from here is communicated with the other 2 sublayers.

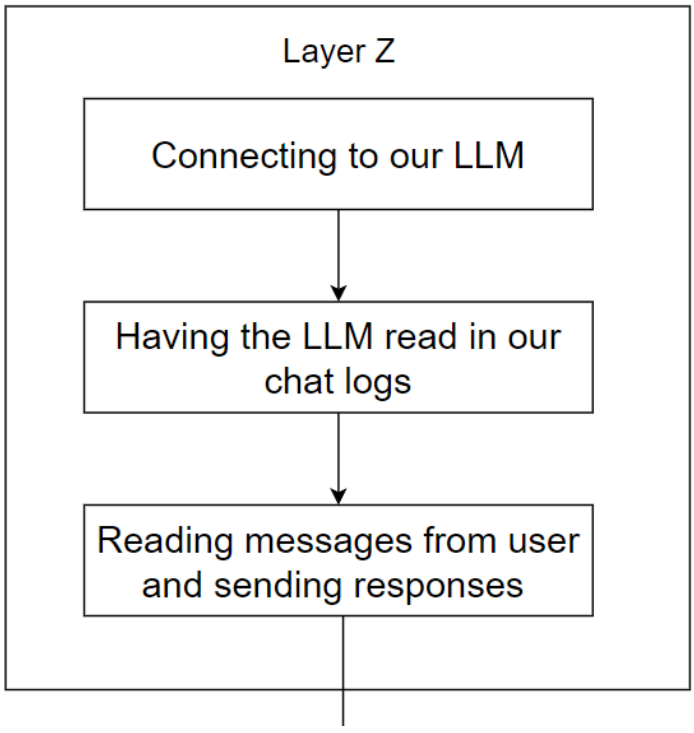

Figure 4: Example subsystem description diagram

5.4.1 Subsystem Hardware

The Raspberry Pi 3

5.4.2 Subsystem Operating System

Raspbian OS

5.4.3 Subsystem Software Dependencies

openai library

5.4.4 Subsystem Programming Languages

Python

5.4.5 Subsystem Data Structures

An api call is being made when we send our openai key to openai. So a packet is sent to do this, probably a TCP packet is sent with our openai key.

5.4.6 Subsystem Data Processing

Nothing too complex is done in this subsystem just sending our openai key to open ai and if there is an error in processing our key we don't execute the rest of the code.

5.5 LLM reading our chatlogs

We will need to train our bot using the large language model to act like a vulnerable widow. Now to do this the most accurate way we can we will feed our LLM a complex set of instructions on how to act and also any chats or data we may think may be useful. We also need to feed our bot any previous chat logs it has already had. Importing this data and feeding it to our chat bot is what this section is dedicated to.

5.5.1 Subsystem Hardware

The Raspberry Pi 3

5.5.2 Subsystem Operating System

Raspbian OS

5.5.3 Subsystem Software Dependencies

openai and os libraries. The OS library being used for file management of the the existing chat logs and instructions which we are feeding our LLM.

5.5.4 Subsystem Programming Languages

Python

5.5.5 Subsystem Data Structures

In each text file that holds the chat logs there are 2 majors sections for each entry. The first section is the "role" whether if this message being feed to openai is from the system/assistant (these are the instructions we give to openai on how to act) or a user (these are the chatlogs for the users). The other section is the "content" or the log itself either containing what the user said or what the bot said in the past in response to what the user said.

5.5.6 Subsystem Data Processing

Before a message is read from the bot it first checks if any preexisting chat logs exist for that user if not it will read our instructions.txt file which will have our instructions to chatgpt on how it should be have plus any additional chat logs we will use to train our bot. If there are preexisting chat logs then it will load them all into openai.

5.6 READING MESSAGES FROM USER AND SENDING RESPONSES

This section of the code will interpret the message the user sends to it and give an appropriate response.

5.6.1 Subsystem Hardware

The Raspberry Pi 3

5.6.2 Subsystem Operating System

Raspbian OS

5.6.3 Subsystem Software Dependencies

openai and os libraries. The OS library being to add each message to the chatlog file for that user.

5.6.4 Subsystem Programming Languages

Python

5.6.5 Subsystem Data Structures

Same data structures as the LLM READING OUR CHATLOGS section, since this section also uses the same text file to form the replies.

5.6.6 Subsystem Data Processing

So after a message is read, that message will be added to that user's chatlog text file. The openai will reread that user's chatlog text file and then form a response from parsing that data. After replying the bots reply will also be added to that user's text file.

6 A Layer Subsystems

This layer is primarily focused on interactions and any communication between the system and the scammers with the external LLM code. And the subsystem uses python, Openai and other software dependencies to run its functions.

6.1 Layer Hardware

The communication layer is executed with the help of a Raspberry Pi 3.

6.2 Layer Operating System

The layer and its subsystems runs on Raspbian OS.

6.3 Layer Software Dependencies

The layer has software dependencies like OpenAi library and standard python libraries that help fetch data/messages and send it to the scammer.

6.4 Subsystem 1 - API Call

The API call layer is a web service designed to manage communication between the chat bot through the integration of the LLM API that generates responses.

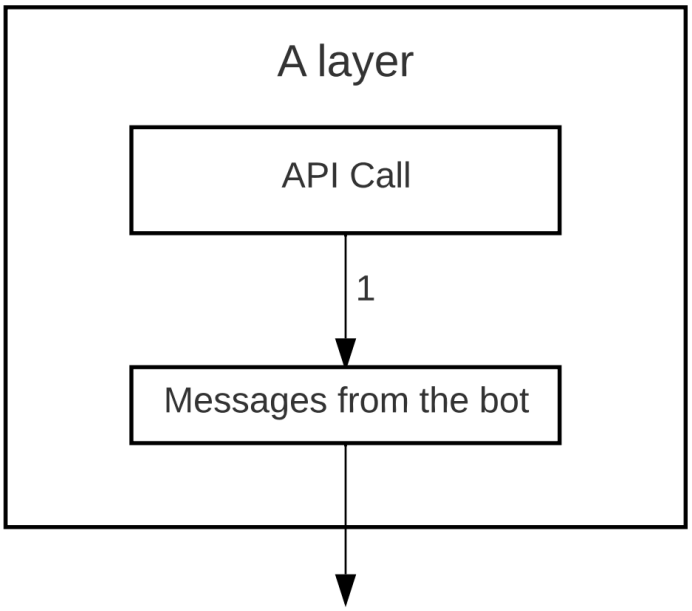

Figure 5: Example subsystem description diagram

6.4.1 Layer Hardware

Raspberry Pi 3

6.4.2 Subsystem Operating System

The subsystem runs on Raspbian OS.

6.4.3 Subsystem Software Dependencies

This subsystem's software dependencies primarily focus on integrating the API with the code base through openai library along with standard Python libraries and package management such as pip,

sys, and subprocess. The layer also utilizes packages such as distro, normalizer, tqdm and httpx that provide synchronous and asynchronous capabilities.

6.4.4 Subsystem Programming Languages

The functions of this subsystem are programmed with the help of Python.

6.4.5 Subsystem Data Structures

In the subsystem, data transmitted to and from the OpenAI API is structured as JSON objects, encapsulating fields like prompts, model specifications, and authentication tokens for requests, and generated text along with metadata for responses. These JSON packets facilitate structured communication between the subsystem and external APIs, ensuring data is correctly formatted and understood by both parties.

6.4.6 Subsystem Data Processing

The subsystem responds to the request that the API makes and successfully converts the response data from json format to a text form.

6.5 Subsystem 2 - Messages from the bot

The messages from the bot is a software component because it retrieves messages from the API and sends those messages to an individual (scammer) and it tailors messages to specific queries from the scammer. It also consists of programmable instructions and logic set in place to fetch data.

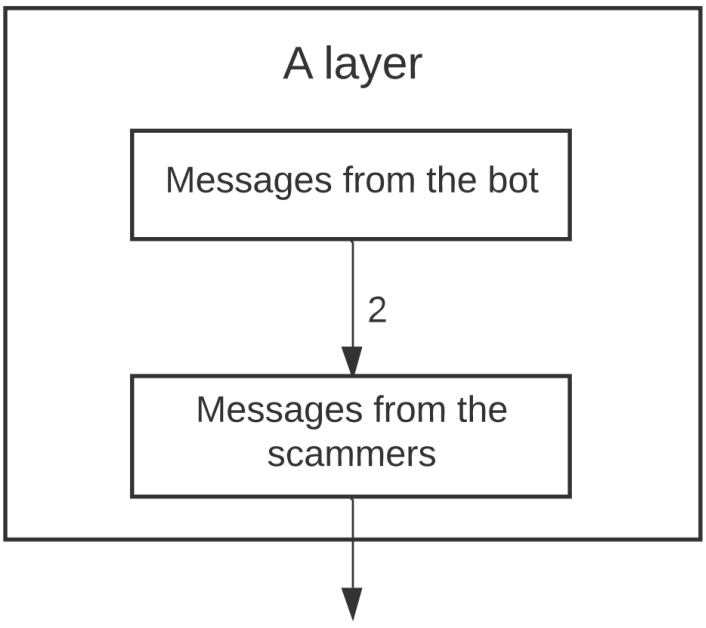

Figure 6: Example subsystem description diagram

6.5.1 Layer Hardware

Raspberry Pi 3

6.5.2 Subsystem Operating System

The subsystem runs on Raspbian OS.

6.5.3 Subsystem Software Dependencies

This subsystem uses the Openai library along with standard Python libraries and Selenium to fetch messages from the API and forward them to the users. It automates the messages on the Facebook web page using selectors to extract the text or content of the conversations between the bot and the scammers.

6.5.4 Subsystem Programming Languages

The functions of this subsystem are programmed with the help of Python.

6.5.5 Subsystem Data Processing

RunMisSpell algorithm: It simulates a scenario where the messages are intentionally misspelled and processed as a response by the bot.

6.6 Subsystem 3 - Messages from the scammers

This subsystem is a software component because it flags potential scammers by analyzing suspicious messages and it involves pattern matching and keyword detection.

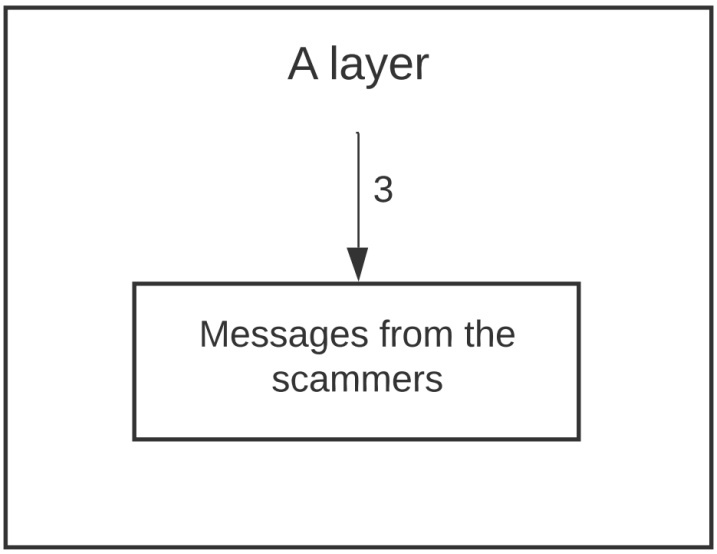

Figure 7: Example subsystem description diagram

6.6.1 Layer Hardware

Raspberry Pi 3

6.6.2 Subsystem Operating System

The subsystem runs on Raspbian OS.

6.6.3 Subsystem Software Dependencies

This subsystem uses the Openai library along with standard Python libraries and Selenium to detect new events in a conversation between the bot and scammer. This helps the bot to appropriately craft messages based on context.

6.6.4 Subsystem Programming Languages

A description of any programming languages used by the subsystem.

6.6.5 Subsystem Data Structures

This subsystem encapsulates all the relevant information about each message received by the scammer to eventually assess if the user is a scammer and if they raise any suspicions.

6.6.6 Subsystem Data Processing

The subsystem collects data from the scammer and analyzes it to make a prediction. Then it does decision logic by using a boolean value to flag true or false with each stored message to detect signs of fraudulent behavior.

7 B Layer Subsystems - Reporting System

7.1 Layer Hardware

Considering the nature of this system being entirely software, there is no specific hardware that will be used.

7.2 Layer Operating System

Windows will be used at the main operating system.

7.3 Layer Software Dependencies

Specific libraries and functions will be used in this system so that the functionality works as intended.

7.4 Subsystem 1

The Scammer Database is a more in depth step that forwards the process of detection and safety. Once the chatbot has detected any semblance of suspicious activity during its interaction with the individual it is in contact with, it will report and store their information into a database that marks them as a scammer.

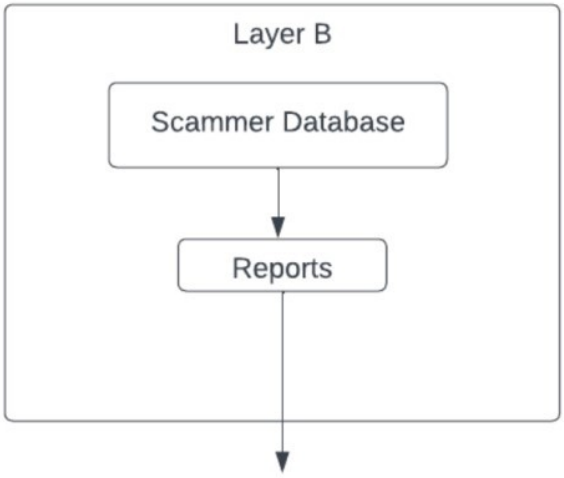

Figure 8: Example subsystem description diagram

7.4.1 Subsystem Hardware

A description of any involved hardware components for the subsystem.

7.4.2 Subsystem Operating System

Windows will be the main OS for this subsystem.

7.4.3 Subsystem Software Dependencies

This will depend on some libraries to make the right calls and to function properly.

7.4.4 Subsystem Programming Languages

The main language used is Python.

7.4.5 Subsystem Data Structures

The data is being received from the chatbot itself which notifies and sends more data into the database. These reports work in a two way structure letting systems connected to update.

7.4.6 Subsystem Data Processing

The reporting system has an algorithm that will reliably detect a potential scammer.

8 C Layer Subsystems

In this section, the layer is described in terms of the hardware and software design. Specific implementation details, such as hardware components, programming languages, software dependencies, operating systems, etc. should be discussed. Any unnecessary items can be ommitted (for example, a pure software module without any specific hardware should not include a hardware subsection). The organization, titles, and content of the sections below can be modified as necessary for the project.

8.1 Layer Hardware

The webscraping code will be run on a Raspberry pi 3.

8.2 Layer Operating System

The Raspberry pi is running a 32 bit version of Raspbian OS.

8.3 Layer Software Dependencies

We will be primarily using some general python libraries and Selenium.

8.4 Vulnerable profile detection

This section provides a detailed overview of the Vulnerable Profile Detection subsystem, a crucial part of the Web Scraping and Analysis Layer. It shows how the subsystem processes Facebook profiles to identify potentially vulnerable individuals and communicates with the Web Scraping Tool.

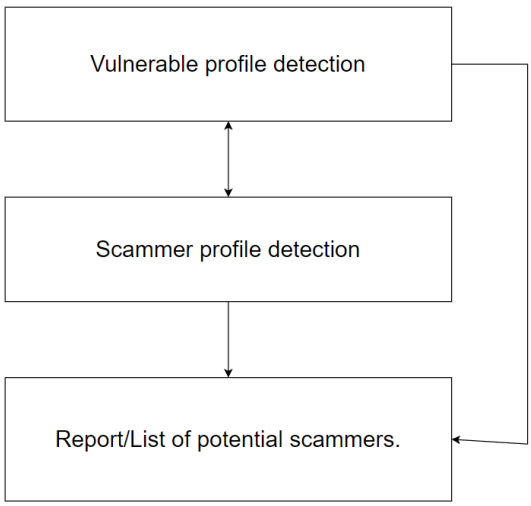

Figure 9: Example subsystem description diagram

8.4.1 Subsystem Hardware

We will primarily use a Raspberry pi 3 for this subsystem.

8.4.2 Subsystem Operating System

The Raspberry pi is running a 32 bit version of Raspbian OS.

8.4.3 Subsystem Software Dependencies

We will be primarily using some general python libraries and Selenium.

8.4.4 Subsystem Programming Languages

Python

8.4.5 Subsystem Data Structures

The datastructures used is that we will document everything collected, onto a database, using firebase.

8.4.6 Subsystem Data Processing

Python libraries and Selenium.

8.5 Scammer profile detection

This section provides a detailed overview of the Vulnerable Profile Detection subsystem, a crucial part of the Web Scraping and Analysis Layer. It shows how the subsystem processes Facebook profiles to identify potentially vulnerable individuals and communicates with the Web Scraping Tool.

8.5.1 Subsystem Hardware

We will primarily use a Raspberry pi 3 for this subsystem.

8.5.2 Subsystem Operating System

The Raspberry pi is running a 32 bit version of Raspbian OS.

8.5.3 Subsystem Software Dependencies

We will be primarily using some general python libraries and Selenium.

8.5.4 Subsystem Programming Languages

Python

8.5.5 Subsystem Data Structures

The datastructures used is that we will document everything collected, onto a database, using firebase.

8.5.6 Subsystem Data Processing

Python libraries and Selenium.

8.6 Report/List of potential scammers

This section provides a detailed overview of the Vulnerable Profile Detection subsystem, a crucial part of the Web Scraping and Analysis Layer. It shows how the subsystem processes Facebook profiles to identify potentially vulnerable individuals and communicates with the Web Scraping Tool.

8.6.1 Subsystem Hardware

We will primarily use a Raspberry pi 3 for this subsystem.

8.6.2 Subsystem Operating System

The Raspberry pi is running a 32 bit version of Raspbian OS.

8.6.3 Subsystem Software Dependencies

We will be primarily using some general python libraries and Selenium.

8.6.4 Subsystem Programming Languages

Python

8.6.5 Subsystem Data Structures

The datastructures used is that we will document everything collected, onto a database, using firebase.

8.6.6 Subsystem Data Processing

Python libraries and Selenium.

9 Appendix A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES