System Requirements Specification

for

AutoPen

**Version 6.0 Approved**

**Prepared by Michael Allen**

**Caleb Hall**

**Calla Robison**

**Myles Scott**

**Joshua Buscher**

**4/14/2024**

**Table of Contents**

[**Revision History**](#_heading=h.6ltov612afoh) **iii**

[**1. Introduction 1**](#_heading=h.1fob9te)

[**1.1 Purpose 1**](#_heading=h.3znysh7)

[**1.2 Document Conventions 1**](#_heading=)

[**1.3 Intended Audience and Reading Suggestions 1**](#_heading=)

[**1.4 Product Scope 1**](#_heading=)

[**1.5 References 2**](#_heading=)

[**2. Overall Description**](#_heading=h.dxquq0idb247) **2**

[**2.1 Product Perspective 2**](#_heading=h.2s8eyo1)

[**2.2 Product Functions 3**](#_heading=)

[**2.3 User Classes and Characteristics 3**](#_heading=)

[**2.4 Operating Environment 6**](#_heading=)

[**2.5 Design and Implementation Constraints 6**](#_heading=)

[**2.6 User Documentation 6**](#_heading=)

[**2.7 Assumptions and Dependencies**](#_heading=) **6**

[**3. External Interface Requirements 7**](#_heading=h.tfoo3e983jbg)

[**3.1 User Interfaces 7**](#_heading=h.2jxsxqh)

[**3.2 Hardware Interfaces 1**](#_heading=)**1**

[**3.3 Software Interfaces 1**](#_heading=)**1**

[**3.4 Communications Interfaces 1**](#_heading=)**2**

[**4. System Features 1**](#_heading=h.nzsk60xmpkla)**3**

[**4.1 System Feature: Automated Penetration Testing 1**](#_heading=)**3**

[**4.2 System Feature: User Profile Management 1**](#_heading=)**4**

[**5. Other Nonfunctional Requirements 1**](#_heading=h.ambpd1yr81kx)**4**

[**5.1 Performance Requirements 1**](#_heading=)**4**

[**5.2 Safety Requirements 1**](#_heading=)**5**

[**5.3 Security Requirements 1**](#_heading=)**5**

[**5.4 Software Quality Attributes 1**](#_heading=)**5**

[**5.5 Business Rules 1**](#_heading=)**6**

[**6. Other Requirements 1**](#_heading=h.xlqospw3fpjr)**6**

[**7. Appendix A: Glossary 1**](#_heading=h.gxd7q0225i42)**8**

[**8. Appendix B: Analysis Models 1**](#_heading=h.cz18rx2b6vcu)**8**

[**9. Appendix C: To Be Determined List**](#_heading=h.r2uwbttfh8a) **19**

# 

# Revision History

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Michael Allen | 9/21/23 | Rewording Section 1.4 and 2 | 1.0 |
| Michael Allen | 9/26/23 | Section 4 and 5 | 1.1 |
| Michael Allen | 9/28/23 | Section 3 and 6 | 1.2 |
| Caleb Hall | 9/29/23 | Overall Editing and formatting | 1.3 |
| Myles Scott | 10/26/23 | Changing title from 1.0 to 2.0, adding use case diagram | 2.0 |
| Caleb Hall | 10/31/23 | Editing 1.5, 3.3 | 2.1 |
| Michael Allen | 10/31/23 | Editing Title Page, Table of Contents | 2.2 |
| Joshua Buscher | 11/21/23 | Editing Section 2.5, 2.7 | 3.1 |
| Caleb Hall | 11/21/23 | System diagram, operating environment, editing | 3.2 |
| Joshua Buscher | 2/5/24 | Removing all usage of words “AI”/”Artificial Intelligence” | 4.0 |
| Caleb Hall | 2/5/24 | Editing, applying change in scope | 4.1 |
| Joshua Buscher | 3/2/24 | Altering Reasons for Changes, editing requirements, | 5.0 |
| Calla Robison | 3/3/24 | Requirements 3.1, 3.2, 3.3, 3.4 | 5.1 |
| Michael Allen | 3/3/24 | Formatting the document | 5.2 |
| Michael Allen and Caleb Hall | 4/14/24 | Worked on implementing missed comments from Version 1 into Version 3 and new comments from Version 2.   Worked on Section 1 Introduction, modifying 1.0. Also Worked on Section 2 Detailing Requirements and Formatting Document. Updated use case diagram and description | 6.0 |

# Introduction

The AutoPen Project is a web-based tool created to revolutionize penetration testing in the fast-changing field of cybersecurity. AutoPen integrates the high-quality tool Burp Suite with a web-hosting service, guaranteeing a versatile testing procedure. This system is designed to facilitate autonomous testing. It includes a user-friendly dashboard for easy monitoring and analysis of results. The AutoPen platform performs assessments as needed, providing a cost-effective, efficient, and comprehensive solution for improving cybersecurity resilience in different companies.

Section 1 is an overview of the document's purpose and intended audience. Section 1 uses subsections to accomplish this. Subsection 1.1 covers the purpose of the document. Subsection 1.2 continues with the formatting conventions of the document. Next, Subsection 1.3 details the audience of the product and which further sections are of interest to them. Subsection 1.4 describes the scope of the product. Finally, Subsection 1.5 lists the references of the product.

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description and comprehensive outline of the AutoPen Project. This document is intended to serve as a foundational guide for the development and implementation of AutoPen, a web-based tool that leverages the capabilities of Burp Suite for advanced, automated penetration testing. The SRS outlines the functional and non-functional requirements, system behavior, user interfaces, and interactions between the system components. It aims to ensure that all project stakeholders, including developers, testers, and end-users, have a clear understanding of the system's objectives, capabilities, and constraints. This document also establishes the framework for ongoing project management and development activities.

## Document Conventions

1. The document will be in Times New Roman size 12 font.
2. The requirements have their own authority.

## Intended Audience and Reading Suggestions

This document is specifically intended for those engaged in the development, implementation, and upkeep of the AutoPen Project. The system's features and capabilities are intended for the primary audience of web developers and penetration testers. The user should prioritize Section 4, which provides a comprehensive overview of the capabilities of the system, and Section 5, which covers important aspects such as system performance, security, and quality specifications. Project managers and IT professionals should thoroughly examine the entire document, paying close attention to Section 2, in order to understand the overall product description, and Section 3, to understand the system specifications. This will facilitate the seamless integration of AutoPen into their IT infrastructure. Stakeholders, such as investors and senior management, may find Section 1 and Section 6 particularly relevant. These sections discuss the purpose of the document and include additional appendices that offer an executive summary of the project's scope and potential business impact. The SRS also fulfills educational objectives for cybersecurity students and researchers, who may find the entire document valuable for comprehending the practical implementation of theoretical concepts.

## Product Scope

The AutoPen project aims to create a web-based automated penetration testing system that utilizes the Burp Suite API for its core testing procedures. In scope functionality is a web platform that enables users to conduct vulnerability assessments and penetration tests using a user-friendly interface. The objective is to offer comprehensive reporting capabilities and immediate feedback mechanisms to facilitate quick decision-making and remediation procedures.

The AutoPen project does not include any functionalities related to Artificial Intelligence for conducting penetration testing or making autonomous decisions based on test results. The project will not create its own virtual machines for web browsing; instead, it will incorporate existing technologies. Additionally, the AutoPen will not possess hardware testing functionalities or provide assistance for testing non-web applications. The program will utilize the Burp Suite API and will not involve the development of testing algorithms beyond this product.

## References

* The URL of the web application is [https://www.autopentest.net](about:blank)/
* Jira site for agile sprint planning can be found at: <https://autopentest.atlassian.net/jira/software/projects/PEN/boards/1>
* Documentation for Burp Suite, the APT used to run penetration testing, can be found at: <https://portswigger.net/burp/documentation>
* Github for the project can be found at: <https://github.com/Caleb-Hall-1015/AutoPen>

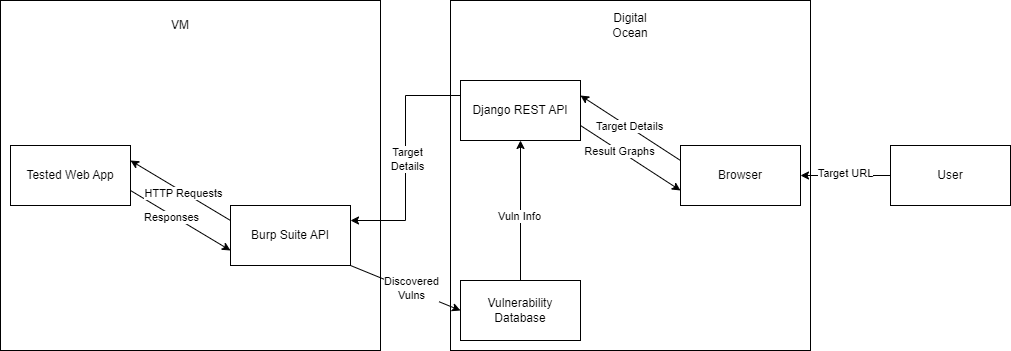
# Overall Description

Section 2 contains information regarding the product use and description. Subsection 2.1 covers the perspective of the product. The product function is located in Subsection 2.2. Subsection 2.3 contains the user information and classes. The operating environment is discussed in Subsection 2.4. Subsection 2.5 details the design and implementation constraints. In Subsection 2.6 there is information on how to find a product tutorial. The assumptions and dependencies are listed in Subsection 2.7.

## Product Perspective

The integration of automated penetration testing represents a new paradigm shift in the field of cybersecurity. By replacing manual systems and innovating with advanced technologies, organizations can enhance their security posture, reduce vulnerabilities, and stay ahead of potential cyber threats. The increased efficiency, accuracy, and scalability offered by automated penetration testing make it an indispensable tool in today's digital landscape. Embracing automation in penetration testing is not only a step towards better security but also a strategic investment in the long-term success and resilience of an organization's IT infrastructure.

.



**Figure 1: System Diagram**

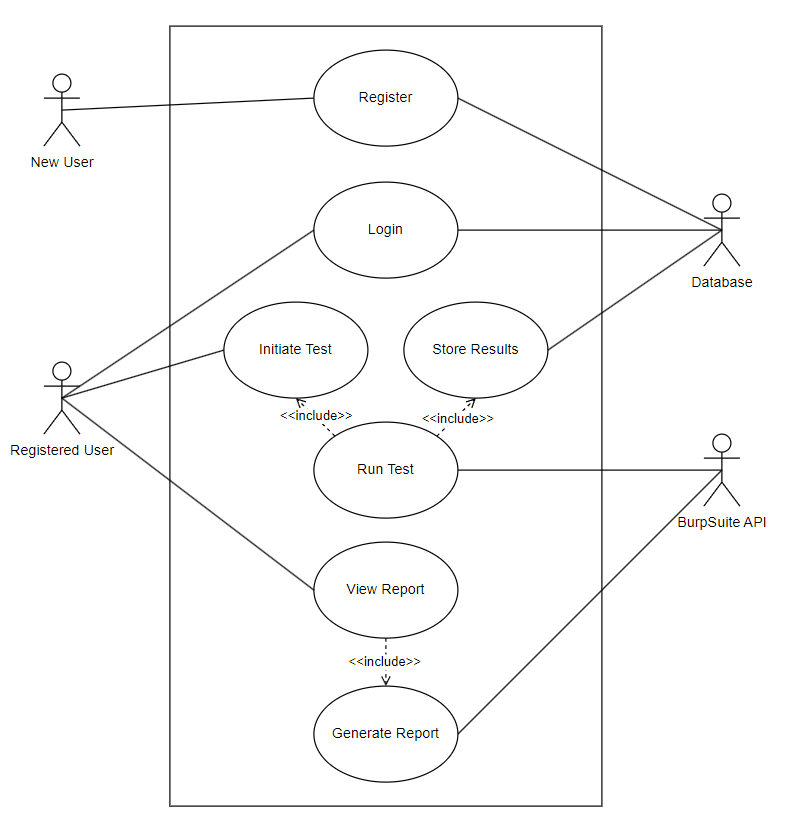
Figure 1 presents the system architecture, illustrating the flow of data and interactions between different components. At the core is a web application tested through a Virtual Machine (VM), which communicates with Burp Suite API to handle HTTP requests and responses, and to identify vulnerabilities. The discovered vulnerabilities are logged into a Vulnerability Database. Additionally, a Django RESTful API is employed to manage target details and vulnerability information, which can be accessed via a browser interface hosted on SiteGround. This browser interface allows the user to input a target URL and visualize result graphs and target details.

## Product Functions

1. The product must allow users to test a website for vulnerabilities
2. The product must be a functional website, allowing any user to visit using the internet

## User Classes and Characteristics

AutoPen has two types of users, a new user and a registered user. Additionally, there are two systems involved in the use cases: the database and the Burp Suite API.



**Figure 2: Use Case Diagram**

Figure 2 outlines the use case flow for AutoPen. It differentiates between registered and unregistered users, showing that registered users can proceed to user login, while new users must undergo user registration. Once authenticated, users can initiate a test or view previous reports. On the other end, the database stores user login records and the scan results. Once a user attempts to view a report, the API will generate the report.

## New User

New user is a user who has never used AutoPen before. They are any web developer, penetration tester, or individual interested in the service of AutoPen.

## Registered User

A registered user is one who has created an account with AutoPen. They are to provide a URL to scan and use the AutoPen system. During the scan, this user is to monitor progress and take manual control to cancel the scan in case of system failure.

## Database

The database is the system where the user records and scan results are stored. This system is to authenticate user login attempts and store and retrieve scan results.

## Burp Suite API

The API is responsible for conducting the vulnerability scans. Once given a URL, this system will perform a security assessment of the site and provide a report containing any vulnerabilities found within.

## Use Case & Relationships

**User Registration:** This use case allows unregistered users to create an account by providing necessary details such as username, email, and password. The system then validates this information and stores it, enabling the user to access the system with their new credentials. No prerequisites exist for this action, and upon completion, the user's data is stored in the system.

**User Login:** Registered users can log into the system by entering their credentials, which the system verifies. If the credentials are correct, the system grants access, establishing a user session. This process requires that a user account already exists and results in the user being logged in.

**Initiate Penetration Test:** Authenticated users can start a penetration test by specifying a target URL and the depth of the test. The system then processes this request and begins the test, requiring the user to be logged in prior to this action.

**Run Test:** Once a user requests a scan to begin, the Burp Suite API will test the site and collect information.

**Store Results:** At the conclusion of the testing cycle, the API will send its findings to the database.

**View Report:** A registered user can request to view the report of a finished test.

**Generate Test Report:** Upon requesting a detailed report from the system, the report will generate. This report includes findings on vulnerabilities, their severity, and suggested remediations. The generation of this report depends on the prior completion of a penetration test

## Operating Environment

To effectively utilize the required system, the user must have the following operating environment:

1. Any form of computer that can run a web page
   1. Windows, Apple, or Linux Desktop/Laptop
   2. Android or Apple Mobile Smartphone
2. Stable Internet Connection
   1. At least 1 Mb/s
3. Up-To-Date Website/Web Browser
   1. Latest version of Chrome, Firefox, Safari, or equivalent browser

## Design and Implementation Constraints

The design and implementation of AutoPen must consider the potential legal and ethical constraints associated with the device. By addressing these concerns, such as intellectual property rights, privacy, consumer protection, plagiarism, and human interaction, AutoPen can be developed and used responsibly. It is crucial to prioritize legal compliance and ethical considerations to ensure the successful integration and acceptance of AutoPen in various industries and domains.

Time is another constraint, as the full software development lifecycle must fall within a single academic year. From project vision to final deliverable, this project can take no longer than 10 calendar months, limiting the scope and depth of the project.

A third constraint is budgetary. Limited funds dictate that all tools used are either provided by the university or are available for a generally low price. As such, the project is unable to offer the full gamut of cybersecurity tools, rather it relies on the professional version of Burp Suite and the basic tier of Siteground.

Similar to financial constraints are technical constraints. The tier of Siteground is low, and as such the project has limited resources. This forces the Burp Suite API to be hosted on a system other than the Siteground cloud server as the scan requires more RAM and CPU cores than the cloud server provides.

The final constraint is skill, as the team is made up of intermediately skilled software engineers. As such, development of in-house penetration testing tools and advanced algorithms is not feasible.

## User Documentation

An instruction panel and video tutorial will be implemented into our website. The AutoPen website aims to enhance user experience, provide clear guidance, and support users throughout their penetration testing journey. This addition will ensure that users can effectively utilize the platform's capabilities and maximize the benefits of automated penetration testing.

## Assumptions and Dependencies

## The system shall adhere to General Data Protection Regulation (GDPR) and similar regional data protection laws.

## The system shall use a hosting platform that has integrated or allows the import of virtual machines.

## The system must use a hosting platform that has the ability to host back-end algorithm tasks.

## The system shall use Django RESTful API to implement back-end functionality.

# External Interface Requirements

Section 3 covers the External interface requirements associated with the product. The format of Section 3 is as follows: Section 3.1 covers the user interface requirements, Section 3.2 covers the hardware interface requirements, Section 3.3 covers the software interface requirements, and Section 3.4 covers the communications interface requirements.

Section 3 outlines the external interface requirements associated with the AutoPen product. This section is structured to cover various aspects of system interfacing, ensuring comprehensive functionality and interoperability. Specifically, Subsection 3.1 details the user interface requirements, Subsection 3.2 addresses the hardware interface requirements, Subsection 3.3 discusses the software interface requirements, and Subsection 3.4 explores the communications interface requirements. Each subsection is designed to provide clear, concise specifications for each interface category to support effective system integration and operation.

## User Interfaces

## Main Interface for Test Execution

The main interface of the software shall include the following elements:

## Input Fields for Target:

## The interface shall include specifically labeled input fields for entering IP addresses or domain names required for test execution, as shown in Figure X.

* + - * 1. Input fields shall accept alphanumeric characters
        2. Input fields shall support copy-paste actions.

## Start Test Button:

## The interface shall include a "Start Test" button

## The “Start Test” button shall begin the penetration test.

## The “Start Test” button shall be colored and positioned centrally on the page for easy access, as shown in Figure X.

## Results Display Area:

## The interface shall include a dedicated area for displaying test results, organizing information into categories such as vulnerabilities found, severity levels, and suggested actions, as shown in Figure X.

## Test results shall be colorized based on severity (red for high, yellow for medium, green for low).

## GUI Standards

The system shall adhere to modern web application design principles, ensuring ease of use and intuitive navigation in the graphical user interface (GUI).

## Consistent Design:

## The GUI shall maintain a uniform font type and size across all pages, using Arial, 14-point for text and 16-point for headings, as shown in Figure X.

## Navigation buttons such as "Help," "Home," and "Settings" shall be located at the top of every page, consistently designed and colored, as shown in Figure X.

## Intuitive Navigation:

## Navigation through the GUI shall be facilitated by a fixed sidebar containing links to all major sections of the application, as shown in Figure X.

## Responsive Layout:

## The layout shall adjust to fit the screen sizes of devices ranging from mobile phones to desktop monitors without losing functionality or aesthetic theme, as shown in Figure X.

## User-Friendly Controls:

## All interactive GUI elements like buttons, dropdowns, and sliders shall respond to user interactions with visual feedback such as highlighting or animation, as shown in Figure X

## Feedback Mechanisms:

## Upon completion of actions within the GUI, users shall receive instant feedback through messages like "Action Successful" or "Error Detected," displayed in a temporary, dismissible pop-up, as shown in Figure X.

## Error Messages

## Error messages shall be displayed as red pop-up banners at the top of the screen, providing clear and concise explanations of the issues encountered, as shown in Figure X.

## Components with UI

The system shall incorporate several critical components with user interfaces, each designed to enhance user interaction and system usability. Here are the specific requirements for each:

## Main Dashboard:

## The system shall display a main dashboard with basic user information (name, email) such as real-time data and key metrics relevant to the user's role.

## The main dashboard shall be customizable, allowing users to arrange and prioritize displayed elements based on individual preferences.

## Settings:

## The system shall include a settings interface accessible from the main navigation.

## The system shall allow users to modify settings related to system behavior, notifications, and personalization.

## Changes made in the settings interface shall persist across user sessions.

## Results Page:

## The system shall present a results page with scan result information on completed processes, tests, or relevant activities.

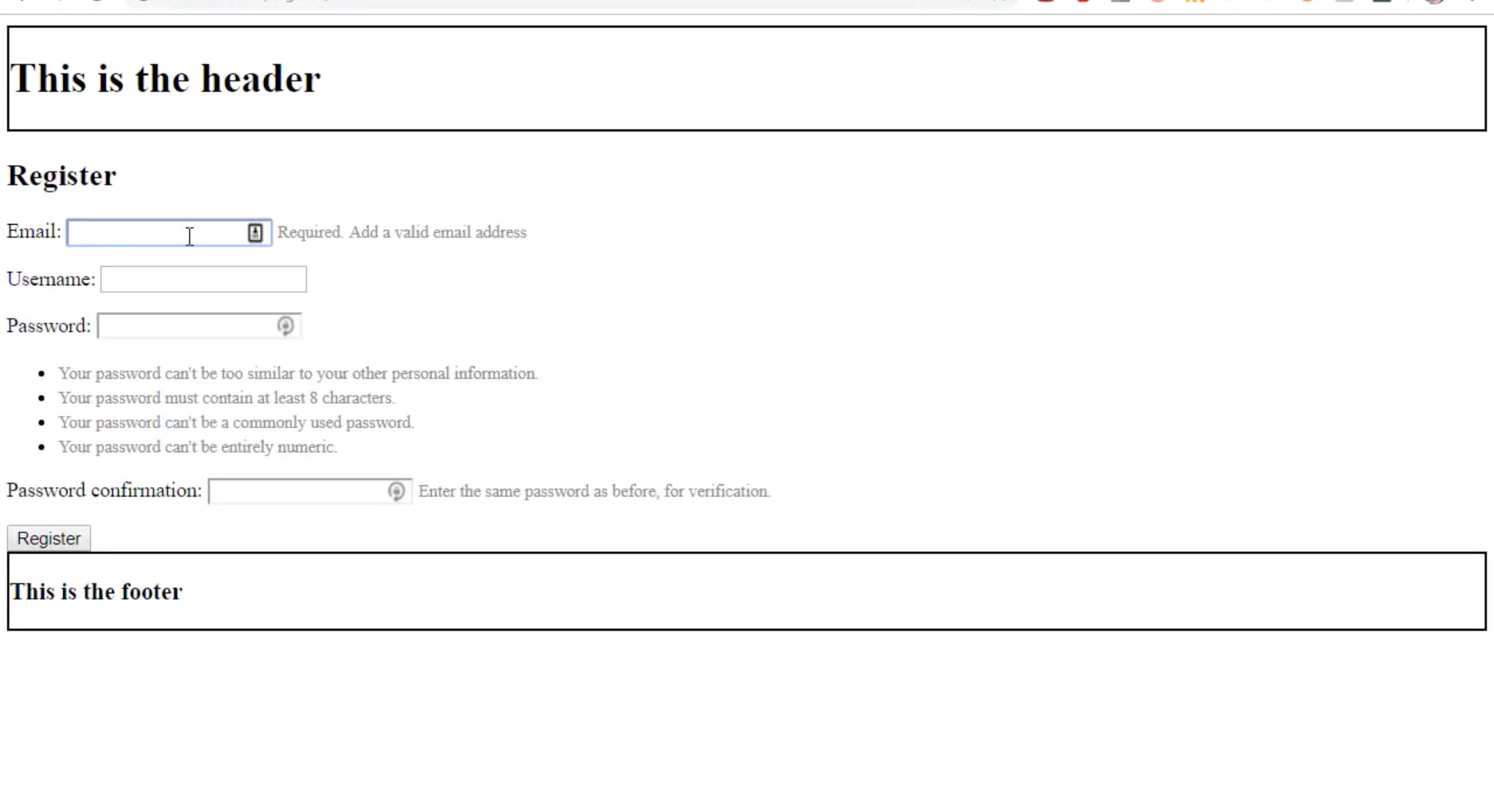
## Users shall be able to filter and sort results based on various parameters such as date, severity, and type.

## User Profile:

## The system shall feature a user profile interface accessible from the main menu.

## The system shall allow users to view and edit their profile information, including but not limited to name, contact details, and password.

## Changes to the user profile shall be reflected throughout the system within 24 hours.



**Figure 3: Example User Interface Screen**

Figure 3 depicts a screenshot of an example user interface for a registration page. The interface includes a header at the top and a footer at the bottom, framing the registration form. The form fields prompt the user for an email address, with a note indicating that a valid email address is required, a username, and a password, along with a password confirmation field. Accompanying the password field are instructions specifying the password criteria, such as a minimum length and restrictions on similarity to other personal information. The layout is simple and user-centric, designed to facilitate an easy and secure registration process.

## Hardware Interfaces

## The interface shall allow users to access the website across multiple devices, including PCs, laptops, tablets, and mobile phones.

## The application shall be hosted on Siteground's web hosting platform.

## User interaction with the system shall primarily occur through the website's servers.

## The system shall allocate a minimum of 20 GB storage to the web server.

## The system shall employ Secure HTTP (HTTPS) for all web communications to ensure data security.

## The system shall produce real-time updates through the use of WebSockets

## Software Interfaces

## Connected Software:

## The system shall utilize a PostgreSQL database hosted on Siteground.

## The system shall operate within Sitegrounds' web hosting environment.

## The system shall be compatible with a Kali Linux VM that uses Siteground cloud services.

## The backend of the system shall be developed using the Python Django Web Framework.

## The frontend of the system shall be developed using the React Web Framework.

## Shared Data Mechanism:

Given the constraints of web space, the system shall prioritize the optimization of data storage and retrieval processes.

## Database caching strategies shall be implemented to enhance efficiency in data storage and retrieval.

## The system shall integrate with the Burp Suite API to facilitate vulnerability and penetration testing services.

## Communications Interfaces

## Functions:

## The system shall support communication functions for email notifications, web browser alerts, and server-to-server communications.

## Message Formatting:

## The system shall utilize JSON for data interchange between different components of the system.

## Communication Standards:

## The system shall employ Secure FTP (SFTP) for secure file transfers.

## HTTPS shall be used for secure web communications, adhering to specific configurations provided by Sitegrounds.

## Security/Encryption:

## All communications within the system shall be encrypted using TLS 1.3 or as supported by Sitegrounds to ensure data security and integrity.

## Data Transfer Rates:

## The system shall be optimized for broadband connections, with a minimum recommended speed of 5Mbps to facilitate efficient data transfer.

## Data In/Out:

## The system shall accept incoming data including user credentials, target specifications, and user configurations

## The system shall generate outgoing data in the form of test results, progress updates, and error messages

## Services & Communications:

## The system shall employ a Django RESTful API for seamless communication between the frontend and backend components

## Comprehensive API documentation shall be made available in a separate document for reference and integration purposes

## Synchronization:

## The system shall employ WebSocket for real-time data synchronization between the client and server components, contingent on Sitegrounds' support for WebSocket technology

# System Features

Section 4 is about the system feature of the product, describing what they do, their stimuli and requirements. The formatting for Section 4 is as follows: 4.1 is about automated penetration testing, and Section 4.2 is about user profile management.

## System Feature: Automated Penetration Testing

## Description and Priority

Allows for the penetration testing of websites. Priority: medium

This section uses a score from 1 to 10 where 1 is the worst and 10 is the highest

Benefits: 9 (Significant user value)

Penalty: 4 (Users may opt for competitors if not implemented well)

Cost: 7 (Complex to implement, but feasible)

Risk: 8 (Potential for misuse or false results)

Overall: 7

## Stimulus/Response Sequences

1. User inputs target specifications.
2. System validates input and prompts confirmation.
3. User initiates the test.
4. System displays progress and provides real-time updates.
5. System presents results upon completion.

## Functional Requirements

## System shall allow users to input the target URL.

## System should provide real-time feedback on the penetration testing progress.

## System must handle potential errors, such as unreachable targets, gracefully by notifying the user.

## System shall ensure user authentication before initiating any penetration tests.

* + - 1. Normal users shall only have the ability to initiate tests and viewing initiated tests.

## System Feature: User Profile Management

## Description and Priority

Allows users to create, edit, and manage their profiles. Priority: Medium

Benefits: 7 (Enhances user experience)

Penalty: 5 (Users might be dissatisfied with static profile settings)

Cost: 3 (Standard feature in web applications)

Risk: 7 (Data privacy concerns)

## Stimulus/Response Sequences

1. Users will log in or sign up.
2. System presents a dashboard or user profile page.
3. User edits profile information.
4. System validates and saves changes, then confirms with the user.

## Functional Requirements

## System shall provide user registration and login functionality.

## System should allow users to edit their profile information, including name, email, and contact details.

## System must securely store and encrypt user passwords.

## System must allow password recovery through a secure method, such as email verification.

# Other Nonfunctional Requirements

Section 5 is about covering all nonfunctional requirements that were not mentioned in earlier sections. The formatting of Section 5 is as follows: 5.1 is about requirements relating to performance, 5.2 is about requirements related to safety, 5.3 is about requirements related to security, 5.4 is about general software quality attributes, and 5.5 is about the business rules related to the product.

## Performance Requirements

## The system shall process user inputs within 2 seconds.

## The system shall make regular, real-time progress updates with no lag exceeding 3 seconds.

## The system shall support simultaneous penetration tests from 100 users without performance degradation.

## The website shall load within 3 seconds.

## Safety Requirements

## The system shall strictly adhere to the target specified by the user.

## The system shall incorporate rate limiting.

## The system shall provide clear disclaimers and guidance to users on ethical use.

## Security Requirements

## The software shall encrypt all user data using industry-standard protocols for password hashing.

## The software shall protect user sessions against session hijacking.

## The software shall protect user sessions against cookie theft.

## The software shall implement multi-factor authentication for enhanced user account security.

## The software shall host all algorithms on a separate server, isolated from the primary website, with direct access to these models by external entities being strictly prohibited.

## The software shall conduct regular security audits.

## The software shall encrypt all data communication.

## The software shall only allow registered and authorized users to view penetration test reports,

## The software shall log and report unauthorized access attempts.

## The software shall have different levels of system functionalities

## Data shall be accessible depending on the user role.

## Software Quality Attributes

The following requirements are basic goals that the systems should attempt to follow, and should not be viewed as rigid requirements like other requirement sections. If any requirements in this section result in conflicts, 5.4.2 and 5.4.7 should be held in a higher priority than all other software quality attribute requirements.

## The software should be able to accommodate changes in penetration testing methodologies or emerging threats with minimal modifications.

## The system should be accessible 99.9% of the time for users when they require penetration testing.

## The penetration testing results shall have an accuracy rate of at least 95%.

## The system shall be able to integrate new tools or functionalities without major architectural changes.

## The system shall be able to seamlessly interact with common third-party platforms or services, if needed, such as vulnerability databases.

## The system shall be web based, with any auxiliary tools or scripts working across different OS platforms.

## The system shall not crash during a penetration test, handling errors gracefully.

## The system shall be designed in a way such that they can be used in different contexts or projects.

## The system shall handle unexpected inputs or situations without failing

## The system shall have provisions to be tested easily, both for individual units and end-to-end functionality.

## The user interface shall be intuitive with a preference towards ease of use, even if it comes with a slight learning curve.

## Business Rules

## Only admins or creators of the pentesting software shall be allowed to alter system mechanisms.

## 

# Other Requirements

Section 6 is about covering all miscellaneous requirements not covered earlier. The formatting for Section 6 is as follows, 6.1 is about requirements relating to reuse objectives, 6.2 is about requirements related to the database, 6.3 is about internationalization requirements, 6.4 is about legal requirements, and 6.5 is about requirements related to accessibility.

## Reuse Objectives

## Components of the system, especially the machine learning models and data processing units, shall be modularly designed for potential reuse in other related projects.

## RESTful APIs shall be made to be able to be used internally and potentially opened up for third-party integrations or other projects.

## Database Requirements

## The database shall be scalable to handle an increasing number of user records and penetration test results.

## The database shall make regular backups, both incremental and full.

## The database shall be scalable to handle an increasing number of user records and penetration test results.

## The database shall have strict role-based access to ensure that only authorized personnel can view or modify the database.

## The system shall use a third-party vulnerability database to retrieve vulnerability metrics.

## Internationalization Requirements

## The system should support multiple languages, starting with English, Spanish, and French.

## The system should be able to handle and convert multiple currencies if there is a payment system in place.

## The system shall have user profiles and scheduling features with accurate user time zones

## Legal Requirements

## The system shall adhere to international data protection regulations like GDPR (General Data Protection Regulation) for European users and CCPA (California Consumer Privacy Act) for California residents.

## The system shall have features to prevent misuse, with users needing necessary permissions to conduct penetration tests on the target.

## The system shall properly license, and follow the terms and conditions of all third party tools, libraries and services.

## Accessibility Requirements

## The website shall be designed following W3C Web Content Accessibility Guidelines.

## The website shall have adequate contrast ratios and readable fonts.

## The website shall be navigable using just the keyboard.

# Appendix A: Glossary

Penetration Testing (Pentest): The practice of testing a computer system, network, or web application to find vulnerabilities that attackers could exploit.

SRS: Software Requirements Specification. A document that describes the features, behaviors, and attributes of a software system.

TLS: Transport Layer Security. A protocol ensuring privacy and data security between two communicating applications.

RESTful API: Representational State Transfer. A set of rules that developers follow when they create their API, allowing for interaction between systems using HTTP.

# Appendix B: Analysis Models

**Stakeholders:**

1. Users (Registered and Unregistered)
2. Developers and IT Administrators
3. External vulnerability databases
4. Payment gateway providers
5. Hosting service providers

**System Boundaries:**

1. The system interacts with external entities, including external vulnerability databases, payment gateways, and cloud storage services.
2. Users interact with the system via a web-based interface, which includes data entry screens and data display screens.
3. The system operates within a cloud-based environment, including cloud servers for hosting and data storage.

**Non-Functional Requirements:**

1. Security and compliance with legal constraints, including access control, encryption, and adherence to relevant laws and regulations.
2. Scalability to handle increased user load and data storage needs.
3. Availability to ensure that the system is accessible and operational.
4. Performance to provide efficient penetration testing and report generation.
5. Usability with a user-friendly web interface.
6. Reliability for accurate and consistent penetration testing results.
7. Maintainability for ongoing updates, patches, and system enhancements.

# Appendix C: To Be Determined List

1. Database Selection: Specific database technology to be used hasn't been finalized.
2. Third-Party Integration: Deciding on whether to integrate third-party vulnerability databases directly.
3. Pricing Model: How users will be charged for the service is TBD.
4. Notification System: How and when users receive notifications about test results or system updates.
5. User Data Retention Policy: Duration and conditions under which user data and reports will be stored.

(These are placeholders and might change based on the actual pending decisions in a real-world project.)