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 6

1.4 References 7

1.5 Overview 7

2 Overall descriptions 7

2.1 Product perspective 7

2.1.1 System interfaces 8

2.1.2 User interfaces 8

2.1.3 Hardware interfaces 9

2.1.4 Software interfaces 9

2.1.5 Communications interfaces 10

2.1.6 Memory constraints 10

2.1.7 Operations 10

2.1.8 Site adaptation requirements 11

2.2 Product functions 11

2.3 User characteristics 12

2.4 Constraints 12

2.5 Assumptions and dependencies 12

2.6 Apportioning of requirements 12

3 Specific requirements 12

3.1 General Overview 13

3.2 Functions 13

3.3 Performance requirements 26

3.4 Logical database requirements 27

3.5 Design constraints 29

3.5.1 Standards compliance (not applicable?) .... **Error! Bookmark not defined.**

3.6 Software system attributes 29

3.6.1 Reliability 29
3.6.2 Availability 30
3.6.3 Security 30
3.6.4 Maintainability 30
3.6.5 Portability 31
3.7 Organizing the specific requirements 31
3.7.1 System mode 31
3.7.2 User class 31
3.7.3 Objects 31
3.7.4 Feature 32
3.7.5 Stimulus 32
3.7.6 Response 33
3.7.7 Functional hierarchy 33
3.8 Additional comments 33
4 Supporting information 33
4.1 Table of contents and index 34
4.2 Appendixes 34

Revision History

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

1 Introduction

1.1 Purpose

This product documents the design and procedure necessary to develop an application for the homeowners of XYZ neighborhood. It is divided into an Introduction, Specific Requirements, and Supporting documentation.

The document will be effective for use by the customer, product owners, and the technical implementation team.

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
MySQL	8.0.28
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 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. 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.
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 remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product. The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences. The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen. The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

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.

This section usually consists of six subsections, as follows:

1. Product perspective;
1. Product functions;
- User characteristics;
1. Constraints;
2. Assumptions and dependencies;
1. Apportioning of requirements.

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. Personnel who will have access to the trend reports is to be determined after discussions with stakeholders about privacy issues and access to sensitive information.

2.1.1 System interfaces

This should list each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system.

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 software packages such as Java Swing, Python GUI, etc.

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:

4. **Design (color and theme) is currently in progress**
5. Assessing the platform through the website or the application will transfer the users to a login page
6. The login page will request the user to enter their chosen username and password.
7. There is a button in the login page which the users can click to be delivered to a page for obtaining their forgotten passwords
    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, crime map, etc.
    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 100 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. The designated leader of the neighborhood watch is provided with special privileges(password) to assign calendar days to the watch members by clicking the add button on the page. The add button directs the watch leader to a new window where the leader can delegate some member of the neighbor watch a day and time of duty.
7. The **Crime Statistics** features trend graphs that should be automatically updated when reports are added to the system. (This needs discussion with customers for specifics on how crimes are displayed and organized).
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. We need to discuss what this will look like in further detail with the stakeholders.

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, and laptops. The mobile device must have access to the internet, and it can either utilize the software as an application or has 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 need to be discussed further with stakeholders.

2.1.4 Software interfaces

The application shall utilize the following software interfaces:

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 interface can be defined in SQLite documentation.

The fields of interest for the database service include several categories:

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

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.

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.

2.1.7 Operations

The application, under normal operation shall allow for 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

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 30 days after creation

2.1.8 Site adaptation requirements

The application shall 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 shall initialize the user accounts upon creation as

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

2.2 Product functions

The application shall have the following functionality.

The ability to log in – Login page –

Log in shall be through the application

Log in shall be through email and password

Security verification – undetermined -

The Home pages

Undetermined

-potential- announcements from the HOA

-potential- quick glance crime statistics what has occurred in the last week

The Crime Statistics page

Ability to view crime reports within a certain radius

The ability to sort crimes based on -undetermined-

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 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

–potentially- The ability to sign up for the watch

The discussion boards

Allow neighbors to start discussions on any issues they may have

Allow neighbors to comment on discussions

-potentially- allow HOA to moderate discussions

Crime Report for police

The ability to print a weekly crime report for the police so the HOA may discuss issues

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. 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 or flood police databases. 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

In the production of the neighborhood watch scheduler the main constraint for our team is is the communication of our software to the local police. 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 or flood police databases. 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 the features most likely to be postponed until later in development are police notification/documentation systems.

3 Specific requirements

This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output. As this is often the largest and most important part of the SRS, the following principles apply:

6. Specific requirements should be stated in conformance with all the characteristics described in 4.3.
2. Specific requirements should be cross-referenced to earlier documents that relate.
1. All requirements should be uniquely identifiable.
1. Careful attention should be given to organizing the requirements to maximize readability.

Before examining specific ways of organizing the requirements it is helpful to understand the various items that comprise requirements as described in 3.1 through 3.7.

3.1 General Overview

Give a summary of application. This should be a general summary of the application that talks about what is in our side menu that users can access and main functionalities. It does not need to be specific that is for 3.2. Very quick and very general

3.2 Functions

(heading 4 under styles for formatting) 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. This should have 1 diagram of entire "happy path".
B. Validity checks. These are the checks the system should go through to make sure the user has valid inputs. Ie. A user must enter a valid address or select their current location. A users 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 include diagrams of how the system responds to incorrect 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. **Simple diagrams may be appropriate.**

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**

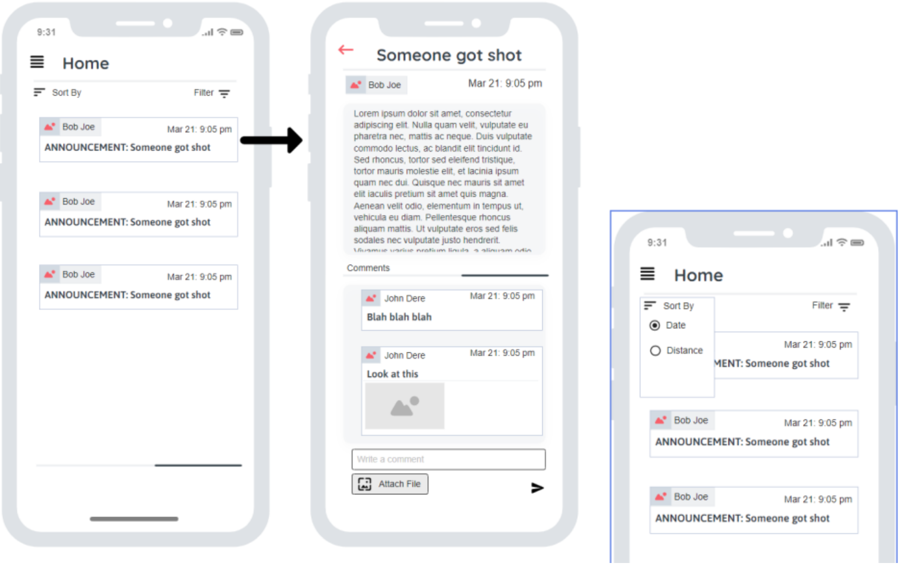
    1. The user shall be able to enter any text into the search box.

C. Abnormal input to the Systems

    1. If the user uses special characters, no error should be thrown.
    2. The system shall use all characters in the search/filtering

D. Relationships of inputs and outputs.

    1. The input to this page is discussion posts
        a. The system shall display posts labeled announcements
    2. When clicked the system shall redirect to the corresponding announcement discussion post



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio

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:
   b. The user will select the frequency of notifications
   c. The user will select how they will receive notifications
   d. The user will enter a desired email address
   e. 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. Notification of direct message to moderators
   d. Notification of abnormal use

B. **Validity Checks**

1. The user will receive an email confirmation any time a change is made to their profile settings.

C. **Abnormal input to the Systems**

1. If the user is not able to enter the password during a change to the settings a notification is sent to the list of moderators.

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

1. The inputs to the system are the selections made by the user.
2. Outputs from the system
   a. verification emails to the user
   b. Notifications of abnormal input
   c. 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 priviledges) 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

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.

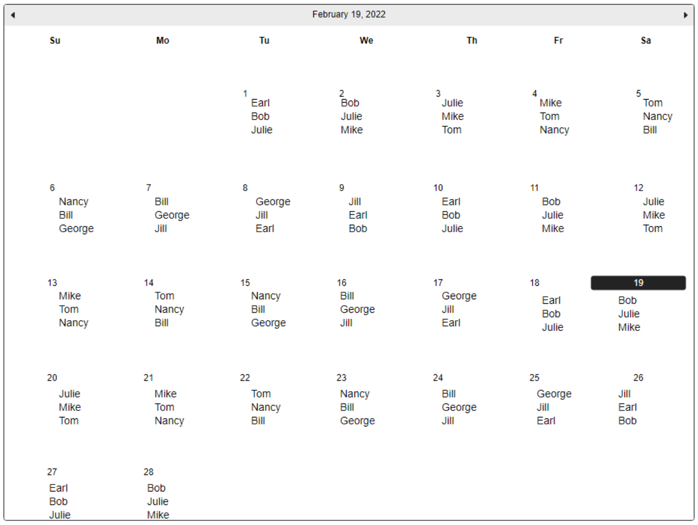
B. Validity Checks

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.



February 19, 2022

Su	Mo	Tu	We	Th	Fr	Sa
		1 Earl Bob Julie	2 Bob Julie Mike	3 Julie Mike Tom	4 Mike Tom Nancy	5 Tom Nancy Bill
6 Nancy Bill George	7 Bill George Jill	8 George Jill Earl	9 Jill Earl Bob	10 Earl Bob Julie	11 Bob Julie Mike	12 Julie Mike Tom
13 Mike Tom Nancy	14 Tom Nancy Bill	15 Nancy Bill George	16 Bill George Jill	17 George Jill Earl	18 Earl Bob Julie	19 Bob Julie Mike
20 Julie Mike Tom	21 Mike Tom Nancy	22 Tom Nancy Bill	23 Nancy Bill George	24 Bill George Jill	25 George Jill Earl	26 Jill Earl Bob
27 Earl Bob Julie	28 Bob Julie Mike					

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.

Below is a view of what the view shall be if a user attempts to sign up for the watch.

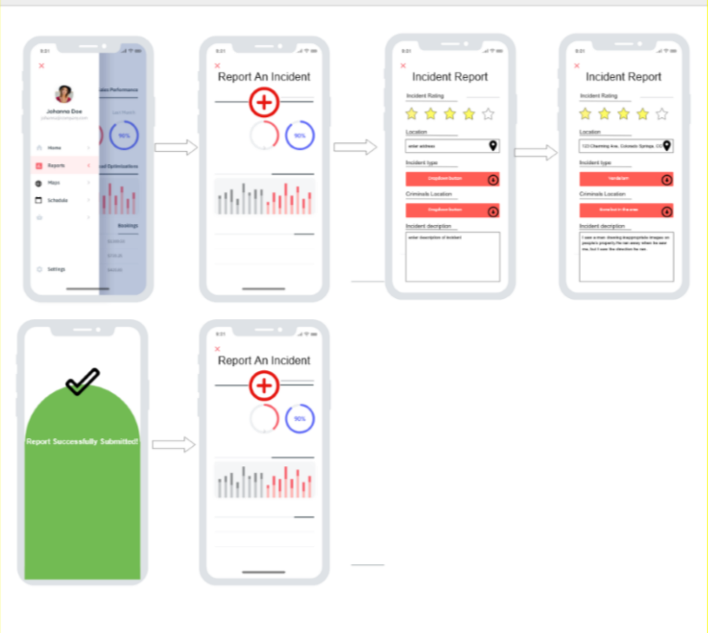
V. *The system shall allow users to make an incident report*

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. The user shall press the Report an Incident button located at the top of the screen of the reports page.
5. The incident report form shall pop up.
6. 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.

7. The user may then press the Submit Incident button.
8. The system shall automatically add the date and time to report based on users system.
9. The user shall be taken to a screen which says report successfully submitted.
10. 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.

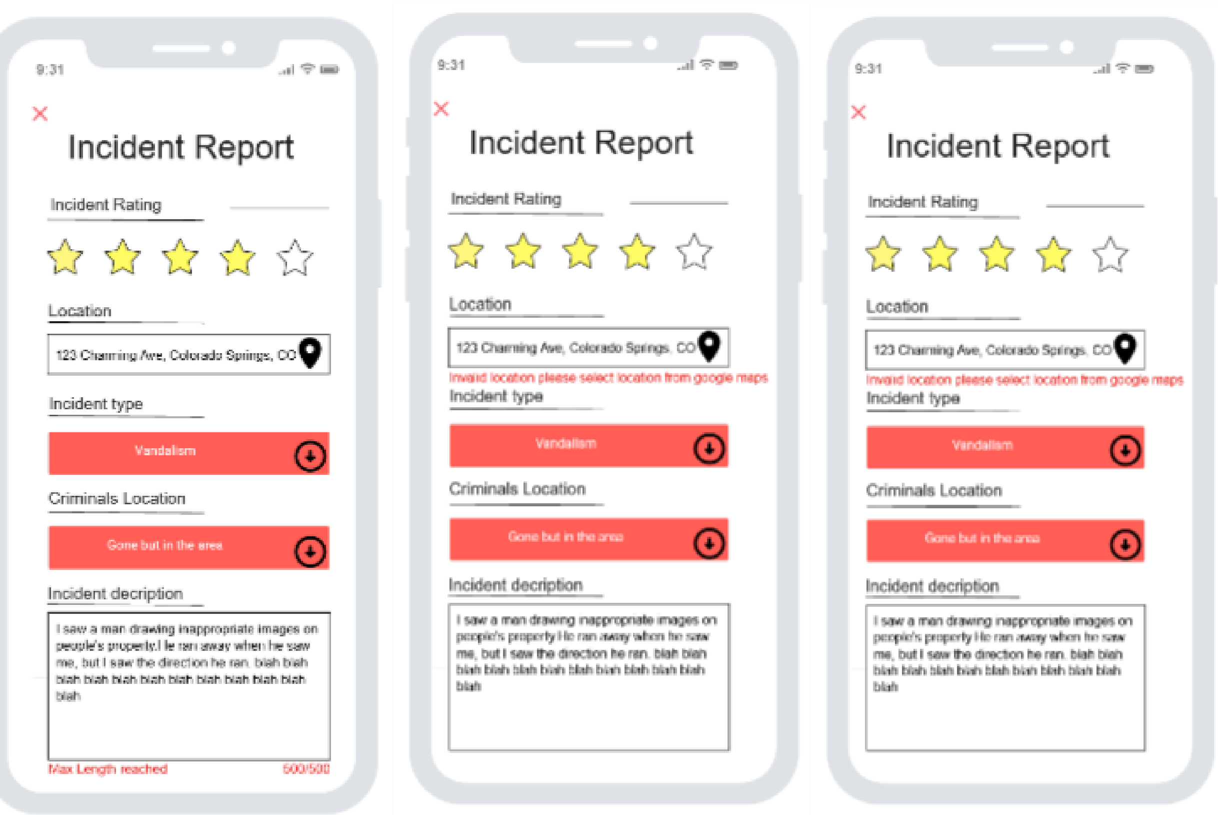
1. The user must enter a valid address or select their current location.
2. The user must select a valid crime from a drop-down list.
3. The user must select a valid Perpetrator Location from 3 given buttons; In progress, done but still in the area, and done and gone.
4. A user may add a detailed report 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”



123 Charming Ave, Colorado Springs, CO

I saw a man drawing inappropriate images on people's property.I le ran away when he saw me, but I saw the direction he ran. blah blah blah blah blah blah blah blah blah blah blah blah

Max Length reached 500/500

123 Charming Ave, Colorado Springs, CO

Invalid location please select location from google maps

I saw a man drawing inappropriate images on people's property.He ran away when he saw me, but I saw the direction he ran. blah blah blah blah blah blah blah blah blah blah blah blah

123 Charming Ave, Colorado Springs, CO

Invalid location please select location from google maps

I saw a man drawing inappropriate images on people's property.He ran away when he saw me, but I saw the direction he ran. blah blah blah blah blah blah blah blah blah blah blah blah

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.

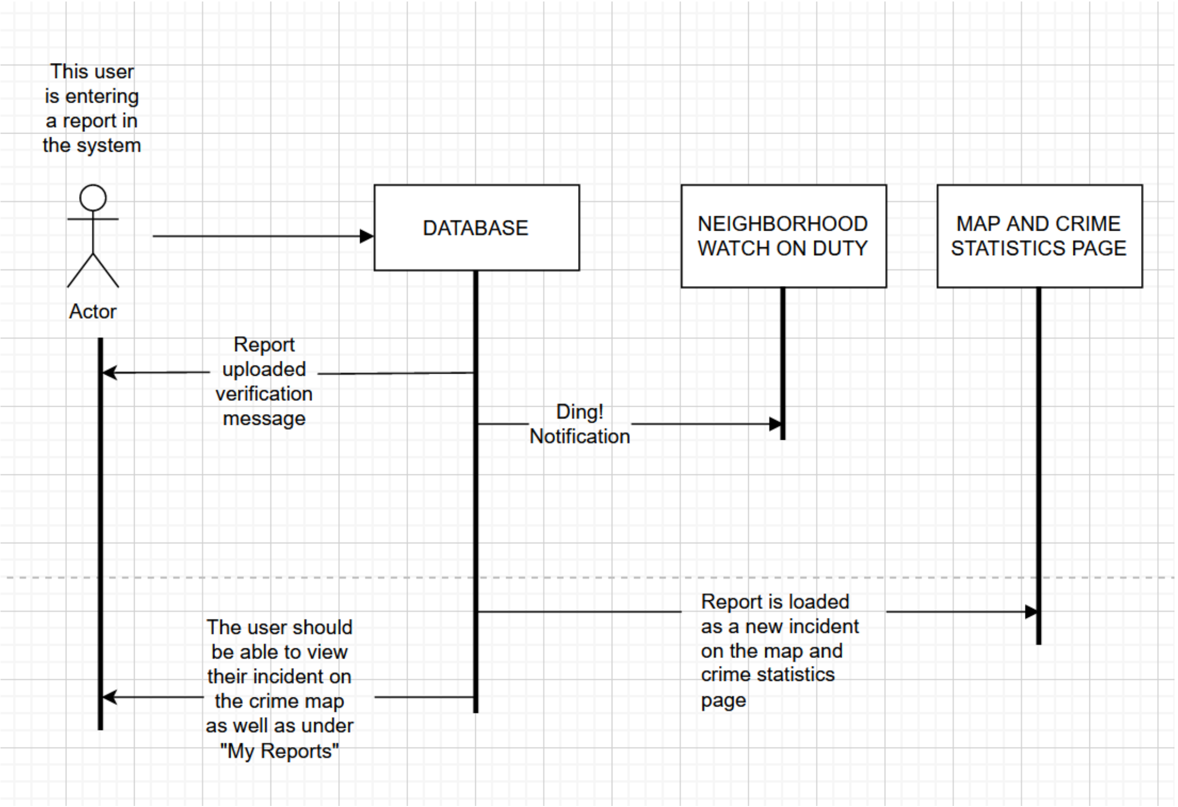
4. Error Handling and Recovery ie. Submitting Report with Missing requirements - If a user attempts to submit a report without all of the proper fields the System shall highlight them in red and mark them as required fields without submitting the report. There shall also be a pop-up which tells them 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”


This user is entering a report in the system
Actor
DATABASE
NEIGHBORHOOD WATCH ON DUTY
MAP AND CRIME STATISTICS PAGE
Report uploaded verification message
Ding! Notification
Report is loaded as a new incident on the map and crime statistics page
The user should be able to view their incident on the crime map as well as 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 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

B. Validity checks

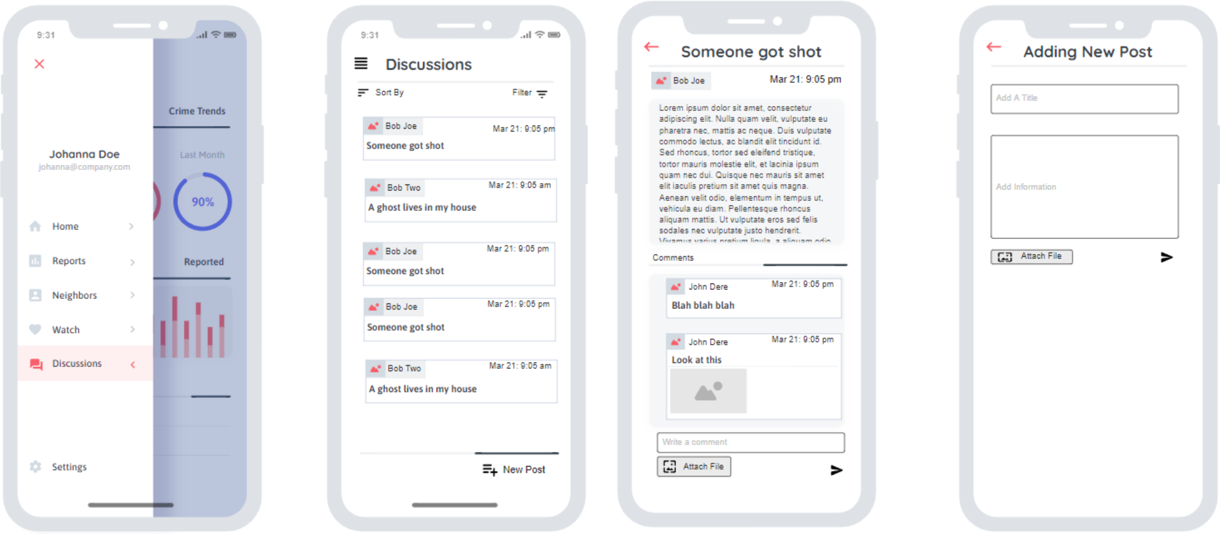
1. Once given a new entry the system shall verify that the content of the data is suitable for discussion board not allowing topics that are not allowed by system rules.

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



Bob Joe Mar 21: 9:05 pm
Someone got shot

Bob Two Mar 21: 9:05 am
A ghost lives in my house

Bob Joe Mar 21: 9:05 pm
Someone got shot

Bob Joe Mar 21: 9:05 pm
Someone got shot

Bob Two Mar 21: 9:05 am
A ghost lives in my house

Bob Joe Mar 21: 9:05 pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit.

John Dere Mar 21: 9:05 pm
Blah blah blah

John Dere Mar 21: 9:05 pm
Look at this

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

The system shall notify members of the neighborhood watch of new incident reports

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
   - **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 (TBD) of a window page. 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 should 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.

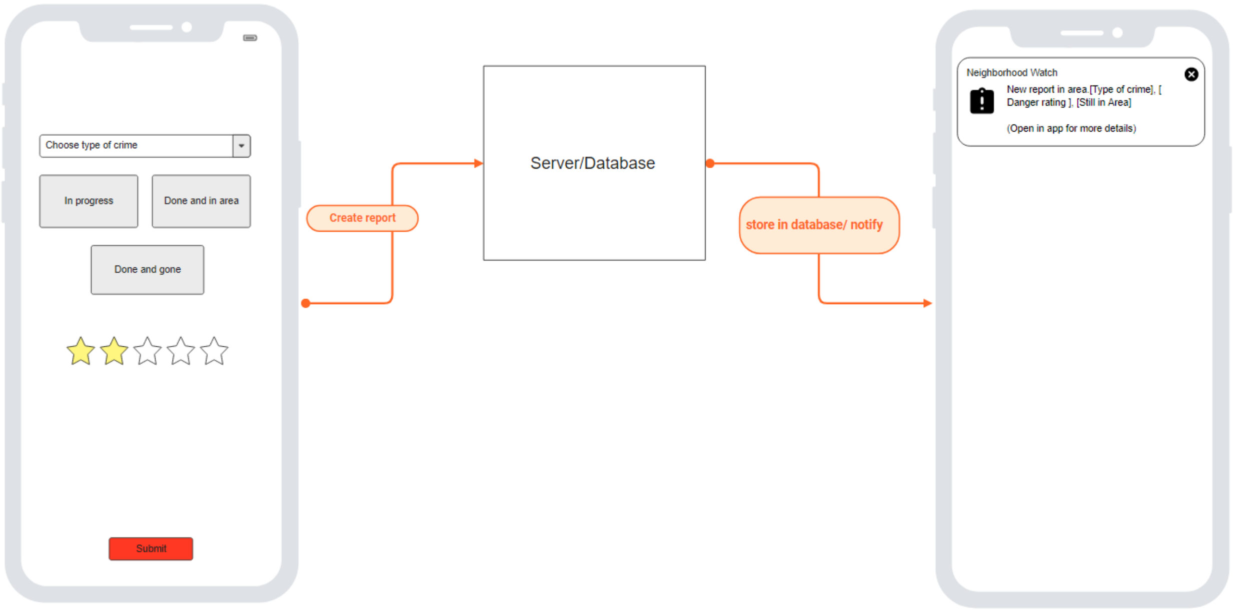
B. Validity Checks

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


Choose type of crime
In progress
Done and in area
Done and gone
Submit
Create report
Server/Database
store in database/ notify
Neighborhood Watch
New report in area.[Type of crime], [Danger rating], [Still in Area]
(Open in app for more details)

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

B. Validity Checks

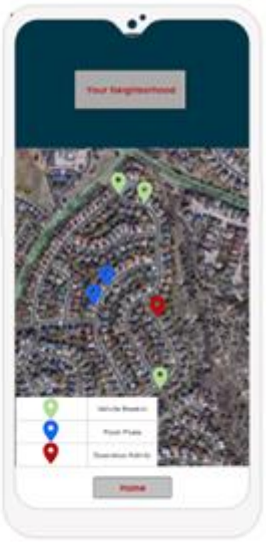
1. Once the map page is opened the system shall ensure that the user is currently on an internet accessing device in order to generate the map
2. The system shall verify the phone number of the user before allowing the map to generate

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 representation of inputs given to the database.



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
2. Street comparison page, which shows the surrounding neighborhood and the safest/least safe streets inside of it.
3. A toggle between weekly, monthly, and annual statistics to show consistency or change in the area.
4. Street incident view page, showing the intensity and severity of accidents reported on nearby streets.
5. Graphs page, which shows trend and analysis data of relevant information

B. Validity Checks

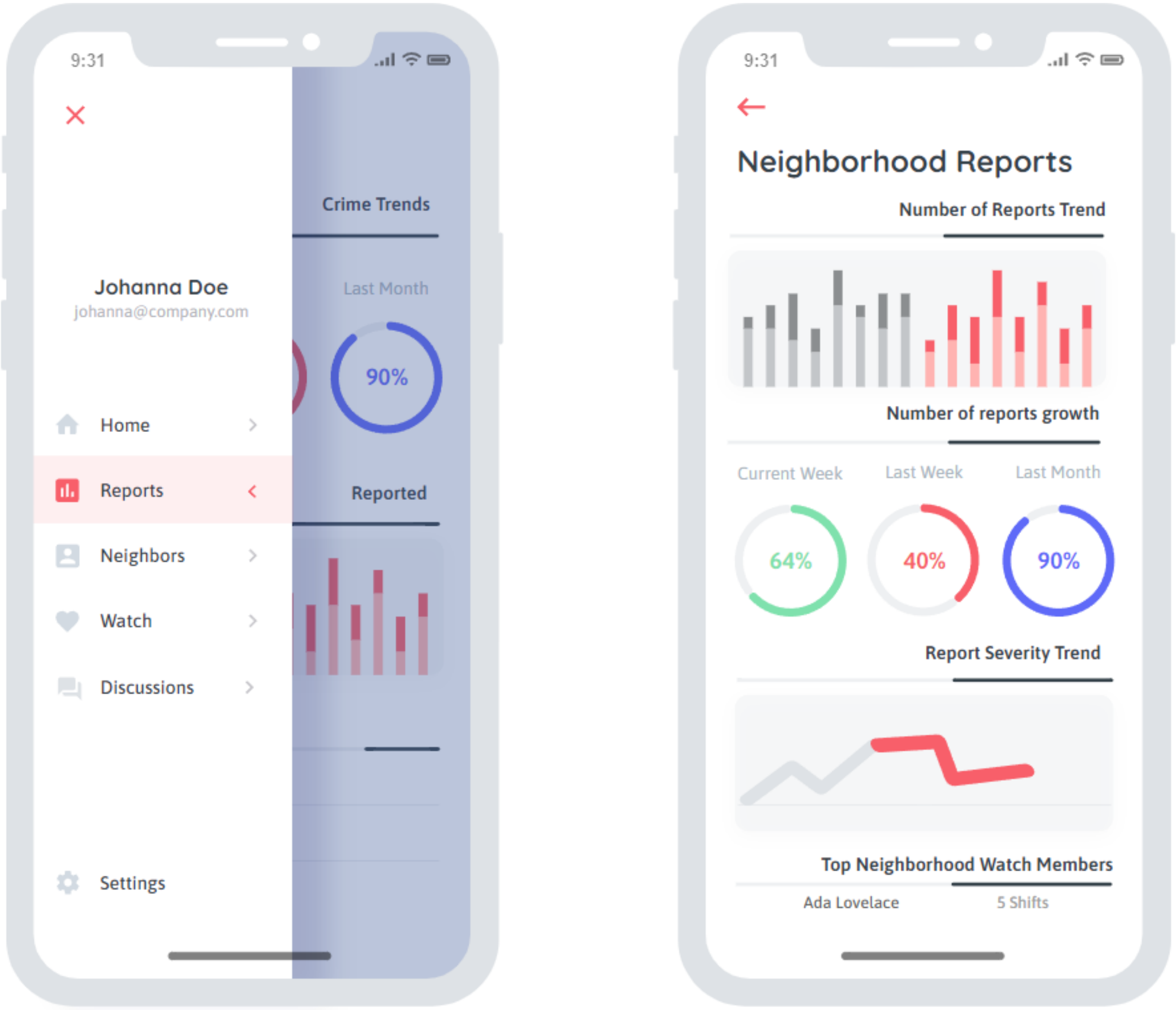
1. When loading the page, the system shall ensure that the user has an internet connection in order to generate the graphs.

C. Abnormal input to the Systems

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

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 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 (to be determined by the HOA) input by users.

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

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

5. The number of terminals to be supported;
3. The number of simultaneous users to be supported;
3 Amount and type of information to be handled.

Static numerical requirements are sometimes identified under a separate section entitled Capacity.

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions. All of these requirements should be stated in measurable terms.

For example,
95% of the transactions shall be processed in less than 1 s.

rather than,
An operator shall not have to wait for the transaction to complete.

NOTE – Numerical limits applied to one specific function are normally specified as part of the processing subparagraph description of that function.

3.4 Logical database requirements

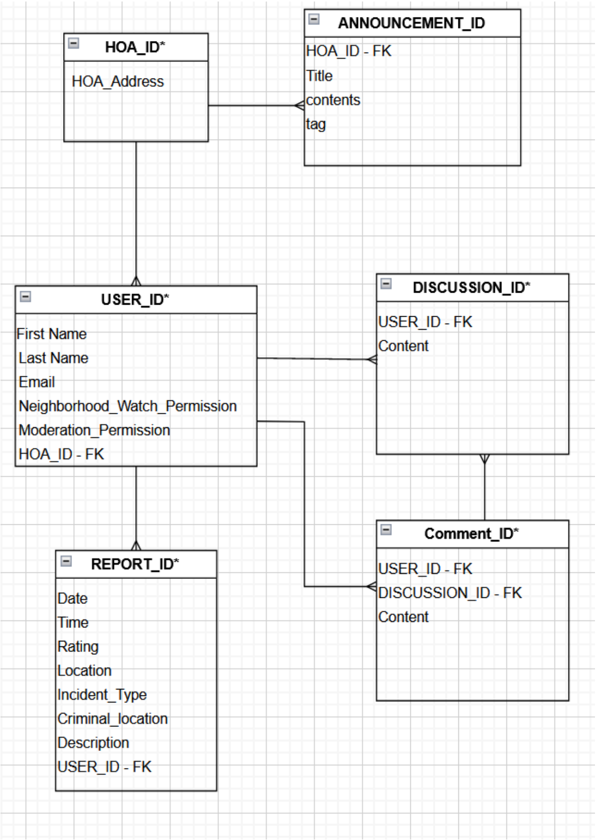
This should specify the logical requirements for any information that is to be placed into a database. This may include the following:

2. Types of information used by various functions;
5. Frequency of use;
4. Accessing capabilities;
A. Data entities and their relationships;
    a. Integrity constraints;
3. Data retention 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.

d



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

This should specify design constraints that can be imposed by other standards, hardware limitations, etc.
This should be like we can't call 911. We must make it an application.
A user should have knowledge of mobile applications other mobile internet-based functionalities i.e. opening an application, understanding how to use mobile text input etc.

Application will need handicap

3.6 Software system attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified. Subclauses 3.6.1 through 3.6.5 provide a partial list of examples.

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. These include cellular or WI-FI internet access, web hosted server resiliency, and access as determined by HOA moderators.

3.6.3 Security

This should specify the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to

2. Utilize certain cryptographical techniques;
(a) Keep specific log or history data sets;
A. Assign certain functions to different modules;
2. Restrict communications between some areas of the program;
3. Check data integrity for critical variables.

3.6.4 Maintainability

The application will have release documentation that includes the following:

Add user
    HOA moderator shall have the ability to manually add users independent of user requests
Remove User
    HOA moderators shall have the ability to remove users when desired
Remove Events from map
    HOA moderators shall have the ability to control the events added to the neighborhood map
Remove comments from discussion board
    HOA moderators shall have discretion regarding the community board posts
Web hosted server management
    Documentation shall include direction for maintaining server capacity
    Documentation shall include instruction on account management
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

1. The application should be portable with iOS and Android.

3.7 Organizing the specific requirements

For anything but trivial systems the detailed requirements tend to be extensive. For this reason, it is recommended that careful consideration be given to organizing these in a manner optimal for understanding. There is no one optimal organization for all systems. Different classes of systems lend themselves to different organizations of requirements in Section 3 of the SRS. Some of these organizations are described in 3.7.1 through 3.7.7.

3.7.1 System mode

1. The system will have 3 modes:
    a. HOA, which will act like a moderator or admin for that region
    b. Volunteer, which will have access to night watch schedules
    c. 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 (2) user classes
    1. The system shall have a 'member' user class
        4. Member users shall be able to take shifts on the neighborhood watch calendar
    3. The system shall have a 'viewer' user class
    4. All users shall be able to view and add incidents
    3. All users shall be able to view neighborhood watch schedule
    a) All users shall be able to view neighborhood watch contact details

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fighters. When organizing this section by user class, the outline in A.3 should be used.

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 provides (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

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. When organizing this section by object, the outline in A.4 should be used. Note that sets of objects may share attributes and services. These are grouped together as classes.

3.7.4 Feature

A feature is an externally desired service by the system that may require a sequence of inputs to effect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs. When organizing this section by feature, the outline in A.5 should be used.

3.7.5 Stimulus

Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc. When organizing this section by stimulus, the outline in A.6 should be used.

3.7.6 Response

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc. The outline in A.6 (with all occurrences of stimulus replaced with response) should be used.

3.7.7 Functional hierarchy

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by either common inputs, common outputs, or common internal data access. Data Flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data. When organizing this section by functional hierarchy, the outline in A.7 should be used.

3.8 Additional comments

Whenever a new SRS is contemplated, more than one of the organizational techniques given in 3.7.7 may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification. For example, see A.8 for an organization combining user class and feature. Any additional requirements may be put in a separate section at the end of the SRS.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, Finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data Flow diagrams and data dictionaries may prove helpful.

In any of the outlines given in A.1 through A.8, those sections called “Functional Requirement” may be described in native language (e.g., English), in pseudocode, in a system definition language, or in four subsections titled: Introduction, Inputs, Processing, and Outputs.

4 Supporting information

The supporting information makes the SRS easier to use. It includes the following: Table of contents;

2. Index;
3. Appendixes.

4.1 Table of contents and index

The table of contents and index are quite important and should follow general compositional practices.

4.2 Appendixes

The appendixes are not always considered part of the actual SRS and are not always necessary. They may include

5. Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
3. Supporting or background information that can help the readers of the SRS;
4. A description of the problems to be solved by the software;
2. Special packaging instructions for the code and the media to meet security, export, initial loading, or

other requirements.

When appendixes are included, the SRS should explicitly state whether or not the appendixes are to be considered part of the requirements.