

# 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)

**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 .....                                | 9  |
| 2.1.6 Memory constraints .....                                       | 10 |
| 2.1.7 Operations .....   | 10 |
| 2.1.8 Site adaptation requirements .....                             | 10 |
| 2.2 Product functions .....  | 10 |
| 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 External interfaces .....  | 15 |
| 3.2 Functions .....  | 15 |
| 3.2.1 The System shall implement a neighborhood watch schedule ..... | 15 |
| 3.3 Performance requirements .....                                   | 26 |
| 3.4 Logical database requirements .....                              | 26 |
| 3.5 Design constraints .....   | 26 |
| 3.5.1 Standards compliance .....                                     | 27 |

|  |    |
|--|----|
| 3.6 Software system attributes.....            | 27 |
| 3.6.1 Reliability.....                         | 27 |
| 3.6.2 Availability.....                        | 27 |
| 3.6.3 Security .....                           | 27 |
| 3.6.4 Maintainability .....                    | 28 |
| 3.6.5 Portability.....                         | 28 |
| 3.7 Organizing the specific requirements ..... | 28 |
| 3.7.1 System mode.....                         | 28 |
| 3.7.2 User class .....                         | 28 |
| 3.7.3 Objects .....                            | 29 |
| 3.7.4 Feature .....                            | 29 |
| 3.7.5 Stimulus.....                            | 29 |
| 3.7.6 Response.....                            | 29 |
| 3.7.7 Functional hierarchy .....               | 29 |
| 3.8 Additional comments.....                   | 30 |
| 4 Supporting information.....                  | 30 |
| 4.1 Table of contents and index.....           | 30 |
| 4.2 Appendixes.....                            | 30 |

[illegible]

# 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.

## 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.        |
|                 | -   |

## 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;
2. Product functions;
3. User characteristics;
4. Constraints;
5. Assumptions and dependencies;
6. 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:

- **Design (color and theme) is currently in progress**
- Assessing 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
- A sign-up button in the login page will direct the users to a page where users can create their accounts
- After signing in, the user is directed to the home page. The home page will display recent neighborhood news and information.
- 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.
- 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.
- 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.

- 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.
- 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).
- 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), DHCP (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 the SQLite [documentation](#).

### 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](#).

## 2.1.6 Memory constraints

This should specify any applicable characteristics and limits on primary and secondary memory.

## 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 page

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 page

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

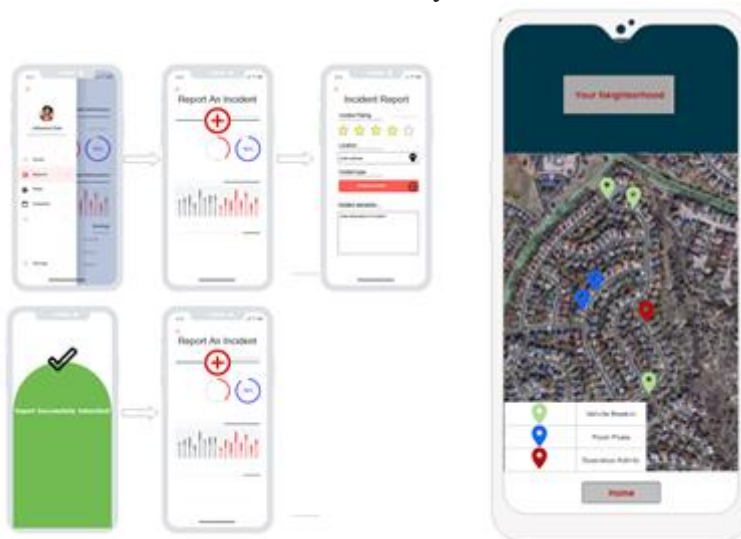
Allow neighbors to comment on discussions

-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

We have some diagrams, but we want to work on them more and get the HOA involved on the design so it looks nice. Here are a few we've already shown, but need a lot of work.



## **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 to use it.

## **2.5 Assumptions and dependencies**

In the production of the neighborhood watch scheduler the main constraints for our team are 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 to use it.

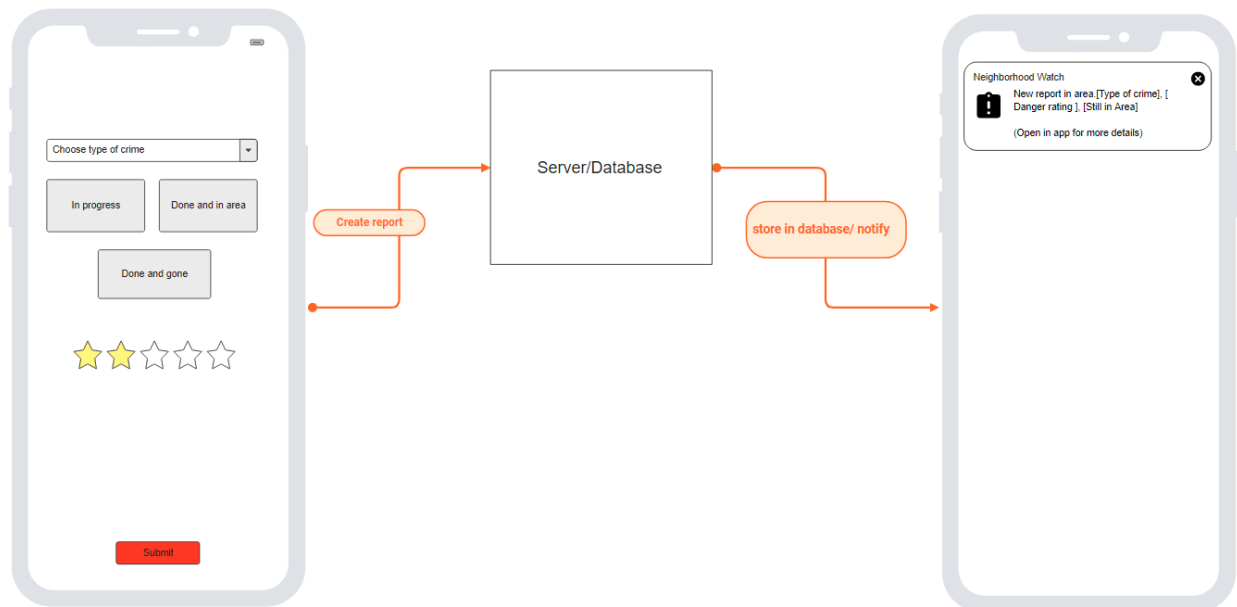
## **2.6 Apportioning of requirements**

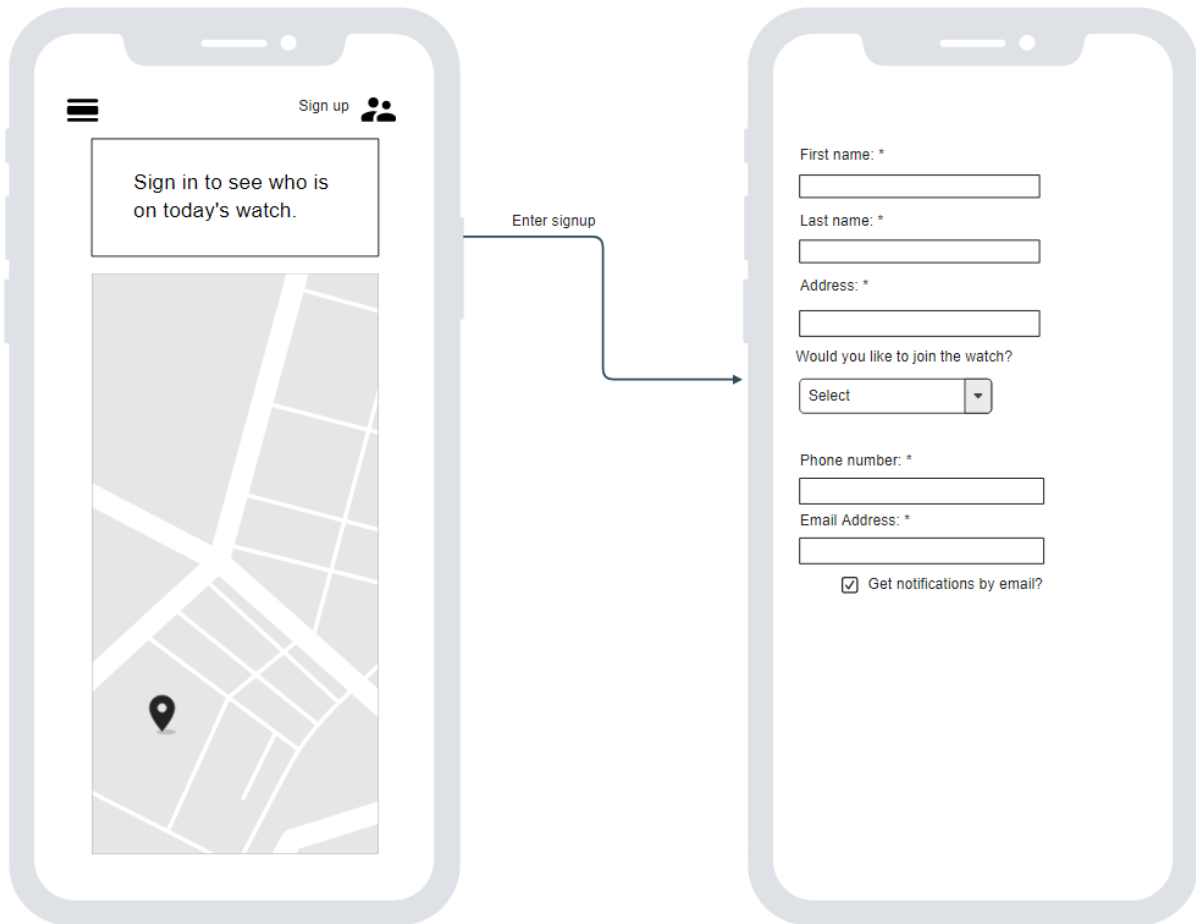
Based on the dependencies above the features most likely to be postponed until later in development are police notification/documentation systems.

## **3 Specific requirements**

\*Temporary placement for low-fi diagrams and storyboards\*

| February 19, 2022            |                             |                               |                               |                              |                              |                            |
|------------------------------|-----------------------------|-------------------------------|-------------------------------|------------------------------|------------------------------|----------------------------|
| Su                           | Mo                          | Tu                            | We                            | Th                           | Fr                           | Sa                         |
|                              |                             | 1<br>Earl<br>Bob<br>Julie     | 2<br>Bob<br>Julie<br>Mike     | 3<br>Julie<br>Mike<br>Tom    | 4<br>Mike<br>Tom<br>Nancy    | 5<br>Tom<br>Nancy<br>Bill  |
| 6<br>Nancy<br>Bill<br>George | 7<br>Bill<br>George<br>Jill | 8<br>George<br>Jill<br>Earl   | 9<br>Jill<br>Earl<br>Bob      | 10<br>Earl<br>Bob<br>Julie   | 11<br>Bob<br>Julie<br>Mike   | 12<br>Julie<br>Mike<br>Tom |
| 13<br>Mike<br>Tom<br>Nancy   | 14<br>Tom<br>Nancy<br>Bill  | 15<br>Nancy<br>Bill<br>George | 16<br>Bill<br>George<br>Jill  | 17<br>George<br>Jill<br>Earl | 18<br>Earl<br>Bob<br>Julie   | 19<br>Bob<br>Julie<br>Mike |
| 20<br>Julie<br>Mike<br>Tom   | 21<br>Mike<br>Tom<br>Nancy  | 22<br>Tom<br>Nancy<br>Bill    | 23<br>Nancy<br>Bill<br>George | 24<br>Bill<br>George<br>Jill | 25<br>George<br>Jill<br>Earl | 26<br>Jill<br>Earl<br>Bob  |
| 27<br>Earl<br>Bob<br>Julie   | 28<br>Bob<br>Julie<br>Mike  |                               |                               |                              |                              |                            |





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:

1. 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.
3. All requirements should be uniquely identifiable.
4. 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 External interfaces

This should be a detailed description of all inputs into and outputs from the software system. It should complement the interface descriptions in 2 and should not repeat information there.

It should include both content and format as follows:

1. Name of item;
2. Description of purpose;
3. Source of input or destination of output;
4. Valid range, accuracy, and/or tolerance;
5. Units of measure;
6. Timing;
7. Relationships to other inputs/outputs;
8. Screen formats/organization;
9. Window formats/organization;
10. Data formats;
11. Command formats;
12. End messages.

## 3.2 Functions

### 3.2.1 The System shall implement a neighborhood watch schedule

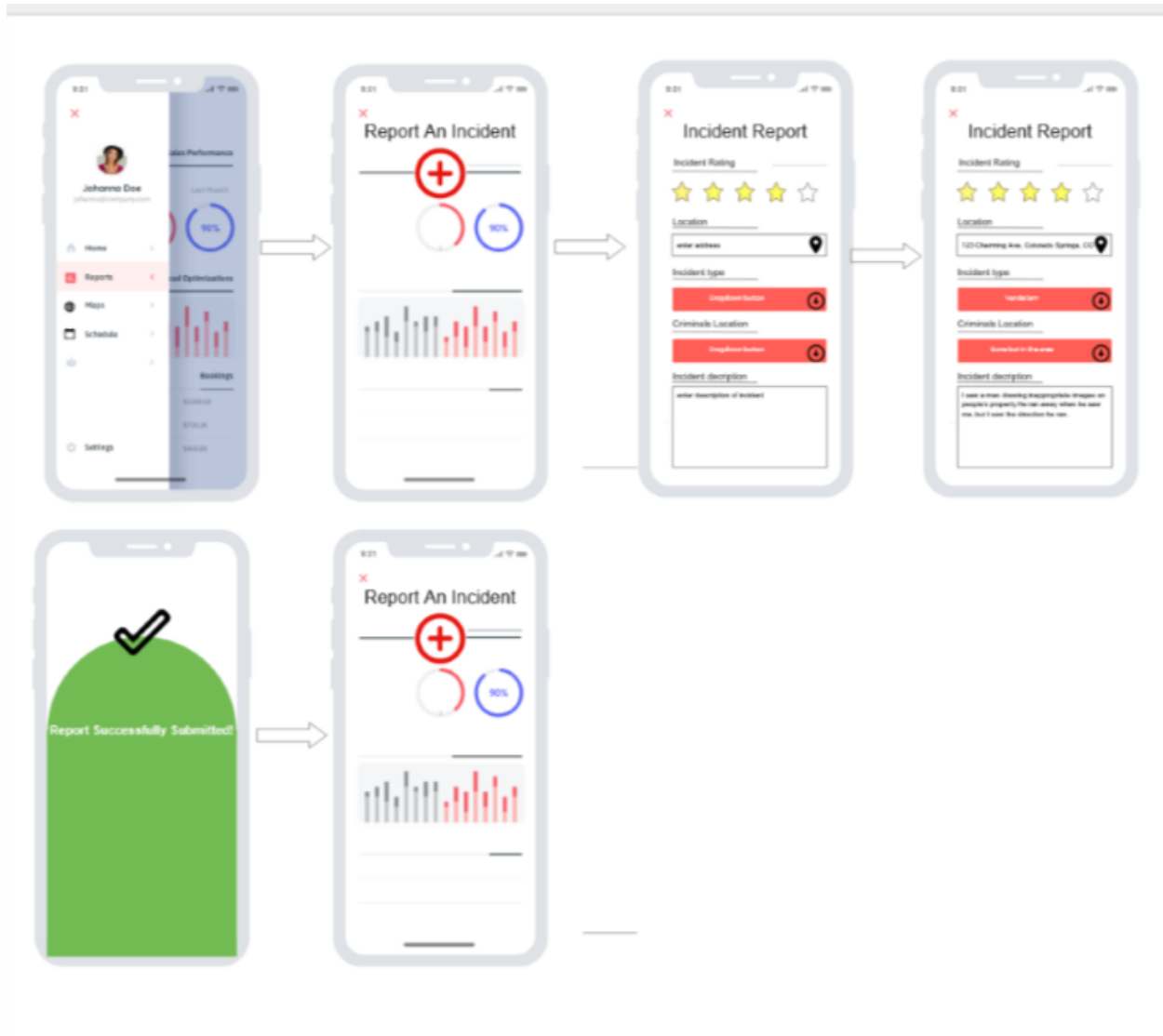
- The system shall allow users to view a calendar of the schedule
- The system shall display a calendar icon within the navigation menu
- The user shall be able to select this icon to view a schedule
- The calendar shall show a monthly calendar of the current month
- Each date on the calendar shall show a user assigned to neighborhood watch during that day
- A user shall be able to select a date
- Upon selection, the system shall display user information for the users assigned
- The system shall allow a 'member' user to claim a shift
- The system shall display a 'claim a shift' button under the calendar
- If the user is not a 'member' user class, the claim a shift button shall not be visible
- The user shall be able to input a date from a calendar input
- The user shall be able to input a time from a time input
- The system shall verify that the date and time selected is open
- The system shall display an error if the date and time is not available
- The user can confirm the acceptance of the shift if there are no errors
- The schedule system shall use the following units:
  - Hours for timing
  - mm/dd for dates

### ***3.2.1.1 The system shall allow users to make an incident report***

Summary:

- A. **Sequence of operations** – The user shall view the following when attempting to create an incident report.
- a) The user shall open the application.
  - b) The user shall open the side menu.
  - c) The user shall select Reports.
  - d) The user shall press the Report an Incident button located at the top of the screen of the reports page.
  - e) The incident report form shall pop up.
  - f) The user may fill out the following in any order:
    - i. **\*Required\*** Location. This shall be the location of the incident in question. The user may select the maps button to upload their current location.
    - ii. **\*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. \*\*\***
    - iii. **\*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.
    - iv. **\*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”
    - v. **\*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.
  - g) The user may then press the Submit Incident button.
  - h) The user shall be taken to a screen which says report successfully submitted.
  - i) The user shall be returned to the Reports screen.





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

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

### C. Abnormal Input and System Responses

#### a) Overflow

- i. 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".

- ii. 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"
- b) 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"

The image displays three sequential screenshots of a mobile application's 'Incident Report' form. Each screen shows a red 'X' icon in the top left corner, indicating an error. The form includes fields for 'Incident Rating' (five stars), 'Location' (with a map icon), 'Incident type' (with a dropdown menu), 'Criminals Location' (with a dropdown menu), and 'Incident description' (with a text area). The first screenshot shows the 'Location' field with the address '123 Charming Ave, Colorado Springs, CO' and a red error message 'Invalid location please select location from google maps'. The second screenshot shows the 'Incident type' field with the option 'Vandalism' selected and a red error message 'Invalid location please select location from google maps'. The third screenshot shows the 'Incident description' field with a red error message 'Max Length reached 500/500'.

### c) Communication Facilities

- i. 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.
- ii. Error Handling and Recovery
- iii. 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**

- a) The report shall be added to the database in given format.
- b) The report shall generate an incident notification for neighborhood watch.
- c) The report shall generate a new mark on the map.
- d) The report shall be added to the general statistics information page.

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

Summary:

##### **A. Sequence of operations**

- a. Neighborhood watch members get notifications on or about their shift.  
Specifications of the notification are listed below
  - i. The notification shall contain a written description (could be blank) of the incident
  - ii. The notification shall contain a user selected type of incident
  - iii. The notification shall contain the severity, a rating from 1 to 5 stars of the incident
  - iv. The notification shall contain the location of the incident
  - v. The notification shall contain information about perpetrator location
  - vi.
- b. 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.
- c. 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.
- d. 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
- e. The system shall retain the notification for the duration of a shift period if a user does not close the notification.

##### **B. Validity Checks**

- a. The system shall only send notifications to members during their "shift"
- b. 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**

- a. 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.
- b. 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**

### ***3.2.1.3 The system shall allow users to sign up for an account***

1. The system shall display input fields for login
1. The system shall display an email input field
1. The system shall verify the input is a valid email
2. The system shall display a password field
1. The system shall obfuscate the input
3. The system shall display a login button
1. If selected by the user, the system shall verify the provided user credentials
1. If the system verifies credentials, the system shall create a new user session
4. The system shall display a 'create account' button
2. The system shall display a create account page when 'create account' button is selected by user
1. The system shall display an email input field
1. The system shall verify the input is a valid email address
2. The system shall display a first name input field
1. The system shall verify the input is non-empty
3. The system shall display a last name input field
1. The system shall verify the input is non-empty
4. The system shall display a phone number input field
1. The system shall verify the input is a valid phone number
5. The system shall display an address input field
1. The system shall verify the input is a valid address
6. The system shall display a password field
1. The system shall obfuscate the input
7. The system shall display a 'password verification' input field
1. The system shall verify the passwords match between fields
1. If passwords do not match, the system shall display a red box around the password field and a message stating the passwords do not match.
2. The system shall verify passwords meet security requirements
1. The password must contain 7 or more characters
2. The password must contain at least one number
3. If the password field does not meet requirements, the system shall display a red box around the password field and a message stating the password requirements.
8. The system shall display a dropdown menu prompting to 'join the watch'
1. The menu shall contain 'Join'
1. When the user selects this option, the system shall mark the account as 'member'
2. The menu shall contain 'no'
1. When the user selects this option, the system shall mark the account as 'viewer'
9. The system shall display a 'create account button'
1. Upon selection, the system shall verify input data
1. The system shall verify the email does not have an account created yet
2. The system shall verify the password field meets password requirements
2. If input data is valid, the system shall create a user account

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

1. Areas in the neighborhood

1. The system shall display the name of the safest streets in a neighborhood
2. The system shall indicate the safest streets in a neighborhood
3. The system shall indicate the dangerous streets in a neighborhood

2. Weekly, monthly, and annual report on crimes

1. The system shall show weekly, monthly, and annual number of violent crimes for the entire neighborhood
2. The system shall show weekly, monthly, and annual amount of property and nonviolent crimes
3. The system shall indicate weekly, monthly, and annual, the street with the greatest number of violent plus nonviolent incidents in the neighborhood

3. Intensity of crimes in the neighborhood

1. The system shall indicate the violent and nonviolent crime that has the most occurrence.
2. The system shall indicate the locations which types of crimes occur most frequently.

4. Graphs

1. The system shall provide a tool which automatically updates as data about crime is inputted in the system.
2. The system shall provide a graph which shows the total crimes in the neighborhood
3. The system shall provide a graph that shows the relationship between violent crimes and areas in the neighborhood

4. The system shall provide a graph that shows the relationship between non-violent and property crimes in the neighborhood
5. The system shall provide graphs that shows the status of types of crimes in the neighborhood.

5. The System shall allow users to communicate to one another with a discussion board

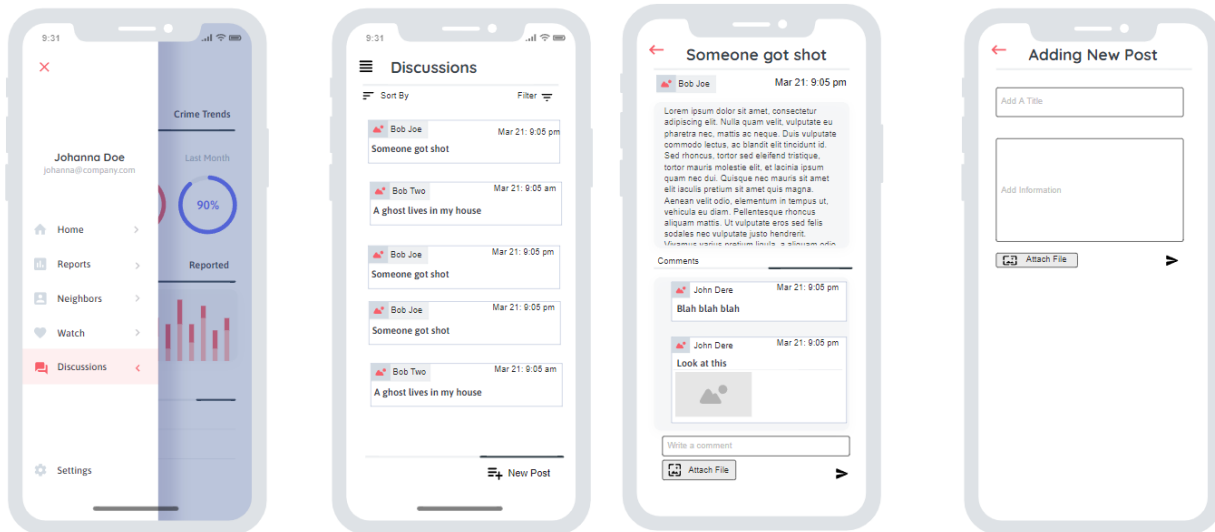
1. The system shall take text, documents, pictures, or other files and post them for others to view and comment on.

1. The system shall display the name of the user who sent the data to the discussion board
2. The system shall display the time the data was posted
3. The system shall allow comments and messages to tag existing posts
4. The system shall check input files and deny the posting of files that are too large

2. The system shall hold and display existing messages for others in the group to view

1. Once given a new entry, the system shall take in the input from the user
2. The system then verifies the size of the data is acceptable to be pushed to the main servers
3. The system will then either give the user an error if the file size is too large or will send the data to the server to be posted on the discussion board
4. Once the data is given to the discussion board it shall be visible and interactable for all members

3. In response to unusual inputs. The system shall alert the user about the unusual input and prevent it from being sent to main servers



6. The system shall allow users to view crime incidents in the neighborhood through an embedded map.
  1. The system shall display crimes categorized by the type of crime
    - 1.The system shall allow the user to filter the map by date
    - 2.The system shall allow the user to display crimes within a user defined radius
  2. The system shall have a popup user interface that displays information on the map
    - 1.The system shall display the date and time of the incident
    - 2.The system shall include the proximity of a reported incident to your home

3. This system shall display the classification of the incidents on the map popup
4. The system shall display a description of the event on the map popup
3. The system shall highlight new incidents since the last log in to the application
  1. The pin noting the map location will be a unique color
4. The system shall allow users to update the map with a new incident
  1. The system shall allow a user to report the date and time of the incident
  2. The system shall allow a user to report the classification of the incident
  3. The system shall allow a user to enter a description of the event
  4. The system shall allow a user to the system shall send a notification when a new incident is added to the map
5. Upon submission the entry will be validated
  1. In response to unusual inputs. The system shall alert the user about the unusual input and prevent it from being applied to the map
  2. The phone number of the user will authenticate before applying the incident to the map





Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as “shall” statements starting with “The system shall...”

These include

1. Validity checks on the inputs
2. Exact sequence of operations
3. Responses to abnormal situations, including
  1. Overflow
  2. Communication facilities
  3. Error handling and recovery
4. Effect of parameters
5. Relationship of outputs to inputs, including

1. Input/output sequences
2. Formulas for input to output conversion

It may be appropriate to partition the functional requirements into subfunctions or subprocesses. This does not imply that the software design will also be partitioned that way.

### **3.3 Performance requirements**

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:

1. The number of terminals to be supported;
2. 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:

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

### **3.5 Design constraints**

This should specify design constraints that can be imposed by other standards, hardware limitations, etc.

### **3.5.1 Standards compliance**

This subsection should specify the requirements derived from existing standards or regulations. They may include the following:

1. Report format;
2. Data naming;
3. Accounting procedures;
4. Audit tracing.

For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

## **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**

This should specify the factors required to establish the required reliability of the software system at time of delivery.

### **3.6.2 Availability**

This should specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart.

### **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

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

### **3.6.4 Maintainability**

This should specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices.

### **3.6.5 Portability**

This should specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include the following:

1. Percentage of components with host-dependent code;
2. Percentage of code that is host dependent;
3. Use of a proven portable language;
4. Use of a particular compiler or language subset;
5. Use of a particular operating system.

## **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**

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency. When organizing this section by mode, the outline in A.1 or A.2 should be used. The choice depends on whether interfaces and performance are dependent on mode.

### **3.7.2 User class**

1. The system shall provide (2) user classes
  1. The system shall have a 'member' user class
    1. Member users shall be able to take shifts on the neighborhood watch calendar
  2. The system shall have a 'viewer' user class
  3. All users shall be able to view and add incidents
  4. All users shall be able to view neighborhood watch schedule
  5. 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**

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;

1. Index;
2. 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

1. Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
2. Supporting or background information that can help the readers of the SRS;
3. A description of the problems to be solved by the software;
4. 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.

