SDA Project Document

For

Rescue 1122 Data Management

Version 1.0 approved

##### **Prepared by**

##### **Capt Shorahbeel**

##### **Capt Abdur Rehman Raza**

##### **Tc Zeeshan Mubarak**

##### **Nc Sehreen**

Military College of Signals

May 26, 2011

Table of Contents

Table of Contents ii

Revision History Error! Bookmark not defined.

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3. Use Cases 5

3.1 System Environment 5

3.2 Conrol Room Wirrless Operator USE Case 6

3.2.1 Request For Emergency Rescue 6

3.2.2 Additional Crew Request 6

3.3 Crew Manager USE Case 7

3.3.1 Assign Crew to Emergency 7

3.3.2 Assign Additional Crew 8

3.4 Crew Informer USE Case 9

3.4.1 Send Crew 9

3.4.2 Report Arrival Back 10

3.4.3 Report Non Availibility of Crew 11

3.4.4 Report Availibility of Crew 11

3.5 Station Incharge USE Case 12

3.5.1 View Emergency Handling Procedure 12

3.5.2 Report Generation 13

3.5.3 Remarks 13

3.6 Crew USE Case 1**Error! Bookmark not defined.**

3.5.1 MO Detailed Report 14

3.5.2 Rescue 112 staff Detailed Report 14

3.5.3 Nursing Staff Detailed Report 15

4. External Interface Use Cases 16

4.1 Login 16

4.2 Submit 16

4.3 Update View 16

4.4 Search 17

4.5 Today's Emergencies 17

4.6 Recent Emergencies 17

4.7 Status Of Crew 18

**5. Patterns Applied……………………………………………………………………………………….20**

5.1 Pure Fabrication…………………………………………………………………………………………………..20

5.2 Singleton……………………………………………………………………………………………………………..21

5.3 Factory………………………………………………………………………………………………………………….21

5.4 Observer………………………………………………………………………………………………………………..22

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the Use Cases, Class Diagrams, Patterns Applied of Rescue 1122 Data Management System.

## Document Conventions

UML Use Case(US)

## Intended Audience and Reading Suggestions

Document is intended for developers, project managers, marketing staff, users, Software test engineers, and documentation experts. The document is sequentially designed.

## Project Scope

The goal of the project is efficient handling of the available crew and providing central staff the information of activity currently in progress. The project deals with various client requirements and facilities, consisting of:

* Central Authentication
* LAN Connectivity
* Database Management
* Crew Management
* Reports
* Status of Emergency & Crews

## Reference

N/A

# Overall Description

## Product Perspective

The product is a replacement of existing system. The existing system maintains the data on registers and all the staff within offices has to inform each other through telephone or personally about some emergency. The product will enable different indoor staff i.e (Control Room Wireless Operator, Crew Manager, Station Incharge) to remain in picture what activity is going on and provide better management of crew.

## Product Features

Following are product main Features:-

* Data Management
  + Initial info record
  + Detailed Emergency Report
  + Remarks of Station Incharge
* Report Generation
* Crew Management
* Status of Emergency
* Search
* Status of Crew

## User Classes and Characteristics

* Control Room Wireless Operator
  + It receives the phone calls for emergency and fills the initial emergency request form. Generally Two types of Calls
    - Request for Emergency
    - Request for Additional Crew
* Crew Manager
  + When an initial request form in filled, Crew manager assigns the crew based on nature of emergency.
* Crew Informer
  + When some crew is assigned to some emergency dealing, it sends them to the location of emergency. And fills out the time of leaving the garage and coming back.
* Station Incharge
  + It is overall supervisor of a rescue 1122 station. He manages all the reports and supervises all the staff.
* Crew
  + Crew deals with emergency physically by going on location. On completion, crew fills out detailed report about the emergency. Crew consists of
    - Medical Officer(MO)
    - Rescue 1122 Staff(driver, helper, rescuer)
    - Nursing Staff

## Operating Environment

All the users office in one central Building Therefore system will be working on LAN and using a central database on server. The users already have the knowledge and working experience on Windows Operating System, therefore product will be made for windows platform.

## Design and Implementation Constraints

The project requires handling of multiple computers. Strict security arrangements are required to cater for the monitoring aspects of the project. Every user will be authenticated and access to only allowed data will be made to all users. Unnecessary access to data will not be granted.

## User Documentation

User manuals and the tutorial will be provided to the users along with the software. Online Help will be provided in case needed.

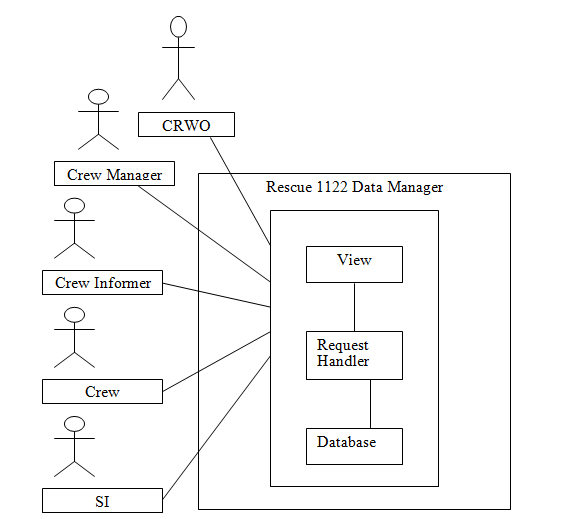
## Assumptions and Dependencies

Various assumptions during the development phase are of great importance such as:

* Level of expertise of the end users
* The system requirements of the project
* The environment in which the project is going to operate .

# Use Cases

## System Environment



The Rescue 1122 Data Manager has 5 Actors one Database. All the User initiate request for a desired operation. The request handler entertain the request and perform required operation after authentication. Request handler insert, delete, update entries from database and provide appropriate data/Form to view so that users can see result/initiate further Action.

## Conrol Room Wirrless Operator USE Case

**3.2.1 Request For Emergency Rescue**

**Brief Description**

The Control Room Wireless Operator on receiving the call of new emergency fills out initial information.

**Initial Step-By-Step Description**

Before this use case can be initiated, the CRWO has to be logged in as CRWO so that he can get associated interface to fill initial information.

1. The CRWO on receiving the call click on new emergency button
2. The System gives initial information form.
3. The CRWO fills the form and click on submit.
4. The Request handler insert the data into database.
5. On completion, Successful insertion message is displayed
6. The CRWO is returned to main page.

**Xref:** Section 4.2.1, Submit

**3.2.2 Additional Crew Request**

**Brief Description**

The Control Room Wireless Operator on receiving the call for additional crew initiates the request.

**Initial Step-By-Step Description**

Before this use case can be initiated, the CRWO has to be logged in as CRWO so that he can get associated interface.

1. The CRWO on receiving the call click on Additional Crew Request Button.
2. The System provides the list of in progress emergencies.
3. The CRWO selects the emergency for which additional crew request was asked.
4. The System provides the view which shows the details of staff already sent for that emergency.
5. The CRWO writes the detail of additional requested crew.
6. The CRWO submit the details.
7. The Request Handler modifies the record of that particular emergency.
8. On completion, Successful insertion message is displayed.
9. The CRWO is returned to main page.

**Xref:** Section 4.2.1, Submit

## Crew Manager USE Case

**3.3.1 Assign Crew to Emergency**

**Brief Description**

When so ever the new Emergency form is submitted by the CRWO Crew Manager Assigns the crew to emergency based upon the nature of emergency.

**Initial Step-By-Step Description**

Before this use case can be initiated, the CREW manager has to be logged in as Crew Manager so that he can get associated interface.

1. The System provides the crew manager with a blank interface.
2. When so ever the CRWO submit the request for crew, the view is updated and detail of emergency is provided.
3. Crew Manager reads the details and assign the crew from list of available crew
4. On completion, Successful message is displayed
5. In case no crew is free, Crew Manager is provided with the list of all the crews and their nature of emergency.
6. Then Crew manager assigns the crew which is expected to fall back soon.
7. The crew manger is returned to main page.
8. Request Handler keep track of assigning of crew. If the assigned crew doesn’t come back soon and some other crew has returned, crew manager is alerted so that he can reassign the crew.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.3.2 Additional Crew Request**

**Brief Description**

When so ever the request for additional crew is submitted by the CRWO, Crew Manager Assigns the crew.

**Initial Step-By-Step Description**

Before this use case can be initiated, the CREW manager has to be logged in as Crew Manager so that he can get associated interface.

1. The System provides the crew manager with a blank interface.
2. When so ever the CRWO submit the request for additional crew, the view is updated and detail of emergency is provided.
3. Crew Manager reads the details and assign the crew from list of available crew
4. On completion, Successful message is displayed
5. In case no crew is free, Crew Manager is provided with the list of all the crews and their nature of emergency.
6. Then Crew manager assigns the crew which is expected to fall back soon.
7. The crew manger is returned to main page.
8. Request Handler keep track of assigning of crew. If the assigned crew doesn’t come back soon and some other crew has returned, crew manager is alerted so that he can reassign the crew.
9. The CRWO is returned to main page.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

## Crew Informer Use Case

**3.4.1 Send Crew**

**Brief Description**

When so ever a Crew is assigned to some emergency, crew informer sends the crew to location.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Crew informer has to be logged in as Crew Manager so that he can get associated interface.

1. The System provides the crew informer with interface which contains list of all the crews.
2. When so ever some crew is assigned, the crew informer view is updated.
3. Crew informer reads out the details and sends the crew to location.
4. Crew informer fills in the time for departure and submit.
5. On completion, Successful message is displayed

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.4.2 Report Arrival Back**

**Brief Description**

When so ever a Crew is back from some emergency, crew informer fills the arrival back time.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Crew informer has to be logged in as Crew Manager so that he can get associated interface.

1. The System provides the crew informer with interface which contains the list of all the crews.
2. When so ever some crew is back, the crew informer fills the report back time
3. Crew informer submits the data
4. On completion, Successful message is displayed

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.4.3 Report Non Availability of Crew**

**Brief Description**

If some crew is not ready because of some problem, the crew informer reports non availability of the crew.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Crew informer has to be logged in as Crew Manager so that he can get associated interface.

1. The System provides the crew informer with interface which contains the list of all the crews.
2. The crew informer marks the crew as non available and fills the reason of non availability.
3. Crew informer submits the data
4. On completion, Successful message is displayed
5. The non available crew is removed from the list of available crews so that crew manager cant assign this crew to some emergency.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.4.4 Report Availability of Crew**

**Brief Description**

If some non available crew is ready, the crew informer reports availability of the crew.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Crew informer has to be logged in as Crew Manager so that he can get associated interface.

1. The System provides the crew informer with interface which contains the list of all the crews.
2. The crew informer marks the crew as available
3. Crew informer submits the data
4. On completion, Successful message is displayed
5. The crew is added to the list of available crews so that crew manager can assign this crew to some emergency

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

## Station Incharge USE Case

**3.5.1 View Emergency Handling Procedure**

**Brief Description**

Station Incharge views the emergency handling procedure to keep check on performance.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Station Incharge has to be logged in as Station Incharge so that he can get associated interface.

1. The System provides the Station Incharge request for emergency procedure view.
2. The System Provides list of current date all completed Emergencies and a search box to search back date events.
3. Station incharge clicks on some current date emergency or search for some back date.
4. Station Incharge is provided with a view of details of the emergency handling.
5. On clicking back page Button Station Incharge return to his main page.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.5.2 Report Generation**

**Brief Description**

Station Incharge views the emergency handling procedure to write report about it.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Station Incharge has to be logged in as Station Incharge and view some emergency handling procedure.

1. After viewing emergency handling Procedure, SI click on Write Report Button
2. SI is provided with report writing view.
3. SI writes the report and submit it.
4. On submission Station Incharge return to his main page.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.5.3 Remarks**

**Brief Description**

Station Incharge views the emergency handling procedure to write his observations and remarks about the procedure adopted.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Station Incharge has to be logged in as Station Incharge and view some emergency handling procedure.

1. After viewing emergency handling Procedure, SI click on Write Remarks Button
2. SI is provided with remarks writing view.
3. SI writes the remarks and observations and submit it.
4. On submission Station Incharge return to his main page.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

## Crew Use Case

**3.6.1 MO Detailed Report**

**Brief Description**

On coming back from emergency, the medical Officer need to report back nature of medical issues handled.

**Initial Step-By-Step Description**

Before this use case can be initiated, the MO has to be logged in as MO so that appropriate view can be provided.

1. MO is provided with the list of all emergences he has handled and yet not written the detailed report.
2. MO write the report and submit it.
3. On submission MO is returned to his main page.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.6.2 Rescue Staff Detailed Report**

**Brief Description**

On coming back from emergency, the Rescue Staff need to report back nature of rescue operation performed and its details.

**Initial Step-By-Step Description**

Before this use case can be initiated, the MO has to be logged in as MO so that appropriate view can be provided.

1. Rescue Staff is provided with the list of all emergences he has handled and yet not written the detailed report.
2. Rescue Staff write the report and submit it.
3. On submission Rescue Staff is returned to his main page.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

**3.6.3 Nursing Staff Detailed Report**

**Brief Description**

On coming back from emergency, the Nursing Staff need to report back the store used in medical operation.

**Initial Step-By-Step Description**

Before this use case can be initiated, the MO has to be logged in as MO so that appropriate view can be provided.

1. Nursing Staff is provided with the list of all emergences he has handled and yet not written the detailed report.
2. Nursing Staff write the report and submit it.
3. On submission Nursing Staff is returned to his main page.

**Xref:** Section 4.2.1, Submit, 4.2.2 Update view

# External Interface Use Cases

**4.1 Login**

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **XRef** | On Sign Out/Start of Application |
| **Trigger** | When users hit Log In Button |
| **Precondition** | 1. The Product must have load correctly 2. Connection to Database Established |
| **Basic Path** | 1. Represents login info view 2. On Click, check username and password in database 3. Allow access to appropriate tab view according to logged in user |
| **Alternative Paths** | If login failed due to wrong password, show message box |
| **Postcondition** | On Successful login, show associated tab |
| **Exception Paths** | On connection failure, show message |
| **Other** | Nill |

**4.2 Submit**

|  |  |
| --- | --- |
| **Use Case Name** | Submit |
| **XRef** | 3.1,3.2,3.3,3.4,3.5,3.6 |
| **Trigger** | When Users Click Submit Button |
| **Precondition** | 1. The User must be logged in |
| **Basic Path** | 1. Checks the Data for Integrity  2. Write the Data to Database |
| **Alternative Paths** | In case of Wrong data Show Error Message |
| **Postcondition** | On Completion take back to main page |
| **Exception Paths** | On connection failure, show message |
| **Other** | Nill |

**4.3 Update View**

|  |  |
| --- | --- |
| **Use Case Name** | Update View |
| **XRef** | 3.1,3.2,3.3,3.4,3.5,3.6 |
| **Trigger** | When so ever some modification to Database is made |
| **Precondition** | The User must be logged in |
| **Basic Path** | 1. Checks for the Required data in Database for the logged in user  2. Update the view |
| **Alternative Paths** | Nill |
| **Postcondition** | Connection to Database must be established |
| **Exception Paths** | On connection failure, show message |
| **Other** | Nill |

**4.4 Search**

|  |  |
| --- | --- |
| **Use Case Name** | Search |
| **XRef** | 3.1,3.2,3.3,3.4,3.5,3.6 |
| **Trigger** | When search button is clicked |
| **Precondition** | The User must be logged in |
| **Basic Path** | 1. Checks for the Required data in Database  2. Update the view |
| **Alternative Paths** | Show Message if no record Found |
| **Postcondition** | Connection to Database must be established |
| **Exception Paths** | On connection failure, show message |
| **Other** | Nill |

**4.5 Today’s Emergencies**

|  |  |
| --- | --- |
| **Use Case Name** | Today’s Emergencies |
| **XRef** | 3.1,3.2,3.3,3.4,3.5,3.6 |
| **Trigger** | When Statistics tab is clicked |
| **Precondition** | The User must be logged in |
| **Basic Path** | 1. Checks for the Current date’s emergencies data in Database  2. Update the view |
| **Alternative Paths** | Show Message if no record Found |
| **Postcondition** | Connection to Database must be established |
| **Exception Paths** | On connection failure, show message |
| **Other** | Nill |

**4.6 Recent Emergencies**

|  |  |
| --- | --- |
| **Use Case Name** | Recent Emergencies |
| **XRef** | 3.1,3.2,3.3,3.4,3.5,3.6 |
| **Trigger** | When Statistics tab is clicked |
| **Precondition** | The User must be logged in |
| **Basic Path** | 1. Checks for the recent emergencies data in Database  2. Update the view |
| **Alternative Paths** | Show Message if no record Found |
| **Postcondition** | Connection to Database must be established |
| **Exception Paths** | On connection failure, show message |
| **Other** | Nill |

**4.7 Status Of Crew**

|  |  |
| --- | --- |
| **Use Case Name** | Today’s Emergencies |
| **XRef** | 3.1,3.2,3.3,3.4,3.5,3.6 |
| **Trigger** | When Statistics tab is clicked |
| **Precondition** | The User must be logged in |
| **Basic Path** | 1. Checks for the data of all the crews in Database and shows their activity record  2. Update the view |
| **Alternative Paths** | Show Message if no record Found |
| **Postcondition** | Connection to Database must be established |
| **Exception Paths** | On connection failure, show message |
| **Other** | Nill |

**Patterns Applied**

**Pure Fabrication**

A Database Handler class is added which is pure fabricated class. It contains all the required data to interact with database, its responsibility is to save, retrieve data from the database, al the class interact to database through this class.

Database Handler is an abstract class and its concrete classes are Submit, Update, and Search.

**Singleton Pattern**

A Database Connection class is made which is responsible to establish connection with the database. To implement concurrency, singleton pattern is implemented so that at any time only one connection exist and only one user can store or retrieve data from database. If connection is open at particular time, then other request waits for connection to close.

class Database\_Connection

{

private static string con\_str = "Data Source=myServerAddress;"

+ "Initial Catalog=myDataBase;User Id=myUsername;Password=myPassword;";

private static SqlConnection \_sqlcon;

private Database\_Connection()

{

}

public static SqlConnection SQL\_Con

{

get

{

if (\_sqlcon == null)

{

\_sqlcon = new SqlConnection(con\_str);

}

return \_sqlcon;

}

}

public static bool isOpen

{

get

{

if (\_sqlcon != null && \_sqlcon.State == ConnectionState.Open)

{

return true;

}

return false;

}

}

**Factory Pattern**

Since creation of particular type of person object such as CRWO, SI ets depends upon the login info i.e (username, password) a separate class Login is implemented which is made responsible for creation of the particular type of object depending upon the info provided thus applying Factory pattern

class Login

{

private SqlConnection sqlcon;

public Login()

{

}

private DataSet Get\_Data()

{

sqlcon = Database\_Connection.SQL\_Con;

if (Database\_Connection.isOpen)

{

//Wait Logic

}

//Data retrival Logic

}

private Person Create(string username, string password)

{

Person p;

DataSet ds = Get\_Data();

//authenticate from database

//return appropriate type of Person

//depending upon the username, password

//such as SI, Crew Manager, CRWO etc

return Person;

}

}

**Observer Pattern**

Observer Pattern is implemented to keep every concerned user updated about the changes happening in the database. To apply this pattern, already existing class hierarchy of Database handler is modified. Update class’s logic is implemented as a method of Submit class. A register method is implemented so that users can register with the class and then each time database is modified, all the register users are informed. Submit class maintains a list of all the register users

class Submit

{

List<Person> per;

public Submit()

{

per =new List<Person>();

}

public bool Register(Person p)

{

if (!per.Contains(p))

{

per.Add(p);

return true;

}

return false;

}

public void submit(string param)

{

//Submission Logic

Inform();

}

public void Inform()

{

for (int i = 0; i < per.Count; i++)

{

per[i].update();

}

}

}