``

Software Requirements Specification

for

Home and Office and Security Scanner (H.O.S.S.)

Version 1.1 approved

Prepared by Garret Stewart, Minsu Lee, Asahel Monroy, Ricardo Castellanos, Luis Cortez, Diego Flores, Ashley Pablo, Guang Wu, Jivan Karapetian, Adrian Holloway

Faculty Advisor: Dr. Huiping Guo

Liaison: Dr. Anson Fong

05/12/2023

**Table of Contents**

Table of Contents............................................................................................................ pg 2

Revision History.............................................................................................................. pg 4

1. Introduction.................................................................................................... pg 5

1.1. Purpose......................................................................................................... pg 5

1.2. Intended Audience and Reading Suggestions.............................................. pg 5

1.3. Product Scope............................................................................................... pg 5

1.4. Definitions, Acronyms, and Abbreviations .................................................. pg 6

1.5. References.................................................................................................... pg 6

2. Overall Description......................................................................................... pg 7

2.1. System Analysis…........................................................................................ pg 7

2.2. Product Perspective....................................................................................... pg 7

2.3. Product Functions......................................................................................... pg 8

2.4. User Classes and Characteristics.................................................................. pg 8

2.5. Operating Environment................................................................................ pg 8

2.6. Design and Implementation Constraints...................................................... pg 10

2.7. User Documentation.................................................................................... pg 10

2.8. Assumptions and Dependencies................................................................... pg 10

2.9. Apportioning of Requirements..................................................................... pg 10

3. External Interface Requirements..................................................................... pg 10

3.1. User Interfaces............................................................................................. pg 10

3.2. Hardware Interfaces..................................................................................... pg 10

3.3. Software Interfaces........................................................................................ pg 10

4. Requirements Specification............................................................................. pg 10

4.1. Functional Requirements............................................................................... pg 12

4.2. External Interface Requirements................................................................... pg 12

4.3. Logical Database Requirements.................................................................... pg 13

4.4. Design Constraints......................................................................................... pg 14

5. Other Nonfunctional Requirements................................................................. pg 14

5.1. Performance Requirements............................................................................ pg 14

5.2. Safety Requirements...................................................................................... pg 14

5.3. Security Requirements................................................................................... pg 14

5.4. Software Quality Attributes........................................................................... pg 14

6. Legal and Ethical Considerations.…............................................................... pg 16

Appendix A: Glossary..................................................................................................... pg 17

Appendix B: Analysis Models......................................................................................... pg 18

Appendix C: To Be Determined List............................................................................... pg 18

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| First Draft | 12/05/2022 | Initial Draft of Document | 1.0 |
| Final Draft | 5/11/2023 | Final Draft of Document | 1.1 |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

Home and Office Security Scanner (which will be denoted as HOSS in our document) is a software application which scans the user’s devices in their proximity and provides a security and privacy score for each device. The purpose of this document is to specify the requirements of HOSS (Version 1) and cover all aspects of its features and capabilities. It shall be clearly outlined what HOSS can and cannot achieve in terms of its network-scanning capabilities and device-grading algorithm. This document will also explain the terminology that will be useful in understanding the structure and function of the application, along with an overview of NMAP, which is the library used in achieving the security scans performed by HOSS. The requirements provided will also define how the software application is meant to be used, both on a hardware level and software level.

**1.2 Intended Audience and Reading Suggestions**

The Software Requirements Specification document (denoted as SRS) is intended for a broad range of readers: such as developers, project managers, marketing staff, testers, and general HOSS users. The SRS will be organized beginning with discussing the scope and perspective of HOSS. After this, the SRS shall explain the classes and requirements of HOSS, and end with a section outlining the legal and ethical considerations and providing a pictorial representation of the structure of HOSS’s database and data flow.

Testers and software engineers are recommended to start reading from Section 3 through Section 5 to have an immediate overview on HOSS’s requirements, and then refer to our ER diagram, relational model, and data flow diagram to have a pictorial perspective on the structure of HOSS. All other readers are recommended to begin with Section 1.3.

**1.3 Product Scope**

The official product we will be producing is HOSS. All other descriptions within the product scope will be within the scope of HOSS. HOSS is a software application that will allow its users to scan the network they are connected to and retrieve a list of all connected devices. The software will provide a 5 point rating security and privacy score for each listed device to its users. Users will be able to see previous devices they have scanned with their scores, as well as a glossary page to provide them with more information that would help them understand their device’s vulnerabilities.

This software is designed to be used by users who would like to have a list of all connected devices on their network and be informed on the security/privacy vulnerabilities of these devices. The user will simply go on the HOSS software app and go to the scanning page to begin their scan. Once clicking the scanning button, HOSS will scan the user's network for all connected devices and return scores for each device’s security and privacy, giving the user a high-level understanding of any risky devices on the network.

**1.4 Definitions, Acronyms, and Abbreviations**

Refer to Appendix: A

**1.5 References**

NMAP API Website - <https://nmap.org/book/nse-api.html>

Fing API Documentation - <https://www.fing.com/images/uploads/general/Fing_Cloud_API_v1.0.pdf>

CVE API Website - <https://nvd.nist.gov/developers/vulnerabilities>

MongoDB Documentation - <https://www.mongodb.com/docs/>

**2. Overall Description**

**2.1 System Analysis**

HOSS provides a way for network managers to view a list of all connected devices. Each connected device will have a security score and a privacy score, letting the network manager know which devices in the network are vulnerable to an attack or unsafe to use. In situations where a variety of computers and IOT devices are connected to the same network, this software application solves the issue of knowing what technology is safe and how your space is being compromised.

Some of the major technical hurdles we may encounter are defining the metrics and grading criteria for the security and privacy scores. IOT devices can have a wide variety of capabilities, so what may be a potential vulnerability in one device may not be present in another. This makes it difficult to choose specific areas or categories that can be used to test every device. As stated before, there are two different scores, privacy and security. The privacy score algorithm was made up of manual research based on the device and brand. As for the security score, we composed it with a mixture of manual research and automatic generation through the use of CVE metrics.

Although devices and their scores are to be manually input into the software application’s database, this approach is not suitable for networks in public areas in which anyone can join. A large variety of unidentified devices may connect and not yet have a proper score, making it difficult for data to constantly be input manually. This can be solved with a simple algorithm or function to give an unidentified device a preliminary score for security and privacy, which can later be manually updated.

**2.2 Product Perspective**

HOSS is fundamentally similar to contemporary device scanners such as Bluetooth®. However, HOSS provides an in-depth security and privacy assessment of each device it detects from its scans. Examples of factors in HOSS’ assessment include, but are not limited to: Encryption methods, authentication methods, default security settings, type of data collected, voice/movement recognition ability, and connectivity.

HOSS uses an NMAP library as a component to scan all devices within its vicinity. NMAP shall be programmed into HOSS’ infrastructure in order to be used seamlessly and concurrently with all of HOSS’ other functions.

Below are two applications which are functionally similar to HOSS.

Privacy Scanner (AntiSpy) - Privacy Scanner Anti Spy was created to check your smartphone whether you are really being spied on.

It detects parental control apps and surveillance apps that are mostly called stalkerware, which might be misused to spy on spouses or friends, using GPS-Track technologies, receive and send sms, read your contacts, read your call history, read your calendar and so on…

HouseCall ™ An increasing number of smart devices - like IP cameras, smart televisions, and storage devices - now connect to home networks. Many devices have security issues that attackers can use to take control of them or the home network itself, which can lead to privacy leaks or much worse.HouseCall for Home Networks scans all of the devices connected to your home network to identify potential risks and offer suggestions about how to eliminate them.

**2.3 Product Functions**

Scan Function

* Scans the network and retrieves information about each connected device
* NMAP library will scan the network for devices
* Devices will be cross referenced with an internal database
* New devices not yet in database will be flagged
* Each device’s security and privacy score will be displayed to users
* Suggestions on how to improve security and privacy will be shown to users

Scoring Function

* Scores devices based on security and privacy
* Security score is an average of brand score and device score
* Privacy score determined by brand privacy policies such as if they sell user’s info, and what sensitive information device stores.
* Security scored based on encryption method, default settings, authentication, universal device identifier.
* Security score also based on past vulnerabilities of a brand from Common Vulnerabilities and Exposures (CVE)

**2.4 User Classes and Characteristics**

There are only two classes for HOSS. The first class shall be the users. Users are defined as customers who are accessing the HOSS software application with the intention of performing a device scan and/or educating themselves on the glossary of security/privacy terms. There is no hierarchy to users and all users shall have the same authentication privilege to accessing HOSS.

The second class shall be the developers of HOSS. Technical expertise, privilege levels, experience, and educational levels are all common amongst HOSS developers since all HOSS developers shall contribute to server and database maintenance. Authentication and privilege levels are also the same amongst HOSS developers, and contributions of HOSS developers shall be explored further when outlining our data flow diagram (DFD).

**2.5 Operating Environment**

As our software is an application that runs using Google Chrome, any hardware capable of running Google Chrome and connecting to the internet will be compatible. As far as external hardware requirements, any external hardware must have either Bluetooth or Wi-Fi connectivity to allow our NMAP library to detect it. If a device does not have either of these connectivity capabilities, HOSS will not be able to detect it, and therefore, not be able to assign it a security/privacy score. The server host used for HOSS shall be DigitalOcean, a popular cloud-based host used to maintain website servers.

**2.6 Design and Implementation Constraints**

Any hardware that is capable of running Google Chrome can use HOSS. This inherently means that the hardware must be capable of having internet connectivity. One constraint that can affect HOSS is the number of devices on a user's network.

Access to an NMAP library is required for HOSS to function. NMAP must run and operate concurrently with HOSS, along with a Mongo database server.

A database programming language, which in this case shall be Mongo, is required for storage of all devices and historical security and privacy information. This data cannot be stored within our Javascript/React code. Javascript + React is required for programming the front-end and user-functionality aesthetics of HOSS, whereas Python shall serve as our backend, where we interact with our database.

**2.7 User Documentation**

While the HOSS software application is extremely simple in terms of user interactivity, since all that is required of the user is to click the Scan button, HOSS shall provide documentation in the form of a glossary. Any security/privacy terminology shall be included in the glossary for users to cross-reference such as the meaning of encryption methods, connectivity methods, authentication types such as two-factor, etc.

HOSS shall also provide documentation in terms of each rating that it provides. For example, if a device receives a 3 out of 5 stars for security, HOSS will elaborate on if this constitutes as a secure device, insecure device, or neutrally secure device. It will document similarly for privacy ratings (I.E. if the user receives a 4 out of 5 stars privacy ratings for a device, HOSS will elaborate on why and provide the required documentation).

HOSS is also inherently a tutorial, because it's a requirement for HOSS to provide recommendations and instructions on improving a device’s security and privacy scores.

**2.8 Assumptions and Dependencies**

Changes to the NMAP library may have an effect on the software, as NMAP is the backbone of the software. Similarly, any changes to Javascript, Python, React, or mongoDB may have effects, though this is not anticipated to happen.

**2.9 Apportioning of Requirements**

There are currently no plans to implement any features not already listed in this document in the future.

**3. External Interface Requirements**

**3.1 User Interfaces**

3.1.1 HOSS will initially present the user with a welcome screen containing information about the web

app functions and a scan button

3.1.2 The user will be able to login from the welcome screen, and will be prompted to do so if they

attempt to scan

3.1.3 HOSS will display all connected devices and their security and privacy scores as a list of items

after completing the scan

3.1.4 Cybersecurity terminology present in the recommendations list will be highlighted for users to click on

3.1.5 HOSS will include a glossary section for defining cybersecurity terminology

**3.2 Hardware Interfaces**

3.2.1 This software application does not have any hardware interface requirements

**3.3 Software Interfaces**

3.3.1 JavaScript

3.3.2 Python

3.3.3 MongoDB

3.3.4 NMAP library

3.3.5 HTML5

3.3.6 CSS

**4. Requirements Specification**

**4.1 Functional Requirements**

4.1.1 Application Requirements

4.1.1.1 The application shall be supported by modern devices

4.1.1.1.1 The application shall run on current desktop computers

4.1.1.2 The application shall provide a main page to the user

4.1.1.2.1 The application shall initially show the main page to the user

4.1.1.2.2 The application shall allow the user to navigate to the glossary from the main

page

4.1.1.2.3 The application shall provide a scan button on the main page

4.1.1.3 The application shall provide a scan results page

4.1.1.3.1 The application shall show the user a list of devices connected to the

user’s network

4.1.1.3.2 The application shall provide the user with a scan button that refreshes the list

of connected devices

4.1.1.3.3 The application shall provide a security and privacy score next to each device

4.1.1.3.4 The application shall display a drop down menu for each device

4.1.1.3.4.1 The application shall display additional information about a device

in the dropdown menu

4.1.1.4 The application shall provide a Glossary page

4.1.1.4.1 The page shall be accessible from any page of the software application

4.1.1.5 The application shall provide a Devices page

4.1.1.5.1 The page shall be accessible from any page of the software application

4.1.1.5.2 The page shall display all previously scanned devices

4.1.1.6 The application shall provide an About page

4.1.1.6.1 The page shall be accessible from any page of the software application

4.1.1.6.2 The page shall display relevant information about the development team

**4.2 External Interface Requirements**

4.2.1 User Interfaces

4.2.1.1 Scan Results Page

4.2.1.1.1 The page shall display a list of all devices connected to the user’s network

4.2.1.1.2 The page shall list the devices in such a way that the device name, security

rating, and privacy rating are visible

4.2.1.2 Glossary

4.2.1.2.1 The page shall display a list of relevant terms related to privacy and security

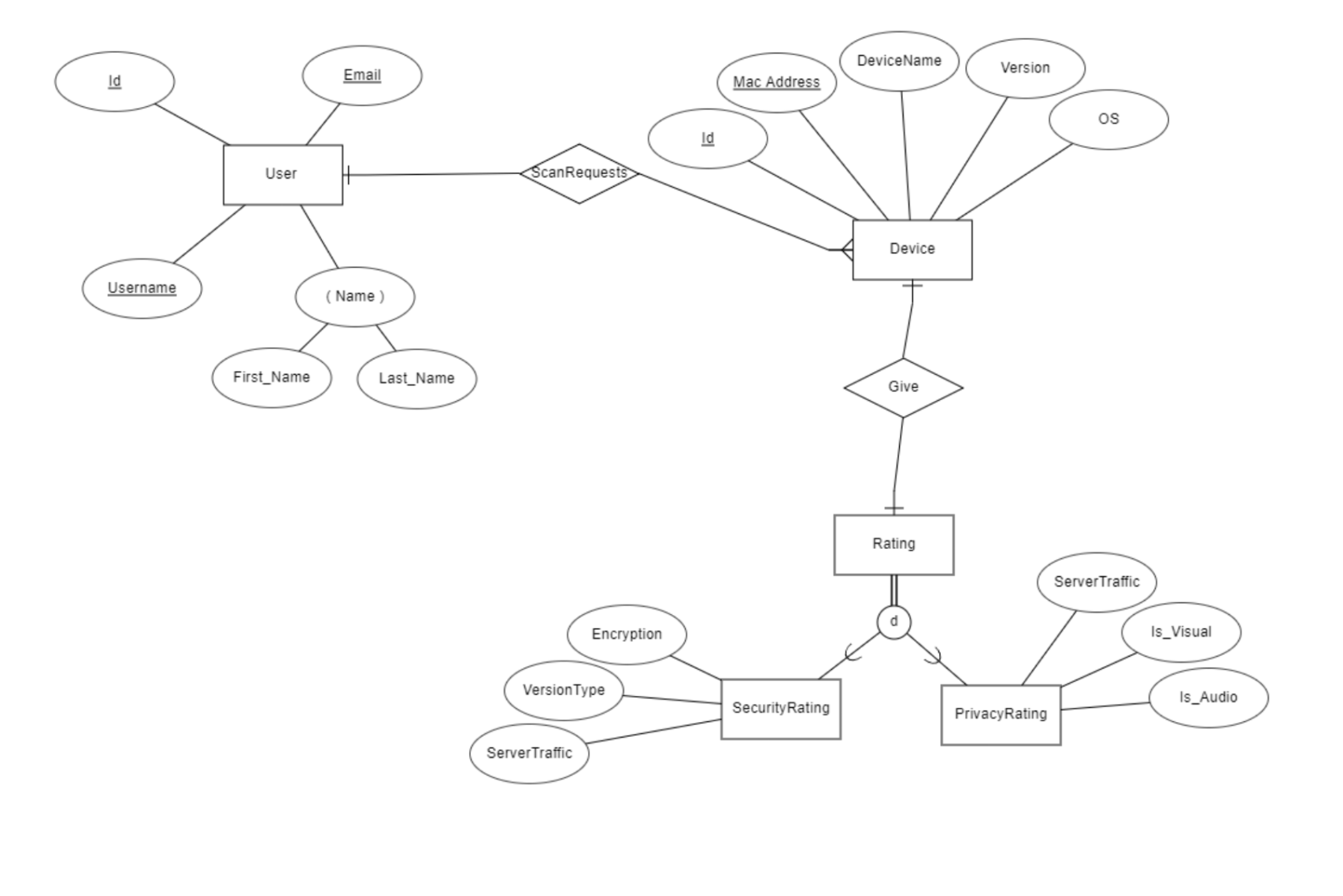
and their definitions

4.2.2 Software Interfaces

4.2.2.1. There are no current External Interface Requirements for the software interfaces used

in this software application

**4.3 Logical Database Requirements**



**4.4 Design Constraints**

4.4.1 There are no known design constraints at the time of this document’s creation.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

5.1.1 The scan request should take mere seconds (time between scan and results appearing)

5.1.2 Around 95% of the transactions shall be processed in less than 15 seconds

**5.2 Safety Requirements**

5.2.1 Due to the nature of this software application, there are no safety requirements provided in this

document.

**5.3 Security Requirements**

5.3.1 Users should not have direct access to database information

5.3.1.1 Users shall have access only to information that has been retrieved by the software

application

5.3.2 Users should not be able to manipulate database information

**5.4 Software Quality Attributes**

5.4.1 Adaptability

5.4.1.1 The database is dynamic, and its contents are updated by users with appropriate

permissions

5.4.2 Availability

5.4.2.1 Users can access the software app online with the appropriate browser and use all features that are provided

5.4.3 Maintainability

5.4.3.1 Content of the system database can be accessed and modified by users with appropriate

permissions

5.4.4 Portability

5.4.4.1 The software application can be accessed from anywhere with a desktop/laptop that has a connection to the internet and a web browser

5.4.5 Reusability

5.4.5.1 Users can use the software application’s features multiple times and retrieve updated

information of connected devices after every scan

**6. Legal and Ethical Considerations**

6.1 Liability

6.1.1 Users should consider following suggestions based on identified vulnerabilities

6.1.1.1 HOSS and its developers are not responsible for any damages or attacks

resulting from a user’s action or inaction to address vulnerabilities

6.1.1.2 Users should consider warning device owners of identified vulnerabilities

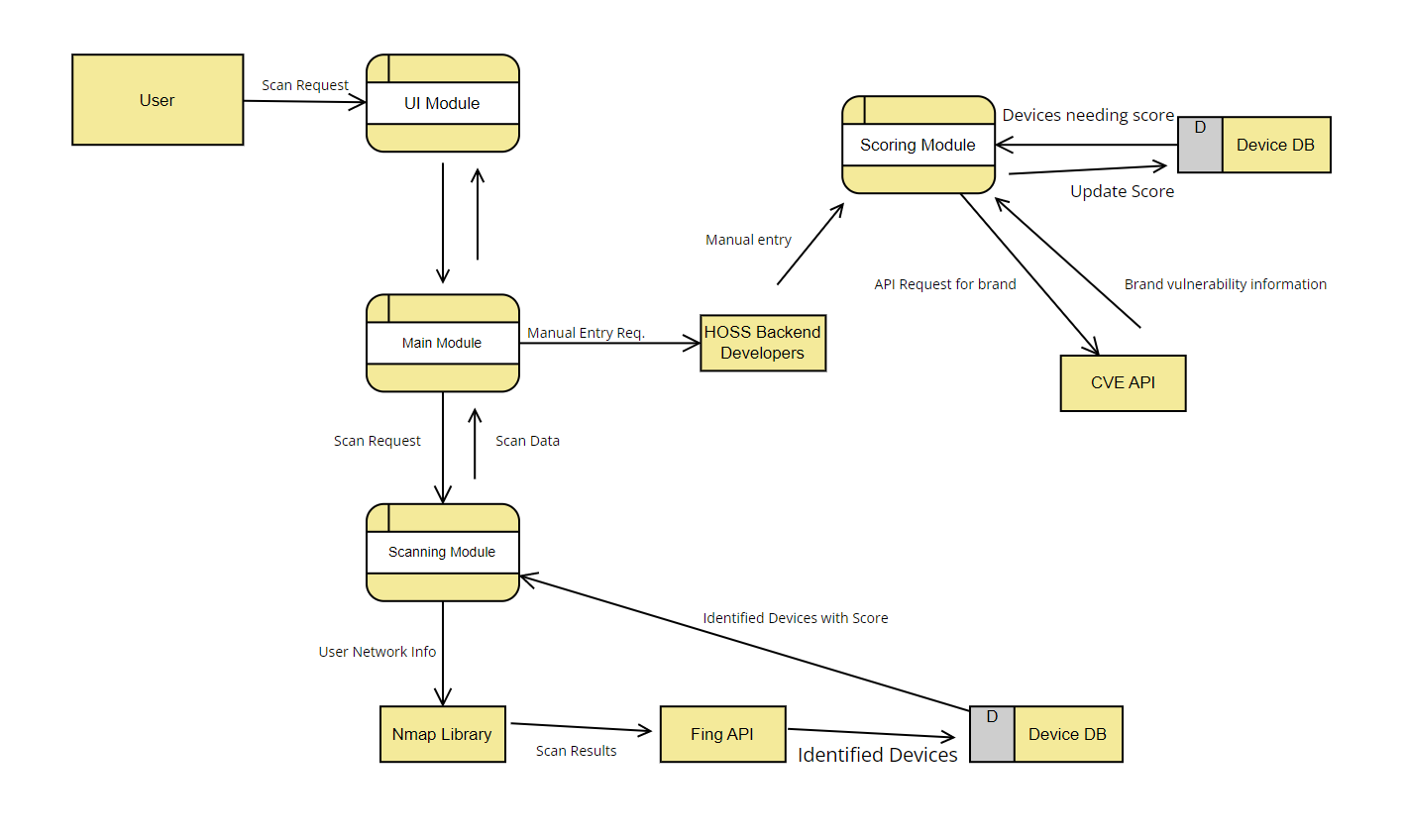
6.1.1.3 HOSS and its developers shall never disclose a device’s information to anyone

besides the user requesting the scan.

**Appendix A: Glossary**

* **Bootstrap** - CSS Framework used for developing responsive and mobile-first websites.
* **CVE** - Common Vulnerabilities and Exposures. A list of vulnerabilities from NVD, a database of vulnerabilities across a large array of devices and vendors
* **Fing DevRecogAPI** - Network scanner that identifies devices based on brand, model, and type.
* **HOSS** - Home and Office Security Scanner
* **IOT** - Internet of Things, which describes the network of physical objects that have software, sensors, and other technologies.
* **JavaScript** - A programming language that creates dynamic and interactive content for web content. It manipulates, calculates, and validates data.
* **MongoDB** - A non-relational database that uses JSON-like documents. It’s designed to store large amounts of data and let the user work with it efficiently.
* **Python** - General purpose programming language used in web development and software development.
* **SRS** - Software Requirements Specification

**Appendix B: Analysis Models**



**Appendix C: To Be Determined List**

No TBD references provided in this document.