DHIRUBHAI AMBANI INSTITUTE OF INFORMATION
AND COMMUNICATION TECHNOLOGY

IT314 - SOFTWARE ENGINEERING

SOFTWARE REQUIREMENT SPECIFICATION REPORT

CRIME AND HAZARD MANAGEMENT SYSTEM

GROUP NO: 30

GROUP MEMBERS

202001410	PATEL AYUSH SANJAYBHAI
202001447	VAKANI HETAV ABHAYBHAI
202001467	JAY GROVER
202001421	PATEL KUNJ RAKESH
202001446	GONDALIYA VENIL CHANDUBHAI
202001275	AYUSH JAIN
202001458	KRIS PATEL
202001264	HARSH SANJAY MAKWANA
202001466	KALP KINJALBHAI PANDYA
202001438	NARODIA JEET NILESHKUMAR
202001457	HITARTH VYAS

Table of Contents

- **Introduction**
  - **Purpose**
    - This document outlines the software requirements for a website designed to assist home buyers and renters in a city to discover a property's crime and hazard profiles. The website provides users with quick and easy access to Crime Safety and Hazard Safety percentile scores for a given address, allowing them to compare it to other properties in the city. Users can choose to continue their search for other properties or dive deeper into the statistics provided on other sections of the website. The goal of this website is to provide users with valuable information to help them make informed decisions when searching for a new home.
  - **Intended Audience**
    - The intended audience for this website are homebuyers and renters in a city who are interested in discovering a property's crime and hazard profiles. This includes individuals and families who are looking to purchase or rent a new home and want to make informed decisions based on the safety and hazard risks associated with a particular property. The website is designed to be user-friendly and accessible to a wide range of users, regardless of their technical expertise or familiarity with real estate terminology
  - **Product Scope**
    - The scope of this website is to provide home buyers and renters in a city with valuable information about a property's crime and hazard profiles. The website achieves this by presenting users with Crime Safety and Hazard Safety percentile scores for a given address, allowing them to compare it to other properties in the city. Users can choose to continue their search for other properties or dive deeper into the statistics provided on other sections of the website. The website is designed to be user-friendly and accessible, providing users with the information they need to make informed decisions when searching for a new home.

- **Overall Description**
    - **Product Functions**
        - **Register:** The register function of a website allows new users to create an account and access personalized content and features. When a user visits the register page, they are prompted to enter their personal information, such as their name and email address, and to choose a username and password. The website then verifies this information and creates a new account for the user.
        - **Login:** The login function of a website allows registered users to securely access their accounts and personalized content. When a user visits the login page, they are prompted to enter their username and password. The website then verifies this information against its database of registered users. If the login credentials are correct, the user is granted access to their account and any personalized content or features associated with it
        - **Search:** The search function of a website that allows users to search through posts provides a convenient way for users to find and access specific content. When a user enters a search query, the website searches its database of posts for content that matches the query. The search results are then displayed to the user, allowing them to easily find and access the posts they are looking for. The website may use algorithms to rank search results based on relevance, so that the most relevant posts are displayed at the top of the results.
        - **Add Post:** The add post function of a website allows users to create and share new content with others. When a user wants to add a new post, they are typically prompted to enter the content of their post, such as text, images, or videos. The website then verifies this information and publishes the new post, making it visible to other users.
        - **Upvote/Downvote facility -** This function allows users to express their approval or disapproval of a particular post. Users can use the upvote button to show that they like a post, and the downvote button to show that they dislike it.
        - **Map Based UI:** This allows users to to browse through the world map and provide precise location information related to various posts.

- **Tools and Technologies used**
    - **Programming/ Scripting Languages used:** Python, HTML5/CSS, JS
    - **Frameworks used:** Django, pymongo, bootstrap
    - **Database:** MongoDB
    - **Tools and other software used:** VS Code, VS Code extensions, GIT/Github, Google Chrome, MapBox API.
    - **Python:** It is a popular language for backend server-side applications due to its versatility and capabilities. It has applications in back-end web development, machine learning and statistics, scientific modeling, system operations, and several enterprise-specific software. Python's English-like syntax, dynamic typing, and many libraries make it an approachable programming language. Python has several frameworks like Django and Flask that can be used to create dynamic server-side web applications. These frameworks allow for fast development of maintainable and secure web applications.
    - **HTML/CSS:** HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are essential tools for creating frontend web applications. HTML provides the structure and content of a web page, while CSS is used to style and layout the page. Together, they allow developers to create visually appealing and user-friendly websites. HTML and CSS are versatile and can be used to create a wide range of web applications, from simple static pages to complex interactive sites. Their capabilities are constantly evolving, with new features being added to improve the user experience
    - **Javascript:** JavaScript is a versatile and powerful programming language that is widely used for frontend web development. It allows developers to add interactivity and dynamic behavior to web pages, enhancing the user experience. JavaScript can be used to create a wide range of web applications, from simple animations to complex single-page applications. Its capabilities are constantly expanding, with new libraries and frameworks being developed to make it easier to create rich and engaging web experiences.
    - **Bootstrap:** Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile-first projects on the web. It makes front-end web development faster and easier and is designed for folks of all skill levels, devices of all shapes, and projects of all sizes.

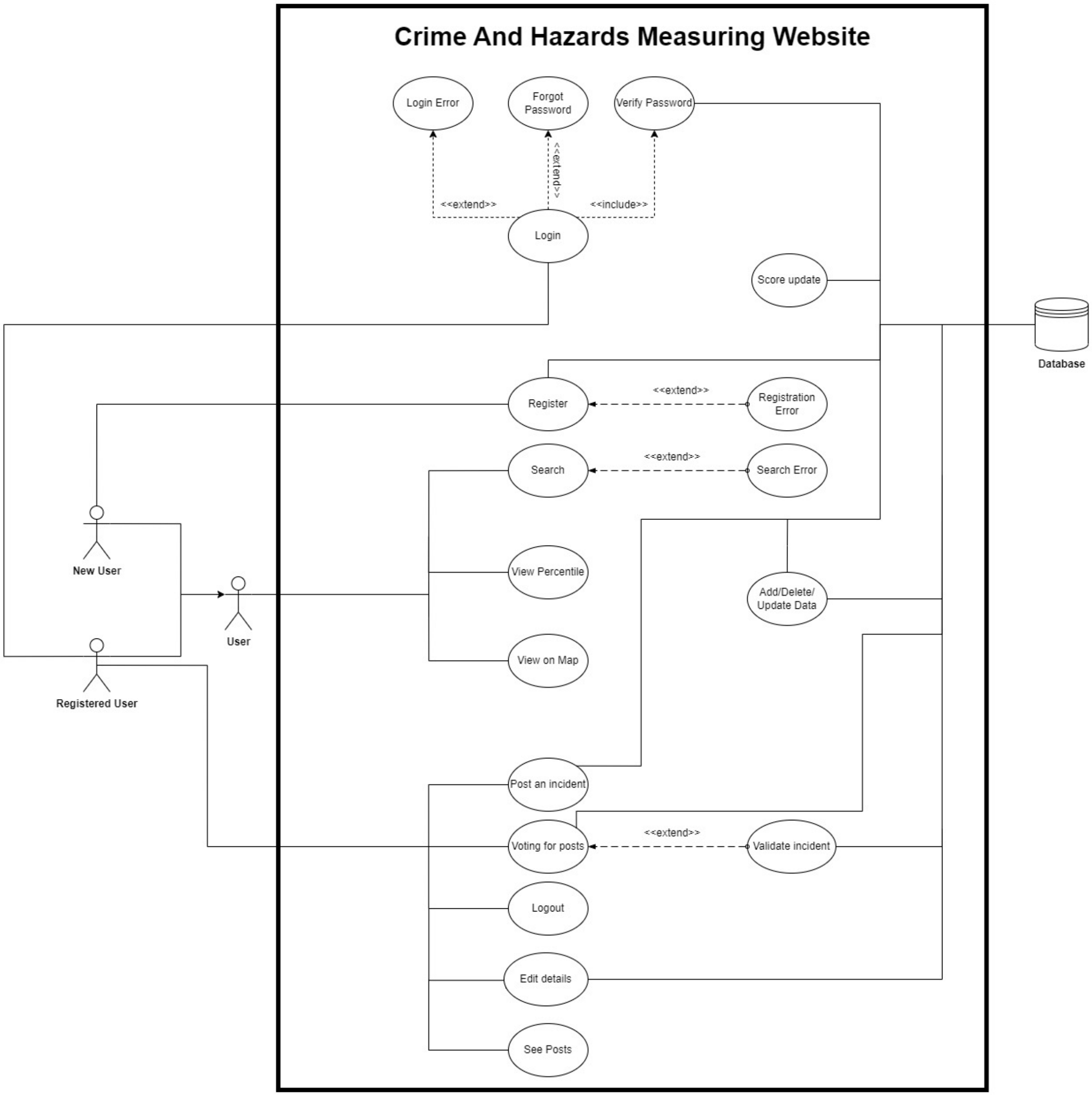
- **PyMongo:** PyMongo is the official MongoDB driver for synchronous Python applications. It provides tools for interacting with MongoDB databases and allows developers to easily connect and use MongoDB from their Python applications[1]. PyMongo offers an intuitive API for accessing databases, collections, and documents. Objects retrieved from MongoDB through PyMongo are compatible with dictionaries and lists, making it easy to manipulate, iterate, and print them
- **Django:** Django is a high-level Python web framework that helps developers create backend infrastructure for web applications. It follows the model-template-view (MTV) architectural pattern and provides a set of tools and abstractions for building scalable and maintainable web applications. Django takes care of many common tasks such as handling user authentication, URL routing, and database schema migrations, allowing developers to focus on writing their application logic. Its modular design and reusable components make it easy to build and maintain complex web applications.
- **MongoDB:** MongoDB is a popular NoSQL database that can be used to create database infrastructure for web applications. It stores data in flexible, JSON-like documents, allowing developers to easily model complex data structures. MongoDB’s dynamic schema makes it easy to adapt to changing requirements and its powerful query language allows developers to easily retrieve and manipulate data. MongoDB also provides features such as horizontal scaling and automatic failover, making it a reliable and scalable choice for web applications.
- **VS Code:** Visual Studio Code (VS Code) is a popular code editor that is widely used for software development. It supports a wide range of programming languages and has a rich ecosystem of extensions that add additional functionality. VS Code’s versatility makes it suitable for a wide range of development tasks, from writing and debugging code to managing source control and deploying applications. Its capabilities include intelligent code completion, syntax highlighting, and integrated debugging tools, making it a powerful tool for developers
- **GIT/Github:** Git is a widely used version control system that helps developers manage and track changes to their code. GitHub is a web-based platform that provides hosting for Git repositories and

additional features such as issue tracking and code review. Together, Git and GitHub provide a powerful set of tools for software development. They allow developers to collaborate on code, track changes, and easily revert to previous versions if needed. Their versatility makes them suitable for a wide range of development workflows, from individual projects to large-scale collaborative efforts. Their capabilities include branching and merging, conflict resolution, and access control, making them essential tools for modern software development

- **Google Chrome:** Google Chrome is a popular web browser that is widely used for web development. It provides a range of tools and features that help developers build and test web applications. Chrome's versatility makes it suitable for a wide range of development tasks, from testing and debugging code to profiling and optimizing performance. Its capabilities include a powerful set of developer tools that provide features such as live editing of HTML and CSS, JavaScript debugging, and network analysis.
- **MapBox - API:** The Mapbox API provides a range of tools and services that can be used to obtain location data for a website. Mapbox's Search API allows developers to perform geocoding and reverse geocoding, converting addresses into coordinates and vice versa[1]. This can be useful for adding location-based features to a website, such as displaying a map or providing directions. Mapbox also provides SDKs and libraries that make it easy to integrate their services into web applications. With its accurate location data and powerful developer tools, Mapbox is a popular choice for adding location-based functionality to websites

- **Functional Requirements**
    - **View Property Score :** Both users who are logged in and those who have not can see the scores of posted properties around the world on a map. Depending on the score, the location will be displayed in a distinct color to enhance the visualization. The higher the score, the better the quality of the property.
    - **Registration:** The website allows for new users to sign up and their details will be saved in the database. Afterwards, the user will be able to access their profile, view their posts, publish content such as incidents or properties, and also vote on other posts.
    - **Log in :** A registered user can sign in on the website and perform the activities mentioned above in registration functionality.
    - **Incident Feed:** Both logged in and non-logged in users have access to an incident feed that contains posts made by other users regarding crimes and hazards.These posts include details such as the title, description, author, and the time of the incident. If the user clicks on the author's name, they can view the author's profile. Furthermore, signed-in users are able to see the number of votes on each post and are able to vote themselves. Once a post reaches a certain threshold of votes, it is verified.
    - **Property Feed:** The property feed, which consists of posts from users about properties, is accessible to both users who have logged in and those who have not. The posts contain information such as the title, description, author,posting time and score of the property. If a user clicks on the author's name, they can see the author's profile.
    - **Search Incident/Property:** The user can search incident/property based on the keyword present in the post like title,author,address etc.
    - **Post an incident:** If a user who has logged in wants to report an incident, they can provide information about it. The property scores will then be adjusted based on the incident details, and the incident report will be visible in the incident feed.
    - **Post an property:** A user who has logged in has the ability to create a new property, which will appear on the map alongside a calculated score, and also be visible in the property feed.

- **Use Case Diagram**



- **Use Case Description**

Identifier	● UC_1
Use Case	● Login
Description	● User will Login into the CHMW*
Actors	● User, Database system
Pre-conditions	● User should be registered
Post-conditions	● CHMW* Display Homepage
Flow	● Users enter the website. ● System show the main page ● User select the Login option ● System requests that the user enter username and password. ● Users enter the username and password. ● The system validates the entered username and password and logs the user into the system.
Exception	● User enter forget password option ● If User enters an invalid username and/or password, the system displays an error message. The actor can choose to either return to the beginning of the Basic Flow or cancel the login.

- CHMW* - Crime and Hazards Measuring Website

Identifier	● UC_2
Use Case	● Register
Description	● Registers new user for system through which one can login
Actors	● New User, Database
Pre-conditions	● Users must be connected to the internet. ● User should be new, existing user can't re-register
Post-conditions	● Users will have login credentials to login in the system. ● New registered user credentials will be saved in the database.

Flow	● User come to home page of system ● Clicks Register ● System asks some user details and ask to create its credentials ● After this, user will have its login credentials
Exception	● In step 2,if user is already registered with an User Name and tries to register again with the same email id then the system displays ○ an error. ● In step 3,if user details are not stored in the database due to some internal issues.

Identifier	● UC_6
Use Case	● View on map/graph
Description	● Any user can see crime and hazards in property location and the neighborhood.
Actors	● Homebuyer/renter, Location Sharing System, Database System
Preconditions	● The homebuyer/renter has successfully logged in to the website. ● The homebuyer/renter has entered valid criteria for property of interest.
Postconditions	● The homebuyer/renter can view property location and surrounding areas on map and make an informed decision based on the data.
Flow	● System retrieves property location information from the database. ● The system displays property location on map. ● The system also shows crime and hazard data in respective areas. ● Users can zoom in or out.
Alternate Flow	● If the property address information is unavailable or invalid, the system displays an error message and prompts the buyer to find another property.
Exceptions	● Your system may be loading slowly or your map may be rendering incorrectly. In such cases, the system should display an error message and provide an alternative option to display property information.

Identifier	● UC_8
Use Case	● Voting for a post
Description	● The user will be able to upvote or downvote a post
Actors	● User
Pre-conditions	● Users should have a valid account. ● Users must be logged in.
Post-conditions	● The web page should display the total count of votes. ● The upvote count of the corresponding post should be updated in the database.
Flow	● Log in to the account. ● Click on the upvote or downvote button.

Identifier	● UC_9
Use Case	● Post an incident
Description	● The user will be able to post information about any criminal or hazardous event in the neighborhood.
Actors	● User,Database
Pre-conditions	● User should be logged into his account. ● A short description about the event should be written. ● Time and location where the incident occurred should be provided and details should be valid (Ex - time should not be in future or location should exist in real world).
Post-conditions	● Other users should be able to read and vote on the incident(UC_8). ● Incident should be validated once the vote threshold is crossed.
Flow	● User logs into account if not logged in. ● User clicks on post incident functionality and provides all the relevant information. ● User clicks on the submit/ post button. ● The system checks if all constraints(description, location, time etc details are provided) are met and posts the incident. ●

Exception	● Authorization of User fails. ○ Error message pops up. ○ User redirected to login page ● Insufficient or Incorrect Information is entered. ○ Warning is given and the user is requested to correct the mistake.

- **Nonfunctional Requirements**
    - **Performance:**
        - In today's world of short time-spans and high speed internet connection a user may find it difficult to access and work with a slow website hence the latency and loading time for the site should be as low as possible for all its available functionalities.
    - **Scalability:**
        - As time passes, the population of any place would increase, so in order to accommodate the new users the system should have the necessary hardware support.
    - **Usability:**
        - The system must have an easy-to-use interface, be intuitive for users, and be as simple as possible to utilize.
        - If a non technical user finds the interface hard to use, he/she may leave the website.
    - **Reliability:**
        - System must be available and it must be dependable having little downtime and a low rate of mistakes and malfunctions.
        - All the updates displayed on the web page should be updated on the server database.
    - **Compliance:**
        - The system must adhere to all applicable legal and regulatory requirements, including security guidelines and data privacy regulations.
        - As the users will be posting and providing possible private information about their neighborhoods or their personal identification information for authentication purposes the storage and handling of data should be done with utmost care, following rules and regulations to avoid any legal liabilities.
    - **Maintainable:**
        - Rolling out new updates should be easier for the developers to do.
        - Minor changes should not affect the system adversely.

- **Incident Validation:**
    - Incidents posted should be verified as soon as possible so users can judge the reliability of information provided on the website.
    - This is required to prevent people from spreading false information about certain communities or neighborhoods.
- **Responsiveness:**
    - The website can be operated efficiently using Desktop as well as mobile phone.

- **Future Scope**
    - The initial offering of Crime Safety and Hazard Safety percentile scores provides users with a quick and easy way to compare properties within a city. This feature alone could attract a significant user base, as safety is a major concern for many people when choosing a place to live.
    - The website includes interactive maps that display crime and hazard data for specific neighborhoods or streets. Additionally Users could also be given the option to view historical data, allowing them to see how crime and hazard levels have changed over time.
    - Another potential feature is the ability for users to customize their search criteria based on their specific safety concerns. For example, a user might be particularly concerned about the risk of flooding or earthquakes in the area where they are considering buying or renting a property. The website could allow users to specify these concerns and provide them with detailed information about the risks associated with each property.
    - The website could also incorporate user reviews and ratings, allowing users to share their experiences and insights with others. This could help build a community of users who can support each other in their search for safe and secure housing.
    - Another potential area for growth is the integration of additional data sources. The website could incorporate information from local government agencies, such as police departments and fire departments, to provide users with even more accurate and up-to-date information about crime and hazard levels. The website could also partner with real estate companies and property management firms to provide users with additional information about the properties they are considering.
    - With the right features and partnerships, such a website could become an essential tool for anyone looking to buy or rent a home in a city.