**PENTESTER TOOLKIT**

**APP**

**TECHNICAL REPORT**



**Supervisor**

Ms. Sidra Shahid

**Submitted By**

Muhammad Ahtesham Sarwar

2019-ag-6068

Muhammad Khawar Azeem

2019-ag-6067

**A TECHNICAL REPORT OF FYP SUBMITTED FOR THE DEGREE OF**

*BACHELOR OF SCIENCE*

*IN*

*COMPUTER SCIENCE*

**DEPARTMENT OF COMPUTER SCIENCE**

**FACULTY OF SCIENCES**

**UNIVERSITY OF AGRICULTURE FAISALABAD**

**DECLARATION**

I hereby declare that the contents of the report **PENTESTER TOOLKIT** are project of my own research and no part has been copied from any published source (except the references). I further declare that this work has not been submitted for award of any other diploma/degree. The university may take action if the information provided is found false at any stage. In case of any default the scholar will be proceeded against as per UAF policy.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Ahtesham Sarwar

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Khawar Azeem

**CERTIFICATE**

To,

The Controller of Examinations,

University of Agriculture,

Faisalabad.

The supervisory committee certify that **MUHAMMAD AHTESHAM SARWAR [2019-ag-6068]** and **MUHAMMAD KHAWAR AZEEM [2019-ag-6067]** has successfully completed his **FYP** for the degree of **B.S. Computer Science**under our guidance and supervision.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ms. Sidra Shahid

Supervisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Ahtesham Sarwar

Member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Muhammad Khawar Azeem

Member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Muhammad Ahsan Latif

Incharge,

Department of Computer Science

**ACKNOWLEDGEMENT**

I thank all who in one way or another contributed in the completion of this report. First, I thank to **ALLAH ALMIGHTY**, most magnificent and most merciful, for all his blessings. Then I am so grateful to the **Department of Computer Science** for making it possible for me to study here. My special and heartily thanks to my supervisor, **Ms. Sidra Shahid**who encouraged and directed me. Her challenges brought this work towards a completion. It is with her supervision that this work came into existence. For any faults I take full responsibility. I am also deeply thankful to my informants. I want to acknowledge and appreciate their help and transparency during my research. I am also so thankful to my fellow students whose challenges and productive critics have provided new ideas to the work. Furthermore, I also thank my family who encouraged me and prayed for me throughout the time of my research. May the **Almighty God** richly bless all of you.

**ABSTRACT**

The **PENTESTER TOOLKIT** app is a mobile application designed to provide a comprehensive set of tools for web application and network penetration testing. This technical report outlines the development and functionality of the app, targeting Cyber Security Engineers and IT professionals in need of a convenient, mobile solution for their penetration testing tasks.

The report begins with an introduction, highlighting the background and motivation behind the development of the app. It emphasizes the current lack of similar solutions for mobile devices and the importance of bringing these essential tools to the fingertips of Cyber Security Engineers. The scope of the project is defined, focusing on the design, development, and testing of the app for Android devices.

The functional requirements section details the specific features and capabilities of the app. It covers tools such as Http Monitor, Site Mapper, Intruder, Repeater, Ping Utility, Port Scan, Encoder, Hasher, Google Dorking, Http Server, Privacy, and Notes. Each requirement specifies the expected functionalities, including monitoring Http requests, performing automated testing, encoding and decoding data, and searching Google using advanced operators.

The report highlights the rapid application development (RAD) approach used for the project, allowing for iterative improvements based on user feedback. The development process prioritizes user-friendliness, ease of use, and compatibility with Android devices running Android 6.0 or higher.

The main contributions of the project include the user-friendly app interface, the implementation of a wide range of tools for penetration testing, compatibility testing on various Android devices, and the incorporation of user feedback for iterative improvements.

In conclusion, this technical report presents the development and functionality of the **PENTESTER TOOLKIT** app. It outlines the problem addressed, the methodology employed, the results achieved, and the main conclusions and recommendations. The app offers a convenient and mobile solution for Cyber Security Engineers and IT professionals engaged in web application and network penetration testing, empowering them with essential tools accessible from their Android devices.

Table of Contents

[**INTRODUCTION** 1](#_Toc136517515)

[1.1 **Background** 1](#_Toc136517516)

[1.2 **Description** 1](#_Toc136517517)

[1.3 **Scope** 2](#_Toc136517518)

[1.4 **Objectives** 3](#_Toc136517519)

[**REQUIREMENTS** 5](#_Toc136517520)

[2.1 **Functional Requirements** 5](#_Toc136517521)

[2.2 **Non- Functional Requirements** 7](#_Toc136517522)

[2.3 **Hardware Requirements** 8](#_Toc136517523)

[2.4 **Software Requirements** 8](#_Toc136517524)

[**METHODOLOGY** 9](#_Toc136517525)

[3.1 **Tools & Technologies** 10](#_Toc136517526)

[**TIMELINE** 11](#_Toc136517527)

[**DESIGN** 12](#_Toc136517528)

[5.1 **Use Case Diagrams** 12](#_Toc136517529)

[5.1.1 Usage Scenario 19](#_Toc136517530)

[5.2 **Class Diagrams** 25](#_Toc136517531)

[5.3 **Data Flow Diagrams** 33](#_Toc136517532)

[5.4 **Database Model** 44](#_Toc136517533)

[**RESULTS & DISCUSSION** 45](#_Toc136517534)

[6.1 **Test Cases** 45](#_Toc136517535)

[6.2 **Conclusion** 51](#_Toc136517536)

# 

# **INTRODUCTION**

## 1.1 **Background**

The **PENTESTER TOOLKIT** app is being developed in response to the need for a convenient, mobile solution for **Cyber Security Engineers** to access important tools for Web Application and Network Penetration Testing. Currently, there are various toolkits available for desktop devices, but there is a lack of similar solutions for mobile devices. The **PENTESTER TOOLKIT** app aims to fill this gap and provide mobility and convenience to Cyber Security Engineers by bringing these important tools to their mobile devices.

## 1.2**Description**

The **PENTESTER TOOLKIT** app will be a mobile application designed to help Cyber Security Engineers in the field of Penetration Testing. The app aims to provide a single, convenient location for accessing a range of important tools for web application and network penetration testing, including:

* Http Monitor
* Site Mapper
* Intruder
* Repeater
* Ping Utility
* Port Scan
* Encoder
* Hasher
* Google Dorking
* Http Server
* Privacy
* Notes

The main goal of the app is to provide mobility and ease to Cyber Security Engineers, allowing them to access these tools from their mobile devices rather than relying on desktop computers. The app will be developed using Java and Android Studio, with a focus on user-friendliness and ease of use.

One of the key challenges in developing the **PENTESTER TOOLKIT** app will be balancing the need for a wide range of features and tools with the need to keep the app lightweight and easy to use. The team will address this challenge by carefully prioritizing and selecting the tools to be included in the app, and by implementing an **RAD (Rapid Application Development)** development process that allows for rapid prototyping and iterative improvements based on user feedback.

The app will be of interest to a wide range of people and organizations, including Cyber Security Engineers, IT professionals, and businesses with a need for web application and network penetration testing. The app will be released on the Google Play Store and will be available for download by anyone with an Android device running Android 6.0 or higher.

## 1.3 **Scope**

The scope of the **PENTESTER TOOLKIT** app project includes the design, development, and testing of a mobile application for Android devices that provides a range of tools for web application and network penetration testing. The app will include the following features:

* **Http Monitor:**

Allows users to monitor Http requests and responses in real-time, including the ability to view request and response headers, bodies, and parameters.

* **Site Mapper:**

Allows users to create a map of a website's structure and links, either through automated crawling or user-directed exploration.

* **Intruder:**

Allows users to perform automated testing of web application vulnerabilities by performing various types of brute force attacks like sniper attack, pitch fork, cluster bomb etc.

* **Repeater:**

Allows users to manually send customized Http requests repeatedly to a server.

* **Ping Utility:**

Allows users to test the availability and response time of a server or website by sending a series of ping requests.

* **Port Scan:**

Allows users to scan a network for open ports, useful for identifying potential vulnerabilities.

* **Encoder:**

Allows users to encode and decode data using various algorithms like Base64, Hex, URL and HTML encoding.

* **Hasher:**

Allows users to calculate hash values for data using various hashing algorithms like MD5, SHA-1, and SHA-2 etc.

* **Google Dorking:**

Allows users to search the internet using advanced search operators and keywords, useful for finding hidden or sensitive information.

* **Http Server:**

Allows users to set up a local Http server, useful for testing and development purposes.

* **Privacy:**

Allows users to mask their identity and location while using the app, useful for preserving privacy and avoiding detection using proxies.

* **Notes**

Allows users to save the notes or payloads for various attacks in one place for later use.

The scope of the project does not include the development of any additional tools or features beyond those listed above. The app will be developed and tested on Android devices running Android 6.0 or higher and will be released on the Google Play Store. The app will be developed using a **RAD (Rapid Application Development)** approach, with a focus on rapid prototyping and iterative improvements based on user feedback.

## 1.4 **Objectives**

The goal of the **PENTESTER TOOLKIT** app project is to develop a mobile application for Android devices that provides a range of tools for web application and network penetration testing. The objectives of the project are as follows:

* Develop a user-friendly app interface that is easy to navigate and use.
* Implement a range of tools for web application and network penetration testing, including Http Request & Response Monitor, Site Mapper, Http Requests Intruder, Http Requests Repeater, Ping Utility, Port Scan, Encoder & Hasher, Google Dorking, and Http Server.
* Test the app on a range of Android devices running Android 6.0 or higher to ensure compatibility and performance.
* Gather user feedback and make iterative improvements to the app based on this feedback.
* Publish the app on the Google Play Store for download by users.

To achieve these objectives, the project will involve the design, development, and testing of the app, as well as the gathering and incorporation of user feedback. The project will follow an **RAD (Rapid Application Development)** development approach, with a focus on rapid prototyping and iterative improvements. The team will work collaboratively to design and develop the app, using RAD practices. The app will be tested on a range of Android devices to ensure compatibility and performance, and user feedback will be gathered and incorporated into future versions of the app. Once the app is complete, it will be published on the Google Play Store for download by users.

# 

# **REQUIREMENTS**

## 2.1 **Functional Requirements**

The functional requirements for the **PENTESTER TOOLKIT** app project are as follows:

**FR01: Http Monitor**

|  |  |
| --- | --- |
| FR01-01 | The app shall allow users to monitor Http requests and responses in real-time. |
| FR01-02 | The app shall display request and response headers, bodies, and parameters. |
| FR01-03 | The app shall allow users to filter and search Http requests and responses by various criteria. |
| FR01-04 | The app shall allow users to copy, save and export Http requests and responses for later analysis. |

**FR02: Site Mapper**

|  |  |
| --- | --- |
| FR02-01 | The app shall allow users to create a map of a website's structure and links. |
| FR02-02 | The app shall allow users to crawl a website automatically to discover links and content. |
| FR02-03 | The app shall allow users to manually explore a website and add links and content to the map. |
| FR02-04 | The app shall display the map in a graphical format, with links and content organized hierarchically. |

**FR03: Intruder**

|  |  |
| --- | --- |
| FR03-01 | The app shall allow users to perform automated testing of web application. |
| FR03-02 | The app shall support various types of brute force attacks like Sniper attack, Pitch Fork attack, Cluster Bomb attack etc. |
| FR03-03 | The app shall allow users to customize and save payloads for use in testing. |
| FR03-04 | The app shall allow users to customize the frequency and number of requests. |
| FR03-05 | The app shall display the results of the requests, including response times and status codes. |

**FR04: Repeater**

|  |  |
| --- | --- |
| FR04-01 | The app shall allow users to manually send customized Http request to a server. |
| FR04-02 | The app shall display the results of the request, including response time and status code. |
| FR04-03 | The app shall allow users to save and export the results of the requests. |
| FR04-04 | The app shall allow users to search for specific queries in the response of request. |

**FR05: Ping Utility**

|  |  |
| --- | --- |
| FR05-01 | The app shall allow users to test the availability and response time of a server or website. |
| FR05-02 | The app shall send a series of ping requests to the target server or website. |
| FR05-03 | The app shall display the results of the ping requests including response time. |
| FR05-04 | The app shall allow users to save and export the results of the ping requests. |

**FR06: Port Scan**

|  |  |
| --- | --- |
| FR06-01 | The app shall allow users to scan a range of ports on a server or device. |
| FR06-02 | The app shall allow users to customize the range of ports to scan and the timeout for responses. |
| FR06-03 | The app shall display the results of the port scan, including open and closed ports. |
| FR06-04 | The app shall allow users to save and export the results of the port scan. |

**FR07: Encoder**

|  |  |
| --- | --- |
| FR07-01 | The app shall allow users to encode and decode data using various encoding algorithms. |
| FR07-02 | The app shall support a range of encoding algorithms, including Base64, Hex, URL and HTML. |
| FR07-03 | The app shall allow users to input and output data in various formats, including text, hexadecimal, and binary. |
| FR07-04 | The app shall allow users to compare the results of different encoding algorithms. |

**FR07: Hasher**

|  |  |
| --- | --- |
| FR07-01 | The app shall allow users to calculate hash using various hashing algorithms. |
| FR07-02 | The app shall support a range of hashing algorithms, including SHA-1, SHA-2, and MD5. |
| FR07-03 | The app shall allow users to input and output data in various formats, including text, hexadecimal, and binary. |
| FR07-04 | The app shall allow users to compare the results of different hashing algorithms. |

**FR08: Google Dorking**

|  |  |
| --- | --- |
| FR08-01 | The app shall allow users to search Google using advanced search operators. |
| FR08-02 | The app shall provide a list of common Google search operators and their usage. |
| FR08-03 | The app shall allow users to customize and save search queries for reuse. |
| FR08-04 | The app shall display the results of Google searches, including links and snippets of text. |

**FR09: Http Server**

|  |  |
| --- | --- |
| FR09-01 | The app shall allow users to create a local Http server on their device. |
| FR09-02 | The app shall allow users to browse and edit the files and directories on the local server. |
| FR09-03 | The app shall allow users to access the local server using a web browser. |
| FR09-04 | The app shall allow users to set permissions for accessing and modifying the local server. |

**FR10: Privacy**

|  |  |
| --- | --- |
| FR10-01 | The app shall allow users to route their Http traffic through the Tor network or a proxy server. |
| FR10-02 | The app shall show users a list of available Proxies / Tor relays. |

**FR11: Notes**

|  |  |
| --- | --- |
| FR11-01 | The app shall allow users to create, edit, and delete payloads for use in testing web application vulnerabilities. |
| FR11-02 | The app shall allow users to categorize payloads and add notes for each payload. |
| FR11-03 | The app shall allow users to import and export payloads from and to external files. |
| FR11-04 | The app shall allow users to search and filter payloads by various criteria. |

## 2.2 **Non- Functional Requirements**

Some non-functional requirements for the **PENTESTER TOOLKIT** app:

|  |  |
| --- | --- |
| **NFR01** | The app shall have a response time of less than 3 seconds for all tools and features. |
| **NFR02** | The app shall have a user-friendly interface with clear navigation and intuitive controls. |
| **NFR03** | The app shall support offline usage, with all necessary tools and resources available without an internet connection. |
| **NFR04** | The app shall be compatible with Android 6.0 or higher devices. |
| **NFR05** | The app shall have minimal battery and resource usage. |
| **NFR06** | The app shall have a small footprint, taking up minimal storage space on the user's device. |
| **NFR07** | The app shall be updated regularly with new tools and features. |
| **NFR08** | The app shall be compliant with relevant regulations and industry standards for cybersecurity. |
| **NFR09** | The app shall be easy to maintain, with clear documentation and a simple update process. |
| **NFR10** | The app shall be user-friendly, with intuitive controls and a clean, easy-to-navigate interface. |
| **NFR11** | The app shall be compatible with a variety of devices and screen sizes. |

2.3 **Hardware Requirements**

Some hardware requirements for the **PENTESTER TOOLKIT** app:

|  |  |
| --- | --- |
| **HR01** | The app shall require a minimum of 1GB of RAM to run smoothly. |
| **HR02** | The app shall require a minimum of 1GB of storage space to install and run. |
| **HR03** | The app shall support devices with a screen size of at least 480x800 pixels. |
| **HR04** | The app shall support devices with a processor speed of at least 1.2GHz. |
| **HR05** | The app shall support devices with Wi-Fi, if any tool or feature requires it. |

## 2.4 **Software Requirements**

Some software requirements for the **PENTESTER TOOLKIT** app:

|  |  |
| --- | --- |
| **SR01** | The app shall support Android 6.0 or higher. |
| **SR02** | The app shall support a variety of file formats for certain tools and features, such as TXT and JSON. |

# 

# 

# **METHODOLOGY**

For the **PENTESTER TOOLKIT** app, the development team has chosen to follow the **Rapid Application Development (RAD)** methodology. This methodology is particularly well-suited for the project because of its focus on rapid prototyping and iterative development.

The team will start by identifying the **key features** and **tools** that should be included in the app, and will then break these down into smaller, more manageable chunks that can be developed in parallel. Each of these chunks, or "**mini-projects**," will be time-boxed to ensure that they are delivered on schedule.

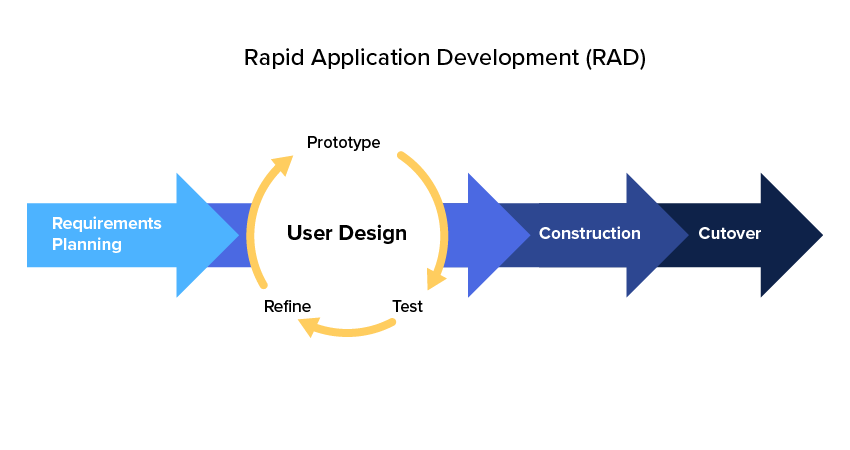
Once these mini-projects are completed, they will be **integrated** into a working prototype that can be tested and refined. This iterative process will continue until the final product is ready for release.

In addition to its focus on rapid prototyping and iterative development, the **RAD** methodology also emphasizes the importance of **feedback** and **collaboration**. The development team will work closely with potential audience to gather feedback and refine the app based on their needs and requirements.

Overall, the **RAD** methodology is a flexible and effective approach that will help the team deliver a high-quality, user-friendly app that meets the needs of cyber security professionals.

Phases of **PENTEST TOOLKIT** RAD (Rapid Application Development) Process**:**

* Requirements Gathering
* User Design
* Construction
* Testing & Integration
* Deployment



## 3.1**Tools & Technologies**

Here are some tools and technologies that may be used in the development of the **PENTESTER TOOLKIT** app:

* **Android Studio:**

This is the primary development environment for Android apps, and will be used to build and test the app.

* **Java:**

This is the primary programming language used for Android app development, and will be used to write the app's code.

* **XML:**

This markup language will be used to define the app's user interface and layout.

* **Git:**

This version control system will be used to manage the codebase and track changes made by the development team.

* **Gradle:**

This build tool will be used to automate tasks such as compiling the code, running tests, and generating the app's final package.

* **Third-party libraries and frameworks:**

The app may make use of various open-source libraries and frameworks to help with tasks such as networking, data storage, and testing.

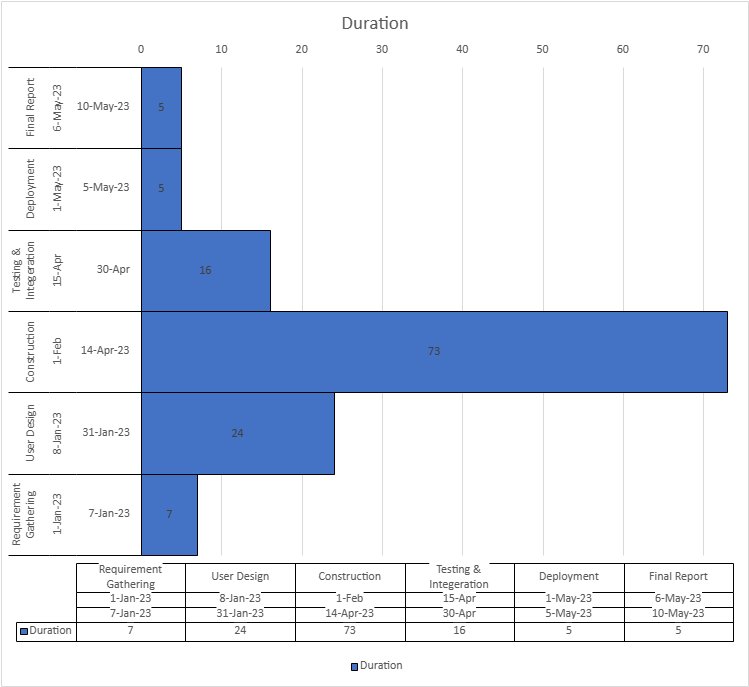
* **Proxy servers:**

The app may allow users to connect to the internet through a proxy server to enhance their anonymity and protect their privacy.

In addition to these tools and technologies, the development team may also use various resources and documentation to learn about best practices for Android app development and security testing. This may include online tutorials, documentation, and forums.

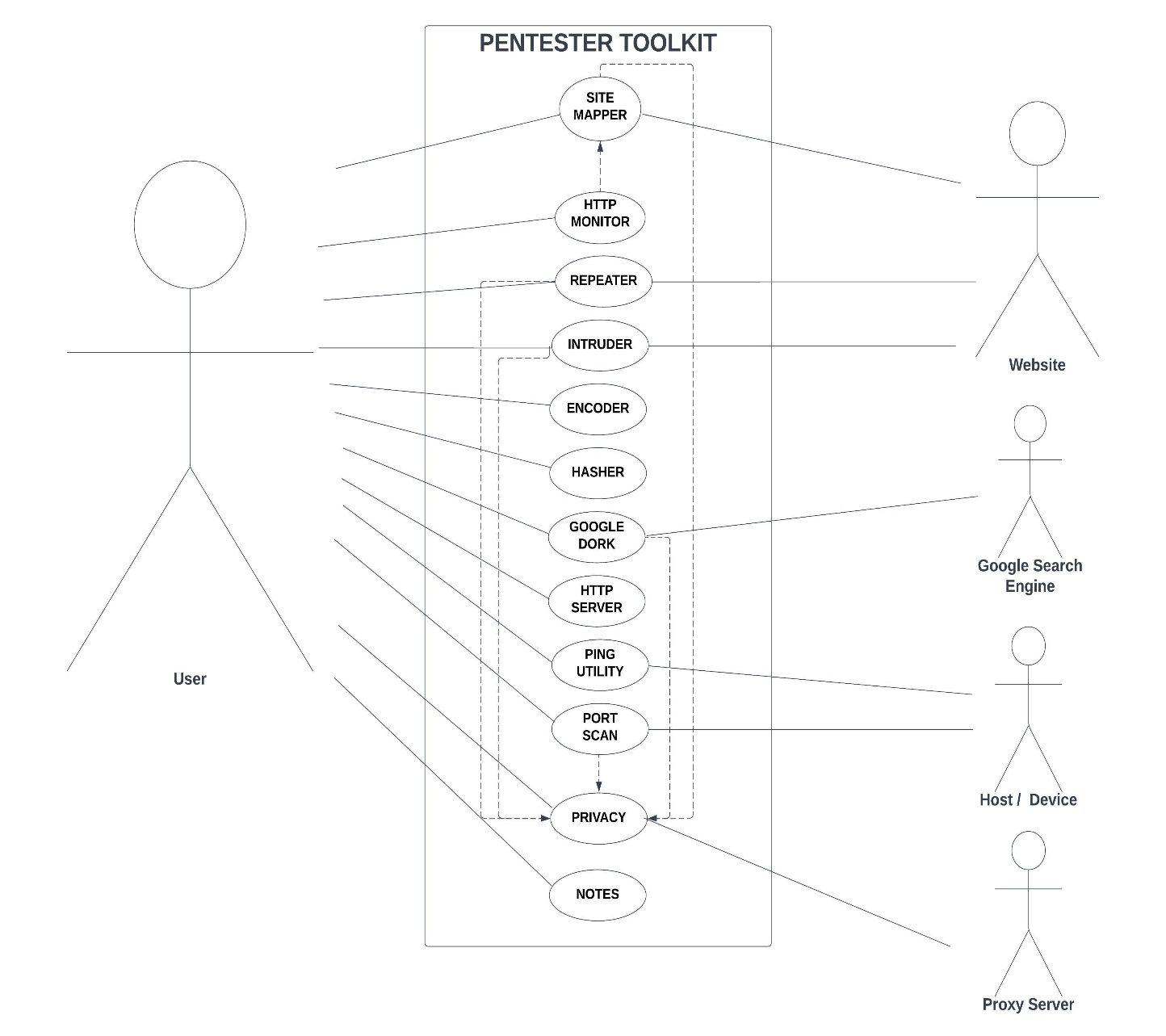
# **TIMELINE**

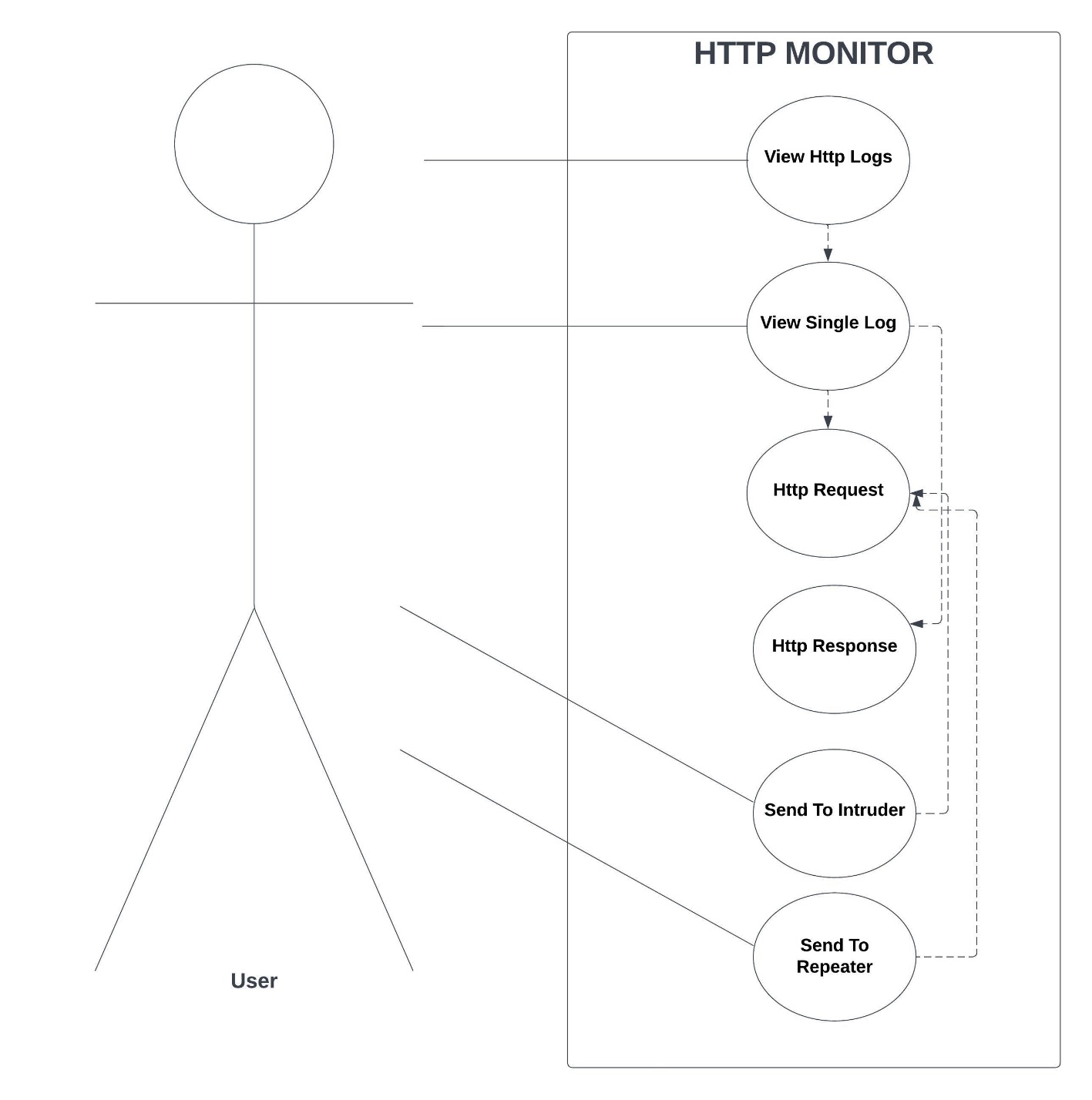
Timeline of the **PENTESTER TOOLKIT** app development process:

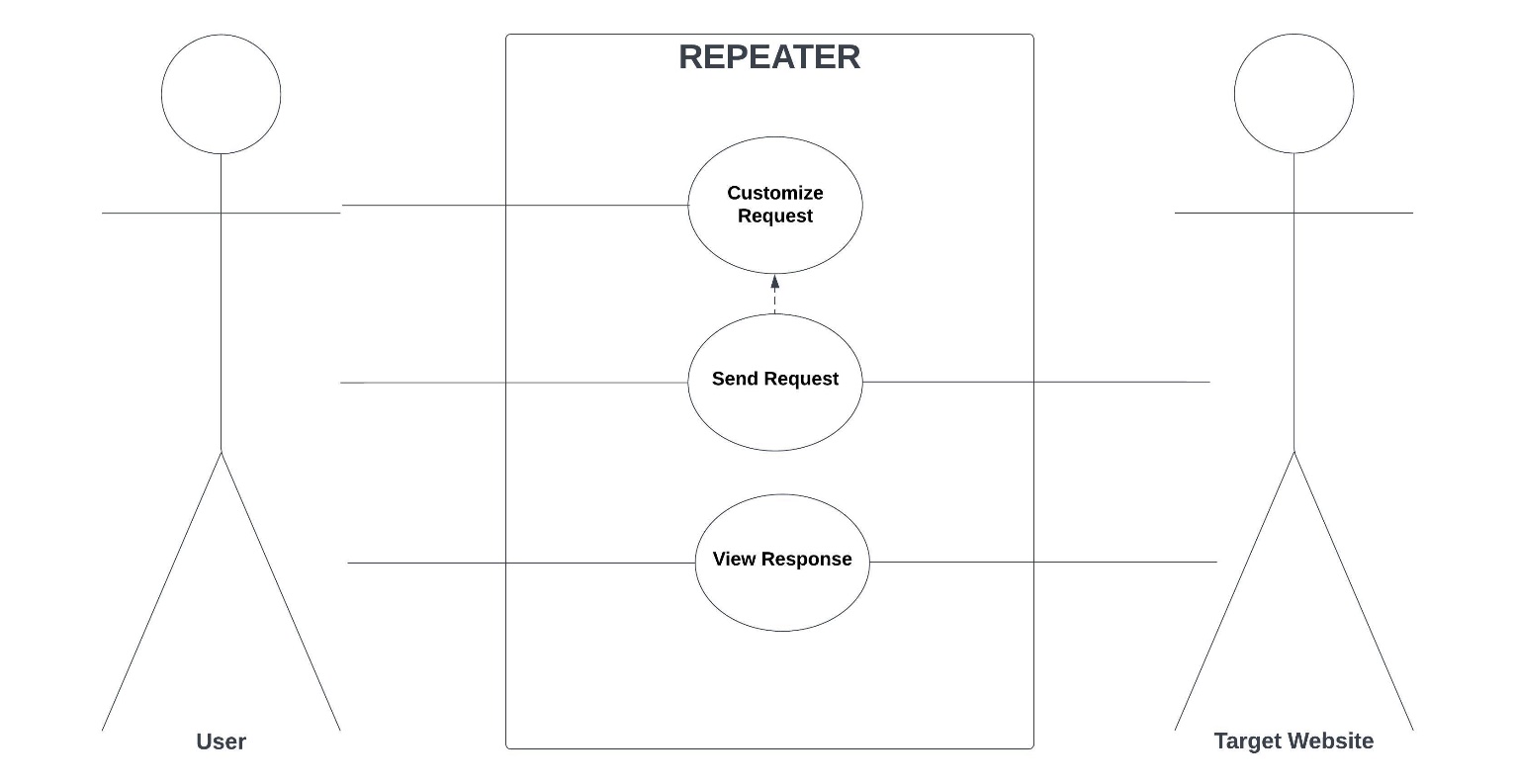


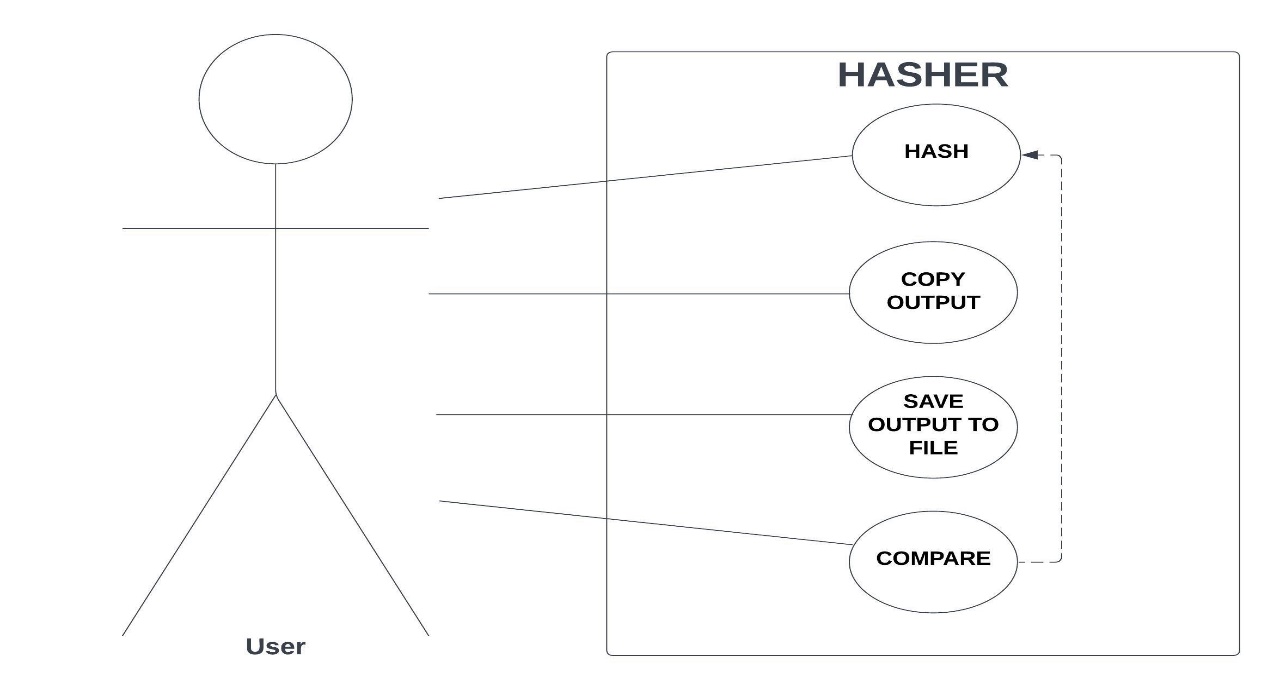
# **DESIGN**

## 5.1 **Use Case Diagrams**

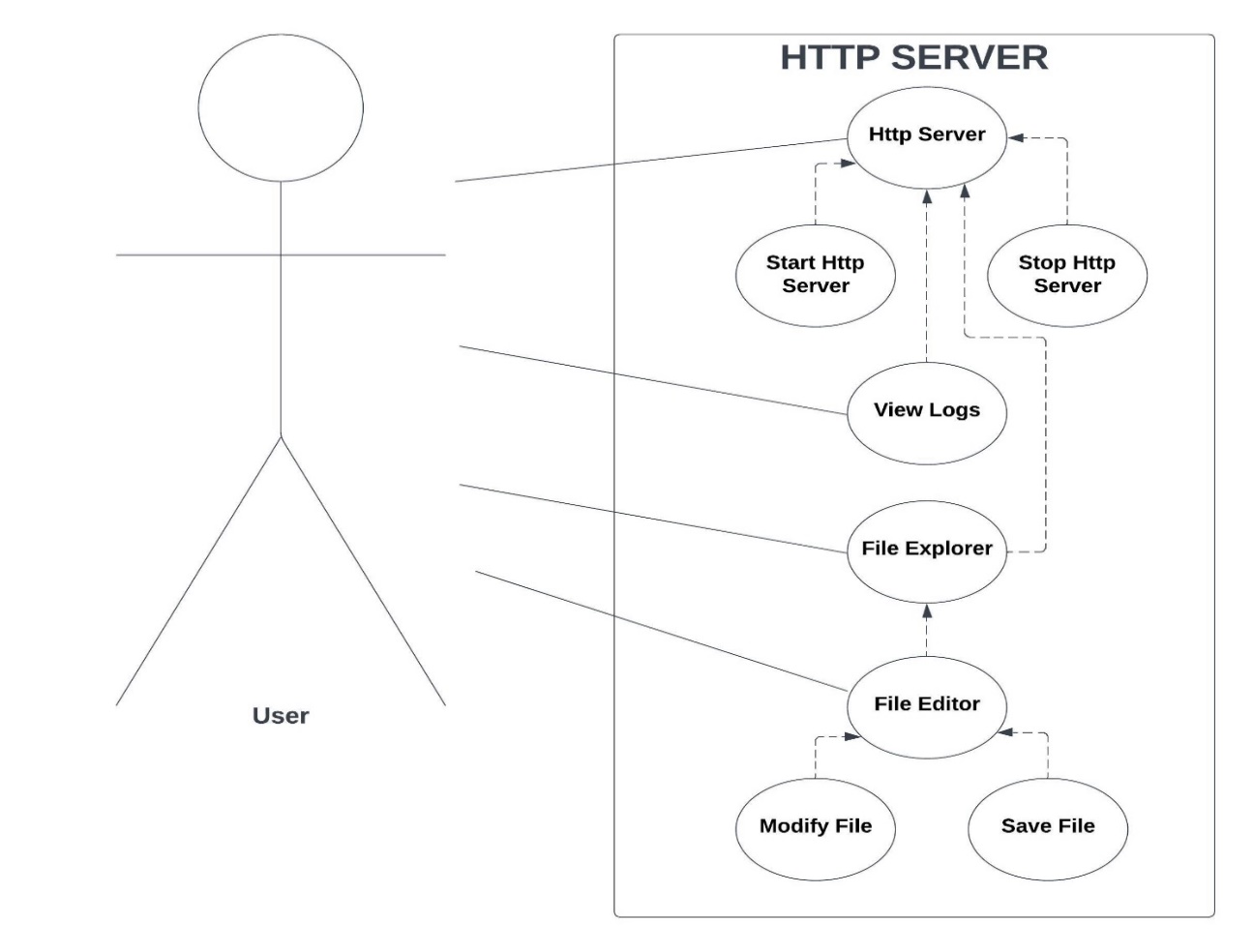
**PENTESTER TOOLKIT:**

**HTTP MONITOR:**

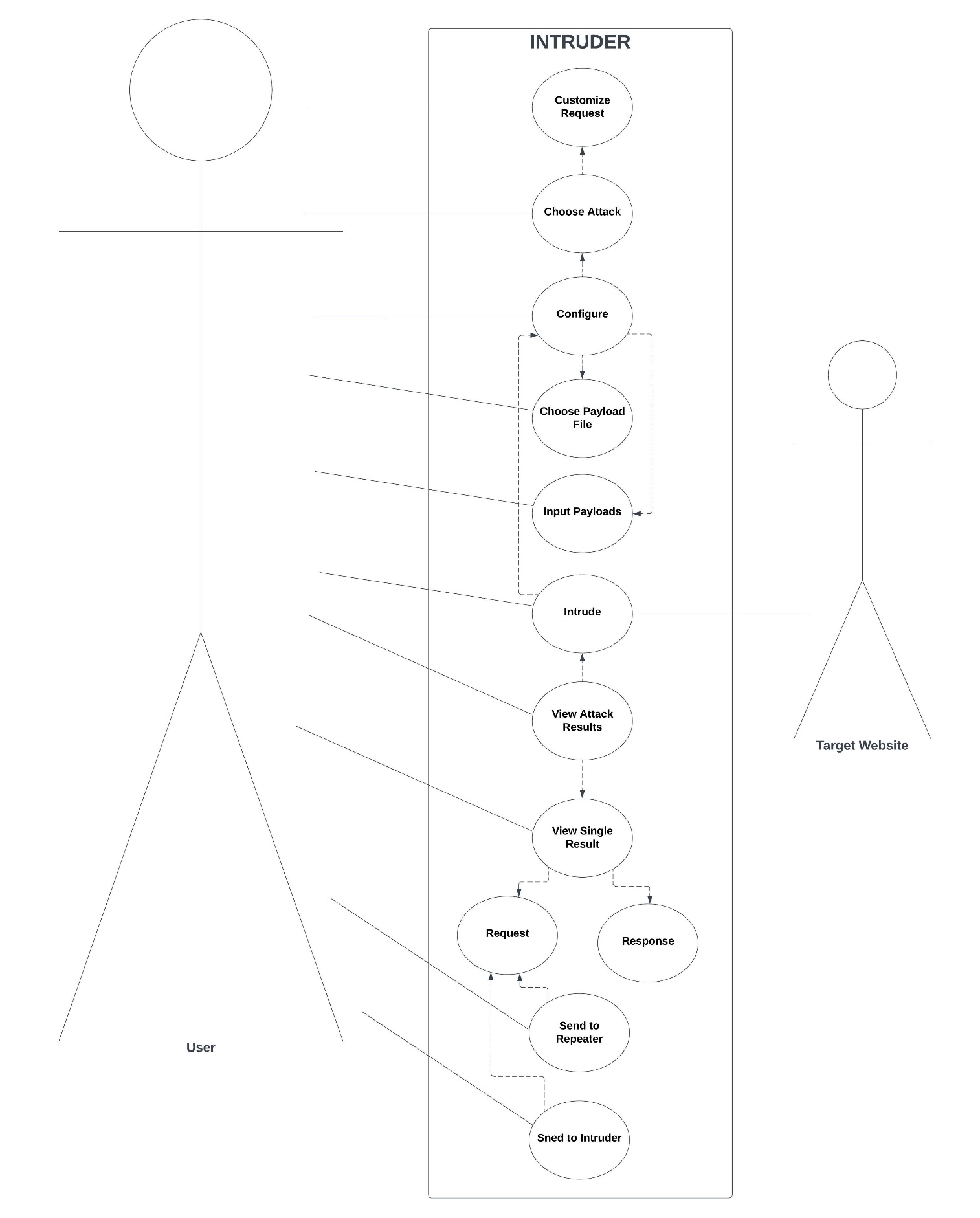
**HASHER:**



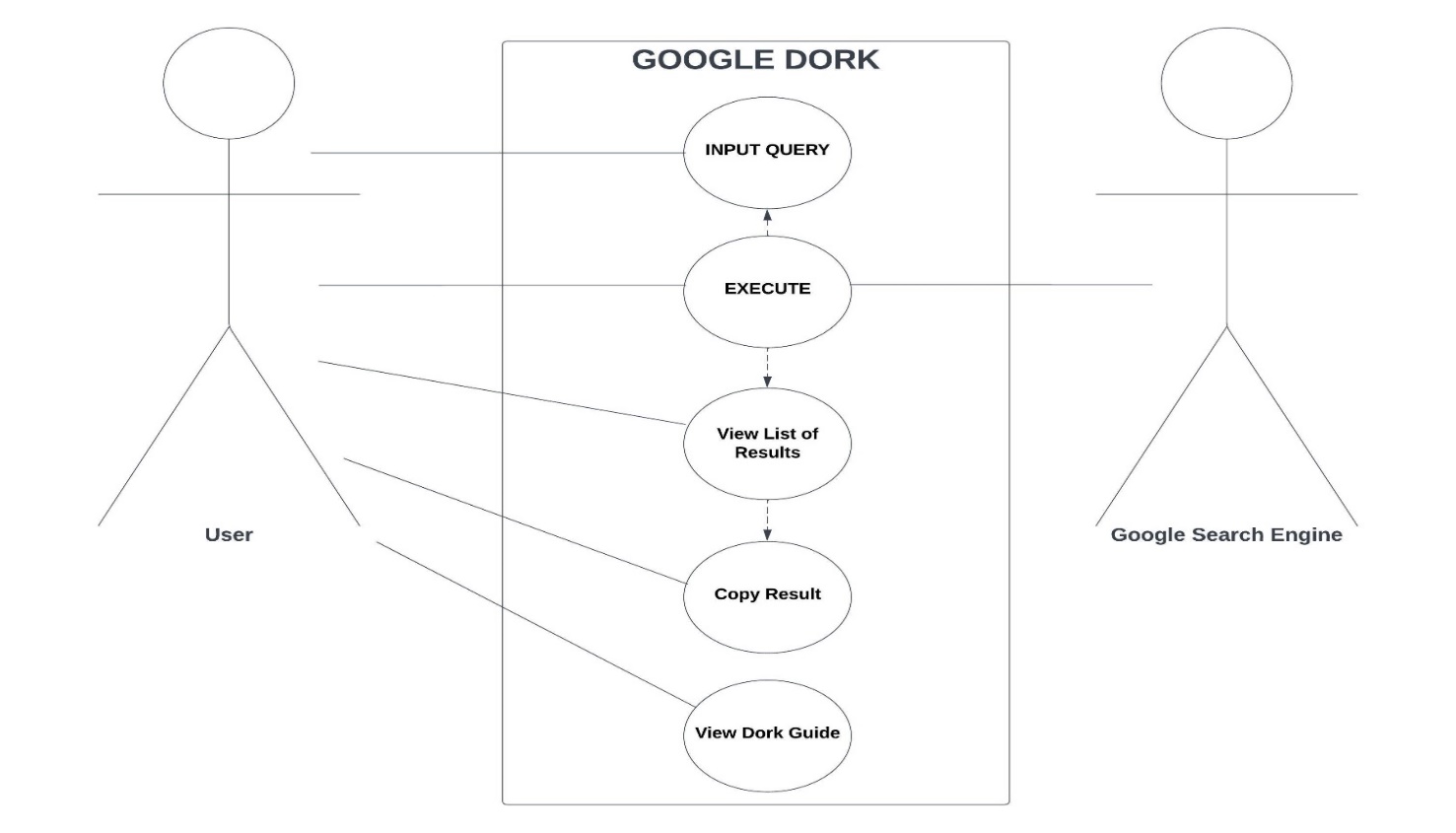
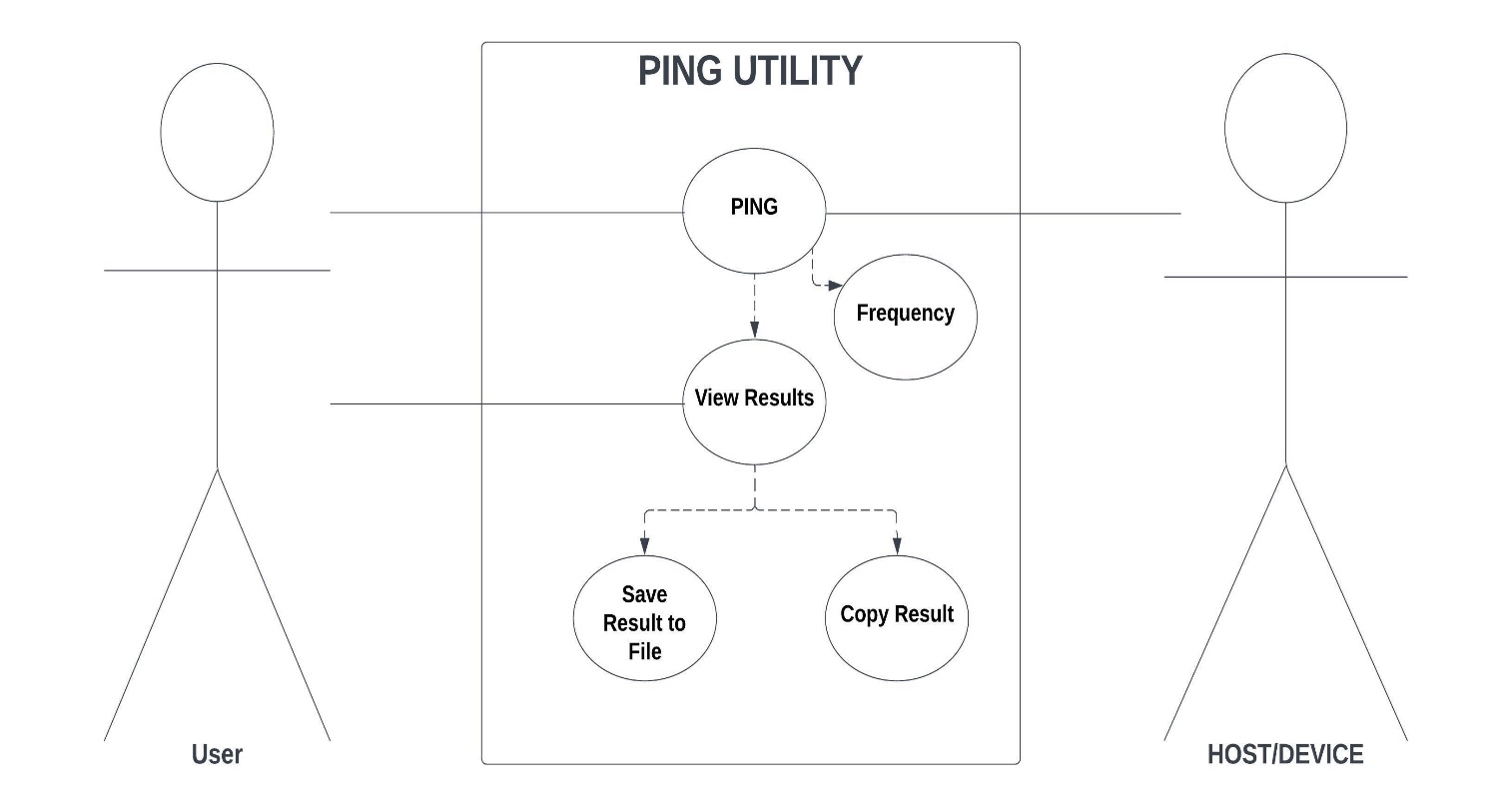
**HTTP SERVER:**

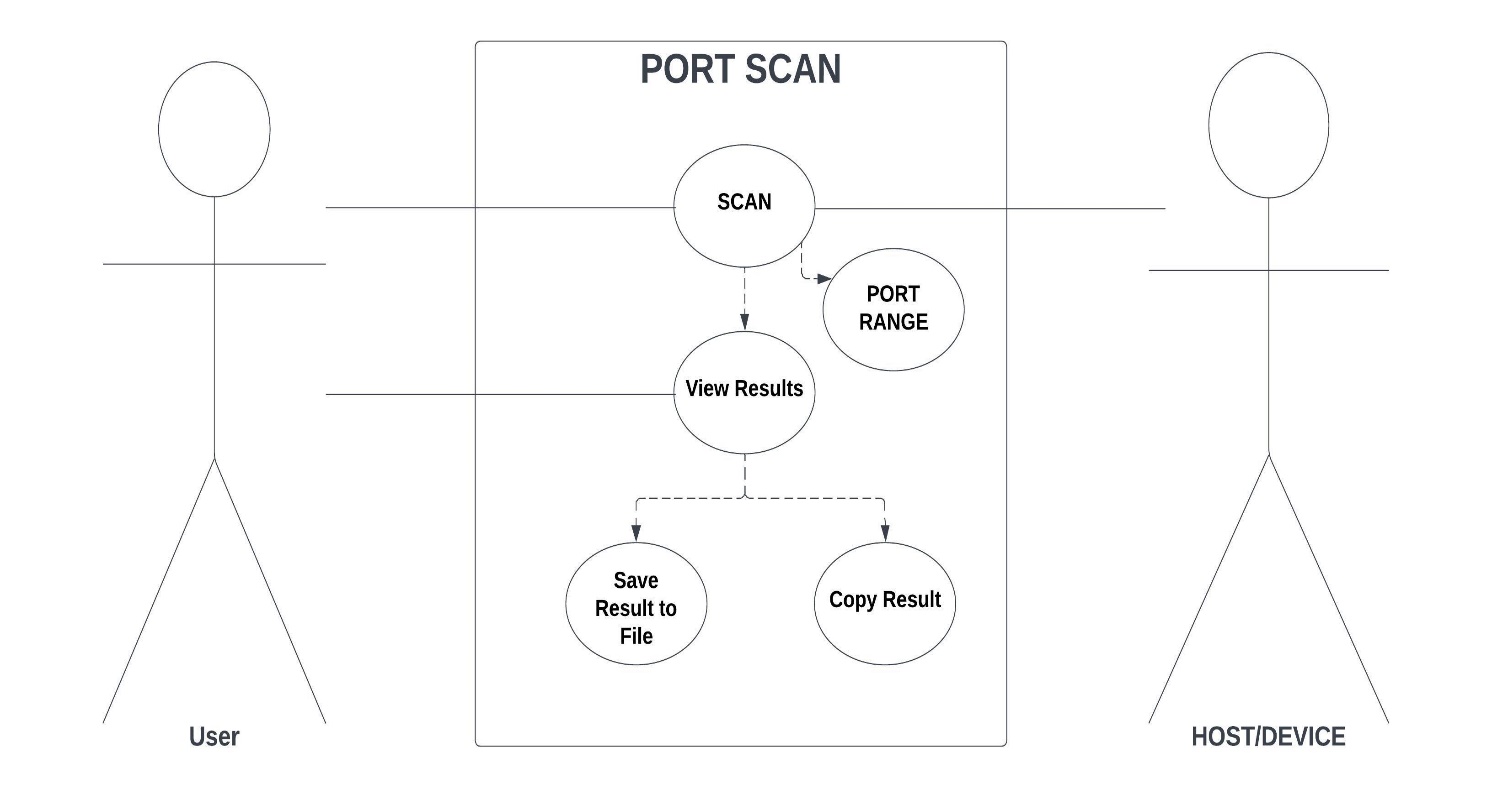
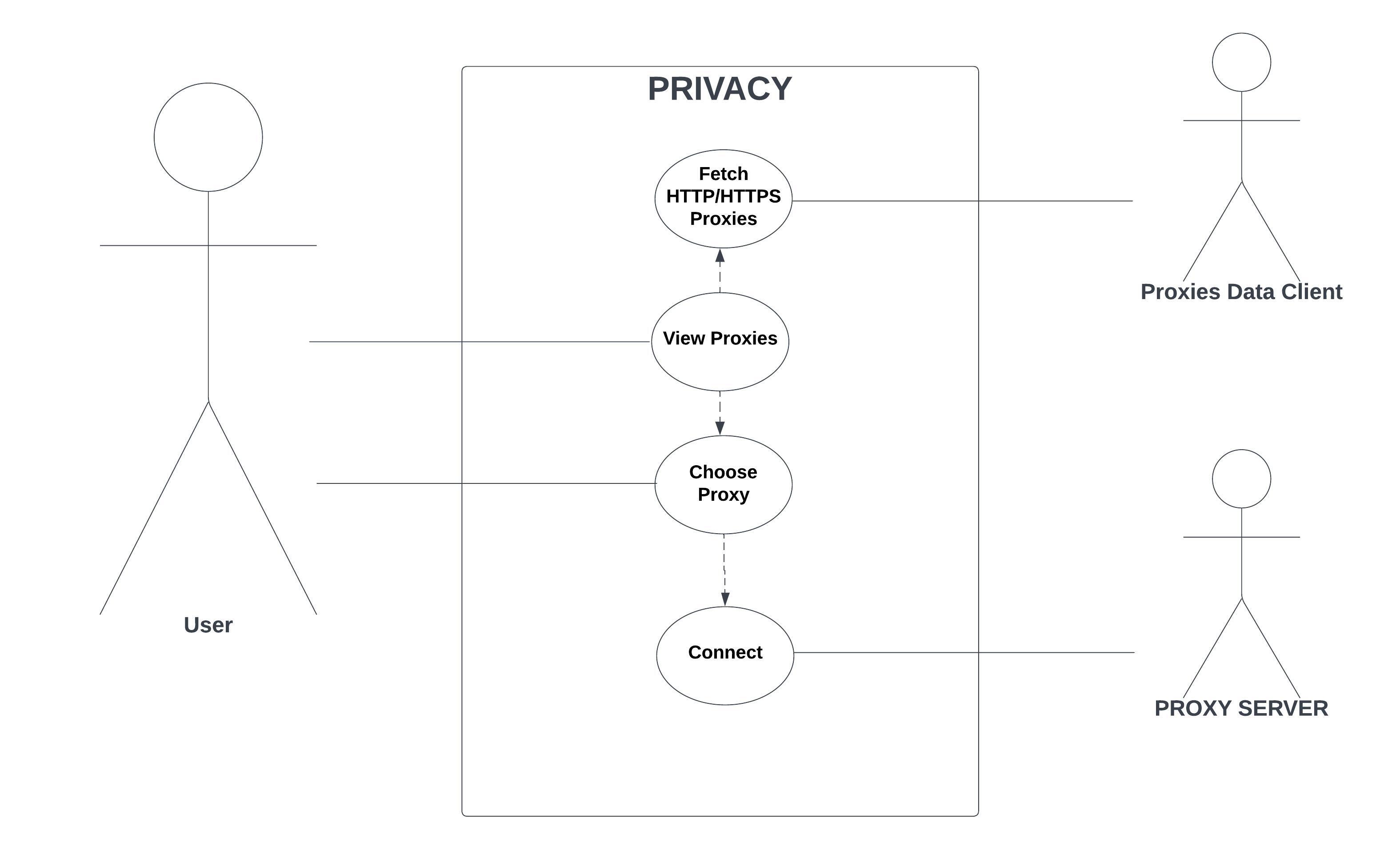
****

**INTRUDER:**

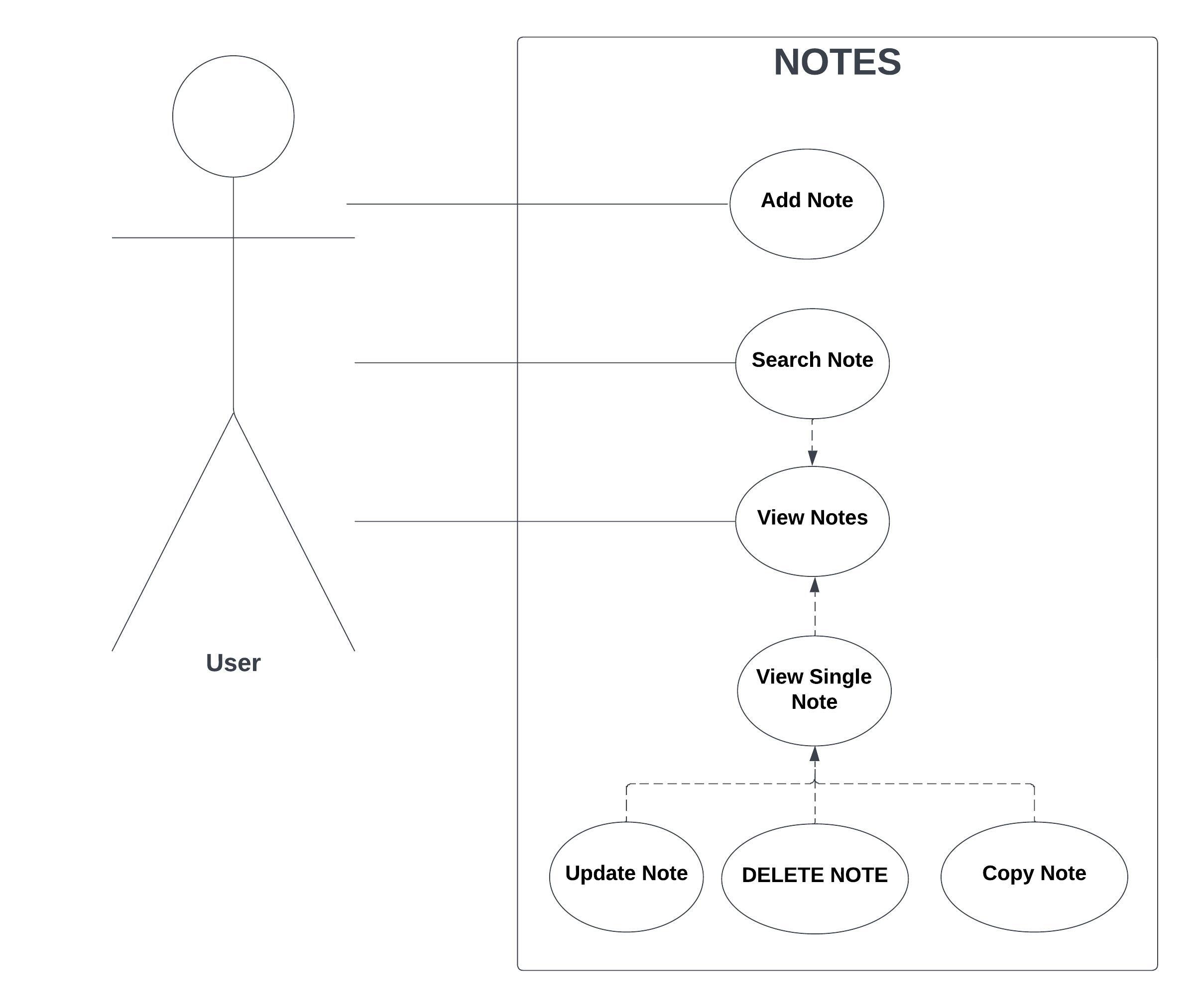


**GOOGLE DORK:**

**PING UTILITY:**

**PRIVACY:**

**NOTES:**

****

### 5.1.1 Usage Scenario

**HTTP MONITOR:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Http Monitor** | |
| **Use Case Id** | 01 | |
| **Requirement Id** | 01 | |
| **Description:** Allows users to monitor Http requests and responses in real-time | | |
| **Pre-Conditions:** User must send request to receive results | | |
| **Task Sequence** | | **Exceptions** |
| 1. At first, user will send request which will be monitored during the process | |  |
| 1. When request is sent then app will display the request | |  |
| 1. User is allowed to search specific http requests | |  |
| 1. When user got required request then user can copy, save and export the request | |  |
| **Post Conditions:** User will be able to copy, save and export the required request. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**SITE MAPPER:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Site Mapper** | |
| **Use Case Id** | 02 | |
| **Requirement Id** | 02 | |
| **Description:** Allows user to create a map of desired site’s structure and links. | | |
| **Pre-Conditions:** User must send provide targeted website to create the map for it | | |
| **Task Sequence** | | **Exceptions** |
| 1. At first, user will provide targeted website to create the map for it | |  |
| 1. When request is sent then app will display an option of user directed or automated spidering. | |  |
| 1. After the required spidering app will show the map of targeted website. | |  |
| 1. App will display the targeted map in graphical form | |  |
| **Post Conditions:** app must show the site map in graphical form | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**INTRUDER:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Intruder** | |
| **Use Case Id** | 03 | |
| **Requirement Id** | 03 | |
| **Description:** Perform automated testing of web development vulnerabilities by performing various brute force attacks | | |
| **Pre-Conditions:** User must select specific brute force attack | | |
| **Task Sequence** | | **Exceptions** |
| 1. At first, user will perform automated testing of web application. | |  |
| 1. User is allowed to select specific brute force attack to test vulnerabilities | |  |
| 1. User will customize and save payloads for use in testing. | |  |
| 1. App will display results of requests including response time and status code. | |  |
| **Post Conditions:** app must show the result of requests with response time and status code. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**REPEATER:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Repeater** | |
| **Use Case Id** | 04 | |
| **Requirement Id** | 04 | |
| **Description:** Allow user to manually send customized request to server repeatedly. | | |
| **Pre-Conditions:** User must come up with customized requests | | |
| **Task Sequence** | | **Exceptions** |
| 1. At first, user will send customized requests to server repeatedly | |  |
| 1. App will show response time and status code of requests | |  |
| 1. User can save and export the results of requests | |  |
| 1. App will allow user to search specific result of a request. | |  |
| **Post Conditions:** User must be able to save and export the results and status codes of request sent to server | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**PING UTILITY:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Ping Utility** | |
| **Use Case Id** | 05 | |
| **Requirement Id** | 05 | |
| **Description:** Allow user to check the availability and response time of a website or a server by sending a series of ping requests. | | |
| **Pre-Conditions:** User must come up with targeted website or a server | | |
| **Task Sequence** | | **Exceptions** |
| 1. At first, user will send customized requests to targeted server or to website. | |  |
| 1. App will show response time and status code of ping requests | |  |
| 1. User can save and export the results of ping requests | |  |
| 1. App will allow user to search specific result of ping request. | |  |
| **Post Conditions:** User must be able to save and export the request with response time of request sent to server | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**PORT SCAN:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Port Scan** | |
| **Use Case Id** | 06 | |
| **Requirement Id** | 06 | |
| **Description:** Allow user to scan the targeted ports | | |
| **Pre-Conditions:** User must come up with targeted website or a server to scan ports | | |
| **Task Sequence** | | **Exceptions** |
| 1. At first, user will scan a range of ports on a server or device. | |  |
| 1. User will customize the range of ports to scan and | |  |
| 1. App will display the result of port scan with open and close ports | |  |
| 1. User will be able to save and export the results of port scan | |  |
| **Post Conditions:** User must be able to save and export the results of port scan | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**ENCODER:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Encoder** | |
| **Use Case Id** | 07 | |
| **Requirement Id** | 07 | |
| **Description:** Allow user to encode and decode the data using various algorithms | | |
| **Pre-Conditions:** User must come up with data to encode and decode by using various algorithms. | | |
| **Task Sequence** | | **Exceptions** |
| 1. At first, user will be able to encode and decode the data using various algorithms. | |  |
| 1. App provide a lot algorithms including Base64, Hex etc. | |  |
| 1. User will be able to input and output data in various formats like text, hexadecimal or binary. | |  |
| 1. User will be able to compare the results of different algorithms. | |  |
| **Post Conditions:** App must be able to compare the results of different algorithms. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**HASHER:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Hasher** | |
| **Use Case Id** | 08 | |
| **Requirement Id** | 08 | |
| **Description:** Allow user to calculate hash using different hashing algorithms | | |
| **Pre-Conditions:** User must be able to come up with different hashing algorithms. | | |
| **Task Sequence** | | **Exceptions** |
| 1. App allow user to calculate hash using different hashing algorithms. | |  |
| 1. App provide a lot of algorithms including SHA-1, SHA-2 etc | |  |
| 1. User will be able to input and output data in various formats like text, hexadecimal or binary. | |  |
| 1. User will be able to compare the results of different hashing algorithms. | |  |
| **Post Conditions:** App must be able to compare the results of different hashing algorithms. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**GOOGLE DORKING:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Google Dorking** | |
| **Use Case Id** | 09 | |
| **Requirement Id** | 09 | |
| **Description:** Allow user to Search on Google using advanced search operators | | |
| **Pre-Conditions:** App must be able to come up with different search operators | | |
| **Task Sequence** | | **Exceptions** |
| 1. App allow user to search on different search operators | |  |
| 1. App must provide a list of common search operators and their usage. | |  |
| 1. User will be able to customize and save search queries for reuse. | |  |
| 1. User will be able to see search results | |  |
| **Post Conditions:** App must be able to compare the results of different search operators. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**HTTP SERVER:**

|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Http Server** | |
| **Use Case Id** | 10 | |
| **Requirement Id** | 10 | |
| **Description:** Allow user to create a local server on their device. | | |
| **Pre-Conditions:** App must be able to create local server on required device | | |
| **Task Sequence** | | **Exceptions** |
| 1. App must be able to create local server on required server | |  |
| 1. App must allow to browse and edit the file and directories on local server | |  |
| 1. User will be able to access local server using browser. | |  |
| 1. User will set the conditions for accessing the local servers | |  |
| **Post Conditions:** User must be able to set the conditions for accessing the local servers. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**PRIVACY:**

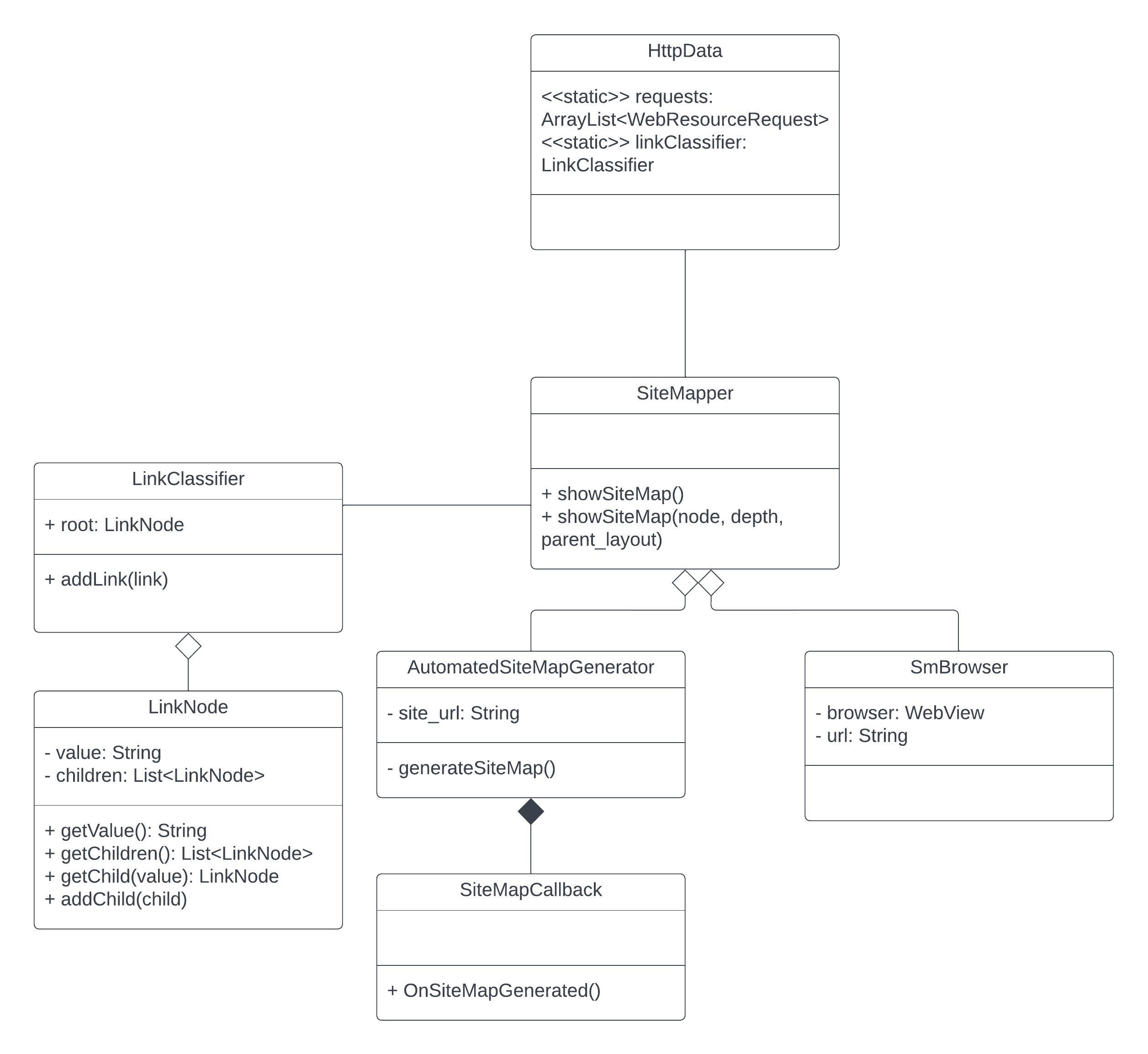
|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Privacy** | |
| **Use Case Id** | 11 | |
| **Requirement Id** | 11 | |
| **Description:** Allow user to mask their identity and location while using the app. | | |
| **Pre-Conditions:** App must be able to create an environment where user can hide its identity and location. | | |
| **Task Sequence** | | **Exceptions** |
| 1. User will be able to mask their identity and location while using the app | |  |
| 1. App must allow to route their traffic using Tor network or proxy | |  |
| 1. User will be able to use tor network or different proxies | |  |
| 1. User will be able to hide their traffic route by using proxies. | |  |
| **Post Conditions:** User must be able to use different types of proxies to get a customized experience for hiding their traffic. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

**NOTES:**

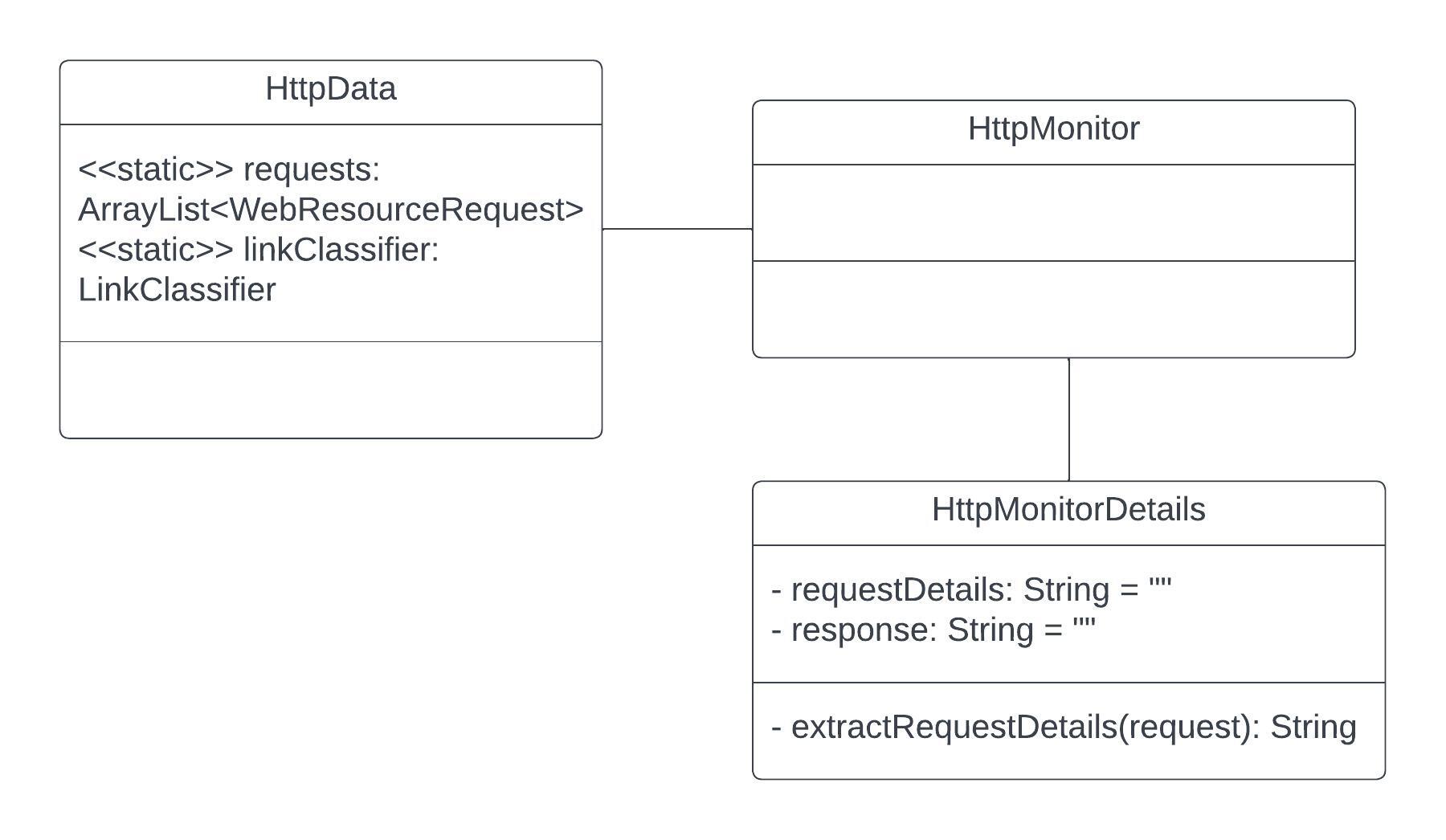
|  |  |  |
| --- | --- | --- |
| **Use Case Title** | **Notes** | |
| **Use Case Id** | 12 | |
| **Requirement Id** | 12 | |
| **Description:** User will be able to create, edit or delete payloads used in testing of web development vulnerabilities. | | |
| **Pre-Conditions:** App must enable user to create, edit or deleting payloads using in testing of web development. | | |
| **Task Sequence** | | **Exceptions** |
| 1. User will be able to create, edit or delete payloads used in testing of web applications | |  |
| 1. User will be able to categorize payloads and then can label them. | |  |
| 1. User will be able to save and export payload categorically. | |  |
| 1. User will be able to search payloads by using different filters. | |  |
| **Post Conditions:** User must be able to search different payloads which are categorize differently. | | |
| **Unresolved issues:** No issue | | |
| **Authority: User** | | |

## 5.2 **Class Diagrams**

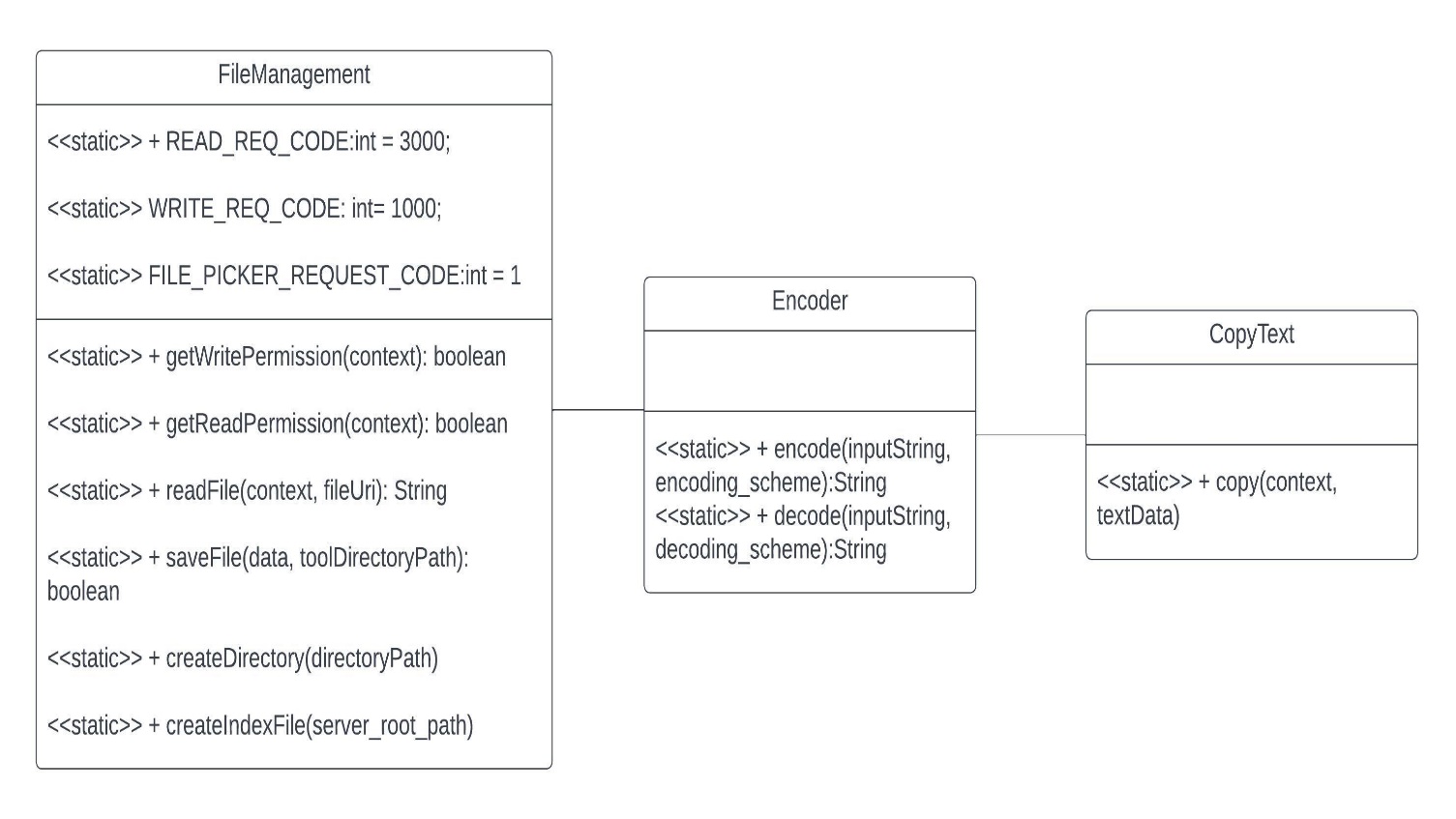
**SITE MAPPER:**



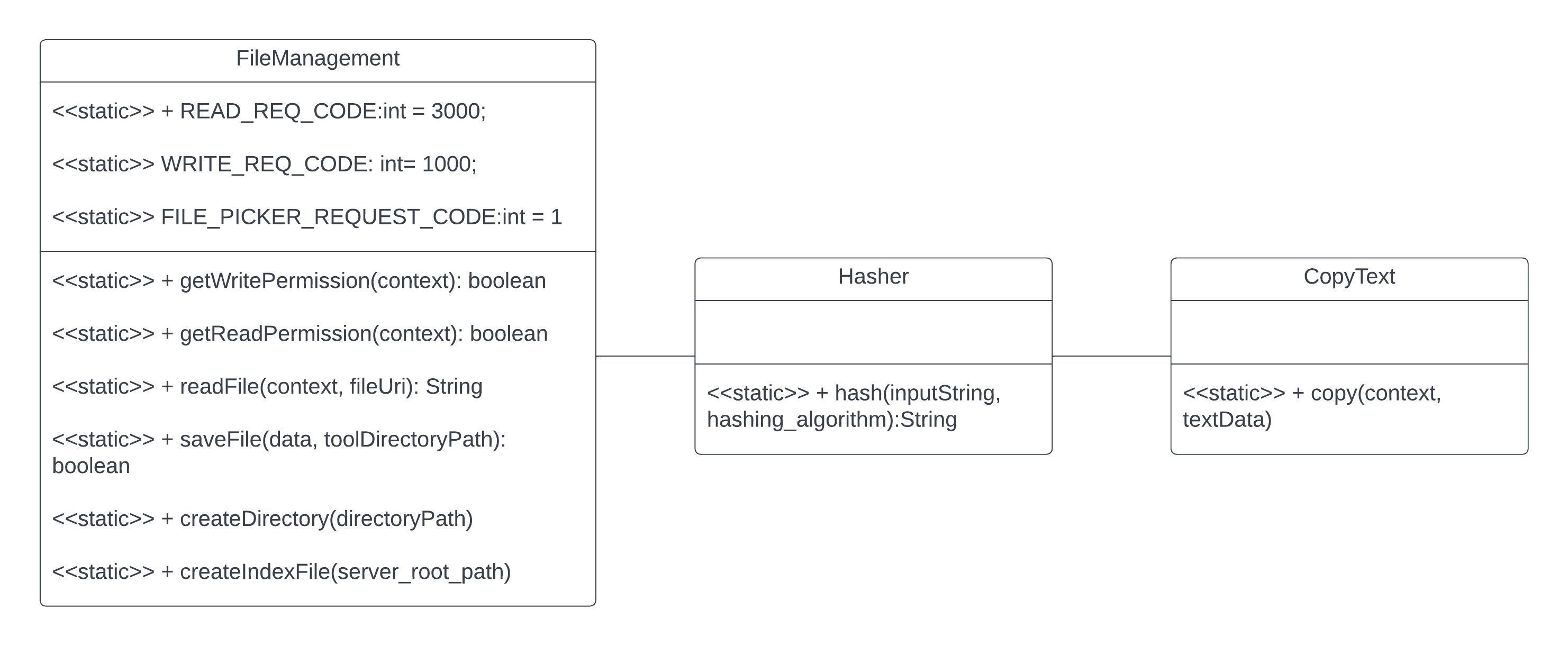
**HTTP MONITOR:**



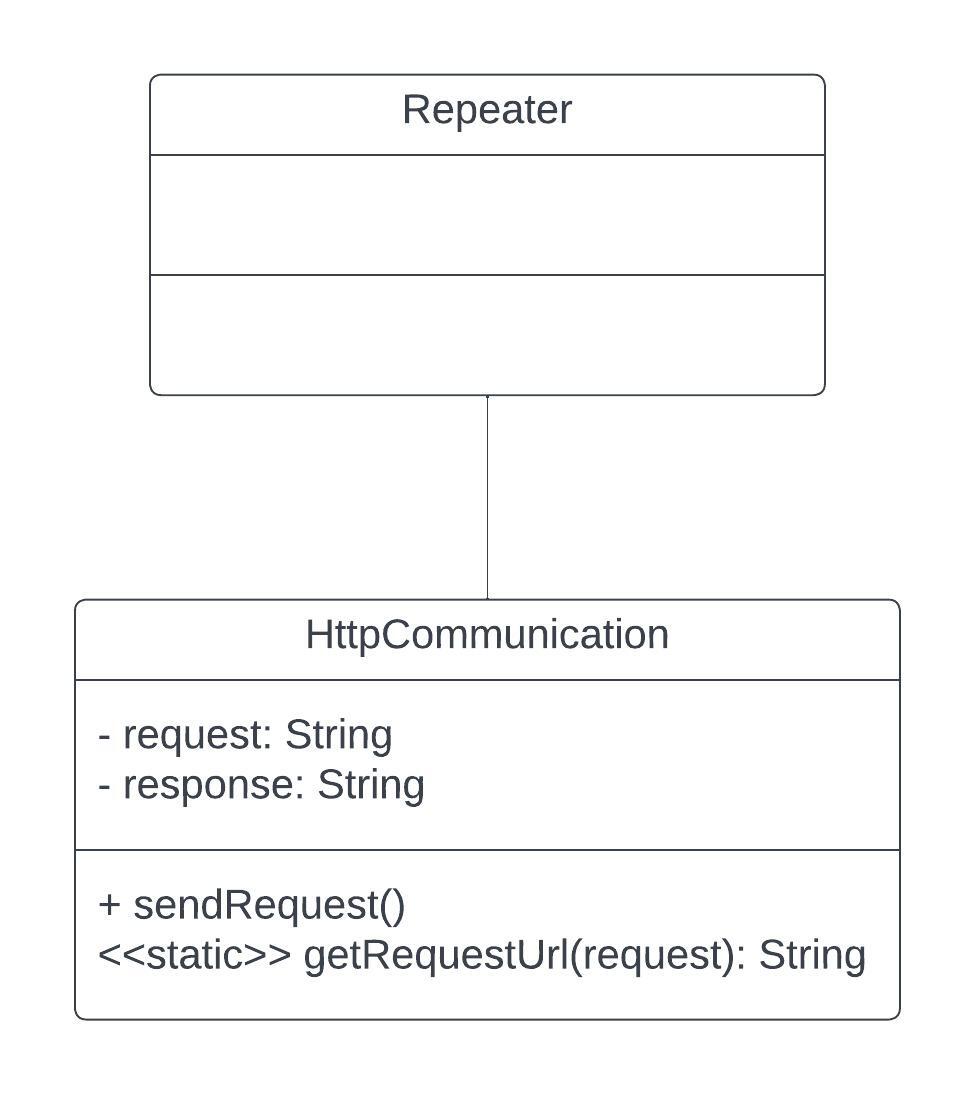
**ENCODER:**



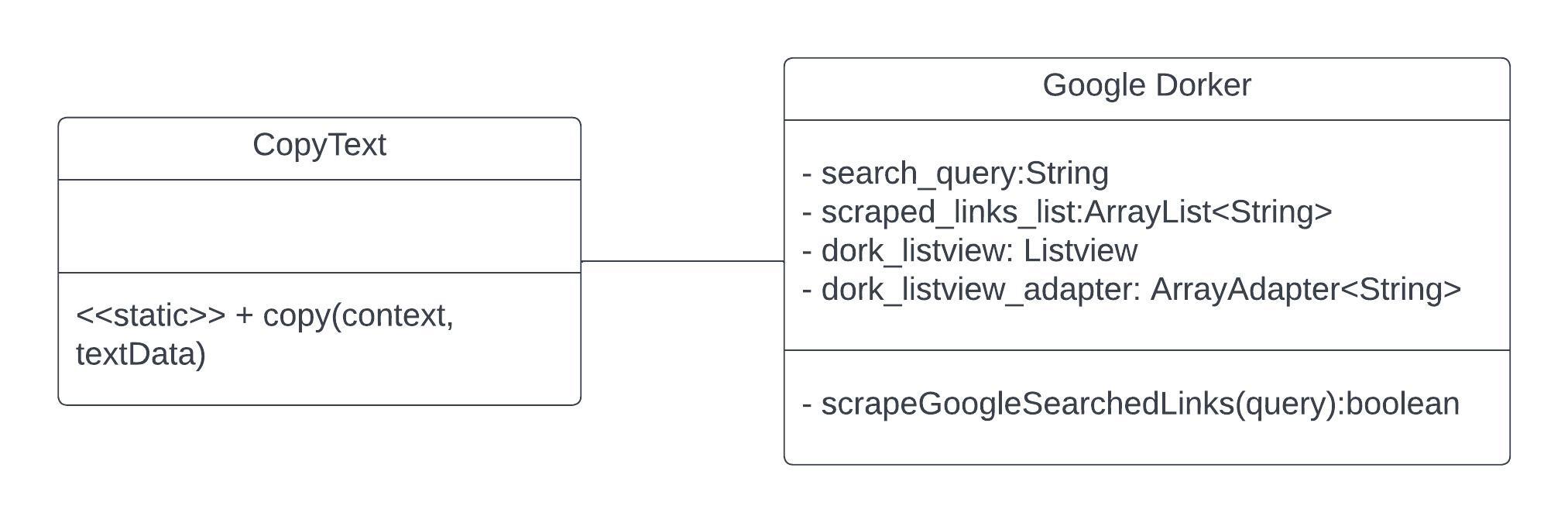
**HASHER:**



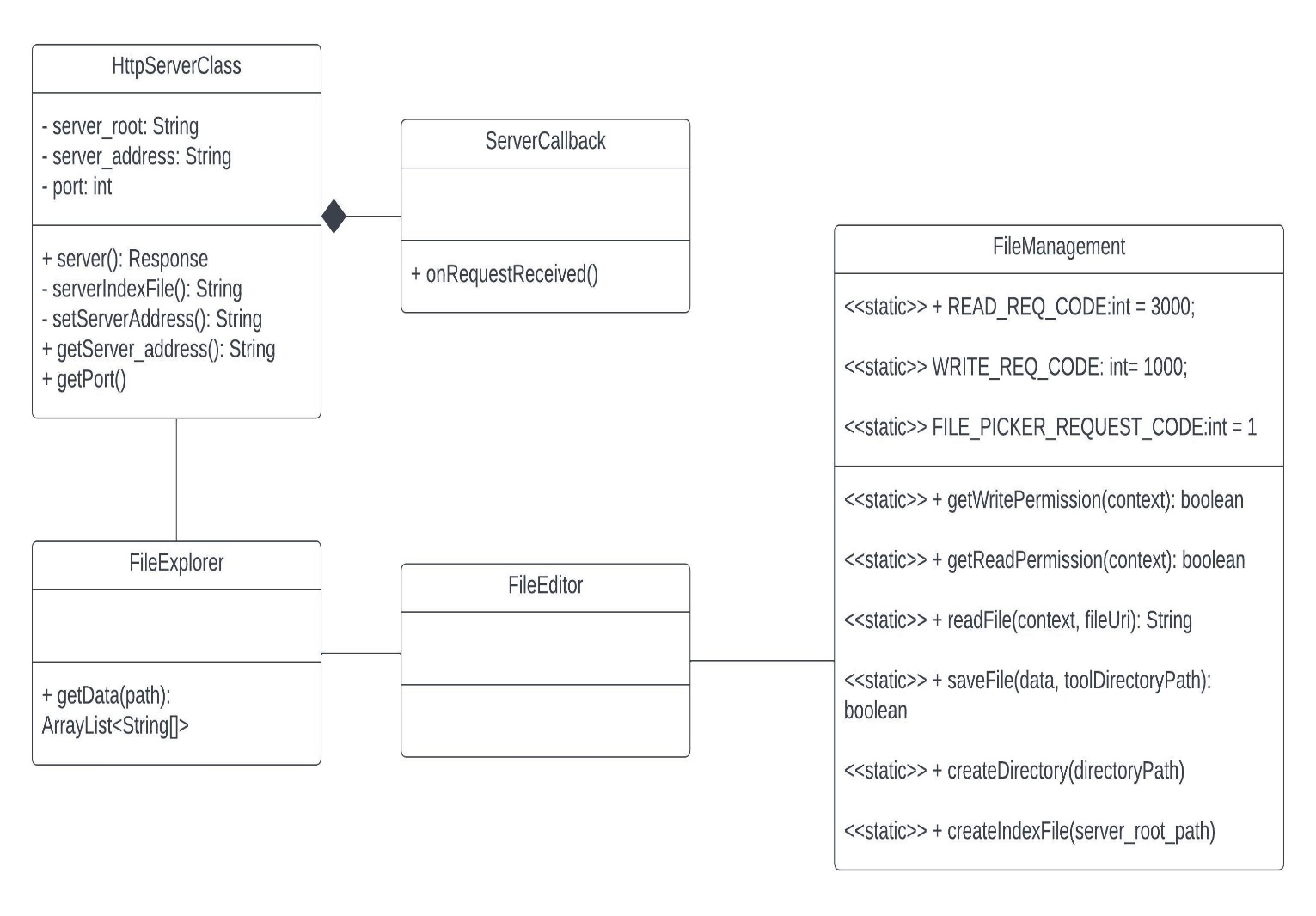
**REPEATER:**



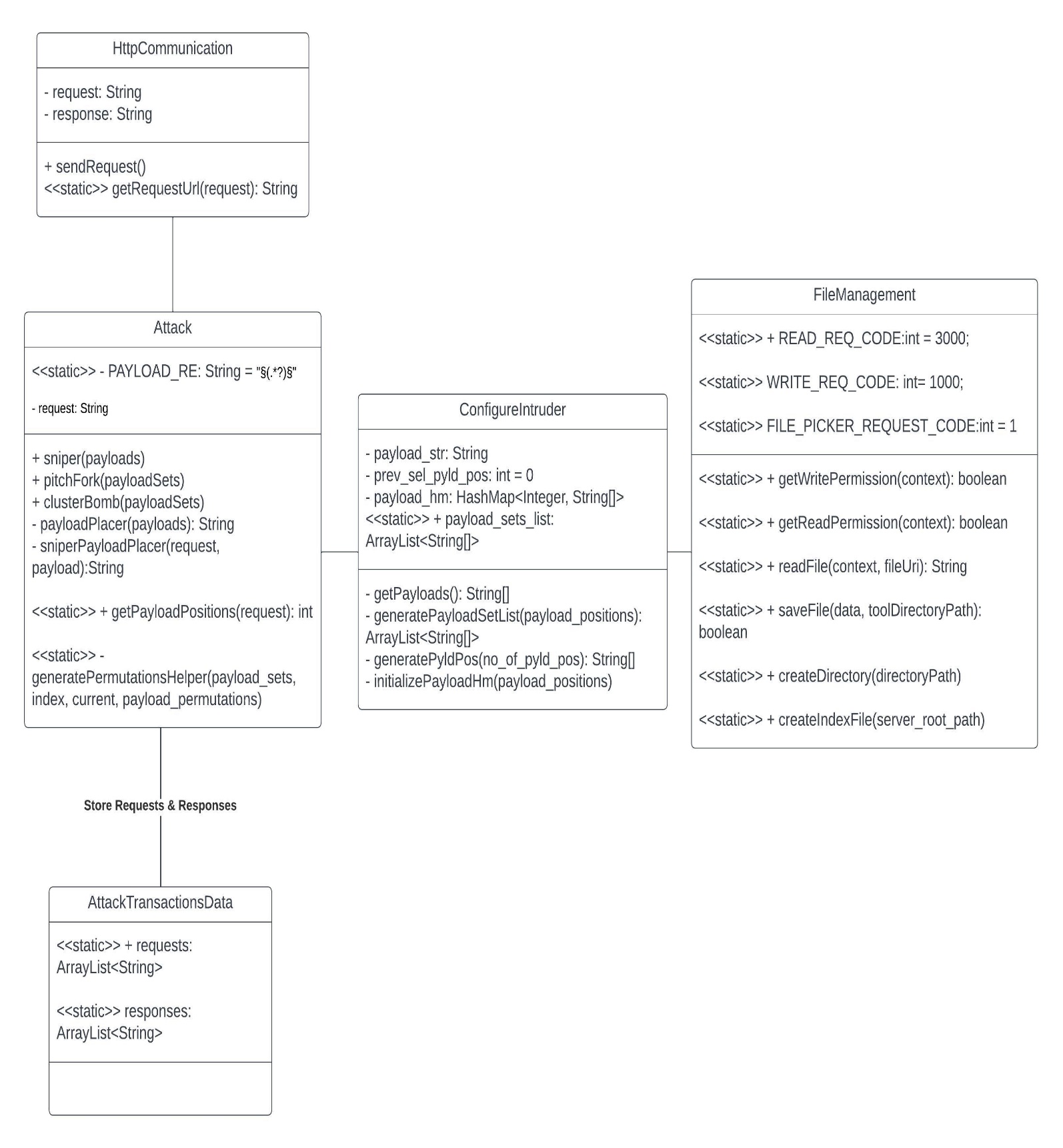
**GOOGLE DORK:**



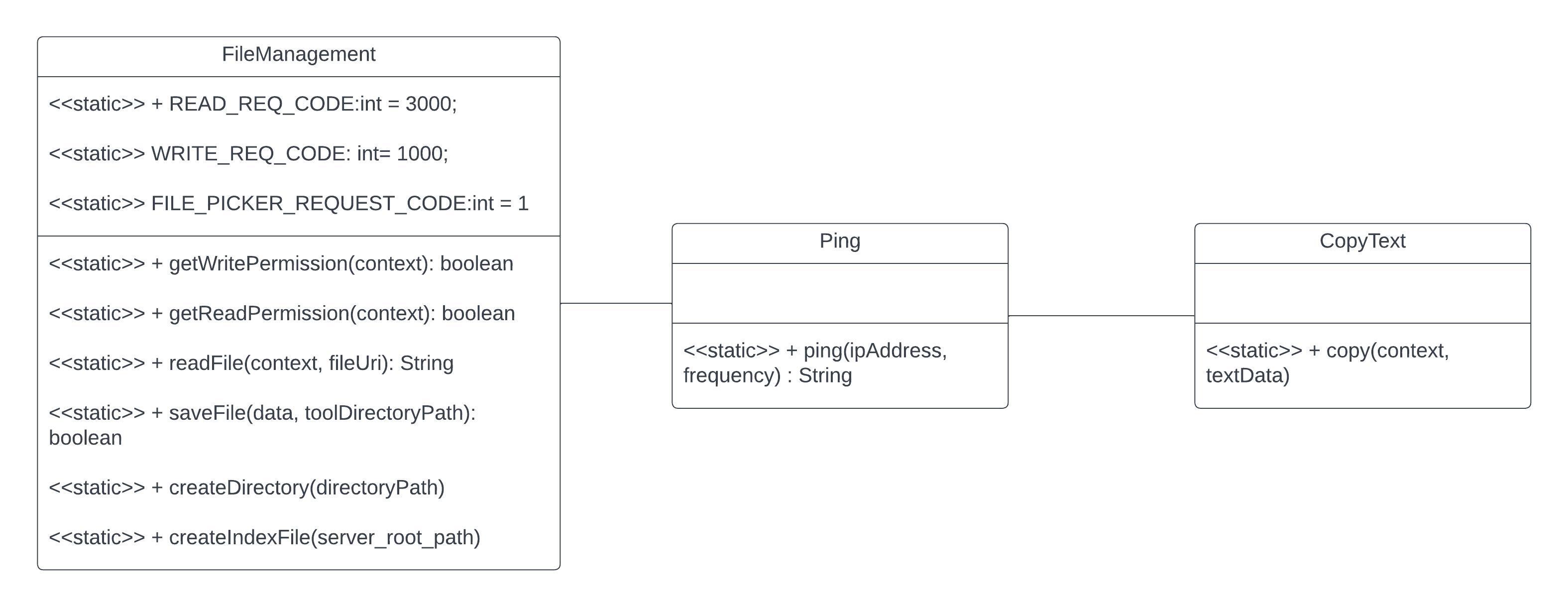
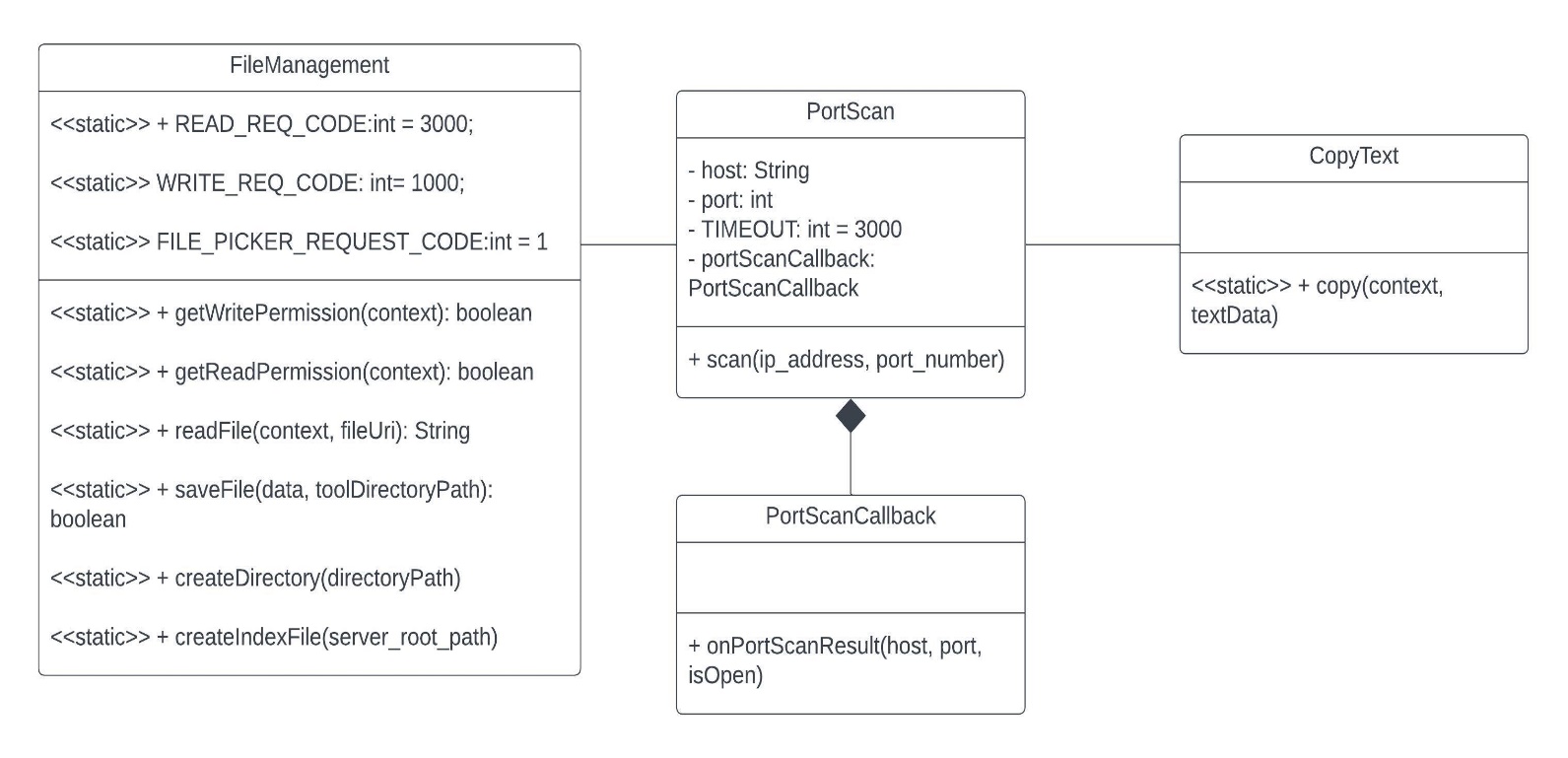
**HTTP SERVER:**

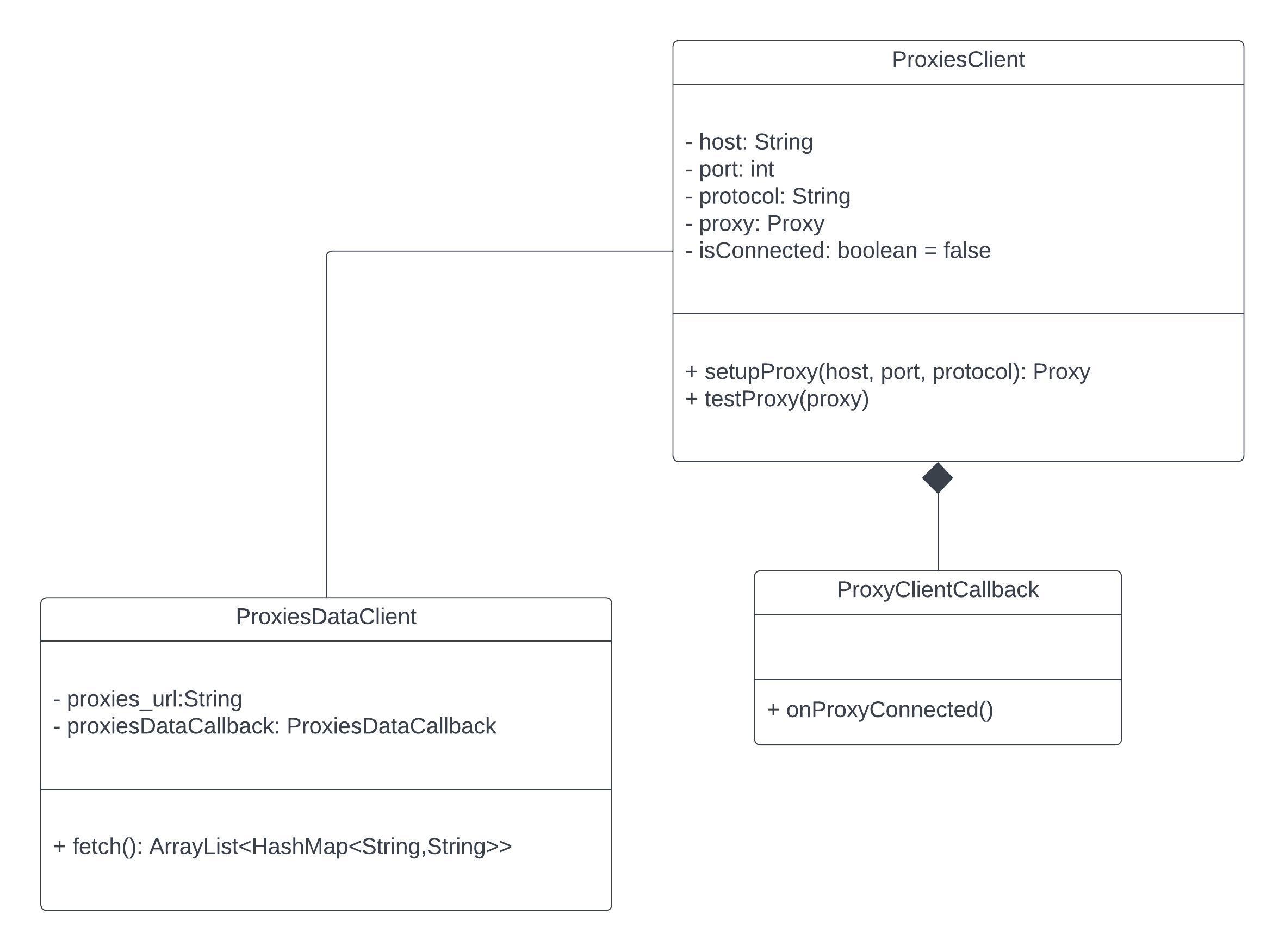


**INTRUDER:**

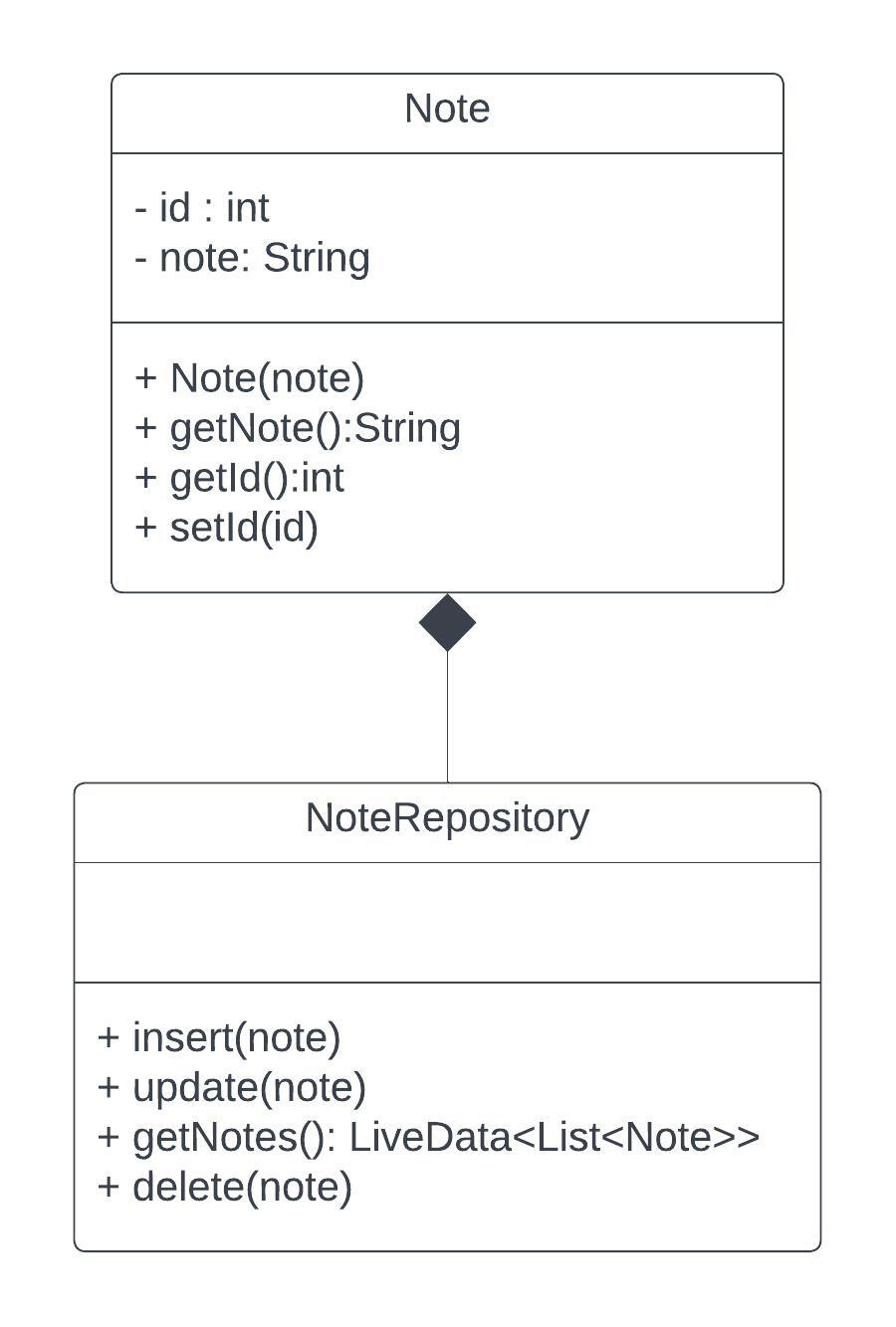


**PING UTILITY:**

**PORT SCAN:**

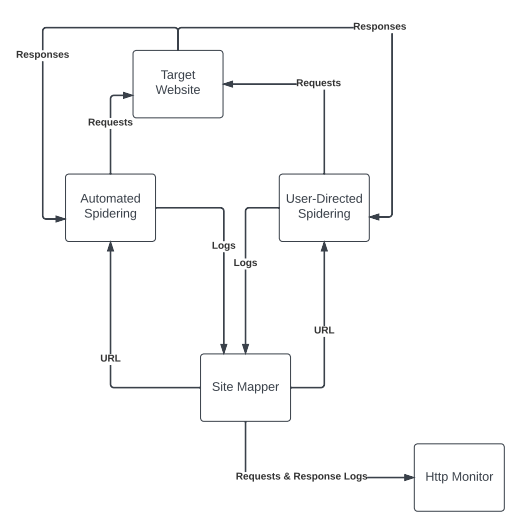
**PRIVACY:**

**NOTES:**

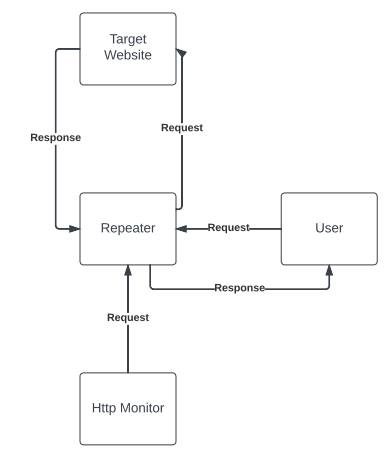


## 5.3 **Data Flow Diagrams**

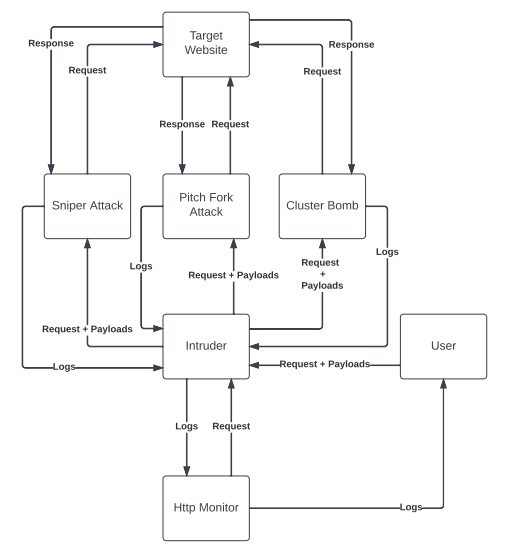
**SITE MAPPER:**



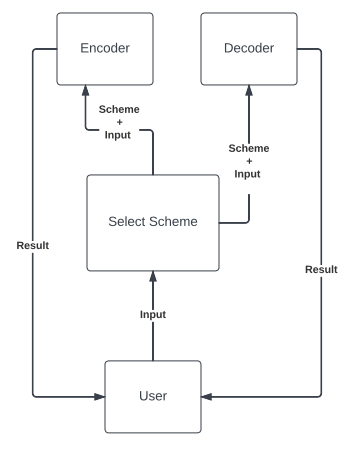
**REPEATER:**



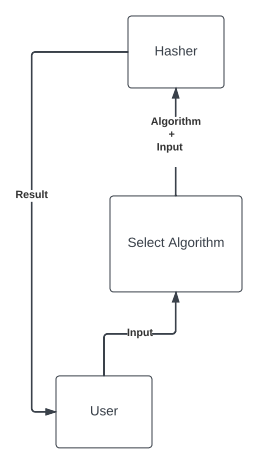
**INTRUDER:**



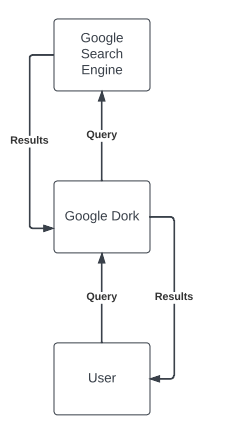
**ENCODER:**



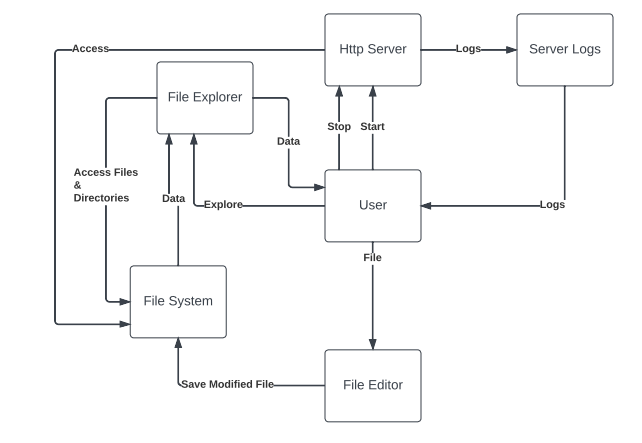
**HASHER:**



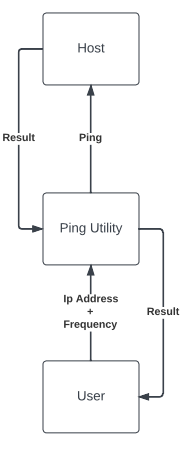
**GOOGLE DORK:**



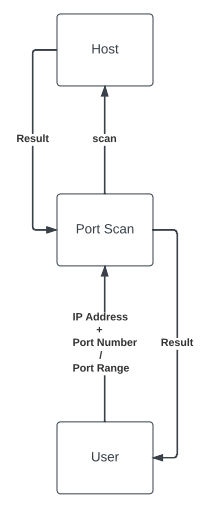
**HTTP SERVER:**

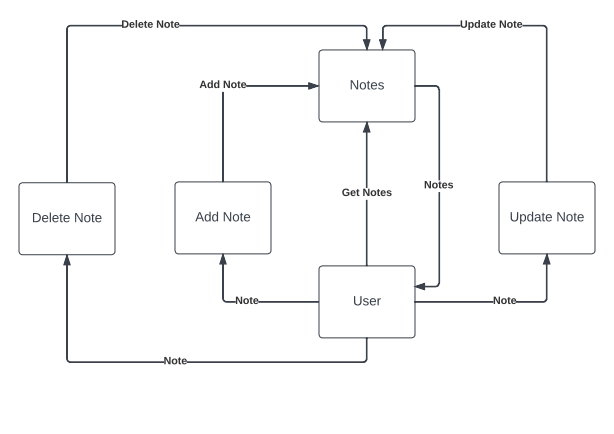


**PING UTILITY:**

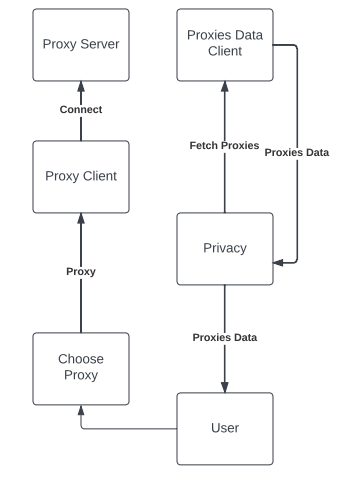


**PORT SCAN:**

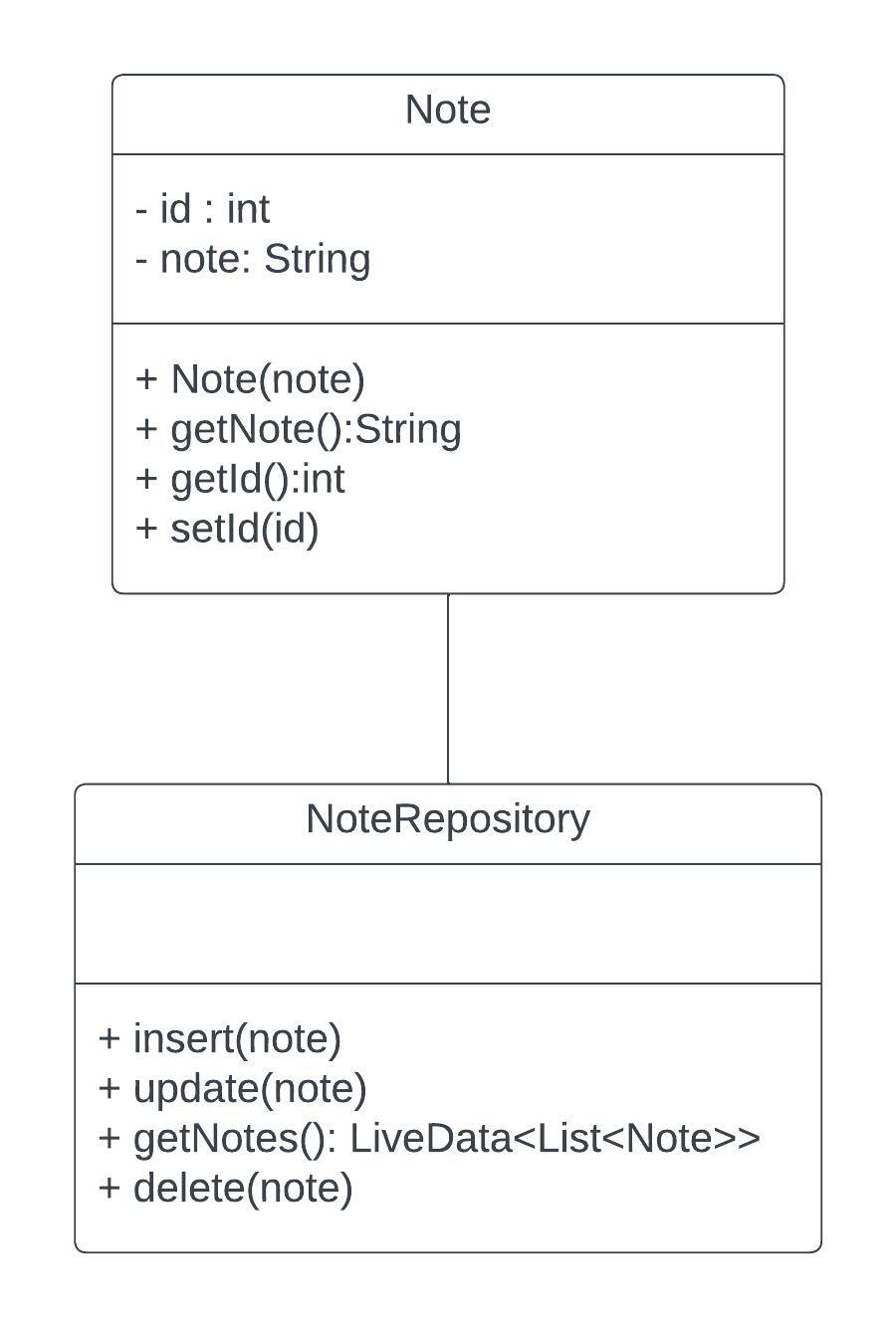


**NOTES:**

**PRIVACY:**



5.4 **Database Model**



# **RESULTS & DISCUSSION**

## 6.1 **Test Cases**

**Home Activity Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **1** |
| **Test Case Title:** | To verify click on every button present in home activity result in transition to right activity |
| **Test Case Priority:** | High |
| **Requirement:** | Home Activity |
| **Test Description:** | This test will verify that every button presents in home activity working right. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click Every button presents in home activity |
| **Dependencies:** |  |
| **Test Steps:** | 1. Click on every button one by one |
| **Test Data** |  |
| **Expected Results:** | 1. System should open home page.  2. Correct activity should be open after clicking a button present in home activity |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show a new activity with respect to the button clicked. |
| **Status: (Pass/Fail)** | Pass |

**Add Note Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **2** |
| **Test Case Title:** | To verify add notes functionality |
| **Test Case Priority:** | High |
| **Requirement:** | Notes Activity |
| **Test Description:** | This test will verify that add notes functionality is working right. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click on “Notes” Button present in Home Activity  3. Click “+” Floating Button present in Notes Activity and new activity for adding notes should be open  4. Click “Add Note” Button |
| **Dependencies:** |  |
| **Test Steps:** | 1. Click “Add Note” without entering text  2. Enter Text and Click “Add Note” |
| **Test Data** | Note Text |
| **Expected Results:** | 1. System should show toast “Note not saved” and transition to Notes Activity  2. Transition to Notes Activity and add Note should be present in the activity |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show the add the note in the Notes Activity. |
| **Status: (Pass/Fail)** | Pass |

**Update Note Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **3** |
| **Test Case Title:** | To verify the update notes functionality |
| **Test Case Priority:** | High |
| **Requirement:** | Notes Activity |
| **Test Description:** | This test will verify that notes are updating. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click “Notes” button present in home activity  3. Click on a note present in the note’s activity  4. Click “Update Note” Button |
| **Dependencies:** |  |
| **Test Steps:** | 1. Click “Update Note” button without editing note text  2. Click “Update Note” button by removing all the note text  3. Click “Update Note” button by editing the note text |
| **Test Data** | Note Text |
| **Expected Results:** | 1. System should transition to “Notes Activity” and the working Note should be present in the notes activity  2. System should transition to “Notes Activity” and the working Note with no text should be present in the notes activity  3. System should transition to “Notes Activity” and the updated Note should be present in the notes activity |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show the updated note in the Notes Activity. |
| **Status: (Pass/Fail)** | Pass |

**Delete Note Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **4** |
| **Test Case Title:** | To verify the delete notes functionality |
| **Test Case Priority:** | High |
| **Requirement:** | Notes Activity |
| **Test Description:** | This test will verify that notes are deleting. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click “Notes” button present in home activity  3. Click on a note present in the note’s activity  4. Click “Delete Note” Button |
| **Dependencies:** |  |
| **Test Steps:** | 1. Click “Delete Note” button without editing note text  2. Click “Delete Note” button by removing all the note text  3. Click “Delete Note” button by editing the note text |
| **Test Data** | Note Text |
| **Expected Results:** | 1. System should transition to “Notes Activity” and the working Note should not be present in the notes activity  2. System should transition to “Notes Activity” and the working Note should not be present in the notes activity  3. System should transition to “Notes Activity” and the updated Note should not be present in the notes activity |
| **Actual Results:** | As above |
| **Post Conditions:** | System should not show the deleted note in the Notes Activity. |
| **Status: (Pass/Fail)** | Pass |

**Copy Note Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **5** |
| **Test Case Title:** | To verify the copy, note functionality |
| **Test Case Priority:** | High |
| **Requirement:** | Notes Activity |
| **Test Description:** | This test will verify that notes can be copied. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click “Notes” button present in home activity  3. Click on a note present in the notes activity  4. Click “Update Note” Button |
| **Dependencies:** |  |
| **Test Steps:** | 1. Click “Copy Note” button without editing note text  2. Click “Copy Note” button by removing all the note text  3. Click “Copy Note” button by editing the note text |
| **Test Data** | Note Text |
| **Expected Results:** | 1. Note text present in database should be copied.  2. Note text present in database should be copied.  3. Note text present in database should be copied. |
| **Actual Results:** | As above |
| **Post Conditions:** | System should copy the note. |
| **Status: (Pass/Fail)** | Pass |

**Privacy Activity Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **6** |
| **Test Case Title:** | To verify list of free proxies are showing in privacy activity |
| **Test Case Priority:** | High |
| **Requirement:** | Privacy Activity |
| **Test Description:** | This test will verify that privacy activity shows free proxies. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application. |
| **Dependencies:** | Internet Availability |
| **Test Steps:** | 1. Click “Privacy” Button present in Home Activity |
| **Test Data** |  |
| **Expected Results:** | 1. A progress dialog should be shown after opening the privacy activity  2. List of free proxies are shown after the progress dialog dismisses. |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show list of free proxies. |
| **Status: (Pass/Fail)** | Pass |

**Connect Proxy Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **7** |
| **Test Case Title:** | To verify the connectivity with the free proxy. |
| **Test Case Priority:** | High |
| **Requirement:** | Privacy Activity |
| **Test Description:** | This test will verify that app’s connectivity with free proxy. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click “Privacy” Button present in Home Activity  3. Click on a free Proxy present in the list shown in Privacy Activity |
| **Dependencies:** | Internet Availability |
| **Test Steps:** | 1. Click on the “Connect” button shown in the popup after clicking one of the free proxy presents in the privacy activity. |
| **Test Data** |  |
| **Expected Results:** | 1. App should be connected to the target proxy server and a toast of “Connection Successful” should be shown.  2. The privacy activity should show the ip address of connected proxy and “Disconnect” button. |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show the ip address of connected proxy in privacy activity and “Disconnect” button. |
| **Status: (Pass/Fail)** | Pass |
| **Other Comments:** | None |

**Disconnect Proxy Test Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **8** |
| **Test Case Title:** | To verify the disconnect proxy functionality. |
| **Test Case Priority:** | High |
| **Requirement:** | Privacy Activity |
| **Test Description:** | This test will verify that app is disconnecting with target proxy. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click “Privacy” Button present in Home Activity  3. Click on a free Proxy present in the list shown in Privacy Activity  4. Click on “Connect” button shown in the popup showed after clicking on a free proxy present in privacy activity. |
| **Dependencies:** | Internet Availability |
| **Test Steps:** | 1. Click on the “Disconnect” button shown in the privacy activity after successful connection with the proxy. |
| **Test Data** |  |
| **Expected Results:** | 1. App should be disconnected from the target proxy server and list of free proxies should be shown. |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show the list of proxies. |
| **Status: (Pass/Fail)** | Pass |
| **Other Comments:** | None |

**Site Mapper Use Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **9** |
| **Test Case Title:** | To verify the site mapper functionality. |
| **Test Case Priority:** | High |
| **Requirement:** | Site Mapper Activity |
| **Test Description:** | This test will verify that app is showing the sitemap of target website. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application.  2. Click “Site Mapper” Button present in Home Activity  3. Click on settings icon present in the Site Mapper Activity |
| **Dependencies:** | Internet Availability |
| **Test Steps:** | 1. Click on “User-Directed Spidering” radio button shown in the popup showed after clicking on the settings icon in site mapper activity.  2. Click on “Automated Spidering” radio button shown in the popup showed after clicking on the settings icon in site mapper activity.  3. Click on “Done” button present in the popup to start mapping the target website. |
| **Test Data** |  |
| **Expected Results:** | 1. App should show the sitemap of target website in a hierarchical manner. |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show the sitemap of target website in a hierarchical manner. |
| **Status: (Pass/Fail)** | Pass |

**Http Monitor Use Case:**

|  |  |
| --- | --- |
| **Test Case ID:** | **10** |
| **Test Case Title:** | To verify the http monitor functionality. |
| **Test Case Priority:** | High |
| **Requirement:** | Http Monitor |
| **Test Description:** | This test will verify that Http Monitor is showing the logs of http communication of the app. |
| **Test Date:** | 05/25/2023 |
| **Pre-Conditions:** | 1. Run the application. |
| **Dependencies:** | Internet Availability |
| **Test Steps:** | 1. Click “Http Monitor” Button present in Home Activity.  2. Click on one of the log from the list.  3. Click on “Send Intruder”.  4. Click on “Send Repeater”.  5. Click on “Response”. |
| **Test Data** |  |
| **Expected Results:** | 1. List of logs should be shown in Http Monitor Activity.  2. Request should be shown when click on a log present in the list.  3. Request of opened log should be sent to intruder activity.  4. Request of opened log should be sent to repeater activity.  5. Response of request should be shown. |
| **Actual Results:** | As above |
| **Post Conditions:** | System should show the logs of app in Http Monitor Activity. |
| **Status: (Pass/Fail)** | Pass |

## 

## 6.2 **Conclusion**

The app demonstrates exceptional performance across all aspects. It has successfully passed all test cases, ensuring its reliability and functionality. The user interface and user experience (UI/UX) of the app effectively meets user needs, providing a seamless and satisfying experience. The app incorporates a well-thought-out design that does not negatively impact the overall user experience.

The UI of the app is intuitively designed, making it easy for users to navigate and interact with its features. It is devoid of any unnecessary complexities, ensuring a straightforward and accessible user interface. Users can effortlessly grasp and utilize all the app's features, as they have been designed with simplicity and user-friendliness in mind.

Additionally, the app excels in its responsiveness, delivering exceptional response times across all its features. This enhances the overall user experience by ensuring swift and seamless interactions. Users can expect quick and efficient results when utilizing any feature within the app.

In summary, the app's outstanding performance, successful test cases, user-centric UI/UX, ease of use, absence of complexities, and remarkable response times collectively contribute to its excellence. It provides users with a seamless and satisfying experience, facilitating their mastery of the app's various features.