CS5310 Software Requirements and Specifications

Software Requirements Specification (SRS) Template

Items that are intended to stay in as of your document are in **bold**; explanatory comments are in *italic* text. Plain text is used where you might insert wording about your project.

The document in this file is an annotated outline for specifying software requirements, adapted from the IEEE Guide to Software Requirements Specifications (Std 830-1998).

Referencing, IEEE Std 830-1998, tailor this to your needs, removing explanatory comments as you go along. *Where you decide to omit a section, keep the header, but insert a comment saying why you omit the data*.

(Neighborhood Watch)
(Neighborhood Watch Developing Team (#2))

(Daniel Botdorf, Emily Bardwell, Jacob Pace, Kayode Gbogi-Emmanuel, Samuel Rossilli)

Software Requirements Specification

Document

(Page Intentionally left blank)

Table of Contents

Revision History 5

1 Introduction 6

1.1 Purpose 6

1.2 Scope 6

1.2.1 Software Version Matrix 6

1.2.2 New Software Capabilities 6

1.3 Definitions, acronyms, and abbreviations 7

1.4 References 7

1.5 Overview 7

2 Overall descriptions 8

2.1 Product perspective 8

2.1.1 System interfaces 8

2.1.2 User interfaces 8

2.1.3 Hardware interfaces 10

2.1.4 Software interfaces 10

2.1.5 Communications interfaces 11

2.1.6 Memory constraints 11

2.1.7 Operations 11

2.1.8 Site adaptation requirements 12

2.2 Product functions 12

2.3 User characteristics 13

2.4 Constraints 13

2.5 Assumptions and dependencies 13

2.6 Apportioning of requirements 14

3 Specific requirements 14

3.1 General Overview 14

3.2 Functions 14

3.3 Performance requirements 29

3.4 Logical database requirements 30

3.5 Design constraints 31

3.6 Software system attributes 31

3.6.1 Reliability 32

3.6.2 Availability 32
3.6.3 Security 32
3.6.4 Maintainability 32
3.6.5 Portability 33
The application should be portable with iOS and Android. 33
3.7 Organizing the specific requirements 33
3.7.1 System mode 33
3.7.2 User class 33
3.7.3 Objects 33
3.7.4 Feature 34
3.7.5 Response 34
3.7.6 Functional hierarchy 35
4 Diagrams 35

Revision History

Name	Date	Reason For Changes	Version
William M. Emmel		Initial Version	1.0

1 Introduction

The Neighborhood Watch Application provides the Homeowners Assocation (HOA), homeowners, and future users with the ability to know what is happening in their neighborhood. The application will offer an interactive format that is inclusive without becoming burdensome, easy to use, and encourages the homeowners to participate in the community.

1.1 Purpose

The Software Requirements Specification (SRS) exists to collect and present the varied ideas that have been presented and will subsequently represent the application as the customer has defined their expectation.

The SRS shall encapsulate the manner in which the application will be used and will serve as a source of truth throughout the development process.

Additionally, the SRS will provide insights for maintenance and be useful for future developers should the application require expanded functionality in the future. The SRS is inclusive of the current customer requests; however, it does not serve as an exhaustive representation of all possibilities. Portions of this document may be disregarded through the creation of the application as deemed appropriate by the customer and development teams respectively.

The SRS defines the application as a deliverable product, including an overall description, Specific Requirements, and Supporting Documentation. As an artifact, the SRS addresses the needs of the user, customer, and future developers throughout the product lifecycle.

1.2 Scope

1.2.1 Software Version Matrix

The software produced will assist in the function of a mobile application. Such software may include:

Software	Version
RHEL 8.5	4.18.0-348
SQLite	3.38.0
Python	3.9

1.2.2 New Software Capabilities

This application will allow the user to see reported violations, document violations, and schedule neighborhood watch taskings. The app will not have a dialing function that works similar to 911**,** but may pull up 911 on the users phone if the user enters a high-level incident report.

1.3 Definitions, acronyms, and abbreviations

Variables Table

Name	Description
<ADMIN_USERID>	The UserID of the individual admin.
<ADMIN_PW>	Admin Password. It must be unique for each admin user.
<USER_USERID>	User Password. Must be unique for each user.
<USER_PASSWORD>	User Password. Must be unique for each user.
<HOA>	Home owners' association
<GPS>	Global Positioning System
<USER>	Someone who interacts with the mobile phone application
<MODERATOR>	A user who is given specific permission for managing and controlling the system

1.4 References

IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications* The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY 10017-2394, USA, ISBN 0-7381-0332-2.

1.5 Overview

The current section of the SRS (Section 1) includes a general layout of the document and how to utilize the contents. It is followed by three additional sections as well as an appendix.

Section Two provides insights into the applications interfaces, constraints, and user characteristics necessary to meet Stakeholder requests. This section will also detail technical and protocol requirements alongside foreseeable dependencies as appropriate to Stakeholders.

Section Three outlines the Requirements Specification as a system interface. The Specification Requirements are presented in a manner that is appropriate for the individual Stakeholder and should be read from the perspective lens of the individual user. Only as an aggregate of users does the Requirements Specification define the application as a functional product.

The Appendix provides further clarification referenced in prior sections of the SRS as well as diagrams that may prove useful in understanding the application architecture.

2 Overall descriptions

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product perspective

The product is a communication platform among the neighborhood watch and other members of a community. Through the application, neighborhood members will be able to view the status of the members of their neighborhood watch. The application provides an easy tool for the neighborhood watch and other members of the neighborhood to report incidents. The reports will immediately alert the community. The application will advise/remind whoever is making a report to immediately call the police if it is a dangerous situation.
The product will document recent incident reports on a map that encompass the entire neighborhood. It will also feature trend reports about the occurrence of criminal activities and their locations.

2.1.1 System interfaces

This section shall provide a short summary of interfaces of the system. Following sections shall procide a deeper overview of the system interfaces.
The system shall have the following interfaces:

1. User interface
    a. Phone application
    b. Web interface or application
2. Communication protocalls
    a. DNS
    b. DHCS
3. Software interfaces
    a. RHEL 8.5
    b. SQLite
    c. Python
4. Communications interfaces
    a. IEEE802.11 Wi-Fi protocall
    b. 3G protocall
    c. Java Database Connectivity 4.3
    d. TCP/IP protocall

2.1.2 User interfaces

Users will be able to access the platform through an application downloaded on their mobile devices. The user interface should be extended to and accessible from browsers such as Microsoft Edge, Google Chrome, DuckDuckGo, etc.

The graphical user interfaces will be implemented using Python 3.9 software packages.

The user interface will utilize the following elements:
Input controls: checkboxes, buttons, text fields, date field, list boxes, dropdown lists
Navigational components: search fields, tags, filters
Information components: icons, progress bars, notifications, message boxes

Navigating the platform:
Accessing the platform through the website or the application will transfer the users to a login page The login page will request the user to enter their chosen username and password. There is a button in the login page which the users can click to be delivered to a page for obtaining their forgotten passwords.

The following may be viewed upon opening the application:

1. A sign-up button in the login page will direct the users to a page where users can create their accounts
2. After signing in, the user is directed to the home page. The home page will display recent neighborhood news and information.
3. The **Home Page** features a dropdown list that lists a series of options including **reports**, **discussion boards**, **neighborhood watch calendar**, **crime statistics**, and **crime map**.
4. The **Report** feature opens a page where users can report an incident. The user must fill every part of the page to complete the report. The first is a severity rating from 1 to 5. The second is a dropdown list that contains possible incident types. The third is a location which can be described by the user or entered through a GPS. The fourth is a text field for a short description with a max length of 500 words. After the users document their incident, they can click the report button which will transfer them to a progress page confirming the action.
5. The **Discussion** feature opens a page where users can connect and communicate with their fellow neighbors. The page lists up to 10 discussions. A user can either create a discussion or comment to a discussion. The page has a button to create a discussion. By clicking the button, users are directed to a page where they fill in the title of the discussion, and their comment. Users can select a discussion where they have an option to reply to other people's comments.
6. The **Neighborhood Watch Calendar** feature directs the user to a new page with a calendar. The calendar will be in standard mm/dd/yy format. Each day on the calendar displays and will simply display the names of those of the watch members who are on duty. Neighborhood watch members may sign up for the watch by selecting a day, where 3 or fewer have signed up for the watch, and agreeing to the terms presented in the next page.
7. The **Crime Statistics** features trend graphs that should be automatically updated when reports are added to the system. It should give neighbors an idea of how safe their neighborhood is by presenting some simple statistical analysis of crime trends i.e. porch pirating is on the rise when the system receives several reports of porch pirating within a few days.
8. The **Crime Map** is a map display that will present pins where crimes have occurred given a certain radius. Users can select a pin to display a small summary of the incident.

Users can sort incidents by distance, incident type, and date from a dropdown list on the window.

The HOA requested that we alert those on neighborhood watch when there is an incident in their area, so those listed as being on watch for any particular night shall receive notifications of incidents in the HOA's specified range. They shall then take action with accordance to HAO policy.

2.1.3 Hardware interfaces

Communication protocols such as DNS (Domain name system), DHCS (Dynamic host configuration protocol), SMTP (Simple mail transfer protocol).

Supported devices include mobile devices such as:

- smartphones,
- tablets
- Laptops.

The mobile device shall have access to the internet. The mobile platform shall utilize the software as an application or a browser that can access the software and maintain the graphical user interface.

Other hardware interfaces such as servers for data security, and database management purposes will be described further in the next sections.

2.1.4 Software interfaces

The application shall utilize the following software interfaces in accordance with definitions found in the Software Version Matrix (Section 1.2.1):

Software	Version
RHEL 8.5	4.18.0-348
SQLite	3.38.0
Python	3.9

Operating System: RHEL 8.5 version 4.18.0-348

This interface will manage hardware resources and provide connectivity between hardware and software resources.

Database Service: SQLite version 3.38.0

The purpose of using this interface is to allow application data to be created, updated, read, and destroyed for data such as users, incidents, and reports. It will act as an interface between application and database services. SQLite is used for compatibility on devices running iOS. The user shall be able to access the database service fields of interest regarding their own account by logging into their account. The user interface shall be implemented on android and Apple iOS enabled devices.

Graphical User Interface: Python version 3.9

Python will be the primary language used in creating the Graphical User Interface (GUI) and, it will provide accessibility to all users by creating compatibility between all interfaces.

The fields of interest for the software interfaces include:

1. Login – user name, password, email address, and physical address.
2. Chat Room logs – username, input, date and time
3. Map – incident date, time, and location, Google API
4. Calendar – Name, date, and time of event

2.1.5 Communications interfaces

The app shall utilize the following communication interfaces, provided by the operating system on the device the app is running on:

IEEE 802.11 Wi-Fi Protocol

3G Protocol

The purpose of using this communication interface is to allow for the application to be available on a mobile device as well as a desktop device.

Java Database Connectivity - JDBC 4.3

The purpose of using this interface is to allow the application to interface and communicate with the database engine. The Java Database Connectivity is used to allow for easy use in the Android operating system.

The interface can be defined in the JDBC documentation.

TCP/IP protocol

2.1.6 Memory constraints

There are no known memory constraints as dictated by the HOA. However reports and old discussions shall be removed from the database after 60 days.

2.1.7 Operations

The application, under normal operation shall allow the user to

- access the application through an iPhone app
- access the application through Android app
- access the application through a webpage
- create incidents to add to the map
- choose data display methods (list vs map)
- sort incidents by date
- sort incidents by distance
- create discussion pages
- allow textual and imagery input in discussion pages

The system automatically shall, under normal operation

- verify user identity with a login
- check user permissions to remove incidents
- remove data from the incident database 60 days after creation

2.1.8 Site adaptation requirements

The application should initialize the user interface with:

- A view of an incident map
- A button to 'login to add incidents'
- Incidents in list view sorted by date

The application should initialize the user accounts upon creation as

- A default 'neighbor' level account without permissions to remove incidents

The application should initialize all the databases as:

- Empty, except the HOA association
    - With no reports
    - With no discussion posts
    - With no user accounts

2.2 Product functions

The application shall have the following functionality defined by users stories generated from engagement surveys, customer requests, and input provided by Stakeholders:

Log in and Home Page

A login page into the user facing application
Log in shall be through email and password
Security verification through administrator account approval
A Home Page displaying recent HOA news

The Discussion Board Page

Allow neighbors to start discussions on any issues they may have
Allow neighbors to comment on discussions
The ability for HOA to moderate discussions

The Crime Statistics page

Quick glance crime statistics what has occurred in the last week
Ability to view crime reports within a certain radius
The ability to sort crimes based on distance from user as well as time posted
The ability to look at crime trends

Crime map page

The ability to view all crime reports on a map
The ability to view a quick summary of the crime when clicking on a pin

The Report pages

The ability to create a report of a crime

The ability to view past reports
The ability to add a location and description of the crime
The ability to alert the neighborhood watch association when a crime is reported
This report should be added to the crime maps page and crime statistics page

The Neighborhood watch calendar page

The ability to view a monthly calendar of who is on duty on any give night
The ability to add different people to the calendar
The ability to take people off the calendar
The ability to sign up for the watch

2.3 User characteristics

Users of the product are of average education, meaning some are college graduates, others have some college and still others in the neighborhood have little to no college. They are not extremely experienced in the technical field and lack any IT department. This is something we will have to carefully consider during the design process and will most likely affect the database interface. They are however able to interface with smart devices such as iPhones with ease and expertise.

2.4 Constraints

In the production of the neighborhood watch scheduler, the main constraint for our team is the transmission of the reports provided by the software to the local police upon request. As this is a major reason for the creation of this software it is crucial that some kind of communication is available, however, we need to ensure that our system is not going to impede police systems in any way. In addition, this system is going to be used for personal safety. We need to ensure that the software is reliable and secure, so the user feels safe using it.

The mobile application is constrained by the system interface to the GPS locating system within the mobile phone. Since there are multiple GPS standards within the mobile device field, the performance and accuracy will likely be variable between manufacturers.

The wireless connection is also a constraint for the application. Since the application fetches data from the database over a wireless network, it is crucial that there is a connection for the application to function. The application will be constrained by the capacity of the network to fetch data. If a large data request is made to the database, the system may have large loading times.

2.5 Assumptions and dependencies

A major assumption is that communication between the local police and the neighborhood would be negotiated by the neighborhood association. A developed platform in the software which can efficiently transfer information between the neighborhood and the local police would further improve the security benefits of the software. However, this is difficult to implement because we require permission from the local police to interface with their communication apparatus without impeding their systems. As this is the case we shall allow the HOA to print reports from the database under the assumption that they shall deliver important information when requested by

the police or when they see it necessary, until the HOA furthers its negotiations with the police. In addition to this, because this system is going to be used for personal safety, we need to ensure that the software is reliable and secure, so the user feels safe using it.

2.6 Apportioning of requirements

Based on the dependencies above features connected to the police interaction shall be postponed until later in development are police notification/documentation systems.

3 Specific requirements

This section of the document shall go over the functional requirements of application.

3.1 General Overview

The user will open the application and the homepage will display HOA announcements and neighborhood news in chronological order. From the homepage, the user will be able to participate in the discussion board or select to navigate to other features provided in the application. On each page of the application there will be a menu option in the top left corner that will display the following tabs: Home, Report, Maps, Schedule, and Settings. Each of these tabs offer a unique functionality defined in Section 3.2 to meet individual Stakeholders needs.

3.2 Functions

Overview and General formatting:
A. sequence of operations. Happy path. How the user is expected to use the application and how the application should respond under normal circumstances.
B. Validity checks. These are the checks the system should go through to make sure the user has valid inputs i.e. A user must enter a valid address or select their current location or a user's input must be under 500 characters.
C. Abnormal input and system responses. This is strongly based on validity checks. How should the system respond to an invalid input. This section should also discuss issues like failure to connect to servers and what error messages a user might receive.
D. Relationships of inputs and outputs How a users actions affect the system as a whole. So if a user reports a crime that crime report should be added to a database and the neighborhood watch person should get a notification. These affects should be noted and cataloged in a logical order.

I. ***The System shall display a home screen, which shall contain announcements from the HOA***

- **A. Sequence of Operations -** The user shall view the following when opening the app after being logged in.
  1. The user shall open the application
  2. If the user is logged in:
     - **a.** They shall see a 'home page' that contains boxes of announcements
       - **i.** Announcements are a discussion post with an 'announcement' tag
       - **ii.** Announcements include a Title and a Description that are displayed

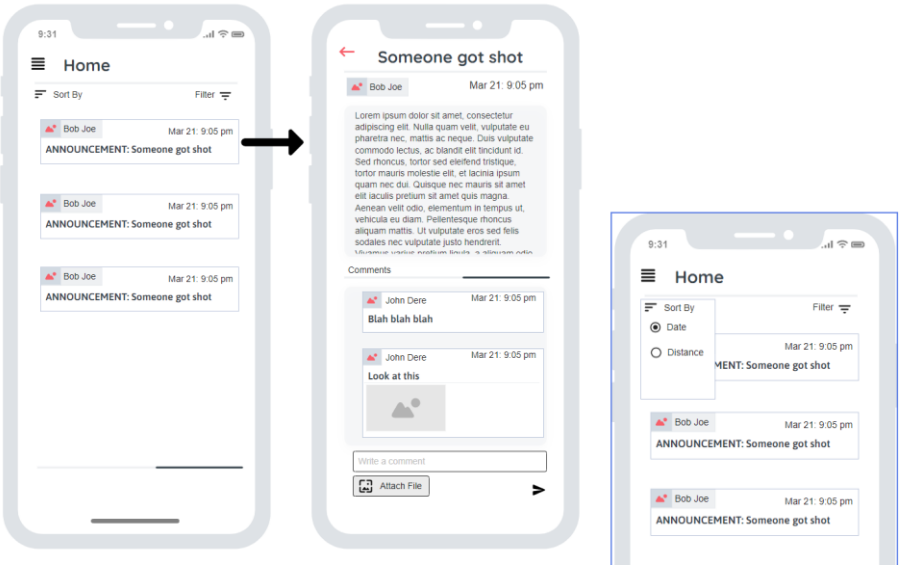
b. They shall see a filter button
  i. When selected, they shall see options to filter by activity type
  ii. They shall see options to sort by date.
c. They shall see a search box
  i. When the user enters text, matching activities with matching titles shall be displayed
d. When a user clicks a recent activity, it shall change the page to the corresponding discussion/announcement page in full detail.

3. If the user is not logged in:
a. The user shall be taken to the sign-up page

B. **Validity Checks**
1. The user shall be able to enter any text into the search box.
2. The text may not exceed 50 characters.

C. Abnormal input to the Systems
1. If the user uses special characters, no error should be thrown.
2. If the user attempts to enter more than 50 characters the box shall be highlighted and below it shall display “Maximum character length reached.”

D. Relationships of inputs and outputs.
1. When the user enters something in the search bar the system shall attempt to locate it within the database. If found the system shall return relative posts to the users screen. If not found the system shall return a null which shall be displayed to the user as “No announcements found”.
2. When clicked the system shall redirect to the corresponding announcement discussion post



Above is a view of the home page of the application. This page should display announcements made by the HOA. Users should be able to sort/filter by using a few different restrictions.

II. ***The System Shall allow a user to change their settings. These settings include adding additional permissions such as moderation, changing password and or email, and changing notification settings of crimes in the area.***

A. Sequence of Operations - the user shall have the option of modifying the application after the initial configuration.

1. The user logs into the application.
2. The user navigates to the settings page on the application.
    a. If the user is not a moderator:
        i. The user will select the frequency of notifications of incidents occurring in their area
            - All incidents, which will notify the neighbor each time an incident report is filed
            - Once per day, which will notify the user once per day of any incidents that occurred. If no incidents occurred no notification shall be sent.
        ii. The user may enter a new desired email address.
        iii. The user will be able to change their password.
3. If the user is a moderator: The user selects additional notification options from the dropdown:
    a. Changes to the neighborhood watch schedule.
    b. Changes to the list of approved moderators.
    c. Notifications of new discussions. This can be set to: all, once per day, or silent.

B. Validity Checks

1. The user will receive an email confirmation any time a change is made to their profile settings.
2. New passwords shall have the same constraints as old passwords, that is at least 7 characters long with 1 capital and 1 special character.

C. Abnormal input to the system

1. If the email is not recognized as a real email by the system the system shall not allow the user to update the email.

D. Relationships of inputs and outputs.

1. When the user inputs a new email the system shall verify the email address is real and then send a verification email message to the given email address. When the user confirms the email address the system shall update the database.
2. The system shall ensure that the password is correct before updating it in the users profile.
3. The system shall only notify the user given the input of the users settings.
4. Outputs from the system:
    a. Verification of email message.
    b. Notifications in accordance with the user specific settings.

III. ***They System shall allow the HOA to change the roles of people directly through the database***

A. Sequence of Operations

1. The HOA shall access the database through a previously created 16 digits alphanumeric and symbol passkey and their username.
2. The HOA user shall select the members profile tab.
3. The HOA shall choose between the neighborhood watch members tab, the moderator (Standard members with special administrative privileges) tab, and the standard member tab.
4. The HOA shall query the name of the individual whose role they intend to change.
5. The HOA shall select the name of the individual.
6. The HOA shall click the role option next to the individual's name.
7. The HOA shall select their desired role for the user between a neighborhood watch member, a standard member and a moderator.
8. The HOA shall select the confirm button.

B. Validity Checks

1. Incorrect Password or username – The HOA shall receive a message on the window if they enter an incorrect password or email
2. Incorrect query – The HOA shall receive a message in the window if the name queried is not found on the database.
3. Abnormal role change – The HOA shall receive a message if they attempt to change an individual to a role which they already belong to.

C. Abnormal input to the Systems

1. The user should not be able to enter anything including files, images, etc other than text into the query search page in the members tabs

D. Relationships of inputs and outputs.

1. Inputs –
   - **a.** HOA user textual inputs such as passwords, usernames, and name queries.
2. Outputs –
   - **a.** Error messages for invalid inputs.
   - **b.** Removed user permissions notification.

IV. ***The System shall allow users to sign up for neighborhood watch on the calendar***

A. Sequence of Operations – The user shall view the following when attempting to sign up for neighborhood watch. *Note the user must be an "approved user" in order to access this feature.*

1. The user shall open the application
2. The user shall open the sidebar menu
3. The user shall select calendar
4. The user shall be presented with a calendar
   - **a.** The calendar shall be in a monthly format

b. The calendar shall have names on dates according to who signed up for watch on that date

c. On dates without watch personal a bolded message “No one on duty” shall be present

5. When the user selects any date, given the date has less than 3 names on it, the user shall be presented with a “sign-up” screen.
6. The user may then select a “sign-up” button on the “sign-up” screen to have their names added to the calendar for neighborhood watch on that day. Or they may select the “go back” button to return to the calendar view.
7. The user shall see a pop-up screen which asks “Are you sure you want to sign-up for neighborhood watch on [date]. You shall be on duty from [time start] to [time end].
8. The user may then choose to validate the option by pressing “Yes I’m sure”.
9. When the option is validated, they will be returned to the calendar view and their name shall be on the calendar for the given day.

B. Validity Checks

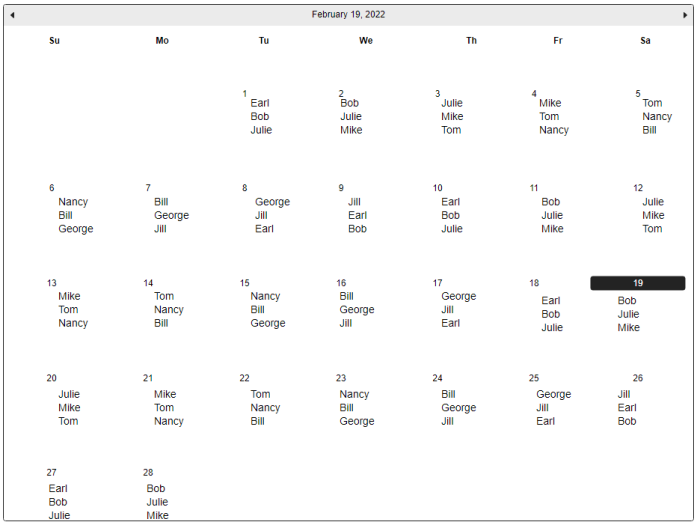
1. The system shall check if the user is an “approved user”. If they are the system shall allow them to view the calendar and all of its functions. If they aren’t the system shall not allow the users to click on a date to sign up. This functionality should only ever be given to people who are approved

C. Abnormal input to the Systems

1. If the user attempts to sign up when a given date is “full” (3 people have alright signed up for that date), the system shall present a pop-up error message informing them, “There are already 3 neighborhood watch members on duty during [date]. Please select another day.”

D. **Relationships of inputs and outputs.**

1. The calendar shall receive data input from the schedule database
    a. The system shall group entries by date and display them on the calendar on the corresponding day
2. The system should utilize a standard UI calendar for output and displaying data



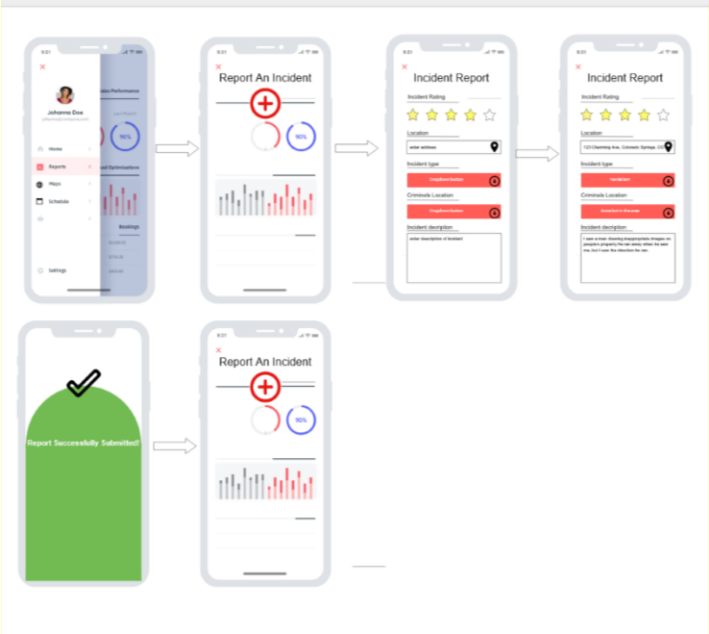
To the left is a fully booked view of the Calendar. As you can see there are 3 people allowed for each day. A person may sign up for multiple days, but may not sign up multiple times for one day.

V. *The system shall allow users to make an incident report*

A. Sequence of operations – The user shall view the following when attempting to create an incident report.

1. The user shall open the application.
2. The user shall open the side menu.
3. The user shall select Reports.
4. On the Reports page the user shall be able to see past reports if any were made by that user.
5. The user shall press the Report an Incident button located at the top of the screen of the reports page.
6. The incident report form shall pop up.
7. The user may fill out the following in any order:
   a. ***Required*** Location. This shall be the location of the incident in question. The user may select the maps button to upload their current location.
   b. ***Required*** Severity. A 5-star scale rating. The user may decide how severe they believe the incident is. 1 is low severity and 5 is high severity. ***** A 5-star severity will ask the user if they want to contact the police now if yes opens 911 on their mobile device. *****
   c. ***Required*** Crime. This shall be the type of incident to occur. It shall be a drop-down menu of various common crimes. It shall have a "other" option for non-listed crimes.
   d. ***Required*** Perpetrator Location. The user shall select one of three possible locations of the criminal from a drop down menu. The possible locations are: "In Progress", "Gone, but in the area", and "Gone, Location Unknown"
   e. ***Optional*** Description. This shall be a more formal description of the incident. The user may add detail to the incident to give their neighbors more insight, but it shall be an optional field if they feel they don't have time or information.
8. The user may then press the Submit Incident button.
9. The system shall automatically add the date and time to report based on users system.
10. The user shall be taken to a screen which says report successfully submitted.

11. The user shall be returned to the Reports screen.



Above is a view of an incident report being made. This is the happy path where all goes well and is processed properly.

B. Validity checks – input that does not meet requirements will not be allowed to enter database as new incident report.

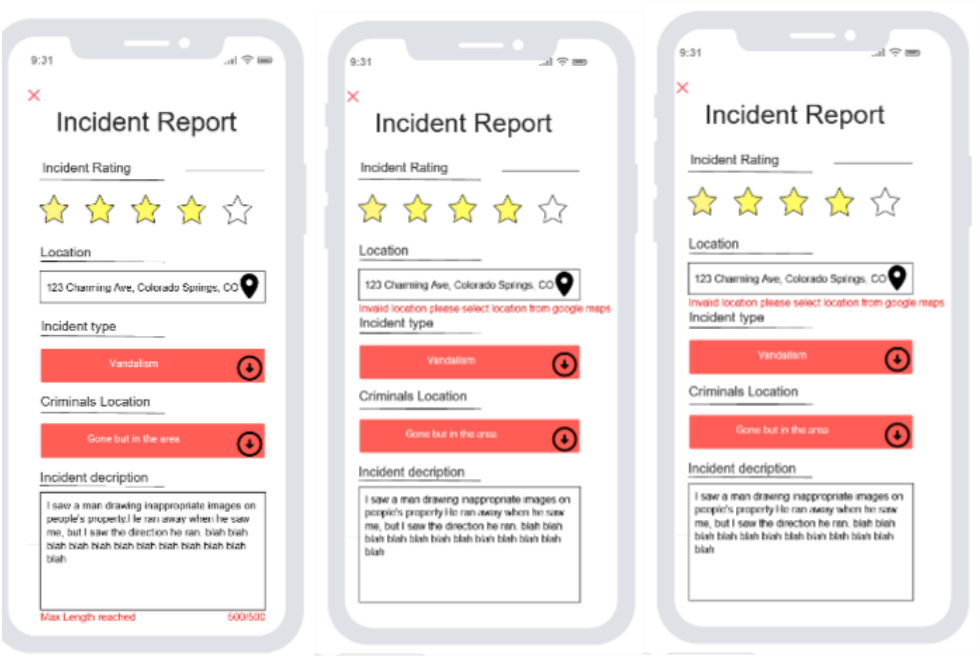
1. The user must enter a valid address or select their current location.
2. The user must select a valid crime from a drop-down list.
3. The user must select a valid Perpetrator Location from 3 given buttons; In progress, done but still in the area, and done and gone.
4. A user may add a detailed report in the description, but this may be left blank.
5. The users detailed description may not be more than 500 characters.

C. Abnormal Input and System Responses

1. Overflow

- **a.** Description Length too long - If the user's description is too long the system will inform them that they have gone over character limit by not allowing the user to enter more characters once 500 characters has been reached. The description's character count will turn red and a small note shall be added: "description length maximum reached".
- **b.** Address length too long – If the address exceeds a set character amount the user shall be asked to either select the location on google maps or select their current location. Error message: "Invalid location please select location from google maps"

2. Location Not Found – If the location is not found on a google maps lookup the user will be asked to select the location on google maps or select their current location. Error Message: "Invalid location please select location from google maps"

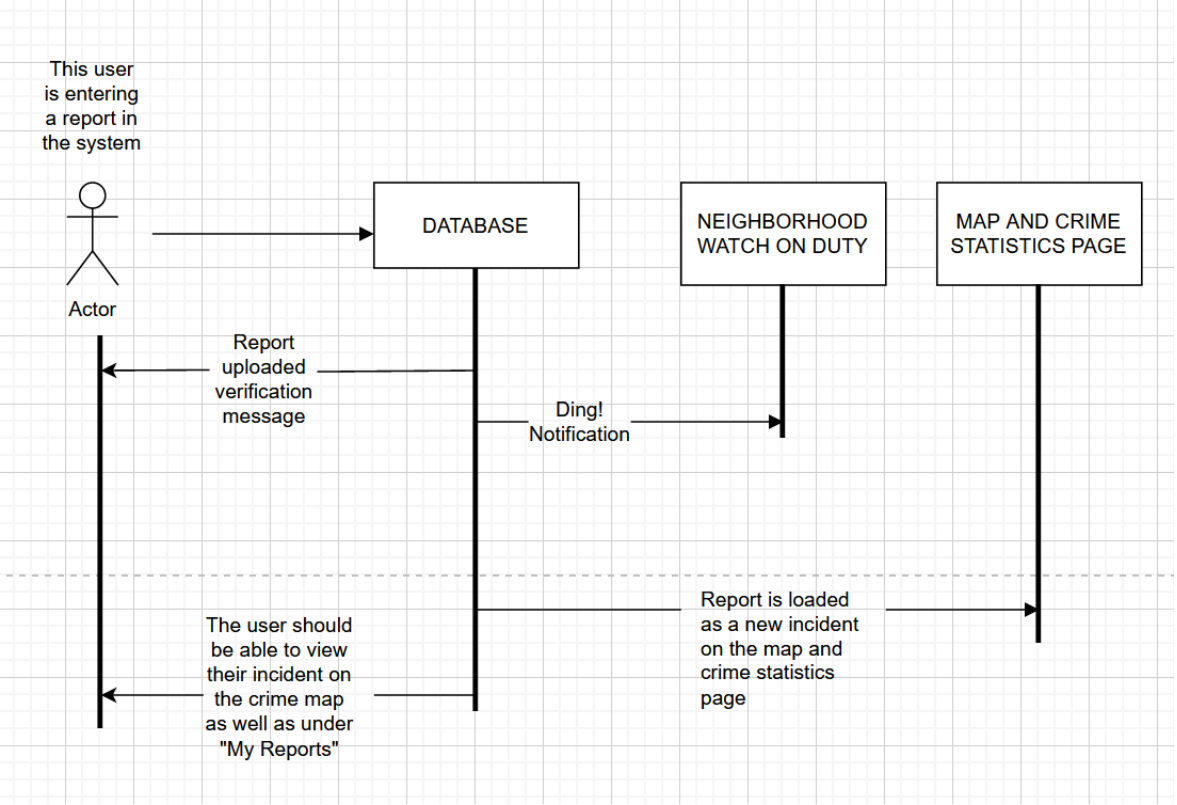


Above is a snippet of what the user might see if input is invalid. The user should always get a message with the invalid input indicating how to fix it.

3. Communication Facilities
    a. Failure to upload incident in database - If the incident report fails to send the user should receive a screen which says “incident report failed to send please try again.” As well as trouble shooting tips such as check your internet connection. It shall then return the user to their incident report.
4. Error Handling and Recovery ie. Submitting Report with Missing requirements - If a user attempts to submit a report without all of the proper fields the System shall highlight them in red and mark them as required fields without submitting the report. There shall also be a pop-up which tells them they are missing required fields.

D. Relationship of Inputs and Outputs

1. The report shall be added to the database in given format.
2. The report shall generate an incident notification for neighborhood watch.
3. The report shall generate a new mark on the map.
4. The report shall be added to the general statistics information page.
5. The report shall be generated under “My Reports”



Above is a sequence of events diagrams, which shows the order in which the system shall process a crime report.

VI. *The system shall allow users to sign up for an account*

A. Sequence of Operations – The user shall view the following when attempting to sign up for an account.

1. The user shall open the app and see a page which asks if they already have an account or if they wish to sign up for a new account.
2. Upon selecting the sign up for a new account button the user shall be taken to a new account page.
3. The page shall display the following input fields as well as a "create new account" button
   - **a.** ***Required**** A First Name input field
   - **b.** ***Required**** A last name input field
   - **c.** ***Required**** There shall be an email input field
   - **d.** ***Required**** There shall be a password input field – this shall be obfuscated with checkbox option to remove obfuscation
   - **e.** ***Required**** There shall be a password confirmation field – this shall be obfuscated with checkbox option to remove obfuscation
   - **f.** ***Optional**** There shall be a special permissions password field. – this shall be obfuscated with checkbox option to remove obfuscation

4. Upon entering the appropriate information the user shall be allowed to press the "create new account" button.
5. The user shall then be brought to the home page and given a welcome message dictated by the HOA.

B. Validity Checks

1. The system shall verify first name and last name field is filled with 2 or more characters each
2. The System shall verify the email address is valid
3. The system shall verify the password is at least 7 characters long with at least 1 capital letter and 1 special character
4. The system shall verify that the two password fields are identical
5. The system shall verify that all fields are appropriate filled out before allowing the user to select create new account

C. Abnormal input to the Systems

1. If the first name or last name field has less than two characters the system shall highlight the field and display message "Invalid input. Name too short."
2. If the first name or last name field contains numbers or special characters the system shall highlight the field and display message "Invalid input. Name may not contain special characters or numbers."
3. If the email address is not valid the system shall highlight the field in red and display a message which says invalid email address
4. If the password does not contain at least 7 letters 1 capital letter and 1 special character the system highlight the password fields in red and display a message "password must contain at least 7 letters 1 capital and 1 special character."
5. If the password fields do not match the system shall highlight the fields in red and display a message "password fields do not match. Please reenter."
6. If the user tried to push the create account button with any bad input the system shall display a message "Unable to create account. Please fix highlighted fields."
7. If the user tries to enter an invalid special permissions password upon selecting the create account button the user shall be shown a pop up message stating "The special permissions password you entered is not valid. Do you wish to continue with account creation or do you wish to reenter special permissions password? You can enter special permissions passwords under settings later on if you need to." and below the user shall be given the choice to "Go Back" or "Create Account".

D. Relationships of inputs and outputs.

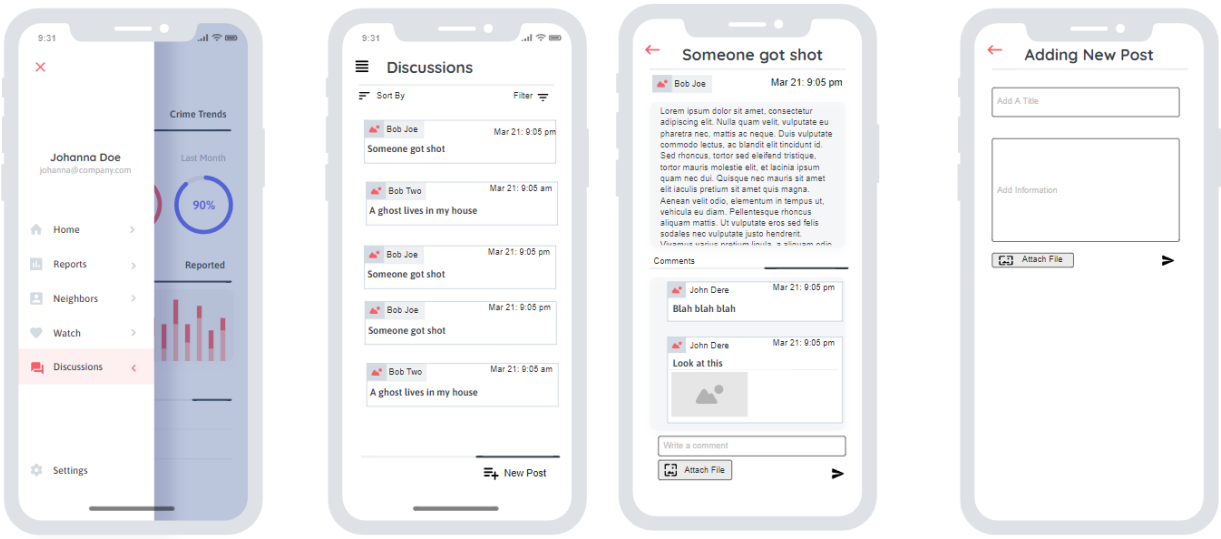
1. Upon creation of an account the user shall be added to the database of users.

VII. *The system shall allow users to communicate via discussion board*

A. Sequence of Operations

1. The user shall navigate to the Discussion Board in the navigation menu.
2. The user shall see discussion board posts dating back to at most 2 months ago.

3. The user shall be able to filter discussion board posts based on time since posted and level of activity.
4. When clicking on an existing discussion post the user shall be able to view the post as well as any comments made on the post.
    a. The user may make a comment of their own by pressing the comment button at the bottom of the screen in a discussion post.
5. Should the user click on the new discussion button the system shall take text documents pictures or other files and post them for others to view and comment on
    a. The system shall display the name of the user who sent the data to the discussion board
    b. The system shall display the date and time it was posted
    c. The system shall allow comments messages and a tag system to exist



Above is an example of the discussion board and how a user might add a new post or comment on an existing post

B. Validity checks

1. Once given a new entry the system shall verify that the content of the data is suitable for discussion board not allowing topics that are not allowed by system rules. Cussing is not allowed, so any discussions with cuss words in them shall not be allowed.

C. Abnormal input to the systems

1. When given abnormal inputs like oversized files or other unacceptable data the system shall give the user an error and not send the data to the main server.

D. Relationships of inputs and outputs

1. Outputs are the viewable backlog of inputs given by the users.
2. When the user creates a post and it has been verified the post shall be added to the database.

3. When the user creates a comment and it has been verified to be appropriate the comment shall be added to the database and posted on the appropriate discussion board post.

VIII. *The System shall notify members of the neighborhood watch of new incident reports*

A. *Sequence of operations*

1. Neighborhood watch members get notifications on or about their shift. Specifications of the notification are listed below
   - ***a.*** The notification shall contain a written description (could be blank) of the incident
   - ***b.*** The notification shall contain a user selected type of incident (home invasion, arson, assault, accidents, etc.)
   - c. The notification shall contain the severity, a rating from 1 to 5 stars of the incident
   - d. The notification shall contain the location of the incident
   - e. The notification shall contain information about perpetrator location
2. The notifications shall be delivered to a watch member's phone in "mini" mode. Mini mode is a medium sized window page in the notification bar. Mini notifications only contain the type and severity of incident, the location of incident, and information about perpetrator location.
3. The system shall implement a feature to expand the notification to a bigger window, "big" mode. The big mode notifications shall include the description written by the reporter.
4. The neighborhood watch member may then open the notification to the application by clicking on it or the neighborhood watch member shall be able to close the notification from the incident report by selecting the "X" icon on the upper right corner of the mini mode notification
5. The system shall retain the notification for the duration of a shift period if a user does not close the notification.

B. Validity Checks

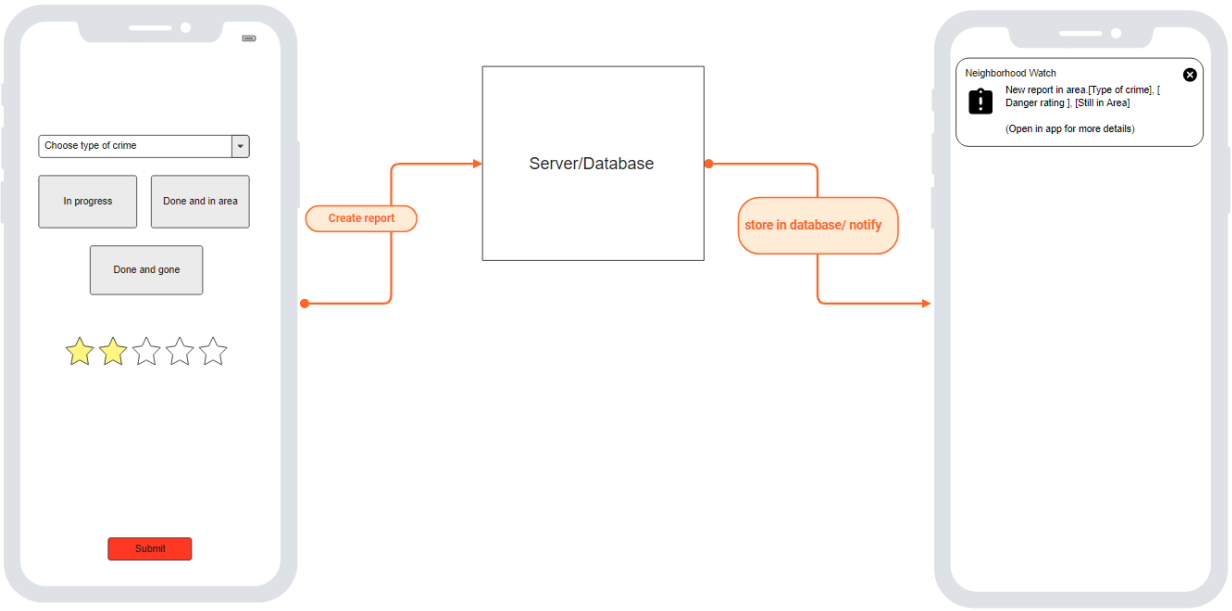
1. The system shall only send notifications to members during their "shift".
2. The system shall implement a reserved standard notification for users signed in as neighborhood watch members. All members signed in as neighborhood watch members have the same ringtone/beep to indicate a received notification status.

C. Abnormal Input and System Responses

1. The system shall prevent watch members from turning off notifications during their shift. Watch members who attempt to change their notifications through the settings option will be denied entry to the notification tab.
2. The system shall not have a feature of changing the notification style for users signed in as the neighborhood watch. Watch members who attempt to change their notifications through the settings option will be denied entry to the notification tab.

D. Relationships of inputs and outputs

1. When a user submits an incident report within the radius the HOA determines a notification shall be sent to the neighborhood watch member on duty.



Above is a simple diagram of the crime report server relationship. It should be noted that if the report is unable to make it to the server the user should be allowed to try sending it again.

IX. *The system shall allow users to view crime incidents in the neighborhood through an embedded map*

A. Sequence of operations

1. From the home page the user shall be presented with the option to see a “map view of area”
2. When this is clicked, the system shall generate a local area map and place data from incident and crime reports on that map.
3. Once generated the system shall be able to resize and zoom the map to fit user needs as well as filter out visible reports.
4. The user shall be able to click on any present map pins, which indicate crimes.
    a. The user shall then be brought to a summary page of the incident which occurred.

B. Validity Checks

1. Once the map page is opened the system shall ensure that the user is currently on an internet accessing device in order to generate the map

C. Abnormal input to the Systems

1. Data provided to this part of the system should be from data inside of our database and should not be abnormal. If an abnormality does occur an error should be sent to the user and the server should be notified of where the abnormality occurred

D. Relationships of inputs and outputs

1. Output is a visual UI map element that is a representation of incident data
2. The input of this system is the database of incidents, queried by location.

To the left is a sample of what the neighborhood map should be. The center is where the user has set their home address to. Different colored pins indicate different types of crimes. The user may click on these pins in order to view the incident report.

X. The system shall allow users to view a crime and incident statistic report of their neighborhood

A. Sequence of Operations

1. When the user enters the crime/incident statistics page, the system shall open up the crime/incident landing page. This page shall contain the following:
    a. Street comparison page, which shows the surrounding neighborhood and the safest/least safe streets inside of it.
    b. A toggle between weekly, monthly, and annual statistics to show consistency or change in the area.

c. Incident by street view page, showing the intensity and severity of accidents reported on nearby streets.

d. Graphs page, which shows trend and analysis data of relevant information.

2. The users shall be able to toggle these settings to their liking.

B. Validity Checks

1. When loading the page, the system shall ensure that the user has an internet connection in order to generate the graphs.

C. Abnormal input to the Systems

1. Inputs from the system would be handled by a different part of the application. And shall be vetted before their entry in the statistics page.

2. The user should not be able to enter anything not allowed by the system, based on the design of the system.

D. Relationships of inputs and outputs.

1. Inputs are used to formulate the output heat maps graphs and statistical data

Above is a view of the crime incident statistics page. This page is meant to give a break down of various crimes occurring within the given radius.

3.3 Performance requirements

The app speed and ease of utility (page load speeds, etc.) shall depend on the internet server of the user I.e., Faster internet services begets faster and access and usability of the app.

For users with extremely slow internet access, the application shall display an error message to use if load time exceeds 8000 ms.

Assuming a standard neighborhood home utilizes an average internet service:

- **A.** The start render time of the application shall be processed in no more than 1000ms.
- **B.** The time to interactive score of the mobile hardware shall be processed in no more than 4000ms

The app shall maintain at least the number of community members simultaneous use of the app (approximately 80 people) before the server begins to lose performance.

Textual inputs and imagery inputs shall be accepted into the app. Moderators employed by the HOA shall be given authority to censor inappropriate information input by users.

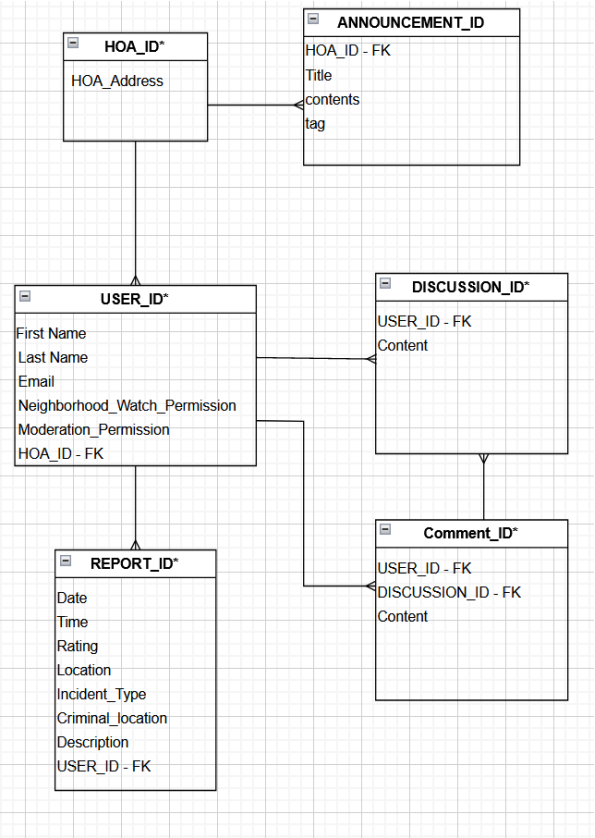
The database shall store all information exchanged between users for the duration of 2 months before it is removed from the system's servers.

3.4 Logical database requirements

The main data being placed into the database are the following:

- User information
    - Generated by the create account function
        - This shall be once per user.
        - The application shall be able to generate at least 50 user accounts per second.
        - Accounts shall be retained forever, even in the case of inactive accounts
            - Unless required to be removed by the user or by law
    - Accessed on user login
        - This shall be used as frequently as the user needs. Usually this would be once during initial login.
        - The application shall be able to verify at least 50 user accounts per second.
    - Updated
        - This shall be used as frequently as the user needs. Specifically, when account details need to be changed
        - The system shall be able to update at least 50 accounts per second
- Incident reports
    - Generated by the 'create incident report' functions
        - This shall be used for every incident. The number of incidents will be infinitely growing. Therefore, at the current state, the system should be able to store and manage at least 1 million entries in the report database.
        - The application shall be able to generate at least 100 incident reports per second
        - Reports will be retained forever.
            - Even if the report is deleted, the system shall still retain a copy of the data without displaying it to users
            - Unless required to be removed by law
    - Accessed when a user view
        - This shall be accessed every view, in map form, list form, or detailed form. Therefore, the number of accesses to this data will be very high.
        - The system shall be able to access at least 20000 report entities per second.
            - This high data requirement is due to the nature of many reports being viewed by many users.
    - Updated when details are added
        - This shall be performed whenever a report is updated.

- The system shall be able to update at least 20000 report entities per second.



To the left is a simple ERD of the Database that is to be implemented for the system. The system shall have only one HOA as per request, but shall be built to accommodate several HOAs should expansion be chosen.

Every user may make multiple reports, multiple discussions, and multiple comments on discussions.

Reports, Discussions, and comments shall belong to only one user.

Every user shall belong to one HOA and until system expansion that HOA shall be the only one and is set by default.

Every HOA shall be allowed to create multiple announcements with a title contents and a tag.

3.5 Design constraints

Design constraints for this system are limited. The majority of the functionality of the system will be on the end of a server that is independent. The main restraint is that our system will not be able to communicate with the police database on a continuous basis and the system will not be able to call the police because flooding the police systems would ultimately be worse that not having the system available in the first place.

3.6 Software system attributes

3.6.1 Reliability
3.6.2 Availability
3.6.3 Security
3.6.4 Maintainability
3.6.5 Portability

3.6.1 Reliability
The application will function as defined in this document at the time of delivery, however, should trouble shooting be required the developer will act in accordance with the contractually agreed upon maintenance contract.

3.6.2 Availability
Availability of the application will rely on dependencies on external products not organic to the developed application. These dependencies include but are not limited to: a functioning device with the application properly installed, access to the internet, and appropriate permissions grated by the HOA moderators. Availability shall not be hindered due to user demand or strained resources unless implemented by HOA moderators. A reduction in availability may indicate the need for increased resources or additional administrators to meet higher demand than defined in this document.

3.6.3 Security
- The system shall use secure sockets in all transactions that include any confidential user information such as password.
- The system shall automatically log out all customers when password information is changed for that user.
- The system shall not leave any cookies on the customer's computer containing the user's password.
- The system shall not leave any cookies on the customer's computer containing any of the user's confidential information.

3.6.4 Maintainability
Documentation will be released for the following operations:

1. Add user
   HOA moderator shall have the ability to manually add users independent of user requests
2. Remove User
   HOA moderators shall have the ability to remove users when desired
3. Remove Events from map
   HOA moderators shall have the ability to control the events added to the neighborhood map
4. Remove comments from discussion board
   HOA moderators shall have discretion regarding the community board posts
5. Web hosted server management
   - Documentation shall include directions for maintaining server capacity
   - Documentation shall include instruction on account management
6. Maintenance/developer continuity
   - HOA will have level 2 support in accordance with sales contract
   - Developer will provide assistance throughout the initial product deployment

3.6.5 Portability

The application should be portable with iOS and Android.

3.7 Organizing the specific requirements

3.7.1 System mode

The system shall have 4 modes:

1. HOA – The admin has complete control over the database systems
2. Volunteer (Neighborhood watch members) - Has assess to creation of incident reports, and other special privileges.
3. Moderator – Has special privileges to observe and manage discussion boards
4. Regular user – which will have access to charts data and other features but will be restricted from higher security sections.

3.7.2 User class

1. The system shall provide (4) user classes
    1. The system shall have a ‘member’ user class. Member users shall be able to take shifts on the neighborhood watch calendar
    2. The system shall have a ‘moderator’ user class. Moderators shall have special permissions to delete inappropriate discussions board posts and comments under discussion boards.
    3. The system shall have an ‘admin’ class which is special to the HOA association. This class shall have complete control over the database.
    4. The system shall have a ‘viewer’ user class
        a) All users shall be able to view and add incidents
        b) All users shall be able to view neighborhood watch schedule
        c) All users shall be able to view neighborhood watch contact details

3.7.3 Objects

1. **User** – A n individual with access to the application with a username and password
   Attributes:
    - **a.** A user shall have a name (user name)
    - **b.** A user shall have a password
    - **c.** A user shall have an email
    - **d.** A user could have special permissions (moderator - standard user- neighborhood watch member)

2. **Servers** – Instrument used to store information and allow connectivity between neighborhood members
   Attributes:
    - **a.** A server shall have a name e.g., server 1, server0x..... etc.
    - **b.** A server shall have a user name e.g. Administrator
    - **c.** A server shall have a password file
    - **d.** A server shall have a port name

e. A server shall have a host name

3. **Mobile hardware** – Instrument used to access the application
   Attributes:
   a. A mobile hardware shall have a name
   b. A mobile hardware shall have internet service
   c. A mobile hardware shall have a location service
   d. A mobile hardware shall have application support

3.7.4 Feature

A feature is a desired service requested of the system that may require a sequence of user stories to create the desired result.

The SRS defines features as requested by the customer along with definitions and user stories in section 3.2. A generalized list of features is provided here:

- The System shall display a home screen, which shall contain announcements from the HOA
- The System Shall allow a user to change their settings.
- They System shall allow the HOA to change the roles of people directly through the database
- The System shall allow users to sign up for neighborhood watch on the calendar
- The system shall allow users to make an incident report
- The system shall allow users to sign up for an account
- The system shall allow users to communicate via discussion board
- The System shall notify members of the neighborhood watch of new incident reports
- The system shall allow users to view crime incidents in the neighborhood through an embedded map
- The system shall allow users to view a crime and incident statistic report of their neighborhood

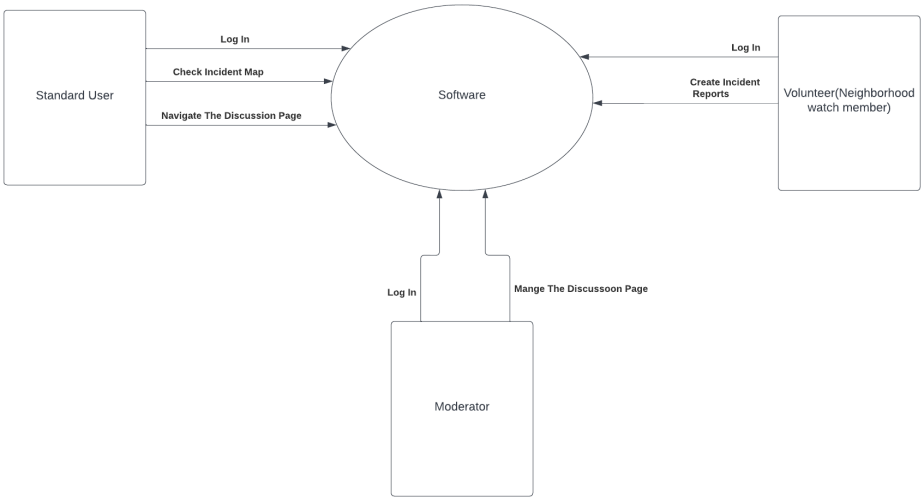
3.7.5 Response

Responses generated by our system will be primarily graphics interpreting the data from the users in an area so our system can be split into:

All functions that graph and chart statistics
All functions that produce the watch schedule
All functions that handle the discussion board
All functions that compare server data to police system data

3.7.6 Functional hierarchy

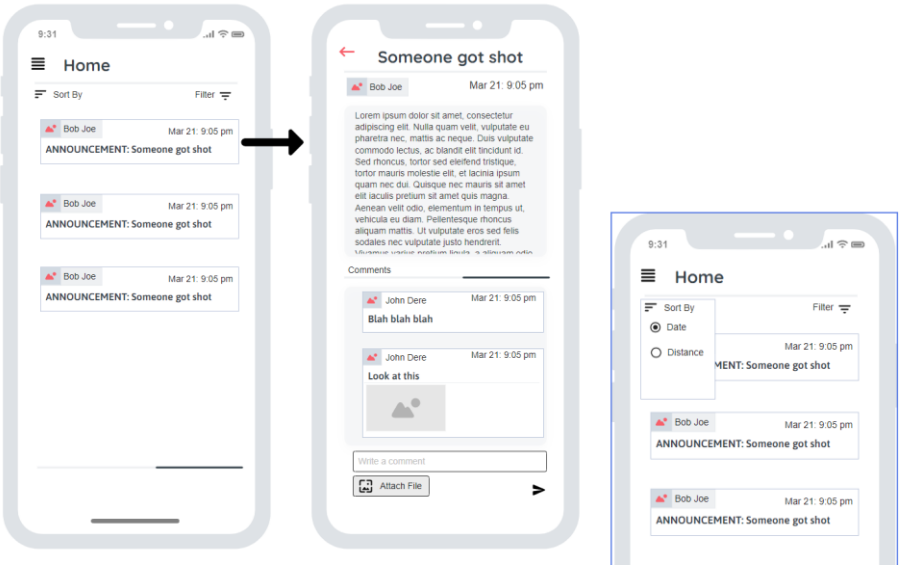
Below is a depiction of the functional hierarchy. There is a standard user, a neighborhood watch member and a moderator all of whom interface with the software via the GUI.



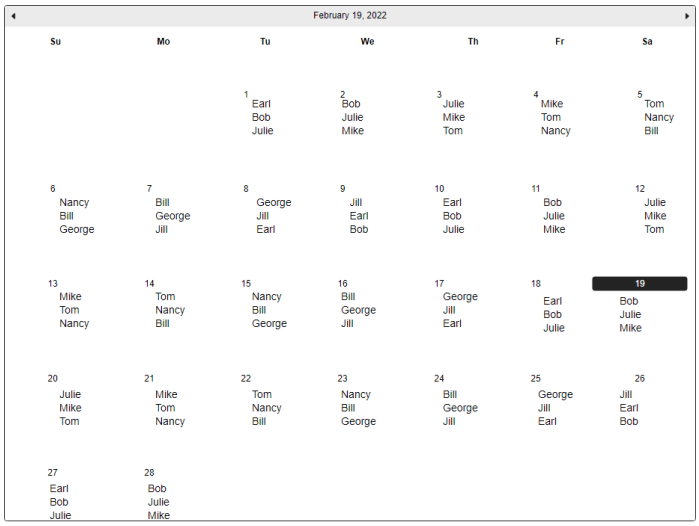
4 Diagrams

The following important images and diagrams are provided within the document for reference:

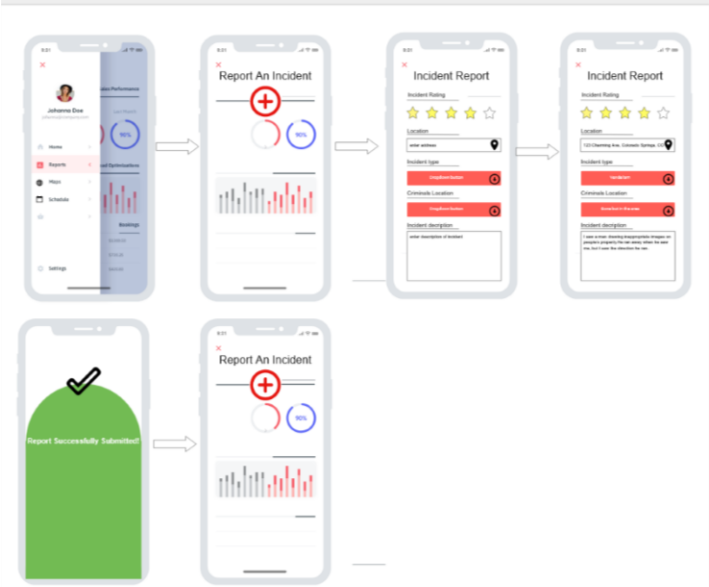
Home page announcements and filtering depiction.



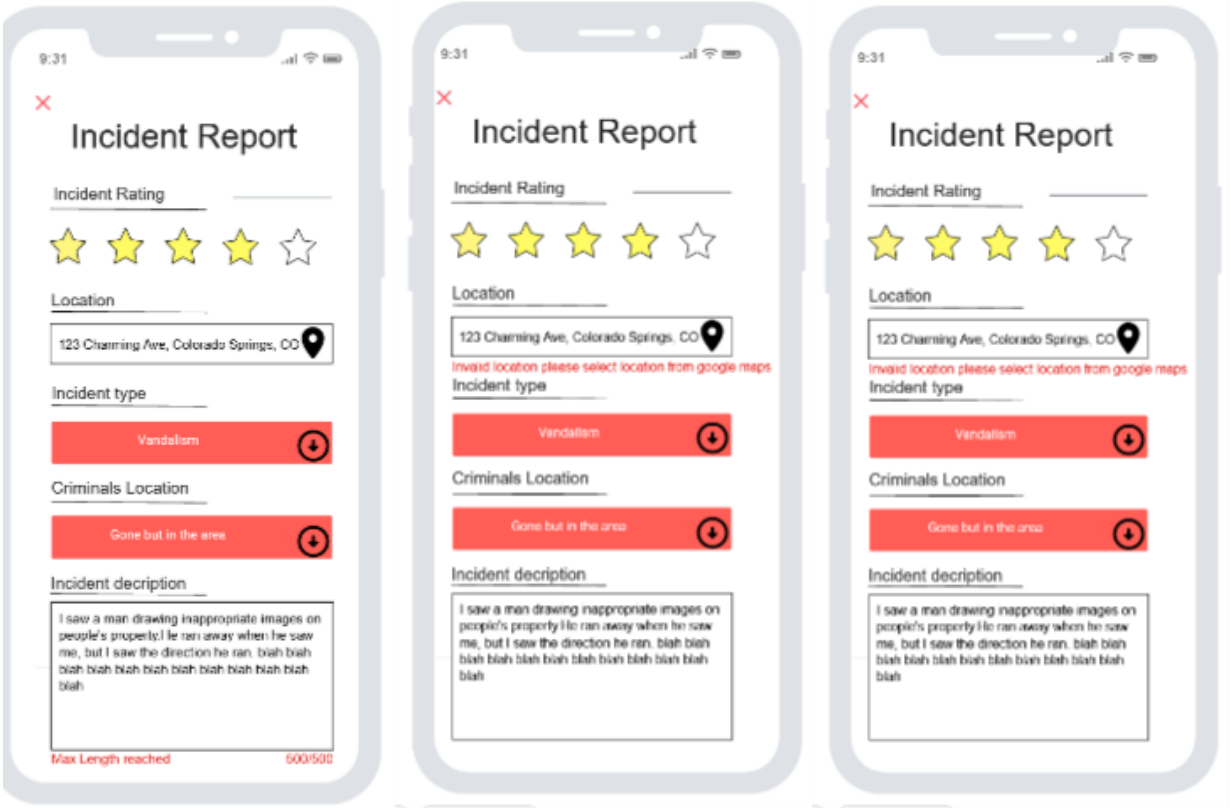
Calendar view of neighborhood watch schedule.



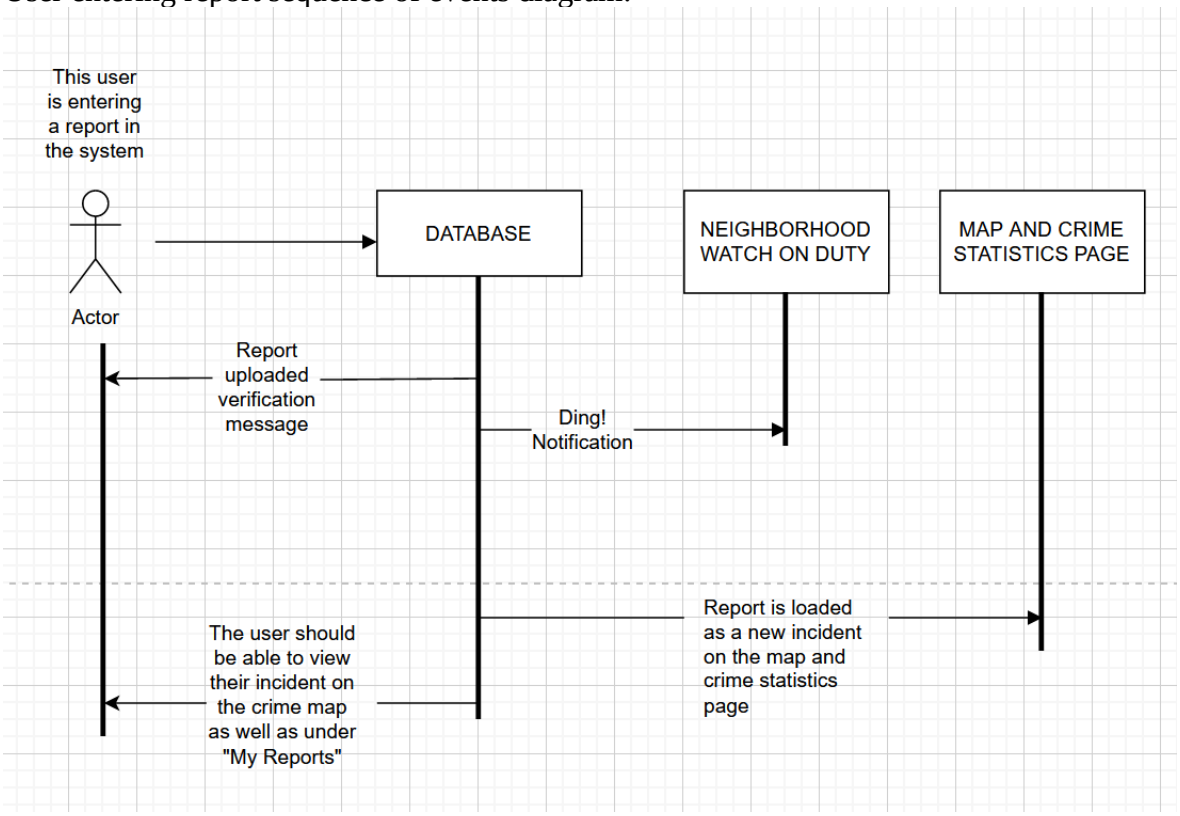
Incident report creation.



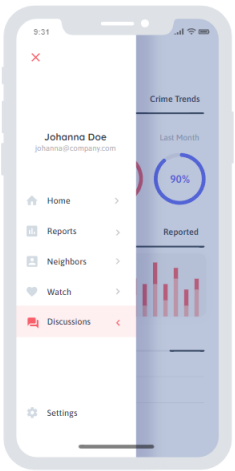
Incident report bad user input.

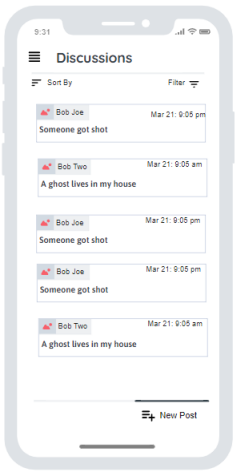


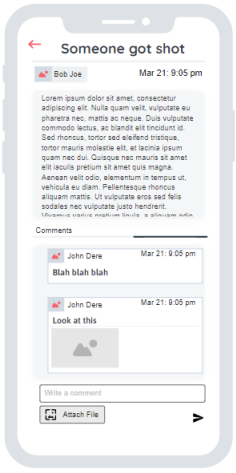
User entering report sequence of events diagram.



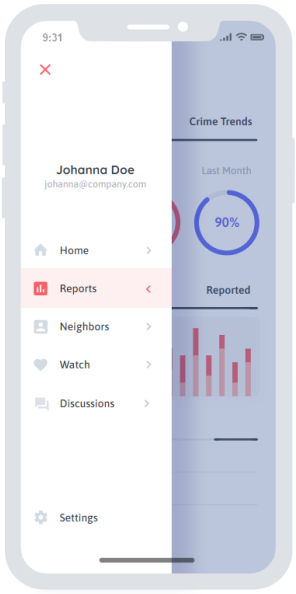
Discussion board depiction.

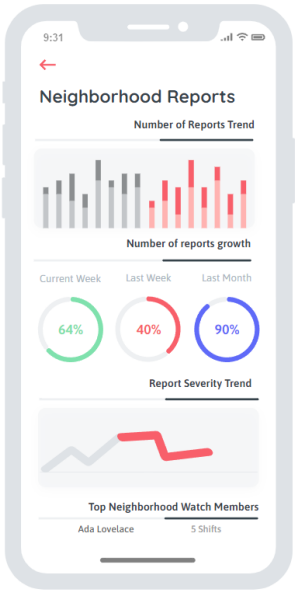






Crime statistics page view.





Database Design