**Software Requirements Specification**

By

**Murphy's Disciples**

|  |  |
| --- | --- |
| **NAME** | **CWID** |
| Ankit Chachada | 802973818 |
| Vandankumar Patel | 890321094 |
| Victor Wei | 891914996 |
| Sirisha Dumpala | 893537035 |

CPSC 541: Standards and Requirements, Section 1, Fall 2015

College of Engineering and Computer Science

California State University, Fullerton

Instructor: Hugh D. Greenbaum

December 7, 2015



Table of Contents

1 Introduction 4

1.1 System Description 4

1.2 Document Scope 4

1.3 Overview 4

2 Product Description 5

2.1 Product Perspective 5

2.2 Product Functions 6

2.2.1 Base Station Radio 7

2.2.2 Status Manager 8

2.2.3 Terminal Manager 9

2.2.4 Map Manager 12

2.2.5 Account Manager 13

2.2.6 Log Manager 14

2.2.7 Display Manager 16

3 Behavioral Requirements 17

3.1 Base Station Radio 17

3.1.1 Receive Update 17

3.1.2 Receive Encoded Message 18

3.1.3 Receive Voice Message 18

3.1.4 Route Voice Message 18

3.2 Status Manager 18

3.2.1 Decode Message 18

3.2.2 Update Message 19

3.2.3 Check Last Update 19

3.2.4 Get Unit Information 19

3.3 Terminal Manager 19

3.3.1 Unit Data Maintenance 19

3.3.2 Get Incident Information 20

3.3.3 Assign Response Units 20

3.3.4 Display Nearest Units 20

3.3.5 Current Unit Display 20

3.3.6 Unit Unavailability Notice 21

3.3.7 Restore Display 21

3.3.8 Decode Voice Message 21

3.3.9 Define Range and Center Point 21

3.3.10 Display Range and Center Point 21

3.4 Map Manager 21

3.4.1 Get Map Feed 21

3.4.2 Add Incident 21

3.5 Account Manager 22

3.5.1 Edit Account 22

3.5.2 Edit Account Information 22

3.5.3 Credential Checking 23

3.5.4 Disable Account 24

3.5.5 Accept Valid Request 24

3.5.6 Re-enable Account 24

3.5.7 Generate Report 24

3.6 Log Manager 24

3.6.1 Generate Communications Log 24

3.6.2 Generate Dispatch Log 25

3.6.3 Generate Audit Trail 25

3.6.4 Manage Online Logs 25

3.6.5 Manage Offline Logs 26

3.7 Display Manager 26

3.7.1 Generate Unit View 26

3.7.2 Display Screen 26

4 Non-behavioral Requirements 27

4.1 Availability 27

4.2 Compliance 27

4.3 Reliability 27

4.4 Maintainability 27

4.5 Recovery 27

4.6 Security 27

5 Trace Table 28

6 Data Model 30

6.1 Data Dictionary 30

6.2 Class Diagrams 54

7 References 64

# Introduction

## System Description

This is an Emergency Services Locator and Messaging System, which consist of handheld and vehicle-mounted radios; base station radios, computer processing equipment, and operator displays and controls. This system allows operators at the base station to send and receive voice messages to/from the mobile radios, to receive coded messages from the mobile radios, to locate each mobile radio and its associated unit on a computer-displayed local map, and to dispatch appropriate units to emergency scenes based upon vehicle unit attributes.

## Document Scope

This document will help the users in understanding the functional and non-functional requirements of this software application. All the features that are in the scope of the project are clearly listed for the users. This will also help the software development team to build software that fulfills all the requirements mentioned in this document.

## Overview

The following is a description of how the remaining sections of this Software Requirements Specification document are organized:

Section 2: This section provides a high level description of the software application, including product perspective and product functions, design constraints if any. It also contains the context diagram and level -1 decomposition of the data flow diagram.

Section 3: This section provides a detailed description of functional requirements of the software application.

Section 4: This section provides a detailed description of f non-functional requirements of the software application.

Section 5: This section contains the Trace Back Table to help identify the origin of each requirement listed in this document, evaluate and incorporate the changes in requirements.

Section 6: This section includes the data flow diagrams and class diagrams of the system.

# Product Description

## Product Perspective

**Context Diagram**

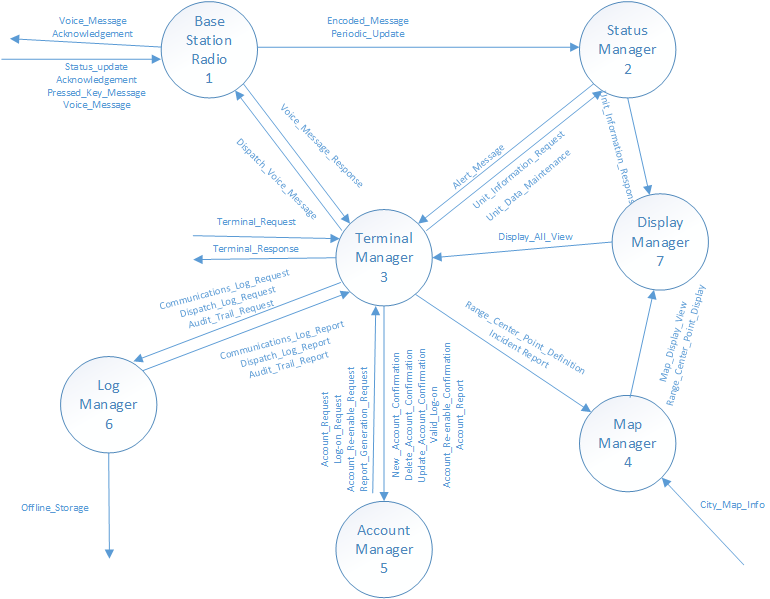


**Description of the relationship of the product to the external interfaces:**

The Emergency Services Locator and Messaging System is an application system that provides locating and dispatching functions for emergency vehicles.  The Emergency Response System is connected to a Base Station Radio, a Terminal System, Offline Storage, and the City Map Feed.  The Base Station Radio is the external interface that is responsible for routing messages between the Emergency Response System and external mobile radios on emergency vehicles.  The Terminal System is the external interface that receives user inputs.  The Offline Storage external interface is responsible for storing all data (logs and audit trail) past 48 hours.  Lastly, the City Map Feed is connected to the Emergency Response System and is responsible for providing up to date city map information to the system.

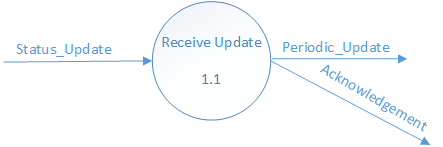
## Product Functions

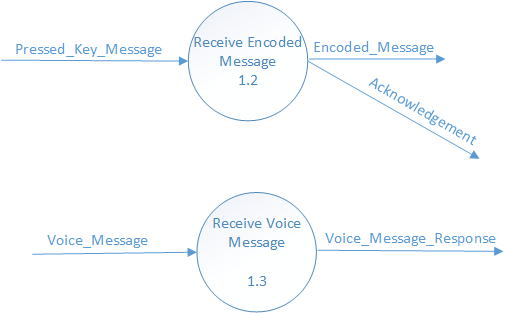
**Data Flow Diagram: Level-1 Decomposition**

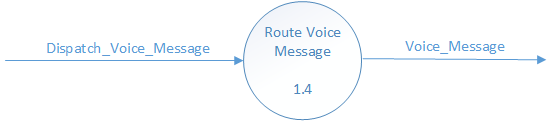


### Base Station Radio

Base Station Radio receives communications from the mobile radio, and sends back the acknowledgement upon the receipt of those communications. Base Station Radio processes these communications and generates messages, which can be used by Status Manager, and Terminal Manager.

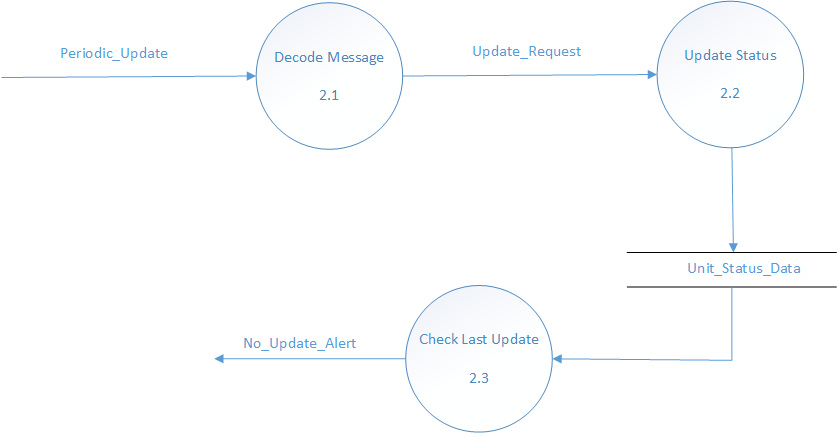


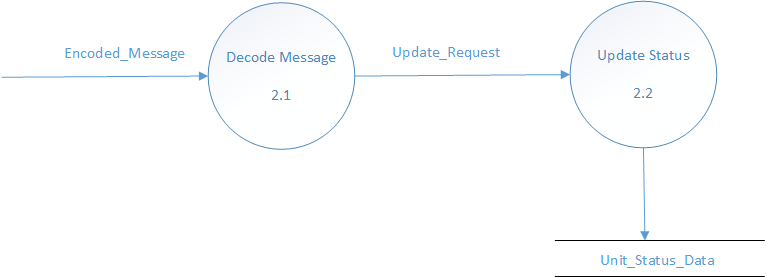


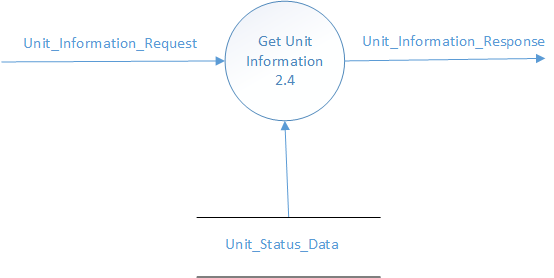


### Status Manager

Status Manager receives encoded messages from Base Station Operator. It decodes these messages and update the status of the associated mobile radio. Status manager generates an alert if no update has been received from mobile radio for more than 5 minutes. Status manager generates the unit-information requested by terminal manager.

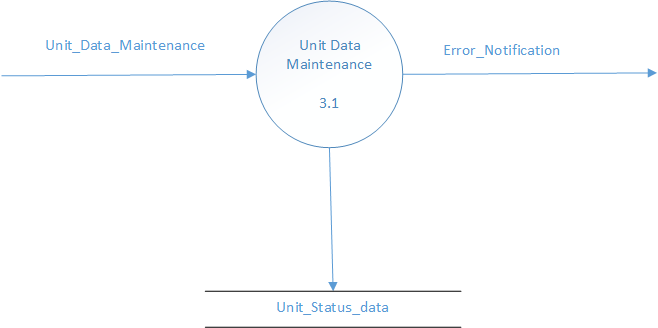


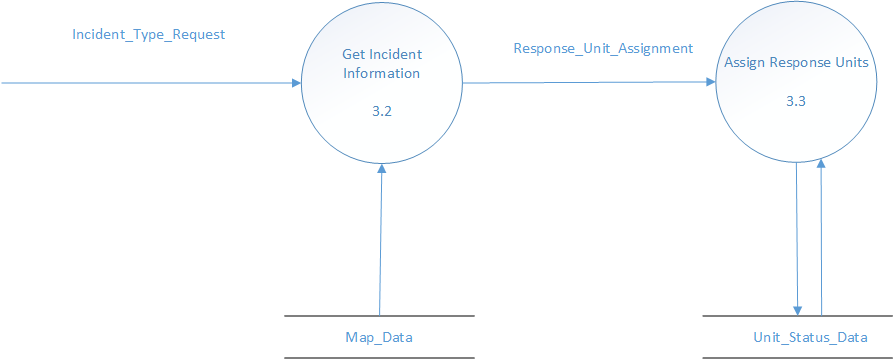


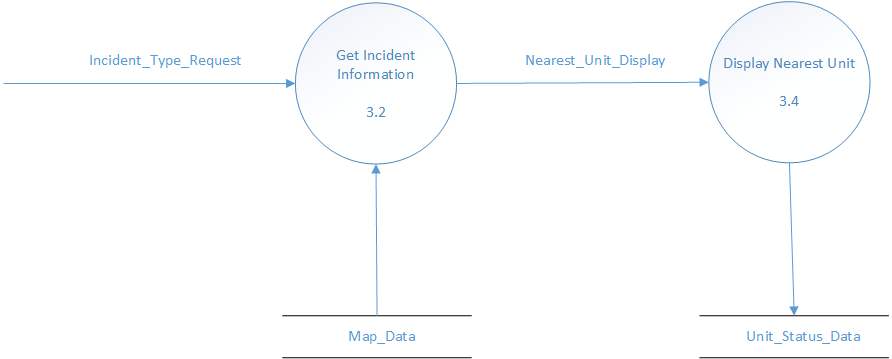


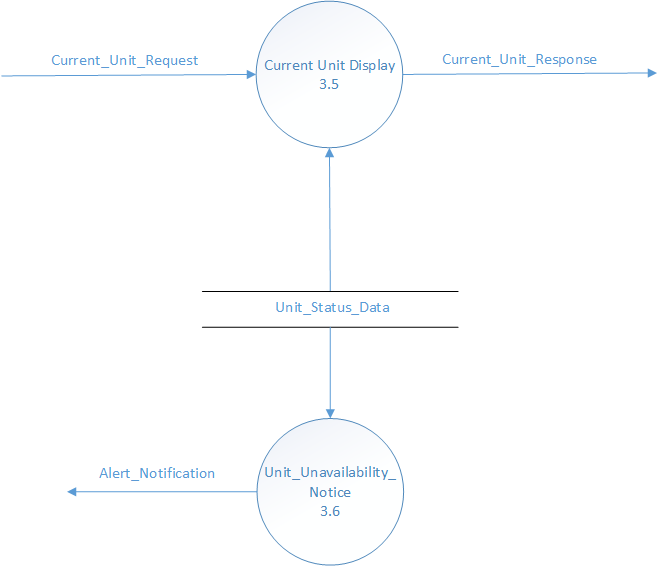
### Terminal Manager

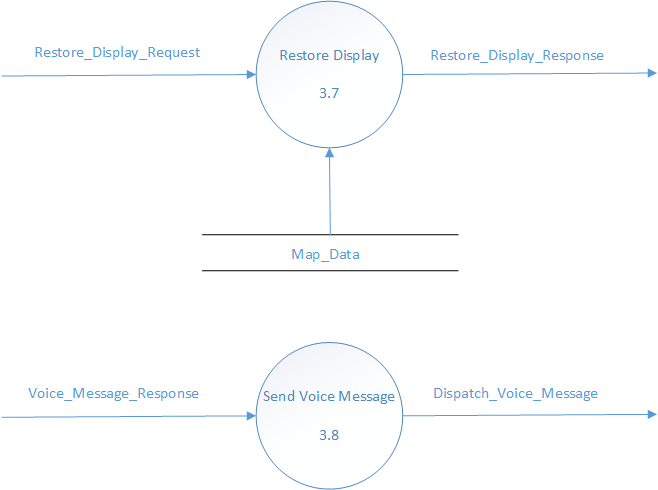
Terminal manager is the central part of the system. It is responsible for managing the operations such as encoding and decoding voice message response, account information, log information and status information. One of its important function is to display all the geographic data.

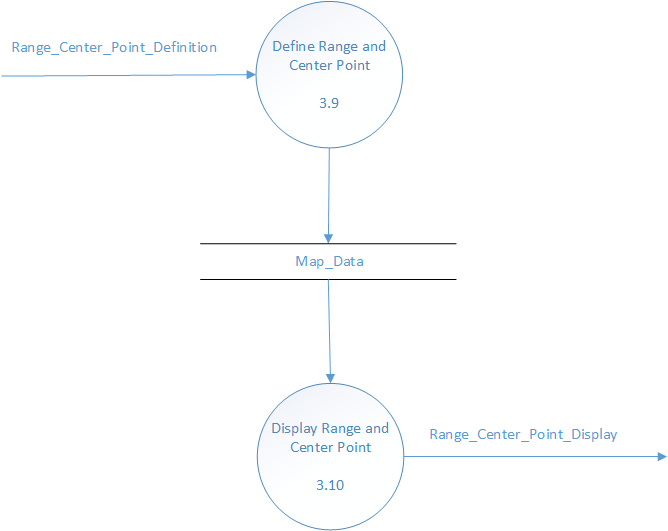






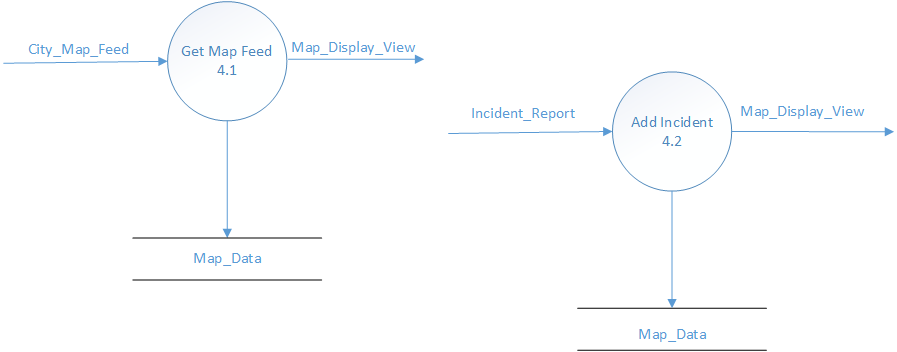






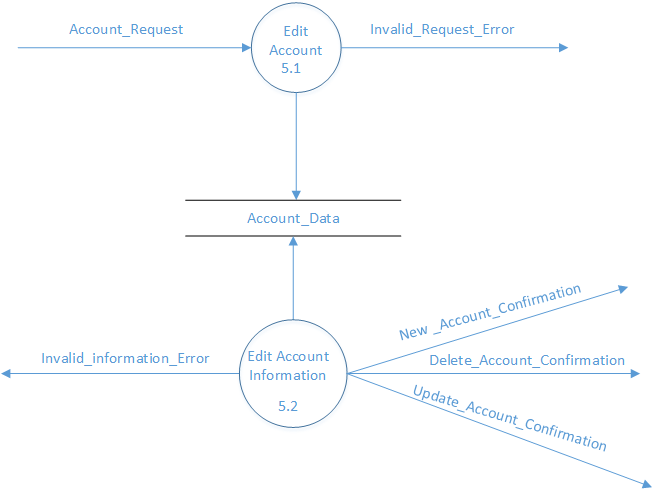
### Map Manager

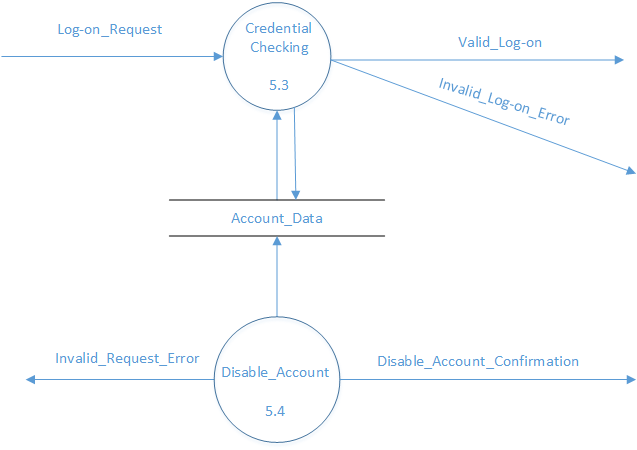
The Map Manager is responsible for providing a map data to display.  The Map Manager is connected to the external interface, City\_Map\_Feed, which receives up to date city map information.

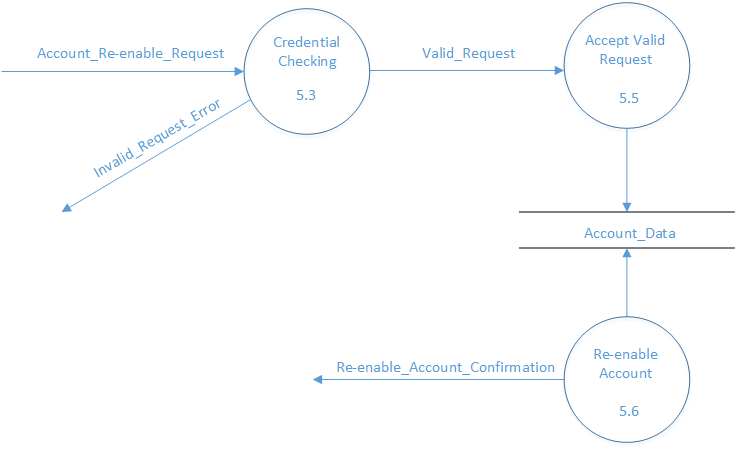


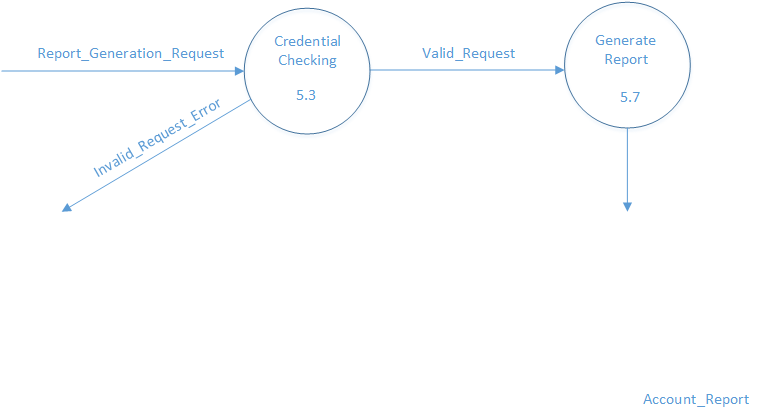
### Account Manager

This function is intended to manage all user accounts in the system.  Activities in the scope of this function include credential checking and providing access to information based on the permissions assigned to each user.



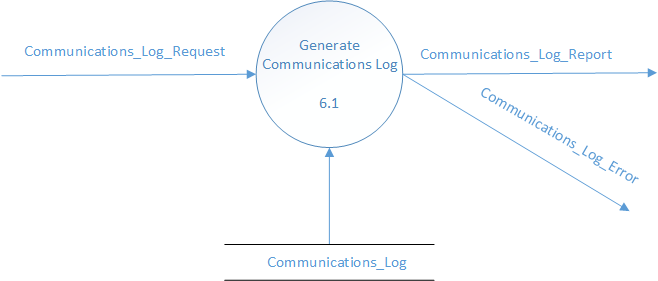


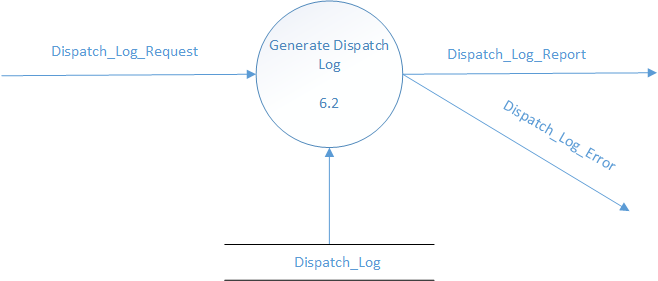


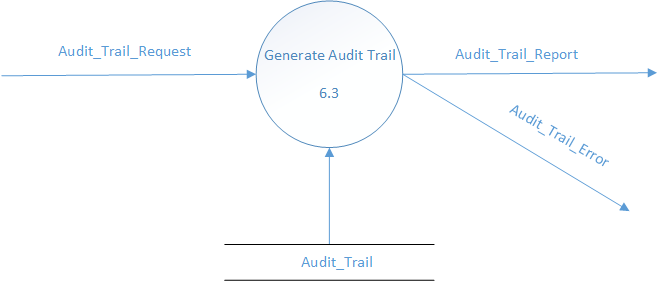


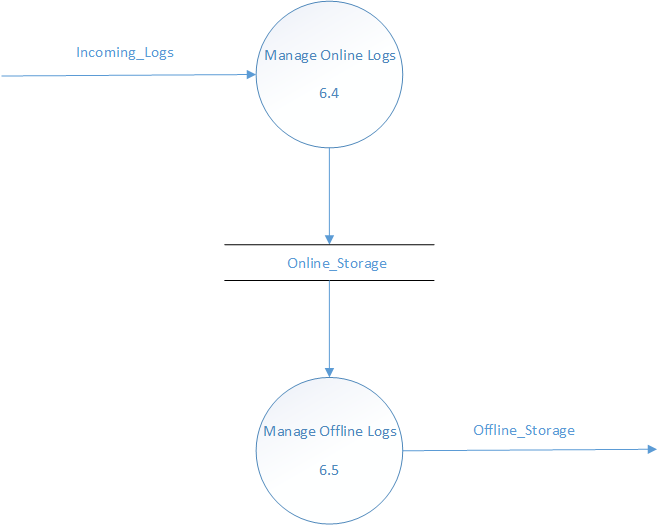
### Log Manager

Log Manager receives requests to generate different logs. Log Manager checks if the requested log exits in the database. If the requested log exists, it generates that log report. If the requested log doesn’t exist, it generates an error.



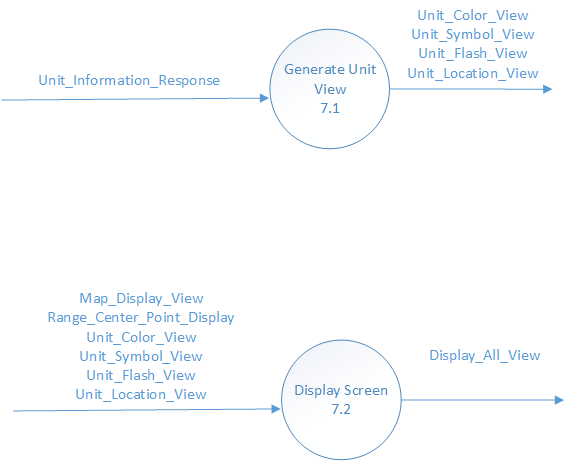






### Display Manager

The Display Manager is responsible for displaying all information to the user.  The Display Manager is connected with the Status Manager and Map Manager to provide the user a real time map of the city and the locations of emergency vehicles.



# Behavioral Requirements

## Base Station Radio

See Section 2.2.1

### Receive Update

Upon receipt of Status\_Update, the product performs the following:

* + 1. The product **SHALL [0001]** generate Periodic\_Update.
    2. The product **SHALL [0002]** create the Acknowledgement to send it to mobile radio.

### Receive Encoded Message

Upon receipt of Pressed\_Key\_Message, the product performs the following:

1. The product **SHALL [0004]** generate Encoded\_Message.
2. The product **SHALL [0005]** create the Acknowledgement to send it to mobile radio.

### Receive Voice Message

Upon receipt of Voice\_Message, the product **SHALL** **[0006]** generate Voice\_Message\_Response

### Route Voice Message

Upon receipt of Dispatch\_Voice\_Message, the product performs the following:

* + 1. The product **SHALL [0095]** find the radio channel assigned to the group(s) for which the operator is responsible.
    2. The product **SHALL [0096]** generate Voice\_Message.
    3. The product **SHALL [0097]** routeVoice\_Message to the radio channel found in 3.1.4.a.

## Status Manager

See Section 2.2.2

### Decode Message

Upon receipt of Periodic\_Update, the product performs the following:

1. The product **SHALL [0008]** decode the message to read and extract the most recently recorded location, operational status of the associated unit, and time of last update.
2. The product **SHALL [0009]** generate the Update\_Request using information of decoded message.

### Update Message

Upon receipt of the Update\_Request, the product **SHALL [0010]** update the Unit\_Status\_Data Table.

### Check Last Update

1. The product **SHALL [0011]** keep checking the time of last update every 10 seconds.
2. The product **SHALL [0012]** generate No\_Update\_Alert if no update has been received for more than 5 minutes for any mobile radio.

### Get Unit Information

Upon the receipt of Unit\_Information\_Request, the product performs the following:

1. The product **SHALL [0018]** check if the requested information exists in the Unit\_Status\_Data table.
2. If the information exists in the Unit\_Status\_Data table, the product **SHALL** **[0019]** generate Unit\_Information\_Response using that information.
3. If the information doesn’t exist in the Unit\_Status\_Data table, the product **SHALL [0089]** generate Unit\_Information\_Error on terminal to notify the operator.

## Terminal Manager

See Section 2.2.3

### Unit Data Maintenance

1. Upon base station operator’s request, the product **SHALL [0020]** display a form to add Unit\_Type, Unit\_Call\_Sign and Operational\_Status.
2. Upon adding these values associated with the radio, the product **SHALL [0021]** store the Unit\_ Type, Unit\_Call\_Sign and Operational\_Status in Unit\_Status\_Data table.
3. The product **SHALL [0022]** validate the format of the Unit\_Call\_Sign.
4. The format of Unit\_Call\_Sign **SHALL [0023]** be the combination of Unit\_Type and a number. For instance “Ambulance-46” where Ambulance is Unit\_Type and 46 is a number.
5. The product **SHALL [0024]** generate error notification if the value entered is incorrect

### Get Incident Information

Upon receipt of Incident\_Type\_Request, the product **SHALL [0026]** retrieve the incident type and location from Map\_Data table.

### Assign Response Units

1. See section 3.3.2
2. Upon receipt of Response\_Unit\_Assignment, the product **SHALL [0027]** be able to assign.
3. The product **SHALL [0028]** dispatch appropriate response units based on incident type.
4. If the primary response unit is unavailable then product **SHALL [0029]** dispatch secondary response unit.
5. If the secondary response unit is unavailable then product **SHALL [0030]** dispatch tertiary appropriate response units viz. primary, secondary and tertiary response unit.

### Display Nearest Units

Upon receipt of Nearest\_Unit\_Display request, the product **SHALL [0032]** display nearest units available based on the incident location (See section 3.3.2).

### Current Unit Display

Upon receipt of Current\_Unit\_Request, the product **SHALL [0033]** retrieve current units available from Unit\_Status\_Data table.

### Unit Unavailability Notice

The product **SHALL [0034]** notify the base station operator if appropriate units are unavailable.

### Restore Display

Upon receipt of Restore\_Display\_Request, the product **SHALL [0035]** restore the display to default center point.

### Decode Voice Message

Upon receipt of Voice\_Message\_Response, the product **SHALL [0036]** be able to decode the voice message.

### Define Range and Center Point

Upon receipt of Range\_Center\_Point\_Definition request, the product **SHALL [0038]** store the map parameters such as range\_scale and center point in Map\_Data table.

### Display Range and Center Point

The product **SHALL [0040]** retrieve Range\_Center\_Point\_Display response from Map\_Data table.

## Map Manager

See Section 2.2.4

### Get Map Feed

Upon receipt of City\_Map\_Feed, the product performs the following:

1. The product **SHALL [0045]** update Map\_Data table.
2. The product **SHALL [0046]** generate Map\_Display\_View.

### Add Incident

Upon receipt of Incident\_Report, the product performs the following:

1. The product **SHALL** **[0047]** update Map\_data table with Incident\_Report.Incident\_Type.
2. The product **SHALL [0048]** convert Incident\_Report.Incident\_Type into Map\_Display\_View.Incident\_Type.

## Account Manager

See Section 2.2.5

### Edit Account

Upon receipt of an Account\_Request, the product performs the following:

1. The product **SHALL [0050]** validate the request by checking the credentials of the user.
2. The product **SHALL [0052]** log the validation of the request in Account\_Data store.
3. If the request is invalid, the product **SHALL [0051]** generate an Invalid\_Request\_Error message.

### Edit Account Information

Upon validating the user and the request made by them, the product performs the following:

1. The product **SHALL [0064]** identify the type of request.
2. When he user makes a ‘New account request’, the product **SHALL [0054]** display a form for user to enter the new account information.
3. The product **SHALL [0056]** generate a New\_Account\_Confirmation message to indicate the successful creation of a new account.
4. When the user makes a ‘Delete account request’, the product **SHALL [0058]** display a form for user to enter the required information to delete the account.
5. The product **SHALL [0060]** generate a Delete\_Account\_Confirmation message to indicate the successful deletion of the selected account.
6. When the user makes a ‘Update account request’, the product **SHALL [0061]** display the selected account.
7. The product **SHALL [0063]** generate a Update\_Account\_Confirmation message to indicate the successful updating of the selected account.
8. If invalid data is entered in any form, the product **SHALL [0055]** generate an Invalid\_Information\_Error message to notify the user.
9. The product **SHALL [0057]** log the edits made to the account in Account\_Data store.

### Credential Checking

Upon the receipt of a Log-on\_Request, the product performs the following:

1. The product **SHALL [0065]** display a login form where the user can enter the access credentials.
2. The product **SHALL [0107]** store the user password in encrypted form.
3. The product **SHALL [0066]** validate the credentials for the user to login.
4. Upon Valid\_Log-on, the product **SHALL [0067]** display the appropriate homepage according to the authorization provided to the user.
5. The product **SHALL [0068]** generate an Invalid\_Log-on\_Error message indicating an invalid credentials by the user.
6. When the user makes ‘Account\_Re-enable\_Request’, the product **SHALL [0069]** validate the request by checking the user credentials for permissions.
7. When the user makes ‘Report\_Generation\_Request’, the product **SHALL [0104]** validate the request by checking the user credentials for permissions.
8. If invalid request is made, the product **SHALL [0070]** generate an Invalid\_Request\_Error message to notify the user.

### Disable Account

If an invalid request for log-on is made, the product performs the following:

1. The product **SHALL [0072]** keep a count of number of consecutive invalid log-on attempts by a single user.
2. The product **SHALL [0073]** disable the user’s account after three consecutive invalid log-on attempts.
3. The product **SHALL [0074]** log the information related to disabling the account in account data store.

### Accept Valid Request

If a Valid\_Request is made, the product **SHALL [0103]** log the data in Account\_Data store.

### Re-enable Account

If the account is re-enabled, the product **SHALL [0071]** generate Re-enable\_Account\_Confirmation message indicating successful completion of the request.

### Generate Report

Upon receiving a Valid\_Request for a report, the product **SHALL [0105]** generate a Account\_Report containing the requested information.

## Log Manager

See Section 2.2.6

### Generate Communications Log

Upon the receipt of Communications\_Log\_Request, the product performs the following:

1. The product **SHALL [0075]** check if the requested log exists in Communications\_Log table.
2. If the requested log exists in the Communications\_Log table, the product **SHALL [0076]** generate Communications\_Log\_Report using that log information.
3. If the requested log doesn’t exist in the Communications\_Log table, the product **SHALL [0090]** generate Communications\_Log\_Error on terminal to notify the operator.

### Generate Dispatch Log

Upon the receipt of Dispatch\_Log\_Request, the product performs the following:

1. The product **SHALL [0078]** check if the requested log exists in the Dispatch\_Log table.
2. If the requested log exists in the Dispatch\_Log table, the product **SHALL [0079]** generate Dispatch\_Log\_Report using that log information.
3. If the requested log doesn’t exist in the Dispatch\_Log table, the product **SHALL [0091]** generate Dispatch\_Log\_Error on terminal to notify the operator.

### Generate Audit Trail

Upon the receipt of Audit\_Trail\_Request, the product performs the following:

1. The product **SHALL [0080]** check if the requested log exists in the Audit\_Trail table.
2. If the requested log exists in the Audit\_Trail table, the product **SHALL [0081]** generate Audit\_Trail\_Report using that log information.
3. If the requested log doesn’t exist in the Audit\_Trail table, the product **SHALL [0092]** generate Audit\_Trail\_Error on terminal to notify the operator.

### Manage Online Logs

Upon the receipt of Incoming Logs, the product **SHALL [0082]** store all the logs in the Online Storage table for the most recent 48 hours.

### Manage Offline Logs

The product **SHALL [0084]** move all the logs, which are older then 48 hours, to the Offline\_Storage table.

## Display Manager

See Section 2.2.7

### Generate Unit View

1. Upon receipt of Unit\_Information\_Response, the product **SHALL [0085]** generate Unit\_Color\_View.
2. Upon receipt of Unit\_Information\_Response, the product **SHALL** **[0086]**generate Unit\_Symbol\_View
3. Upon receipt of Unit\_Information\_Response, the product **SHALL** **[0087]**generate Unit\_Flash\_View
4. Upon receipt of Unit\_Information\_Response, the product **SHALL** **[0088]**generate Unit\_Location\_View

### Display Screen

1. Upon receipt of Map\_Display\_View, the product **SHALL** **[0108]**generate Display\_All\_View from Map\_Display\_View attributes.
2. Upon receipt of Map\_Display\_View, the product **SHALL** **[0109]**convert Map\_Display\_View attributes into  Display\_All\_View.
3. The product **SHALL [0110]** generate Display\_All\_View from Unit\_Color\_View, Unit\_Symbol\_View, Unit\_Flash\_View, and Unit\_Location\_View.

# Non-behavioral Requirements

## Availability

1. The downtime for the product **SHALL** [0112] be no more than 53 minutes per year

## Compliance

1. The product **SHALL [0113]** be compliant with international standards for units.

## Reliability

1. The product **SHALL [0114]** not display any debugging messages or exceptions in error handling.
2. Response time for the product **SHALL** **[0115]** be less than one (1) second.
3. The product **SHALL [0116]** be designed to support on-line recording of sixty (60) hours of data.

## Maintainability

1. The product **SHALL [0117]** include full documentation to allow for future enhancements.

## Recovery

* 1. The product **SHALL [0118]** recover from its last backup within five minutes from crashing.
  2. The product **SHALL** **[0119]** recover from a disaster situation within 12 hours.

## Security

1. The product **SHALL** **[0121]** have a dedicated frequency line for all its communications.
2. The product **SHALL** **[0122]** be installed along side an anti virus software.

# Trace Table

|  |  |  |  |
| --- | --- | --- | --- |
| **System Requirement** | **Behavioral Requirement** | **Non-Behavioral Requirement** | **Test Type** |
| 0001 | 0001, 0008, 0009, 0010 |  | Inspection |
| 0002 | 0011, 0012 |  | Demonstration |
| 0003 | 0014, 0015, 0010, 0004 |  | Inspection |
| 0004 | 0010 |  | Analysis |
| 0005, 0031 | 0082 |  | Analysis |
| 0006, 0032 | 0084 |  | Analysis |
| 0007 | 0018,0019, 0020, 0021, 0022, 0023, 0024, 0089 |  | Demonstration |
| 0008 | 0036 |  | Inspection |
| 0009 | 0095, 0096, 0097 |  | Inspection |
| 0011 | 0045, 0046 |  | Demonstration |
| 0012 | 0088, 0086 |  | Demonstration |
| 0013 | 0085 |  | Demonstration |
| 0014 | 0087 |  | Demonstration |
| 0015 | 0038,  040 |  | Demonstration |
| 0016 | 0035 |  | Demonstration |
| 0017 | 0026, 0032 |  | Demonstration |
| 0018 | 0026, 0027, 0028, 0029, 0030 |  | Used |
| 0019 | 0032 |  | Demonstration |
| 0020 | 0033 |  | Used |
| 0021 | 0065, 0107, 0066, 0067, 0068, 0070 |  | Inspection |
| 0022 | 0050, 0051, 0064, 0054, 0056, 0058, 0060, 0061, 0063, 0055 |  | Inspection |
| 0023 | 0052 |  | Analysis |
| 0024 | 0107 |  | Analysis |
| 0025 | 0052, 0057, 0104, 0074, 0103, 0105 |  | Demonstration |
| 0026 | 0072, 0073 |  | Inspection |
| 0027 | 0069, 0071 |  | Inspection |
| 0028 | 0075, 0076, 0090 |  | Demonstration |
| 0029 | 0078, 0079, 0091 |  | Demonstration |
| 0030 | 0080, 0081, 0092 |  | Demonstration |
| 0033 | Refer to Class Diagram |  | Analysis |
| 0034 | 0002 |  | Inspection |
| 0035 | 0005 |  | Inspection |
| 0036 | Refer to Class Diagram |  | Analysis |
| 0037 | 0006 |  | Inspection |
|  |  | 01120112 | Inspection |
|  |  | 0113 | Analysis |
|  |  | 0114, 0115 | Demonstration |
|  |  | 0116 | Inspection |
|  |  | 0117 | Inspection |
|  |  | 0118 |  |
|  |  | 0119 |  |
|  |  | 0121 |  |
|  |  | 0122 |  |

# Data Model

## Data Dictionary

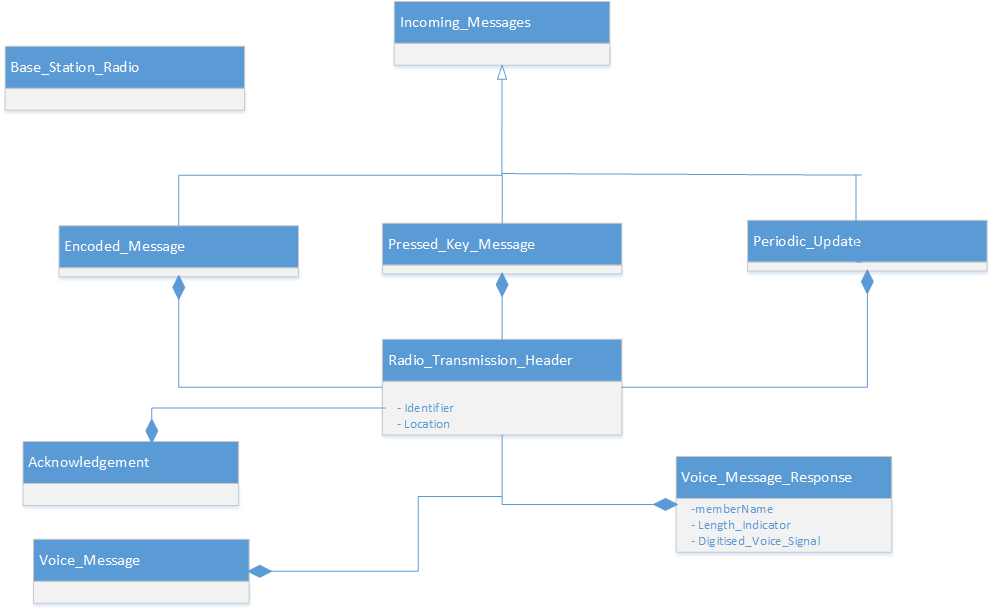
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Kind** | **Description** | **Range** | | **Units** |
| Unit\_Data\_Maintenance | Class | Defines unit data such as unit type, unit call-sign and operational status | N/A | | N/A |
| Unit\_Type | String | It stores type of vehicle associated with each radio | 1-250 | | ISO/IEC 10646 |
| Unit\_Call\_Sign | String | It stores the call sign assigned to a particular unit type | 1-250 | | ISO/IEC 10646 |
| Unit\_Operational\_Status | String | It stores the operational status of a particular unit | 1-250 | | ISO/IEC 10646 |
| Incident\_Type\_Request | Class | Pulls the information on type of incident from Map\_Data | N/A | | N/A |
| Response\_Unit\_Assignment | Class | Assigns response units such as primary, secondary and tertiary | N/A | | N/A |
| Nearest\_Unit\_Display | Class | Displays the information on nearest available units based on incident type from map data | N/A | | N/A |
| Range\_Center\_Point\_Definition | Class | Defines range and center point | N/A | | N/A |
| Range\_Center\_Point\_Display | Class | Displays range and center point for the incident | N/A | | N/A |
| Current\_Unit\_Request | Class | Request to display current unit statuses and current unit locations | N/A | | N/A |
| Current\_Unit\_Response | Class | Displays current unit statuses and current unit locations | N/A | | N/A |
| Alert\_Notification | Class | Alerts the operator if no appropriate units are determined | N/A | | N/A |
| Restore\_Display\_Request | Class | Request to display map to the default center point position | N/A | | N/A |
| Restore\_Display\_Response | Class | Displays the map with default center point position | N/A | | N/A |
| Encoded\_Voice\_Message | Class | It is a voice message with radio transmission header, length indicator and digitized voice signal | N/A | | N/A |
| Voice\_Message | Class | It is a decoded voice message | N/A | | N/A |
| Length\_Indicator | Time | Indicates the length of a voice message | >0 | | Milliseconds |
| Digitized\_Voice\_Signal | Long | It is a series of bits representing a digitized voice message | >0 | | Bits |
| Range\_Scale | Long | It is a ratio of a distance on the map | 1:40,000 to 1:90,000 | | Units |
| Center\_Point | Long | Calculates the center point based on the location of the incident | N/A | | N/A |
| Default\_Position | Long | It restores the map display to its default position | N/A | | N/A |
| Unit\_Information\_Request | Class | Defines unit information request from the Status Manager | N/A | | N/A |
| Unit\_ID | Double | Defines the ID for a specified mobile radio unit. | 1-10,000 | | Unit |
| Unit\_Location\_Request | Class | Defines request for unit location from the Status Manager | N/A | | N/A |
| Unit\_Type\_Request | Class | Defines request for the unit type from the Status Manager | N/A | | N/A |
| Unit\_Operational\_Status\_Request | Class | Defines request for the unit’s operational status from the Status Manager | N/A | | N/A |
| Unit\_Emergency\_Request | Class | Defines request for whether the unit is in an emergency from the Status Manager | N/A | | N/A |
| Unit\_Information\_Response | Class | Defines unit information response from Status Manager | N/A | | N/A |
| Unit\_Location\_View | Class | Contains the location of specified unit. | N/A | | N/A |
| Location | Class | Defines the location of an object on a map | N/A | | N/A |
| X\_Coordinate | Float | Defines the latitude location on a map | 180° E ↔ 180° W | | Degrees |
| Y\_Coordinate | Float | Defines the longitude location on a map | 90° N ↔ 90° S | | Degrees |
| Unit\_Symbol\_View | Class | Defines how the unit symbol looks according to the unit type. | N/A | | N/A |
| Unit\_Color\_View | Class | Defines what color the unit symbol is according to the unit’s operational status. | N/A | | N/A |
| Unit\_Flash\_View | Class | Defines whether the unit is flashing according to its emergency status. | N/A | | N/A |
| Display\_All\_View | Class | Defines what will be shown on the terminal display | N/A | | N/A |
| Map\_Display\_View | Class | Defines which map information will be displayed on the terminal display. | N/A | | N/A |
| Streets | Geospatial | Defines all street information that will be displayed on the terminal display. | N/A | | Units |
| Highways | Geospatial | Defines all highway information that will be displayed on the terminal display. | N/A | | Units |
| Fire\_Stations | Geospatial | Defines the location of all fire stations that are displayed on the map. | N/A | | Units |
| Police\_Stations | Geospatial | Defines the location of all police stations that are displayed on the map. | N/A | | Units |
| Hospitals | Geospatial | Defines the location of all hospitals that are displayed on the map. | N/A | | Units |
| Fire\_Hydrants | Geospatial | Defines the location of all fire hydrants that are displayed on the map. | N/A | | Units |
| Shut\_Off\_Valves | Geospatial | Defines the location of all shut-off valves that are displayed on the map. | N/A | | Units |
| Incident | Geospatial | Defines the location of the specified reported incident on the map. | N/A | | Units |
| Map\_Manager | Class | Responsible for managing the data flow to and from map information. | N/A | | N/A |
| Display\_Manager | Class | Responsible for managing what data is displayed on the screen. | N/A | | N/A |
| City\_Map\_Information | Class | Defines the map information taken from the city map feed | N/A | | N/A |
| Map\_Data | Data | Data store that contains information of all map objects | 0-50,000 | | Each |
| Incident\_Report | Class | Defines information for incident to be displayed on the map. | N/A | | N/A |
| Incident\_Type | Integer | Defines what type of incident – such as fire, robbery, medical. | 1-100 | | ISO/IEC 10646 |
| Status\_Manager | Class | Defines the system that manages the status of the units | N/A | N/A | |
| Radio\_Transmission\_Header | String | Each radio message (voice or data) is preceded by a header | 1-256 | ISO/IEC 10646 | |
| Message | String | It describes the prompt message displayed | 1-256 | ISO/IEC 10646 | |
| Identifier | String | It identifies the transmitting radio | 1-256 | ISO/IEC 10646 | |
| Time\_Interval | Integer | It stores the interval of periodic update | 1-5 | Minutes | |
| Encoded\_Message | Class | Defines vehicle-mounted radio’s transmission when any of one of the keys is pressed | N/A | N/A | |
| Pressed\_Key | String | It stores the key pressed on the mobile radio | 1-1 | ISO/IEC 10646 | |
| Update\_Request | Class | Defines the request to update the status of the unit | N/A | N/A | |
| No\_Update\_Alert | Class | Defines the alert to be generated if no updates is received from a mobile radio within 5 minutes | N/A | N/A | |
| Unit\_Status\_Data | Class | Defines the storage for status data of the unit | N/A | N/A | |
| Unit\_Location | Class | Defines the location of the unit | N/A | N/A | |
| Unit\_Operational\_Status | Class | Defines the operational status of the associated unit | N/A | N/A | |
| Status\_Type | Enum | It stores the valid operational statuses of a unit | N/A | N/A | |
| Unit\_Last\_Update | Class | Defines the time of last update of the radio | N/A | N/A | |
| Unit | Class | Defines the transmitting radio and associated unit | N/A | N/A | |
|  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time | DateTime | It stores the time of the last update received | N/A | Milliseconds |
| Log\_Request | Class | Defines the request for logs | N/A | N/A |
| Log\_Manager | Class | Defines the system that manages the logs | N/A | N/A |
| Communications\_Log\_Request | Class | Defines the request for communications log | N/A | N/A |
| Dispatch\_Log\_Request | Class | Define the request for dispatch log | N/A | N/A |
| Audit\_Trail\_Request | Class | Defines the request for audit trail | N/A | N/A |
| Account\_Report\_Request | Class | Defines the request for account information | N/A | N/A |
| Log\_Report | Class | Defines the response of the log request | N/A | N/A |
| Communications\_Log\_Report | Class | Defines the response of the request for communications log | N/A | N/A |
| Dispatch\_Log\_Report | Class | Defines the response of the request for dispatch log | N/A | N/A |
| Audit\_Trail\_Report | Class | Defines the response of the request for audit trail | N/A | N/A |
| Account\_Report | Class | Defines the response of the request for account information | N/A | N/A |
| Online\_Storage | Class | Defines the online storage for logs | N/A | N/A |
| Incoming\_Logs | Class | Defines the logs generated by log manager | N/A | N/A |
| Communications\_Log | Class | Defines the storage for Communications Logs | N/A | N/A |
| Dispatch\_Log | Class | Defines the storage for the Dispatch Log | N/A | N/A |
| Audit\_Trail | Class | Defines the storage for the audit trail | N/A | N/A |
| Account\_Data | Class | Defines the storage for Account Information | N/A | N/A |

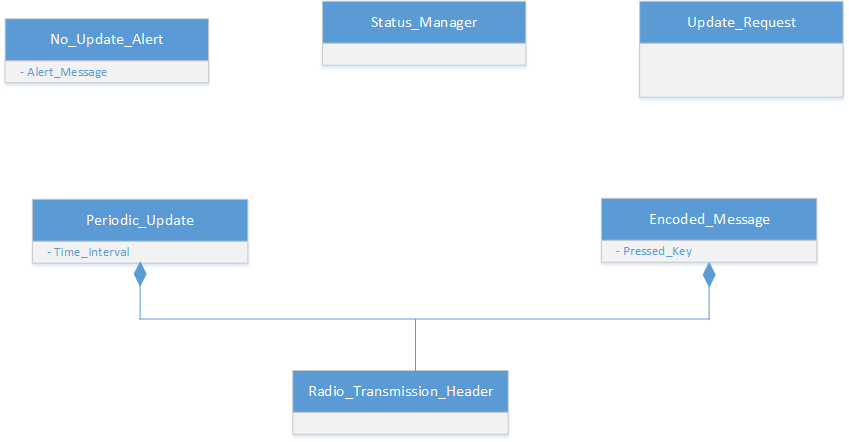
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transmission\_Duration | Class | Defines the duration for each transmission over a specific time period | N/A | N/A |
| Dispatch\_Request | Class | Defines each dispatch request | N/A | N/A |
| Time\_Period | Class | Defines the specific time-period | N/A | N/A |
| Results | Class | Defines the dispatch requests for a specific time-period |  |  |
| User\_ID | Class | Defines the user who made change to system tables | N/A | N/A |
| Time\_Of\_Change | Class | Defines the time of the change made to system table | N/A | N/A |
| Account | Class | Defines the account information of the users who use the system | N/A | N/A |
| Valid\_Log-on |  | Defines the valid Log- on into the system | N/A | N/A |
| Log-off |  | Defines the Logging off from the system | N/A | N/A |
| Unsuccessful\_Log-on |  | Defines the unsuccessful log in attempts to the system | N/A | N/A |
| Session\_Duration |  | Defines the duration of the log in session | N/A | N/A |
| Account\_Manager | Class | Responsible for managing user account information | N/A | N/A |
| Base Station Radio |  |  |  |  |
| Account\_Request | Class | Defines request for adding or making changes to user accounts | N/A | N/A |
| Invalid\_Request\_Error | Class | An error message prompting the user the | N/A | N/A |
| New\_Account\_Confirmation | Class | A prompt message indicating successful creation of a new account | N/A | N/A |
| Delete\_Account\_Confirmation | Class | A prompt message indicating successful deletion of an account | N/A | N/A |
| Update\_Account\_Confirmation | Class | A prompt message indicating successful updating of an account | N/A | N/A |
| Log-on\_Request | Class | Defines a request for logging into the system | N/A | N/A |
| Valid\_Log-on | Class | Defines successful log-on to the system | N/A | N/A |
| Account\_Data | Class | Data store that contains all the user account related information | N/A | N/A |
| Disable\_Account\_Confirmation | Class | A prompt message indicating that user account has been disabled after 3 consecutive unsuccessful login attempts | N/A | N/A |
| Account\_Re-enable\_Request | Class | Defines the request for re-enabling a disabled account | N/A | N/A |
| Valid\_Request | Class | Validates a request based on the credentials and authorizations | N/A | N/A |
| Re-enable\_Account\_Confirmation | Class | A prompt message indicating that user account has been successfully re-enabled | N/A | N/A |
| Report\_Generation Request | Class | Defines the request for generating a report showing account related events like valid log-on, session durations | N/A | N/A |
| Account\_Report | Class | Report containing account information | N/A | N/A |
| Username | String | Unique sequence of characters used to identify the user | 0-15 | ISO/IEC 10646 |
| Password | String | Unique sequence of characters used to authenticate the user | 8-15 | ISO/IEC 10646 |
| Role | String | Defines the role assigned to the user in the system | 0-20 | ISO/IEC 10646 |
| Count | Integer | Keeps a count of consecutive invalid or unsuccessful attempts to login by the user | 0-3 | Each |
| Status\_Update | Class | Defines the status update of a specific unit | N/A | N/A |
| Acknowledgement | Class | Confirms the receipt of a message | N/A | N/A |
| Pressed\_Key\_Message | Class | Defines the message generated by pressing a specific key on the radio | N/A | N/A |
| Voice\_Message\_Response | Class | It is the voice message sent by the base station radio to the operator at the terminal | N/A | N/A |
| Dispatch\_Voice\_Message | Class | Encoded voice message sent to the base station radio by the operator with dispatch instructions | N/A | N/A |
| Start\_Time | DateTime | It stores the starting time of the login | N/A | Milliseconds |
| End\_Time | DateTime | It stores the ending time of the login | N/A | Milliseconds |
| Error | Class | It is the class representing errors | N/A | N/A |
| Unit\_Information\_Error | Class | It represents the error thrown when requested unit information is not found | N/A | N/A |
| Communications\_Log\_Error | Class | It represents the error thrown when requested communications log is not found | N/A | N/A |
| Dispatch\_Log\_Error | Class | It represents the error thrown when requested dispatch log is not found | N/A | N/A |
| Audit\_Trail\_Error | Class | It represents the error thrown when requested Audit Trail is not found | N/A | N/A |

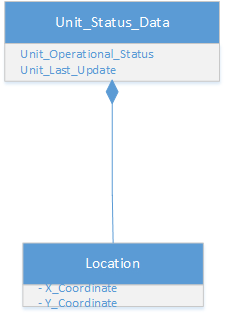
## Class Diagrams

**Base Station:**

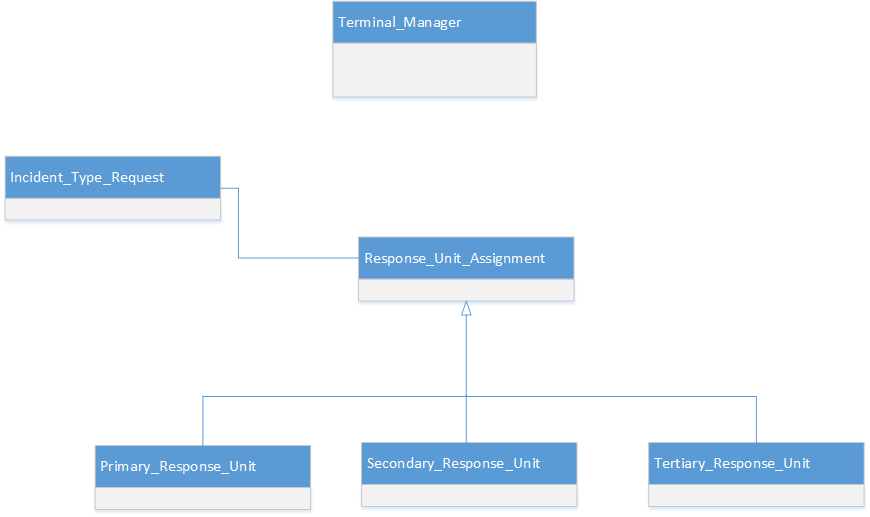


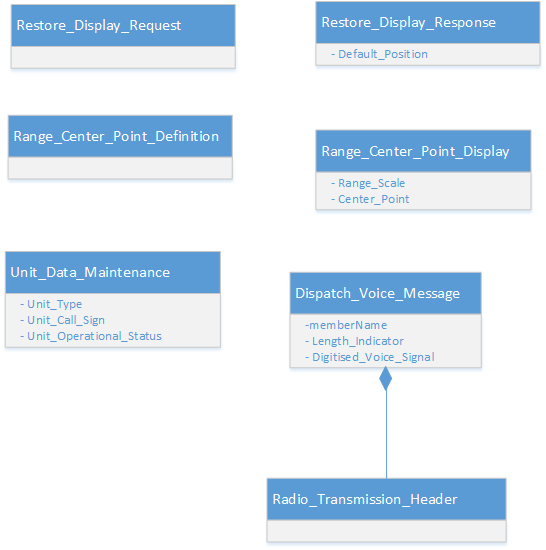
**Status Manager:**

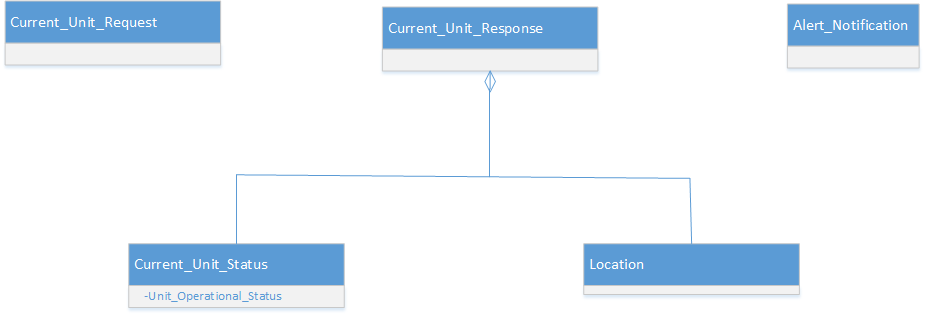




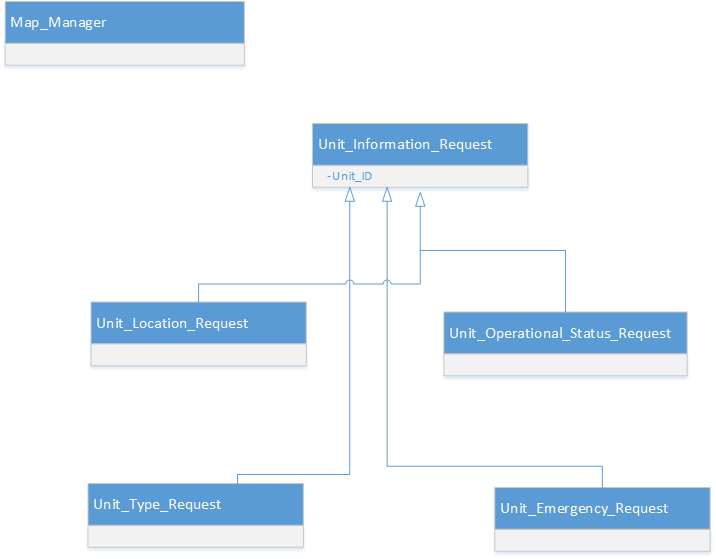
**Terminal Manager:**

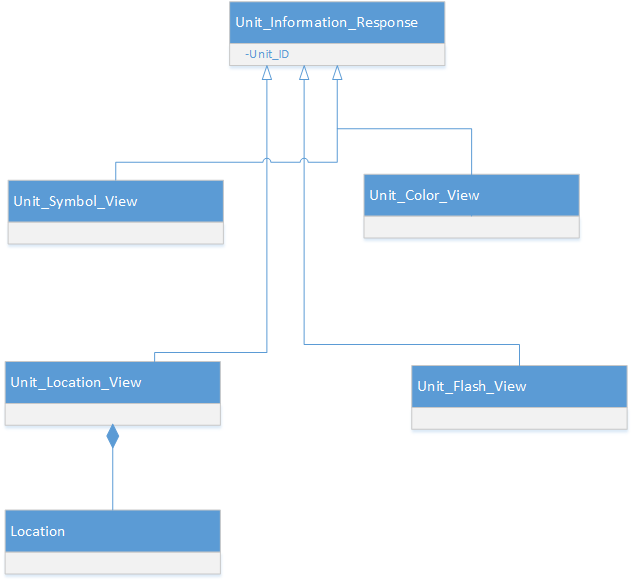


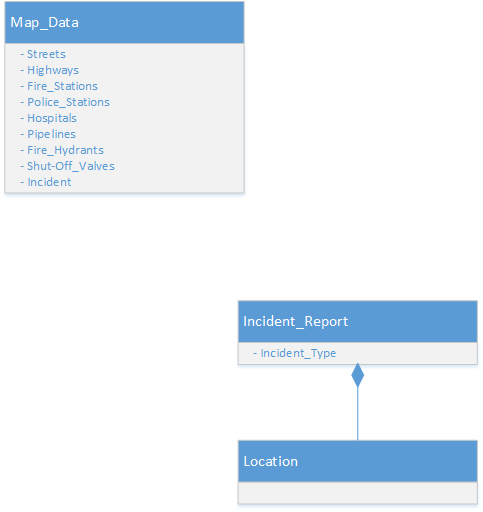


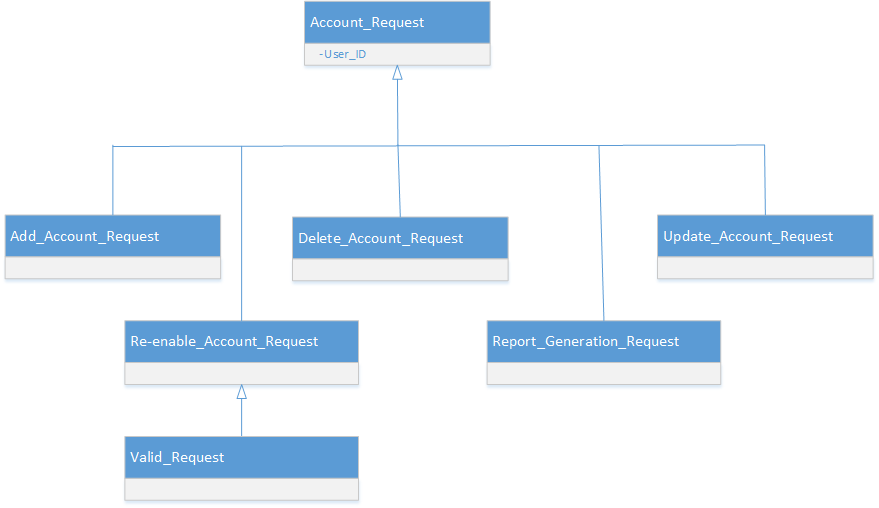


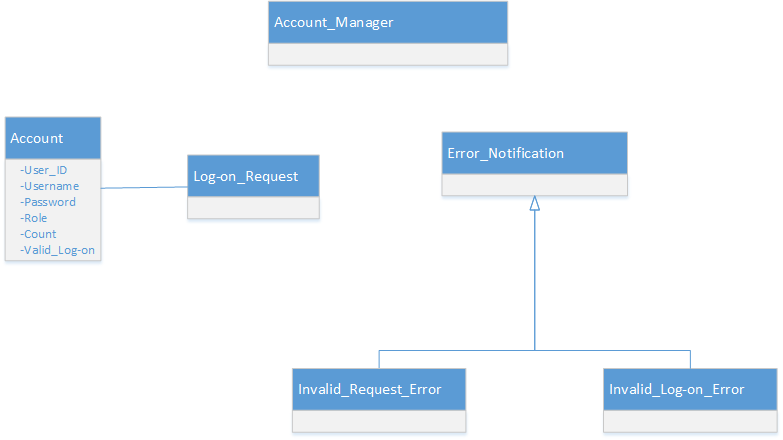
**Map Manager:**

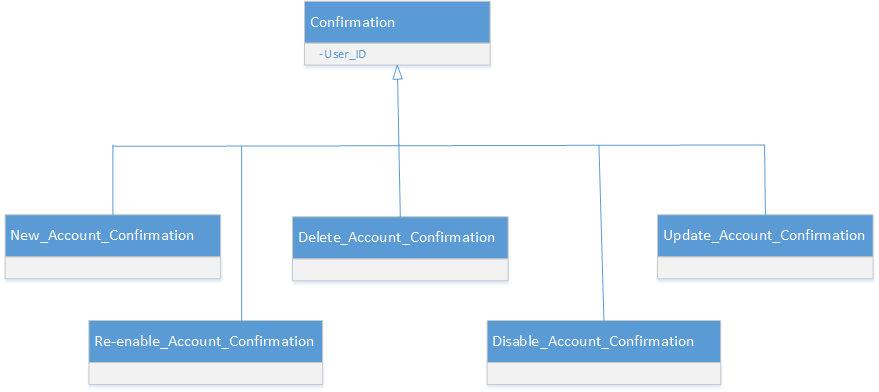


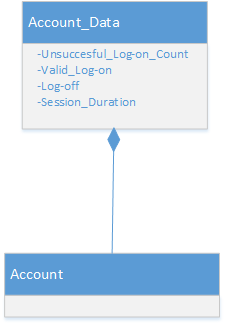




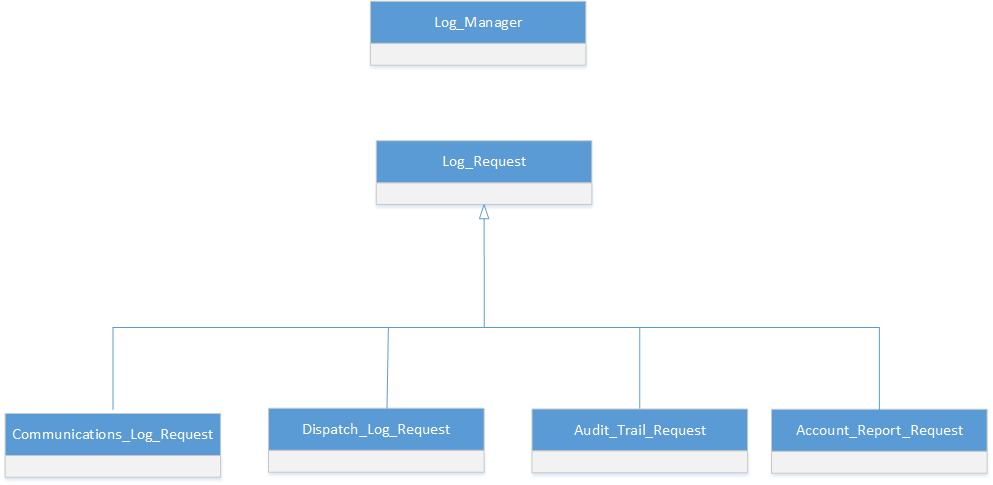


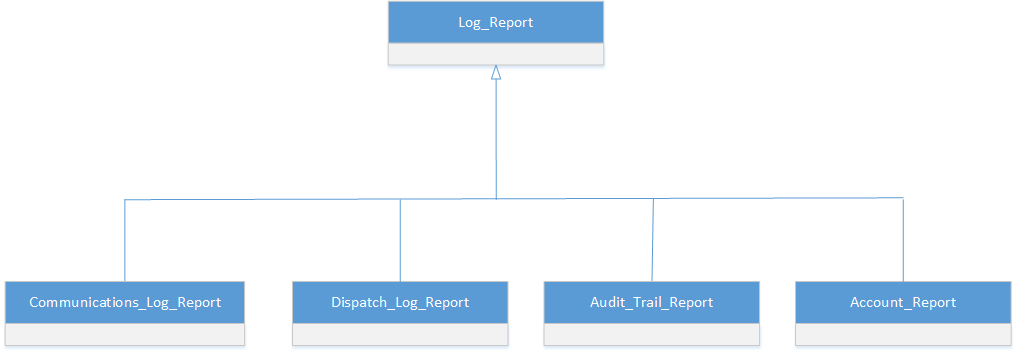


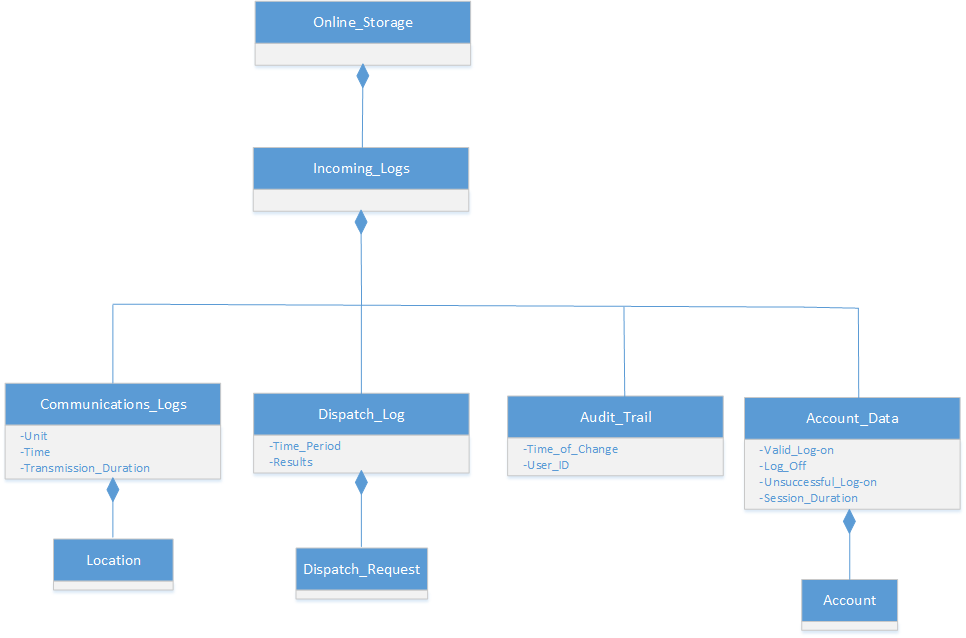


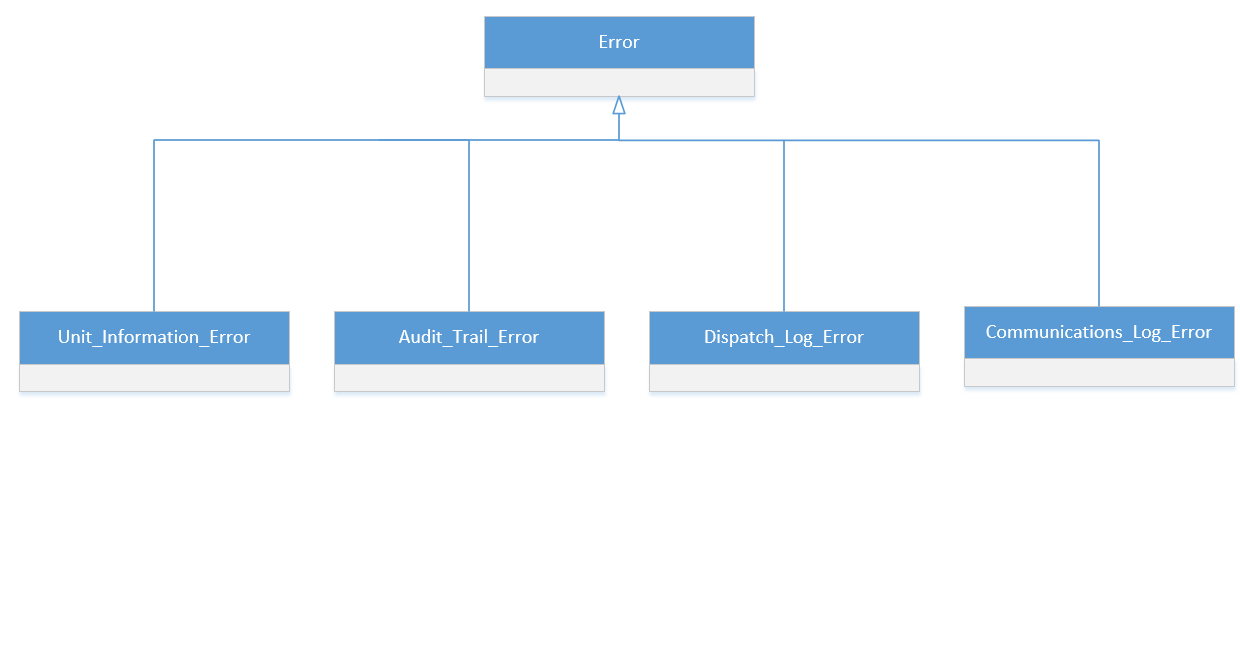


**Log Manager:**

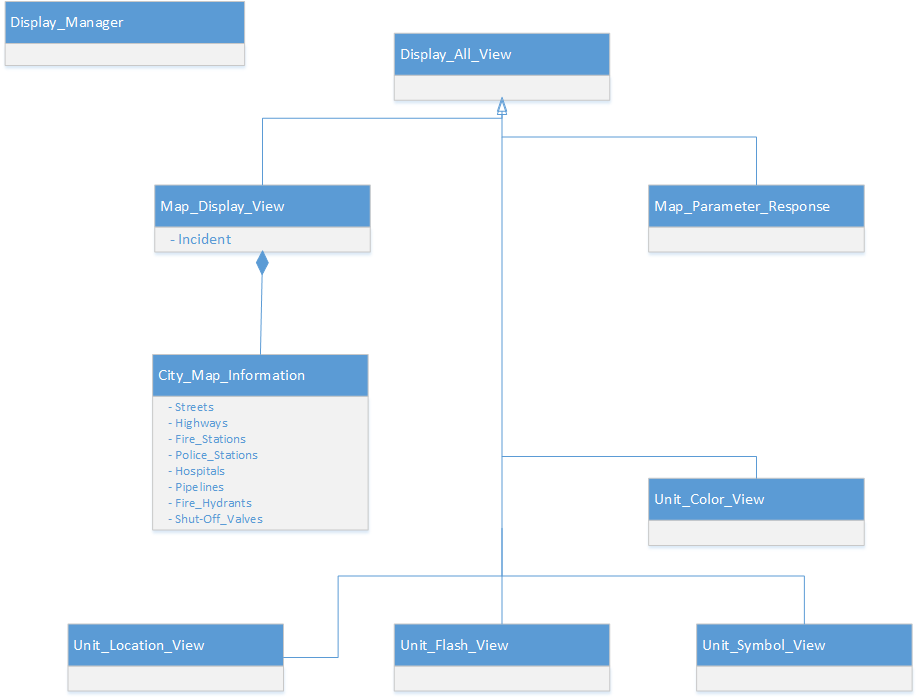








**Display Manager:**



# References

Wiegers, K., & Beatty, J. (n.d.). *Software Requirements* (3rd ed.). Redmond, Wash.: Microsoft Press.

**A critique of our efforts:**

**What was difficult?**

Thinking on an abstract level and not focus on the design aspect when writing the data dictionary and functional requirements was the most challenging task. Also, for the second part of the project we did not have enough meetings with the professor for feedback due to Thanksgiving break.

**What went well?**

Our team was very diligent in completing the required tasks. Regular meeting with the professor (acting as our customer) and getting regular feedback was very helpful.

**What would you do differently?**

Better time management as we noticed that we spent more than we planned for for some sections.

**How should the assignment be changed for the next semester?**

We had a difficult time in determining the scope of the project. Determining what was part of the system and what was external to the system was a difficult task. We think a session in the class to go over the general requirements and understand the scope might be helpful before the teams start working on the projects.