Software Requirements Specification

for

<SecureSpot>

Prepared by

Batch:A

Deepanshu Agarwal	**<2021300002>**	**deepanshu.agarwal@spit.ac.in**
Kunal Bhatia	**<2021300010>**	**kunal.bhatia@spit.ac.in**

Instructor:	**Prasenjit Bhavathankar**
Course:	**Software Engineering**

Date:	**04/09/23**

1 Introduction

Introducing "SecureSpot" - A Central Hub for Crime Awareness! The user-friendly website SecureSpot was created to keep you informed and involved in local safety. By authoring articles about local crimes, you may report, discuss, and learn more about them here. Stay updated on local crime incidents with our subscription feature. You may also express your ideas and comments in the "Community Insights" section that appears beneath each post, which promotes communication and collaboration. When it comes to staying informed, exchanging ideas, and enhancing local safety, SecureSpot is the place to be.

1.1 Purpose

This paper is like a roadmap for making the SecureSpot (Crime Management Website) better. The website is all about helping people getting crime alerts , reporting crimes , crime analysis and staying safe. This guide explains exactly what the website should do and how it should work. It's like giving directions to the people building the website so they know what to create and how to make it work well. It's like a plan that helps everyone working on the website understand what needs to be done. This way, the website can become a strong tool for reporting crimes and making communities safer.

1.2 Product Scope

The SecureSpot Management Website is a digital platform that helps communities deal with crime and safety more effectively. It acts as a bridge between individuals, local authorities, and the community, making it easy to report crimes, get real-time updates, and communicate efficiently. This website's purpose is to empower citizens by giving them a secure and user-friendly way to report incidents and stay informed about what's happening in their area. It benefits both users and authorities, promoting collaboration for a safer community.

This solution focuses on users, making it quick and easy for citizens to report incidents. It also provides an interface showing reported incidents, increasing community awareness and providing timely information. Local authorities receive notifications, allowing them to respond promptly to incidents and maintain law and order. Overall, the SecureSpot Website aims to create safer neighborhoods by encouraging community involvement, improving crime reporting, and boosting confidence among users and authorities. Additionally, it offers a detailed crime analysis feature, including statistical data and visualizations of reported crimes within the state, enhancing users' understanding of local crime trends.

.

1.3 Intended Audience and Document Overview

The document is geared toward a variety of readers, each of whom has a unique role and set of interests:

- Developers: The technical specifics, system design, and implementation facets of the "SecureSpot" project will be of interest to developers.

- Users: Users are interested in understanding how to use the website effectively, including how to report crimes, subscribe to updates, and engage in discussions. They seek a user-friendly experience.

- Professor: In an academic context, they may be interested in the project's objectives, methodologies, and the demonstration of theoretical and practical knowledge.

This document contains the functional and non-functional requirements of the system and the Data Flow Diagram (DFD) of the system in the following pages.

1.4 Definitions, Acronyms and Abbreviations

- WA : Web Application
- IEEE : Institute of Electrical and Electronics Engineers
- API: Application Programming Interface
- GUI: Graphical User Interface
- HTTP: Hypertext Transfer Protocol
- HTTPS: Hypertext Transfer Protocol Secure
- JSON: JavaScript Object Notation
- SRS: Software Requirements Specification
- UI: User Interface
- URL: Uniform Resource Locator
- UX: User Experience

1.5 Document Conventions

In general this document follows the IEEE formatting requirements. Used Time New Roman font size 11, or 12 throughout the document for text. Used italics for comments. Document text is single spaced and maintains the 1” margins found in this template. For Section and Subsection , their titles are bold and font size 14.

1.6 References and Acknowledgments

This SRS document refers to the following documents and resources:

- O'Hara, J.M. and Fleger, S., 2020. *Human-system interface design review guidelines* (No. BNL-216211-2020-FORE). Brookhaven National Lab.(BNL), Upton, NY (United States).

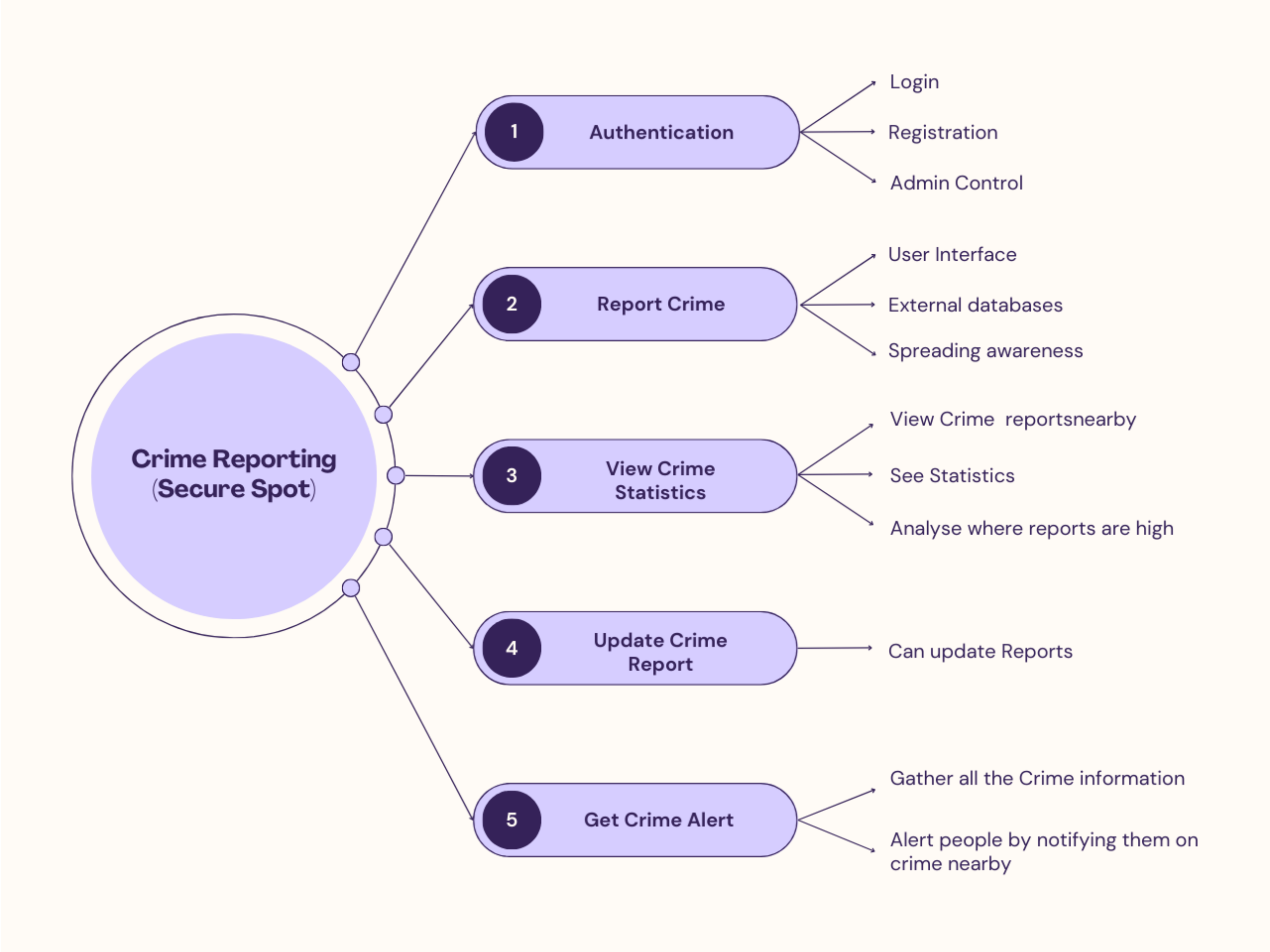
- Hazra, D. and Aranzazu, J., 2022. Crime, correction, education and welfare in the US–What role does the government play?. *Journal of policy modeling*, *44*(2), pp.474-491.

- Mamak, K., 2022. Categories of fake news from the perspective of social harmfulness. In *Integrity of scientific research: Fraud, misconduct and fake news in the academic, medical and social environment* (pp. 351-357). Cham: Springer International Publishing.

2 Overall Description

2.1 Product Perspective

"*SecureSpot*" is a standalone product that serves as a comprehensive solution for community-based crime reporting and awareness. It is not part of an existing product family, nor does it replace any existing systems. The software provides a self-contained platform for users to report crimes, subscribe to updates, and engage with their community regarding safety matters. It operates independently, interfacing with user devices (web and mobile) and external databases for crime data while maintaining its own functionality and features.



2.2 Product Functionality

"SecureSpot" offers a range of key functionalities to empower users and enhance community safety. These include:

- *Crime Reporting*: Users can easily report details of known crimes in their locality, providing essential information such as location, date, time, and incident description.

- *Subscription Feature*: Users can subscribe to receive real-time updates on crimes occurring in their area of interest, ensuring they stay informed about safety concerns.

- *Community Insights*: The platform provides a space for users to share their thoughts, opinions, and insights on specific crime reports, promoting community engagement and collaboration.

- *User Profiles*: Users can create and manage their profiles, facilitating personalized interactions with the platform and tracking their contributions.

- *Notification System*: "SecureSpot" offers a notification system to keep users informed about new crime reports, comments on their articles, and updates relevant to their subscriptions.

- *Search and Filtering*: Users can search for specific crimes or filter information based on criteria such as location, crime type, and date, making it easy to find relevant information.

- *Mobile Accessibility*: The platform is accessible via web browsers and mobile devices, ensuring users can access critical information and updates while on the go.
- *Privacy Measures*: Robust privacy and security measures are implemented to protect user data and maintain anonymity when required. It also allows users to post anonymously.

- *Crime Analysis*: Users can access a dedicated analysis dashboard that presents statistical data and visualizations, showcasing the types and frequency of reported crimes within their state or region.

2.3 Users and Characteristics

In the context of Crime analysis and reporting, it's essential to identify and understand the diverse user base that will interact with the system. The primary user categories include:

- *Regular Users:* These are members of the community who use "SecureSpot" on a daily basis to report crimes, remain informed, and take part in conversations. Depending on their interests and nearby events, they may use the platform intermittently. Regular users must be able to browse both web and mobile applications, therefore they often have basic to moderate technological skills. They are given basic user rights that let them participate in debates, report crimes, and subscribe to updates.

- *Documentation Writers :* Articles and crime reports can be used as primary sources by document writers to learn more about previous incidents and interactions within the community.They are able to create thorough user manuals and instructions using the platform's real-world case studies as inspiration.

- *Researchers*: Researchers can examine the crime statistics provided by "SecureSpot" to investigate hotspots, trends, and patterns of crime in various cities. They can also collect feedback from the general public on issues like crime prevention and safety.

- *Law Enforcement Agencies*: Law enforcement agencies can access crime reports on the platform to respond quickly to incidents and gather valuable information for investigations. They can collaborate with the community to improve safety measures

2.4 Operating Environment

Platform minimum requirements:

- *Hardware*: Internet-connected desktop or laptop computers, tablets, cellphones, or laptops.

- *Operating system*: Compatible with popular desktop and mobile operating systems like Windows, macOS, Linux, and iOS and Android.

- *Web browsers*: Current web browsers like Microsoft Edge, Apple Safari, Mozilla Firefox, and Google Chrome.

- *Internet connection*: In order to use and access the features of the platform, you must have a dependable internet connection.

- Software Components: The software interacts seamlessly with various software components and libraries, including web servers (e.g., Apache, Nginx), database management systems (e.g., MySQL, PostgreSQL, MongoDB), and web frameworks (e.g., Node.js, Django, Ruby on Rails). It also integrates with standard authentication protocols (e.g., OAuth, LDAP) to provide secure user access

2.5 Design and Implementation Constraints

The design and implementation of SecureSpot will be subject to several constraints that will shape the development process:

- *Cross-Browser Compatibility:* The platform must function correctly on a variety of mobile devices and web browsers. To ensure a consistent user experience, ensuring compatibility with various operating systems and browsers raises issues for design and testing.

- *Data security:* It is crucial due to the sensitive nature of crime reporting. To safeguard user data and uphold privacy, "SecureSpot" must include strong encryption, access controls, and user authentication procedures. It is crucial to adhere to security legislation and requirements.

- *Scalability:* "SecureSpot" should be able to scale effectively to meet growing traffic and storage needs as the user base and data volume rise. To avoid performance bottlenecks, scalability considerations must be incorporated into the design process.

- *Programming Languages and Frameworks*: The project will be limited by the predetermined programming languages and development frameworks used by the customer's organization. The development stack for the program might need to be compatible with the organization's current technological environment.

- *Mobile Responsiveness*: In particular on mobile devices, "SecureSpot" must be built to be responsive to different screen sizes and orientations. Due to this restriction, responsive web development techniques and careful UI/UX design are required.

- *Third-Party Services*: The incorporation of third-party services (such as notification services) may impose restrictions and dependencies on the accessibility of the service, modifications to the APIs, and compliance with data formats.

- *Maintainability*: The client's company may be in charge of maintaining the program, even though the development team may have created the initial version. To guarantee long-term maintainability and make it simple to make improvements in the future, adherence to code standards, documentation procedures, and design guidelines is essential.

2.6 User Documentation

In order to help customers get the most out of the platform, "SecureSpot"'s user manual includes a comprehensive collection of resources. This offers comprehensive user guides for subjects like criminal reporting, subscriptions, and community involvement. Within the application, a real-time, context-sensitive help system is available. Users may immediately become familiar with the features of the platform with the help of interactive tutorials and step-by-step instructions. The Frequently Asked Questions (FAQs) section responds to frequent questions and offers quick answers. Video guides provide visual demonstrations of important features in addition to written literature. All documentation complies with industry standards for usability and accessibility, making it simple for users to get the details they require to get the most out of "SecureSpot."

2.7 Assumptions and Dependencies

The following assumptions could have an impact on the requirements listed in the SRS for "SecureSpot":

- *Integrations with Third Parties*: The platform's operation may be impacted if third party services that are used for crime data retrieval and notifications alter or stop being compatible with it.

- *User Accessibility*: It is essential to assume that the majority of users have internet connection and contemporary gadgets. The reach and usability of the platform may be impacted if a sizable section of the target group is unable to access these resources.

- *Legal and Regulatory Compliance:* It's critical to make the assumption that the target regions' legal and regulatory standards for data privacy and crime reporting stay mostly constant. Changes to these rules can call for substantial platform modifications.

- *Data Sources*: An important consideration is making the assumption that external data sources for crime statistics are trustworthy and regularly updated. The accuracy of the information supplied on the platform can be impacted if these sources turn out to be inaccurate or out of date.

- *External Libraries and Frameworks*: It is envisaged that some components from open-source or for-profit sources will require access to external libraries, frameworks, or other resources. These dependencies could become unstable or vulnerable, which could have an impact on the software's stability and security.

- *Internet Connectivity*: User access to the software assumes stable and reliable internet connectivity. Any disruptions or outages in internet services could hinder users' ability to access and utilize the tool.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

SecureSpot will feature a user interface (UI) designed for intuitive and efficient interaction. The primary user interfaces are as follows:

- *User Home Page*: Upon login, users will be directed to a HomePage displaying recent crime reports, updates on subscribed areas, and quick access to essential features.

- D*ark & Light Mode:* Users will be able to choose between light and dark mode according to their preference.

- *Crime Reporting Page*: Users can report crimes through a form where they provide incident details, including images, location, date, time, and description. The form will follow an intuitive layout with clear input fields and options. It will incorporate a user-friendly text editor.

- *Crime Search and Filtering*: A search interface will enable users to search for specific crimes, with filtering options based on location, crime type, and date. Search results will be displayed in an organized manner.

- *Crime Details Page*: Clicking on a specific crime report will lead users to a detailed view, displaying all available information about the incident, along with comments and discussions from other users.

- *User Profile Management*: Users will have a profile management interface to update personal information, preferences, and view their activity history.

- *Discussion Forums*: Users can participate in discussions related to specific crime reports. The discussion interface will display comments, user interactions, and an option to contribute to the conversation.

- *Notifications Center*: Users will access their notification preferences and view updates in the notifications center. This interface will list real-time crime alerts and comments on their posts.

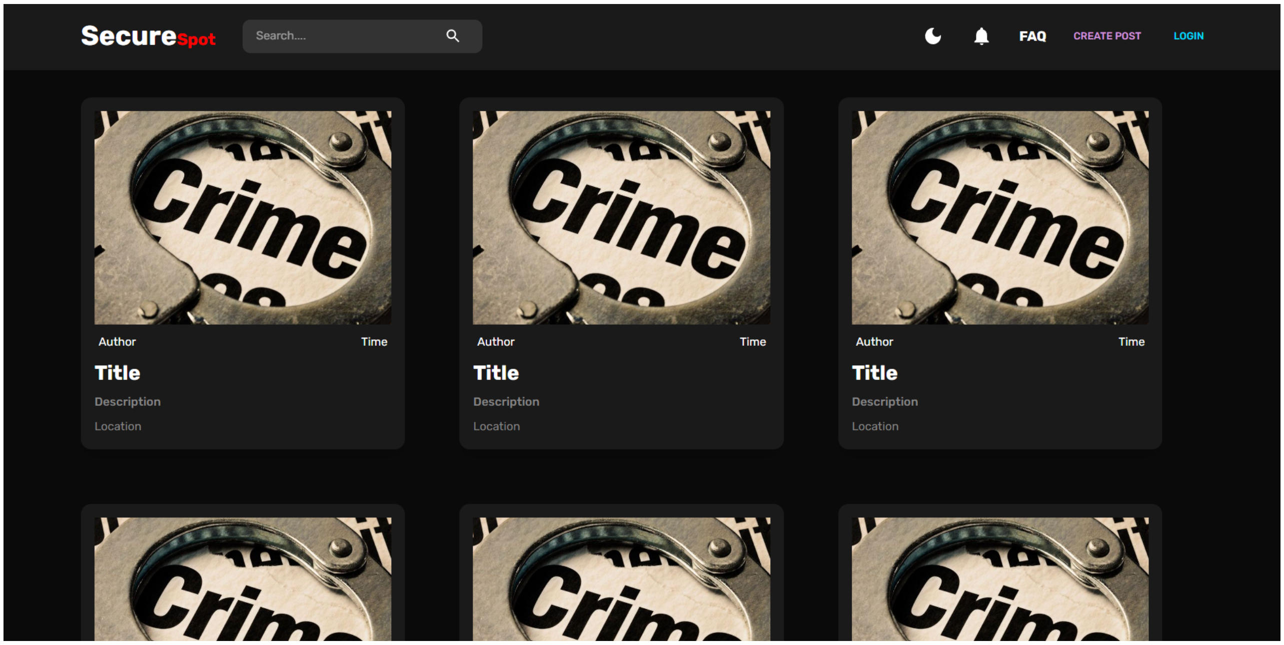
- *Analysis Dashboard*: A dedicated dashboard will present statistical data and visualizations regarding reported crimes in the user's state or region.

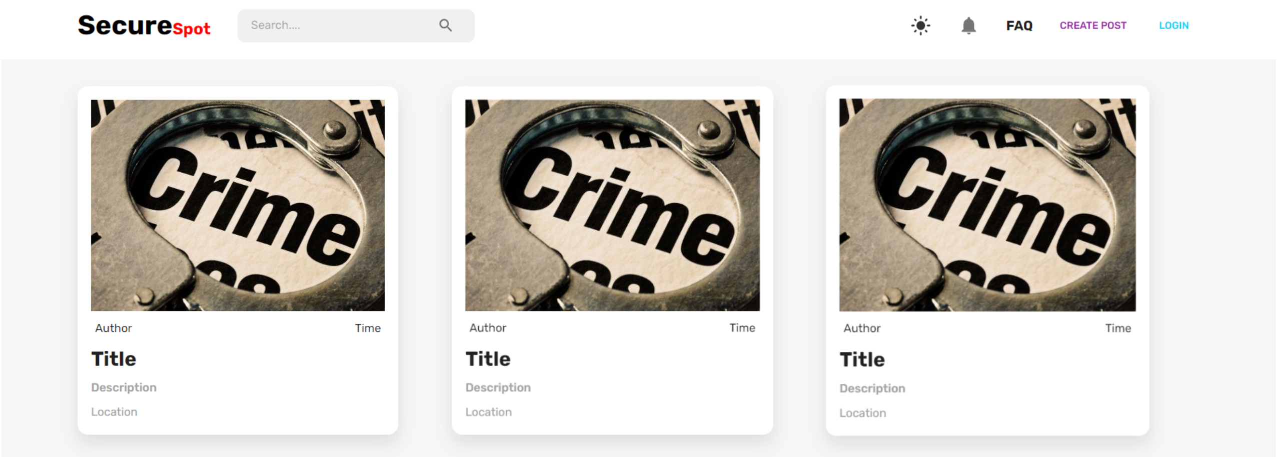
- *Help and Support*: An intuitive help center will provide assistance and FAQs, while error messages will follow standard conventions for clarity and guidance.
- *Navigation Menu*: A consistent navigation menu will be available on every screen, ensuring easy access to all sections of the platform, including Home, Report Crime, Notifications, Profile, and Analysis.

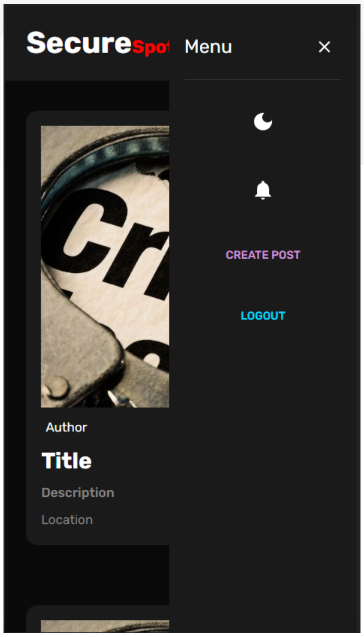
- *Logout*: A logout function will be accessible from any screen to ensure secure session termination.

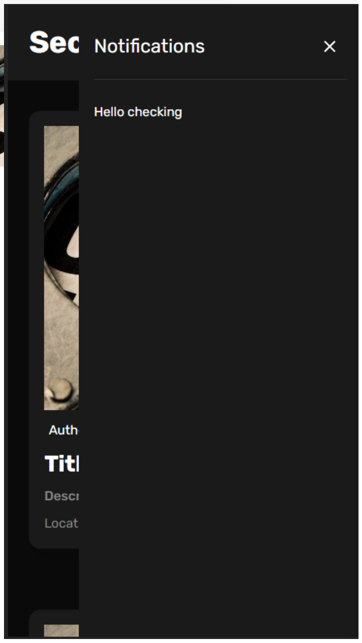
While textual descriptions provide an overview of the user interface components, optional graphical user interface screenshots can be provided to illustrate the design and layout of these screens, enhancing user understanding and providing a visual representation of the interface's logical characteristics.

Graphical User Interface screenshots

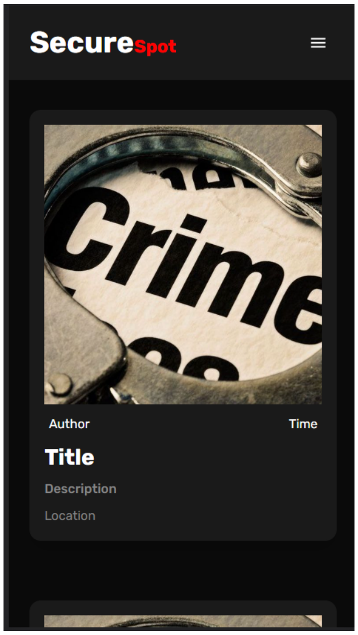








Hello checking



3.1.2 Hardware Interfaces

A list of the logical and physical features of the "SecureSpot" hardware interfaces:

- *Devices Supported*: "SecureSpot" is usable on a range of user devices, including PCs, laptops, tablets, and smartphones, enabling widespread accessibility.

- *Web-based Interaction*: Users interface with the platform through web browsers on their devices, making it simple to input data, access information, and use features.

- *Server Management*: The application manages user accounts, data processing, and communication using common web protocols on a web server hosted in a data center or cloud architecture.

- *External statistics Sources*: To improve the platform's informational value, "SecureSpot" connects with external databases or APIs to retrieve real-time crime statistics.

- *External notification systems:* They are used for real-time warnings and updates, ensuring that consumers are kept informed about security issues effectively.

These features demonstrate the software's adaptability to a variety of user devices, its reliance on web-based user interfaces, and its capacity to seamlessly integrate with external data sources and notification systems to offer users useful information and functionality.

3.1.3 Software Interfaces

The software runs under Linux or Windows operating systems. This web application uses two different databases for users and articles. Following are the salient elements outlining the relationships between "SecureSpot" and other software parts:

- *Software Component*s: "SecureSpot" communicates with Node.js, Express.js, MongoDB, React, and npm packages (such as Nodemailer, Material UI, Multer, and Morgan).

- *Operating Systems*: Windows, Linux, and macOS are just a few of the operating systems on which the program runs without a hitch.

- *User inputs*: Through web browsers, users communicate with the platform and provide information like crime reports and comments.

- *Data Flow*: Node.js and Express.js back-end components that interface with MongoDB for data retrieval and storage process inputs.

- *Additional Service*s: Npm packages (Multer, Nodemailer, Material UI, Morgan) make it easier to access services like file handling, email notifications, and user interface elements.

- *Communication*: Data transport and interactions are made as efficient as possible by using HTTP-based APIs.

- *Cross-Platform Compatibility*: The software is designed to maintain cross-platform compatibility, minimizing platform-specific code and enhancing overall maintainability and scalability.

3.1.4 Communications Interfaces

To ensure effective data exchange, "SecureSpot" relies on communication standards that have been created. It makes use of HTTP/HTTPS for smooth communication between client devices and the server, guaranteeing both speed and security through encryption. To secure sensitive content during transmission, email notifications that are controlled by SMTP are taken into consideration for encryption. Multer speeds up safe file uploads, allowing users to confidently share photos and documents. The software will provide RESTful APIs (Representational State Transfer) to enable integration with external applications and services. These APIs will follow industry best practices for data formatting, using widely accepted standards like JSON (JavaScript Object Notation) for message formatting.. While user information is protected throughout its path through the system by encryption and other security measures, data transfer rates are adjusted to preserve responsiveness.

3.2 Functional Requirements

In order to comprehensively capture the intended behavior of the web application, we can categorize the functional requirements into several key areas, each addressing specific aspects of the software's functionality:

1. Managing user accounts:

    - *User Registration:* Users should be able to sign up for an account by giving a legitimate email address and password.

    - *User Login:* Users should be able to safely log into their accounts after registering.

    - *Password Reset:* If users forget their login information, they should have the ability to reset their passwords through email.

    - *Update Profile:* Users should be able to modify their profiles, alter their profile images, and update their personal information.

    - *Account Deactivation*: Users ought to be able to delete or deactivate their accounts with the proper confirmation procedures.

2. Reporting of Crime:

    - *Incident Submission*: Users should have the ability to submit occurrences by including information about the location, the date, the time, the type of crime, and a description.

    - *Image Upload*: Users should have the option to upload files that are connected to the reported occurrence in order to provide further details.

    - *Edit and Delete Reports*: Users should have the choice to amend or delete their own incident reports if necessary.

    - *View and Comment on Reports*: In order to promote conversation and provide more information, users should have the ability to view and comment on incident reports that have been submitted by others.

3. Crime Analysis:

    - *Crime Statistics*: The system should provide statistical data and visualizations on the types and frequency of reported crimes within a user's state or region.

    - *Filter and Search*: Users should be able to filter and search for specific types of crimes, locations, or time periods to access relevant crime data.

4. Notifications and Alerts:

    - *Real-time Alerts*: Users should receive real-time notifications and alerts about crimes reported in their subscribed areas.

    - *Email Notifications:* Users should receive email notifications for important updates, new discussions, and messages.

    - *Notification Settings*: Users should be able to customize their notification preferences, including frequency and types of alerts

5. Typical Functioning

    - *Responsive Design*: The platform should have a responsive design and be usable on a variety of gadgets, such as desktops, laptops, tablets, and smartphones.

    - *Help and Support*: For help utilizing the platform, users should have access to a FAQ section and a support center.

    - *User Authentication*: To safeguard user accounts and data, the system should use secure user authentication techniques.

    - *Data Privacy*: The platform needs to respect data privacy laws and safeguard user data.

3.3 Behaviour Requirements

3.3.1 Use Case View

For creating crime reporting website, we identify the primary actors and their interactions with the system:

Description of Use Cases and Actors:

- *Register User* (Actor: Guest): This use case allows a guest user to register for an account, providing necessary information to create a new user profile.
- *Login* (Actor: User): Registered users can log in using their credentials to access the system.
- *Reset password* (Actor:User): Users who forget their login information can send an email asking to have their passwords reset.
- *Subscribe to Crime Updates* (Actor: User):Users can sign up to receive notifications and real-time crime updates for particular locations or interest areas. This use case makes sure that consumers are aware of their surroundings.
- *View Crime Statistics* (Actor: User): Users can access statistical data and visualizations depicting the types and frequency of reported crimes within their state or region. This use case provides insights into crime trends.

- *Administer Platform* (Actor: Administrator): Administrators have access to an admin dashboard through this use case. They can manage user accounts, monitor content, and oversee discussions, ensuring the platform's smooth operation.
- *Deactivate Account* (Actor: User): Users have the option to deactivate or delete their accounts through this use case. It involves appropriate confirmation steps to ensure the user's intention is validated before account deactivation.
- *Comment on Incident* (Actor: User): Users can take part in conversations by leaving comments on other people's incident reports. User participation and the exchange of ideas and insights are made possible by this use case.

These use cases capture the essential interactions between users and the "SecureSpot" platform, providing a comprehensive view of the system's functionality and user interactions.

4 Other Non-functional Requirements

4.1 Performance Requirements

Performance requirements are crucial for ensuring that the issue tracking and project management tool functions optimally under varying circumstances. Here are five distinct performance requirements based on client input and system needs:

- *Incident Reporting Response Tim*e: Performance Requirement: The system should respond to incident report submissions within 5 seconds of the user's request. This requirement ensures that users experience quick and efficient incident reporting, promoting active participation and timely data collection.
- *User Authentication Speed*: During the login procedure, user authentication shouldn't take more than three seconds. Swift user authentication guarantees a smooth and convenient login process.
- *Real-time Notification Delivery*: Within two seconds of an incident being reported, subscribers should receive real-time crime notifications and updates.For users to keep updated about situations in their chosen regions and improve their situational awareness, timely notification delivery is essential.
- *Data Retrieval Efficiency*: Data retrieval for crime statistics and incident reports should not exceed 3 seconds, even when handling a large volume of data.
- *Image Upload Processing Time*: The system should process and display uploaded photos and documents related to incident reports in under five seconds. Rapid image processing improves the accuracy and usefulness of incident reports by making visual evidence easily available.

4.2 Safety and Security Requirements

Safety prerequisites

- *User Data Privacy*: "SecureSpot" is required to abide by data privacy laws, such as GDPR, to protect user data and get consent for data processing. Encryption should be used for both storage and transport of user data. Users should be able to manage their data and delete their accounts and the related data with them.

- *Safety and content moderation*: "SecureSpot" needs to implement effective content moderation with systems in place to stop and deal with hate speech, harassment, and harmful content in order to guarantee a secure and courteous online community. Moderators should quickly assess and respond to reported content. Users ought to be able to mark objectionable information for review.

- *Safety Requiremen*t: "SecureSpot" shall implement user identity verification procedures for users who desire to participate in discussions or report occurrences in order to maintain a trustworthy platform.Users can confirm their identity by phone number or email. This lessens the chance of platform abuse.

- *Data Backup and Recovery*: Regular automated data backups with redundancy must be in place to ensure data integrity and availability in case of system failures or data corruption.
- *Third-party Security*: Third-party integrations and plugins must adhere to security standards and undergo security assessments to prevent vulnerabilities.

4.3 Software Quality Attributes

4.3.1 Usability

- *User-Friendly Interface Is A Must*: The user-friendly interface of "SecureSpot" will have simple navigation, useful features, and a responsive design. To make sure that users can carry out typical actions quickly and clearly, usability testing will be carried out with a sample of users.

- *Accessibility*: According to the description, "SecureSpot" must enable screen readers, keyboard navigation, and alt text for images in order to be usable by people with impairments. To ensure adherence to accessibility requirements, accessibility testing will be carried out using assistive technology.

4.3.2 Reliability

- *System Uptime*: With a minimum uptime of 99.5%, "SecureSpot" guarantees that users may access the platform with minimal interruptions. This reliability metric will be measured and maintained using system monitoring and uptime tracking.

- *Data Integrity is a second requirement*: Through routine backups and data validation checks, "SecureSpot" will ensure the integrity of user data, preventing data loss or corruption.To ensure the dependability of the data, regular data integrity checks and backup and recovery tests will be carried out.

4.3.3 Maintainability

- The software shall be developed following modular and well-documented code practices. Code comments and documentation should be maintained to facilitate future development and troubleshooting.
- Changes and updates to the software should be achieved with minimal impact on existing functionality. The design should allow for easy modifications and extensions, following the principle of "design for change”.

Appendix A – Data Dictionary

Item	Description	Operations	Requirements
User	Represents a registered user of the system	Create, Read, Update, Delete (CRUD) operations, Authentication	Must provide valid login credentials
Project	A project within the system	Create, Read, Update, Delete (CRUD) operations	Project name must be unique
Task	Represents a task within a project	Create, Read, Update, Delete (CRUD) operations	Task must have a unique identifier
Issue	An issue submitted by a user	Create, Read, Update, Delete (CRUD) operations	Issues must have a priority level

Audit Log	Records of a system actions and events	Read operations to view logs	Logs must be timestamped and secure
User Interface (UI)	The graphical interface of the application	User interaction, display, and navigation operations	UI should be intuitive and responsive.
Database	Stores data for the application	CRUD operations, data retrieval, and indexing	Database must be backup up regularly
API	Enables external integrations	Data exchange, authentication, and data manipulation	API should be secure and well-documented

The data dictionary serves as a valuable reference for the development team to ensure that all elements and their properties are well-defined and meet the specified requirements.