Incident Reporting System (IRS)

By

Maj Sardar Muhammad Ali
Maj Umer Riaz
Capt Hafiz Muhammad Bilal
Capt Haris Javaid

Supervisor

Lt. Col Dr Adil Masood

Submitted to the Faculty of Computer Software Engineering, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering

MAY 2017

ABSTRACT

The aim of incident reporting system was to develop mobile and Web based system to support various activities with rescue, relief and rehabilitation process during any minor/ major emergency and to set up an IT infrastructure to respond to different situations/ emergencies. The IT infrastructure would constitute various organizations providing their services (resources) to perform various post management tasks. This system particularly helps rescue management authorities and concerned agencies to collaborate and cooperate with each other. Moreover, it also helps the authorities to efficiently locate and allocate required resources to a particular rescue/relief operation.

The system is accessible through a Web interface and alert feature can be access through android app. User authentication and a User Management Service controls access management. Rescue/ Control Management Service is the key component of the system, which consists of various subsystems including Workflow System. This system appoints tasks to different organizations and sets priorities and dependencies between different tasks. A Capability Assessment Engine suggests resources of different organization based on incident specific parameters. Resource Management Service has the responsibility to update status of resources. Communication Service allows different organizations and officials involved in a task to communicate with one another. Incident/ Disaster Support Repository is a normalized relational database managing the records of the entire system.

The above tasks have been accomplished by establishing Web Services in C# language using ASP.NET WCF Service Application. The Communication between Web service and its clients is through SOAP using XML. WCF uses SOAP as message protocol. Specification of SOAP describes the envelope composition, its encoding rules and the ways of expressing Web service request and response. Request and response are used as SOAP messages over HTTP. The interface consists of ASP pages, for dynamic data handling without reloading the page we have also employed JQUERY, AJAX, JSON. Methodologies have been used to prevent SQL Injection attacks.

A comprehensive testing methodology including Prototype Testing, unit testing, and integration testing is followed. Usability analysis is also carried out by following different usability heuristics.

CERTIFICATE OF CORRECTNESS AND APPROVAL

Certified that work contained in this thesis “Incident Reporting System” carried out by Maj Sardar Muhammad Ali, Maj Umer Riaz, Capt Hafiz Muhammad Bilal and Capt Haris Javaid under the supervision of Lt. Col Dr Adil Masood for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

_____

(**Lt. Col Dr Adil Masood**)

Dated: _____

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of **ALLAH**, the Most Merciful, The Most Beneficent

To our teachers and parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible.

To Google, which made it all possible.

ACKNOWLEDGMENTS

We are eternally grateful to Almighty Allah for bestowing us with the strength and resolve to undertake and complete the project.

We gratefully recognize the continuous supervision and motivation provided to us by our supervisor Lt. Col Dr Adil Masood for his continuous and valuable suggestions, guidance, and commitment towards provision for the undue support throughout this project.

We will also like to thank other faculty members of the department of Computer Software Engineering for their continuous support and help. It was their help and guidance which helped us complete project in due time.

We would also like to thank our class mates and last but not the least our family members for their support throughout the project in every way possible.

TABLE OF CONTENTS

LIST OF FIGURES

Appendix A

Appendix B

Chapter 1

Introduction to Incident Reporting System

1.1 Introduction

Natural and manmade minor and major incident/ disasters bring destruction, chaos and misery to human life and society. Under such circumstance, need for effective and timely communication, collaboration of different organizations and people becomes very much important to carry our various activities of rescue, relief and rehabilitation. Incident Reporting System is a resource management and collaboration platform to support such activities specifically rescue and relief operations after any incident. Incident Reporting System has been developed using open standards like XML, SOAP and Web services for interoperability and cross platform support to facilitate organizations in integrating their existing IT solutions.

1.2 Background

Accidents, fire and other small/ big disasters are mostly naturally occurring events that have a negative effect on human life. These can be of two types, due to geological activities on surface of earth or human errors. Geological activities are occurring at all times its only when it causes a significant damage to human life it's considered a disaster. Disasters that occur in low vulnerability areas such as uninhabited areas are usually not called disasters. Natural disasters can include floods, tsunamis, tornadoes, volcanic eruptions and so on. While incidents due to human error can be like fire, oil spills, accidents, nuclear waste and so forth. In the last decade, hundreds of thousands of people have died in these disasters and countless other have suffered their impacts and are still suffering. Tsunami in 2004 and 2011, earthquake in Pakistan in 2005 and floods in 2010 have made a huge impact on human life. Many people died while still many

became homeless. Disaster Management is vast and critical subject in this scenario because if a disaster occurs and post disaster activities are not managed properly it can not only expand the impact of disaster but cause further disasters. However the impact of any incident/ disaster can be minimized by using calamity mitigation strategies. These include implementation of calamity/ incident early warning systems, preparing development plans to provide resilience for difficult incidents, mobilizing resources and expediting rehabilitation and post calamity activities.

Unfortunately after suffering two of the biggest disasters in last decade, Pakistan still doesn't have an IT infrastructure for management of post accident activities. To manage these post activities, Pakistan needs an effective IT solution and that is the reason of development of IRS. Since a large number of organizations are active in Pakistan, it is necessary that the said solution should be easy to integrate and communicate with existing and upcoming solution. That is why IRS has been built using web services.

1.3 Problem Statement

Natural calamities and daily mishaps cannot be avoided. However efforts can be made to minimize their effects. Mismanagement can lead to more damage. In the recent years a number of disasters have occurred in the world. Pakistan itself has faced two of the biggest calamities in recent history in shape of earthquake in 2005 and floods in 2010 which has triggered the need of developing a system to gain control of situation quickly and to effectively plan rescue and relief operations and execute them. IRS addresses this problem by efficiently managing these post activities to provide timely and effective help to the effected people.

1.4 Goals and Objective

IRS has been developed to facilitate in post-accident activities. The objectives of IRS is to alert the system using mobile app, to locate and allocate resources and helping organization to collaborate with each other

in post incident activities of rescue and relief. The mandatory functions of IRS are locating and allocating resources, assigning tasks to different organizations and developing a workflow for different rescue and relief activities. The goal of the project is to create a new project once any mishap has occurred, develop workflow(s) for different post incident activities, assign activities and resources to different organizations, monitor their performance and finally grade how efficiently they performed a task.

1.5 Deliverables

1st Progress Report: Including SRS Document and Literature Review.

2nd Progress Report: Including System Design Document.

3rd Progress: Including progress of the system in the form of demo.

4th Progress: Including Complete Demonstration of Project and Project Documentation.

1.6 Document Organization

This document provides basic knowledge about IRS. Initially the description of project has been given, the next chapter discusses related software products and their functionalities and how IRS is different from them. Requirement specifications have been covered in chapter 3, Design specifications are discussed in chapter 4. Implementation is described in chapter 5 and further chapters explains analysis of the software, testing techniques employed and suggestion about further work in enhancing the functionalities and capabilities of the system.

1.7 Summary

IRS is a resource management and collaboration tool to assist in post incident rescue and relief operations and comes under the umbrella of web engineering. This chapter is an introduction to IRS, goals and objectives

which have been defined in order to develop this software. A brief introduction to document contents is also included.

Chapter 2

Literature Review

2.1 Introduction

Main objective of this project is to help and support the Pakistani government in post-incident activities. For this purpose, National Disaster Management Authority (NDMA) has developed a comprehensive disaster management framework. Salient features of this framework and working of NDMA is discussed in this chapter. This framework is the foundation of the proposed Incident Reporting System. Different existing systems to support the disaster management and the limitations of these systems are also discussed.

2.2 Accident/ Crisis Management in Pakistan

Pakistan's inability to deal with large calamities was highlighted during the earthquake of 2005. It was after observing this deficiency, National Disaster Management Authority was formed. The framework for dealing with disasters laid down by NDMA lays down SOPs for dealing with different disasters. It also calls for establishment of a forum to increase collaboration between armed forces and other governmental and non-governmental organizations. The said framework also calls for strengthening PDMAs and DDMAs as they are currently very weak. Under this framework, NDMA has also established Emergency Operations Centers at national, provincial and district level.

Emergency Operations Centers at regional and district level are responsible for continuously monitoring different risks, hazards and vulnerable conditions. They also coordinate complete spectrum of disasters in their regions and are responsible for spreading awareness and education on disaster reduction and response.

NDMA has laid down the foundations for efficient post and pre calamity activities in its framework. However, the support infrastructure has not yet been provided by the concerned agencies. The core issues under such conditions are of communications, information sharing, access to right and correct information and collaboration between different agencies. An IT infrastructure along with the relevant software systems can greatly help these operations. However, in Pakistan, we still don't have the central repository that contains information about the capabilities and resources available that can be used under such conditions. Therefore, the proposed IRS can help the NDMA and the concerned organizations in this regards.

In the next sections we will discuss the available disaster management systems that have been used by different relief and welfare organizations worldwide for the post-disaster relief and rescue operations.

2.3 IT based tools for Disaster Management

The following section provides a brief description of tools already developed for disaster management and the last section highlights the limitation of these systems.

2.3.1 SAHANA FOSS Disaster Management System

SAHANA is a free and open source system built for disaster management after 2004 Sri Lanka tsunami. The system was initially deployed by Sri Lanka's CNO (Center of National Operations) . The project has since grown with deployment in Pakistan Earth Quake , Philippine Mudslide (2006) and Indonesian Earthquake (2006) as examples. The system was initially developed for missing person registry while as an open source project rest of the modules have been developed by different communities all over the world. While the issue of scalability has been recently identified and people are working on exposing SAHANA FOSS functions as web services.

2.3.2 Disaster Management Information System

This is a web based tool developed by International Federation of Red Cross and Red Crescent societies. The biggest disadvantage is that this tool can only be used by members of Red Cross and Red Crescent societies. The tool basically provides information to federation about disaster trends, resource both internal and external and different tools for information management [3].

2.3.3 Limitations

SAHANA FOSS's limitation is that it was developed as a database for missing person so the focus from the start wasn't on disaster management activities. Secondly SAHANA didn't support scalability as such because it is not based on web services. Though now a project has been started to shift SAHANA FOSS to web services.

Disaster Management Information System can be only used by volunteers of International Red Cross and Red Crescent Federation.

2.4 Enabling Technologies

Accessibility, interoperability and open standards can help the organizations to develop systems that can be integrated together. In the case of disaster management, the main monitoring agency has to work with different organizations including health services, armed forces, civil organizations, met office and NGOs. These organizations may have their own systems which include the information related to their available resources, held stock levels of goods, competencies and team/project management system. Thus, the integration of these systems at National level is a major challenge. Moreover, accessibility to a centralized system is also a major concern. Therefore, appropriate tools and technologies are required to develop an integrated solution that can help in various post-disaster activities.

The following sections give a brief description of various tools and technologies that can be helpful in developing an open, interoperable and extendible incident reporting system.

2.4.1 Service Oriented Architecture

IRS has been built on service oriented architecture, publishing functionalities as web services.

Service oriented architecture is in essence design principles which are used during computer system development and integration. A system developed on SOA will have different functionalities offered as interoperable services which can be used within multiple, separate systems from several business domains.

SOA also generally provides a way for consumers of services, such as web-based and mobile based applications, to be aware of available SOA-based services. For example, several disparate departments within a company may develop and deploy SOA services in different implementation languages; their respective clients will benefit from a well understood, well defined interface to access them. XML is the most common method to access services.

A service has three fundamental attributes: a description of service, this is the interface of the service; a method to access the service by invoking its interface and an implementation of the service.

2.4.2 Web Services

Web services provide a mean of interoperating between different software applications that might or might not run on the same platform or web service.[6] To access a web service a client has to subscribe to a web service first and once authorized only then it can access its functionalities.

2.5 Summary

This chapter introduces us to existing system for incident/ disaster management; their features and limitations. The section of enabling technologies provides an introduction to various technologies that can be used to develop IRS so that overcomes limitations of existing systems.

Chapter 3

System Requirements

3.1 Introduction

System requirements for the proposed Incident Reporting System (IRS) are gathered based on the framework proposed by the National Disaster Management System (NDMS) and the general project management principles. Moreover, these requirements are specified based on the principles of Computer Supported Collaborative Work (CSCW), collaborative networks and Virtual Organizations. The proposed system helps in post disaster activities of rescue and relief. When a disaster occurs the system helps in developing workflows for different post disaster activities and assigns tasks based on the workflow to different teams thus enabling teams to collaborate and share resources to perform different rescue and relief tasks effectively and efficiently. This chapter provides specifications that IRS has to fulfill so it can perform in the best possible manner.

3.2 Project Scope

The main objective of this project is to develop a Web based decision support system to support various activities associated with rescue and relief processes during any incident/emergency situation, to develop mobile app for alerting and to provide situational awareness through maps. This system will particularly help the management authorities and concerned agencies to collaborate and cooperate with each other. Moreover, it will also help authorities to efficiently locate and allocate required resources to a particular rescue or relief operation. The scope of the project is restricted to rescue and relief operations only.

3.3 Product Features

IRS has specific features which have been designed by focusing on how the rescue and relief operations take place. These features are the main functionalities the system will provide and form the basis for the system design and implementation. The required features are elaborated in more detail in the following sections.

3.3.1 Competency/ Resource repository of organizations

IRS allows the maintenance of a complete database of resources/competencies of different organizations. For this purpose IRS allows the organization registration process. Moreover, the system supports the features that allows automatic updating of the central repositories whenever a particular organization updates the status of its resources in their own system.

3.3.2 Competency Analysis Module

IRS supports the competency analysis of the organizations. This feature can be used whenever a new project is created, and a set of required competencies or required resources need to be generated. Based on the tasks involved in a particular operation, the system helps in identifying the concerned organizations with relevant skills/competencies as well as the available resources.

3.3.3 Team formation and Task management (Workflow System)

IRS helps in developing a workflow to support the tasks associated with a particular type of disaster.

3.3.4 Project Tracking

The system provides means for progress tracking for the tasks that can be performed in a particular project.

3.3.5 Assessment/Grading

IRS supports the evaluation of various participating organizations for future reference and the overall performance measurement of a particular operation. Time of completion, number of resources available/provided can be used as criteria for the evaluation.

3.3.6 Team Selection

This feature allows user to select a team to undertake a task of rescue or relief operation once a disaster has struck.

3.3.7 Appoint Tasks

This feature allows appointing tasks to different teams or team leads. Once a workflow is generated based on incident type, the different tasks enlisted in the workflow are appointed to teams based on their areas of expertise.

3.3.8 Project Status Check

Each organization that are subscribed to IRS can be able to publish its status on IRS, so that a track of current activities can be kept. This feature also helps in planning and executing future tasks.

3.3.9 Login/Access Right

IRS allows users to login based on their hierarchy in the organization. Different users have different features visible to them based on his role in the organization.

3.3.10 Record maintenance and Report generation

This feature allows user to maintain records of organizations registered with the system. IRS have complete record of resources available, tasks and services performed by different teams during rescue and relief operations. The records also be useful in future incidents as well as in studies of disaster management. Different users have different access

level to database repository. The database is designed in such a way to protect data integrity and confidentiality.

3.3.11 MAP

IRS has fully integrated mapping functionality which allows any location-based data to be visualized on a map. Maps provide situational awareness which is essential when either planning to prepare for or respond to a disaster.

3.3.12 Messaging

In the complex domain of incident/ emergency Management, communication is critical. IRS provides support for messages to be sent by Email and SMS. Distribution Groups can be set up to allow messages to be easily sent to many people at once. Users are able to search for specific information and subscribe to receive update messages when new information is added. Interactive messages can also be set up to allow people to send short message queries to IRS and receive automatic responses.

3.3.13 Alert App

A mobile based app that alert the system by the mobile user when the particular accident occurs.

3.4 Assumptions and Dependencies

Some basic assumptions have been made in the development of IRS. This section describes assumptions and dependencies based on which IRS has been developed.

3.4.1 Basic Assumptions

The system is robust and available 24/7 for the user. Because any emergency/ incident can happen anytime the system should be always available. In case of disaster, the number of users using the system can increase drastically so software should use computer resources optimally.

Also it should be able to serve a large number of requests. The user of the system should have basic knowledge of web applications.

3.4.2 Operating System

IRS has to be built using C# asp.net, WCF Service and SQL Server. Existing IT infrastructures can integrate with it because it's based on SOA.

3.4.3 Web Server

IRS should be able to run on a machine having ISS and SQL Server in it.

3.5 System Features

System features based on the functional requirements extracted from NDMA framework and Virtual Organization (VO) concepts are as follows:

3.3.1 Competency/Resource repository of organizations

IRS allows the maintenance a complete database of resources/competencies of different organizations. For this purpose IRS allows the organization registration process. Moreover, the system supports the features that allow the automatic updating of the central repositories whenever a particular organization updates the status of its resources in their own system.

3.3.2 Competency Analysis Module

IRS supports the competency analysis of the organizations. This feature can be used whenever a new project is created, and a set of required competencies or required resources need to be generated. Based on the tasks involved in a particular operation, the system helps in identifying the concerned organizations with relevant skills/competencies as well as the available resources.

3.3.3 Team formation and Task management (Workflow System)

IRS helps in developing a workflow to support the tasks associated with a particular type of disaster.

3.3.4 Project Tracking

The system provides means for progress tracking for the tasks that can be performed in a particular project.

3.3.5 Assessment/ Grading

IRS supports the evaluation of various participating organizations for future reference and the overall performance measurement of a particular operation. Time of completion, number of resources available/provided can be used as criteria for the evaluation.

3.3.6 Team Selection

This feature allows user to select a team to undertake a task of rescue or relief operation once a calamity has struck.

3.3.7 Appoint Tasks

This feature allows appointing tasks to different teams or team leads. Once a workflow is generated based on disaster type, the different tasks enlisted in the workflow are appointed to teams based on their areas of expertise.

3.3.8 Project Status Check

Each organization that are subscribed to IRS can be able to publish its status on IRS, so that a track of current activities can be kept. This feature also helps in planning and executing future tasks.

3.3.9 Login/Access Right

IRS allows users to login based on their hierarchy in the organization. Different users have different features visible to them based on his role in the organization.

3.3.10 Record maintenance and Report generation

This feature allows user to maintain records of organizations registered with the system. IRS have complete record of resources available, tasks

and services performed by different teams during rescue and relief operations. The records also be useful in future incidents as well as in studies of disaster management. Different users have different access level to database repository. The database is designed in such a way to protect data integrity and confidentiality.

3.3.11 MAP

IRS has fully integrated mapping functionality which allows any location-based data to be visualized on a map. Maps provide situational awareness which is essential when either planning to prepare for or respond to a disaster.

3.3.12 Messaging

In the complex domain of Disaster Management, communication is critical. IRS provides support for messages to be sent by Email and SMS. Distribution Groups can be set up to allow messages to be easily sent to many people at once. Users are able to search for specific information and subscribe to receive update messages when new information is added. Interactive messages can also be set up to allow people to send short message queries to IRS and receive automatic responses.

3.3.13 Alert App

A mobile based app that alert the system by the mobile user when the particular disaster occurs. This app would contain speech to text converting feature.

3.6 External Requirements

IRS is a web application so it has some specific interface requirements elaborated in this section.

3.6.1 User Interface

IRS is a web application and will perform actions based on user input. This requires that the interface of IRS should have an easy learning curve for

the user. Most of the important features should be visible to the user and no functionality should be hidden. It should be easy for user to perform basic actions and data available to the user should be displayed in a meaningful way.

3.6.2 Hardware Requirements

The hardware requirements for IRS should be high enough so that the system can perform optimally without any compromise in performance under different levels of loads and stress.

3.6.3 Software Requirements

The Software required for our system are: Visual Studio 2013, SQL Server 12, Java Development Kit, a pre-requisite for android platform, NetBeans IDE for Java, Android-SDK version 4.2 of Level 18 API.

3.7 Other Non-Functional Requirements

Certain other functionalities are required based on performance and response of IRS. IRS has to be efficient in its response and operation. The product domain requires that the software is optimized in terms of performance. The data flow should happen in the most efficient way. The system performance shouldn't be affected when a new disaster happens and a new project is created.

3.8 Software Quality Attributes

Quality attributes of IRS are described in this section. By following these attributes quality of IRS has to be improved.

3.8.1 Runtime System Qualities

At run time IRS has to provide its users functionalities so that they can perform post disaster activities of rescue and relief efficiently and effectively. Some of the runtime qualities that should be considered in development of IRS are described here.

3.8.1.1 Functionality

IRS must provide functions to create project and track project status. It should also provide functionalities to develop a work flow. IRS must provide functions of task and resource appointment, status of tasks and resources.

3.8.1.2 Performance

IRS performance shouldn't be affected in case a new project is created and different organizations login at the same time.

3.8.1.3 Availability

IRS should be available 24x7 since any incident can happen at any time.

3.8.1.4 Usability

Usability is an important criterion in development of IRS. The system should present all functionalities in such a way that nothing is missed by user. IRS will also provide a lot of data inputs and outputs to the end user so the interface present data in a proper standard format.

3.8.2 Non-Runtime System Qualities

These are qualities of IRS which are required to make this software useful for further enhancements and future development as well as extending system to different environments.

3.8.2.1 Modifiability

IRS must support modifiability so any further improvements or features are easy to incorporate

3.8.2.2 Portability

IRS should be able to run in different computer environments. The IRS server should be platform independent and support interoperability.

3.8.2.3 Reusability

The different functionalities IRS is providing should also be available as a stand-alone project. So if another system is being developed which needs functionalities of IRS, the system (IRS) should be easy to understand to reuse it.

3.8.2.4 Integrate-ability

Different components of IRS should work together in correct manner so that IRS can be used in the most useful manner. Also different IT infrastructures should be able to integrate with IRS.

3.8.2.5 Testability

Different quality tests should be performed so that IRS is free from faults and perform according to requirements specified in this chapter.

3.9 Other Requirements

IRS should be robust enough so that it can perform in the best manner when the system is under different loads and stress. If during a disaster the system fails, a backup server must be present. The database backup timings should also be defined. System should also validate any data user enters. IRS database repository should also maintain data integrity and confidentiality.

3.9 Summary

This chapter describes the requirements of the system as described by Project Supervisor. It includes interface, functional and nonfunctional requirements along with the main features, system would provide to the end user. These requirements have been set after checking the feasibility of the system. These requirements have been considered as the fundamental principles for testing and standardization of the product.

Chapter 4

System Design

4.1 Introduction

This chapter describes the design specifications of IRS. The design specifications have been developed using requirements described in chapter 3. This chapter provides details about system structure and architecture as well as database architecture.

4.2 System Overview

IRS is a software system to facilitate in post disaster activities of rescue and relief. IRS provides certain functionalities to help organizations carry out rescue and relief activities efficiently and effectively. Also it helps to locate and allocate resources effectively during these activities and develop an effective work plan to carry out these activities.

4.3 Assumptions and Dependencies

Basic assumption for development of IRS is that system should be available 24x7 since a disaster can happen at any time. The server should be able to handle a large number of requests especially when a disaster occurs.

4.4 System Requirements

The software is Windows platform dependent since programming language used to build system is C#. But since it's a web application browser version greater than Internet Explorer 6 or Mozilla Firefox 3 or Chrome or equivalent are recommended.

4.5 General constraints

IRS has to provide functionalities described in chapter 3 but in order to enhance software usability few constraints are applied which are described below.

The server machine must use IIS based application. The web application has been designed for browsers supporting CSS 3.0. The user must have basic knowledge of Web applications and database repository is developed on SQL Server 12.

4.6 Architectural Strategies

IRS is a distributed application and is based on 3 tiers architecture. Figure 4-1 shows an overall picture of IRS. Layered Architecture (3-TiIRS) is used to implement IRS. From a high level perspective, a service-based solution can be seen as being composed of multiple services, each communicating with the others by passing messages. Conceptually, the services can be seen as components of the overall solution. However, internally, each service is made up of software components, just like any other application, and these components can be logically grouped into presentation, business, and data layers. Other applications can make use of the services without being aware of the way they are implemented.

IRS has also been built using service oriented architecture, publishing functionalities as web services. Service oriented architecture is in essence design principles which are used during computer system development and integration. A system developed on SOA will have different functionalities offered as interoperable services which can be used within multiple, separate systems from several business domains [5].

The major modules of IRS are designed as Web Services providing an interface to the client to subscribe and then invoke the necessary service and use the functionalities it provides.

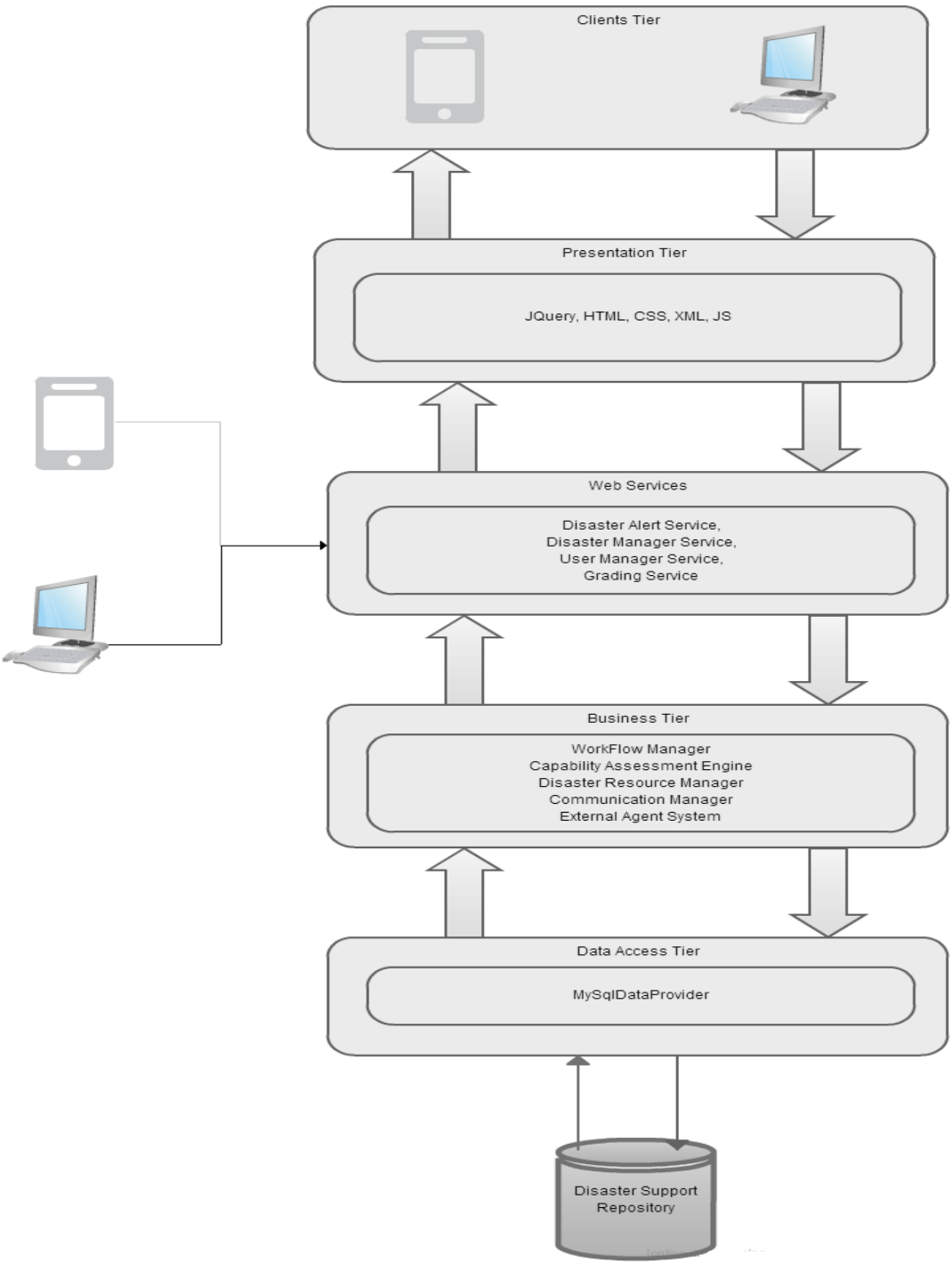

Figure 4 - 1 Three Tier Architecture

4.7 System Architecture

To manage an emergency through IRS, a new project will be created. There are certain details the system needs as an input to create a new project and then develop a work flow for different rescue and relief

activities. IRS functionality is exported to the interface by web services. These web services classes. Which further access database to insert or retrieve data.

The component diagram of IRS is shown in figure 4-2.

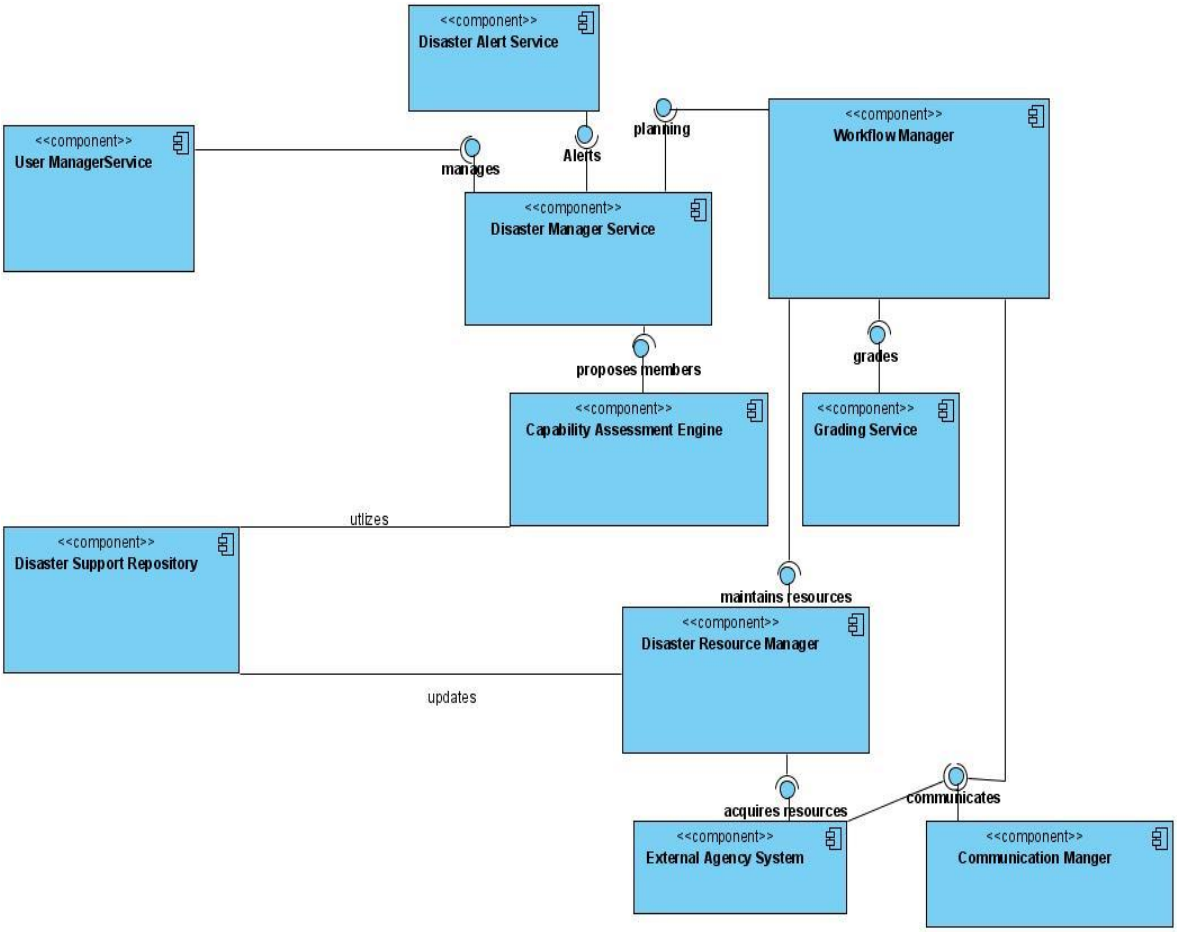

Figure 4 - 2 Component Diagram

Once a disaster occurs the user enters parameters for the disaster to setup a work flow/plan for rescue and relief efforts. The project is open till the time the tasks given to various teams are not completed. When a project is open it is constantly monitored and status of different teams and tasks are updated regularly once all tasks are performed and are reported as complete the project is closed. Then the tasks are reviewed whether a team performed the task successfully or not or whether the results of the tasks were satisfying. Based on these results the teams are graded. This

is helpful in future activities where task allotment could be made easy by allotment of tasks on basis of team's past record and task's complexity level.

Once a rescue or relief operation is completed Grading service is used to grade organization on that particular task. This will not only help in evaluation of tasks performed and performance of different organizations but also help in future projects by giving complex tasks to more competitive organizations. It will also help organization to learn and improve their operations.

The classes within the components are further connected to database repository.

4.7.1 User Support Service

User Support Service and Disaster Support Service is a web service provides functionality for system login/access as well as for user registration.

4.7.2 Disaster Manager Service

This web service is the gateway between the classes that maintain the disaster related information and the front end. It provides web methods for all information on different disaster projects like creating a new project, viewing existing projects, creating new tasks and viewing existing tasks, resource management, updating status on different tasks and inbox for viewing messages as well as creating messages.

4.7.3 Disaster Alert Service

This component is developed as a web service and it contains operations related to providing alerts and notifications to clients which have subscribed to IRS.

4.7.4 Workflow Manager

This component contains classes which generate a workflow for a new disaster management project. The workflow tasks are then assigned resources and forwarded to respective teams.

4.7.5 Capability Assessment Engine

Capability Assessment Engine calculates capability of different teams based on their performance on different rescue and relief activities and how effectively they performed those activities.

4.7.6 Resource Manager

This component contains classes which are collectively used to manage resource. This component's classes are accessed by web methods defined in Disaster Manager Service. The different methods allow IRS to manage resources it has acquired of different organizations and also functions for organizations to update their resources and their status.

4.7.7 Communication Manager

This component contains classes which handles communication between different organizations which are working on the same project. It provides functions to send messages to different organizations as well as to post information and updates on a message board.

4.7.8 Grading Service

Grading service is a web service consisting of functions defined in capability assessment engine. It's the interface between client and capability assessment engine, exposing all function which are related to grading an organization based on its performance in various activities.

4.7.9 External Agency System

This component contains classes which handles project related information. This package's classes are accessed by web methods defined in Disaster Support Service.

4.7.10 Support Repository

This is the database of the whole software system. All the user, organization and project related data is maintained. It has been designed by keeping data integrity and confidentiality principles in mind. Also database normalization principles are applied during database design.

4.8 Use Case Diagram

The use case diagram of IRS has been given as Figure 4-3. This diagram describes the interaction of user and the system.

4.8.1 Basic Flow of IRS

The software has three types of users, admin, manager operations (MO) and organizations. All the three types of users have different access level to the system and its data and can perform functions assigned to their respective roles.

The admin user can create new user or organizations or register organization. It can also deem a user as active or inactive based on some policy. The MO user is basically the project manager; he/she can also manage more than one project at a time. It can perform all actions related to a project like project creation, appoint tasks, check task status, send receive messages and grade users/ organizations based on their performance on a given task.

The organization user can view project and tasks it has been appointed to, it can also update the organization resources whether new resources are added or existing resources are occupied or busy. It can update MO on task status and can send and receive messages to communicate with others on the same project.

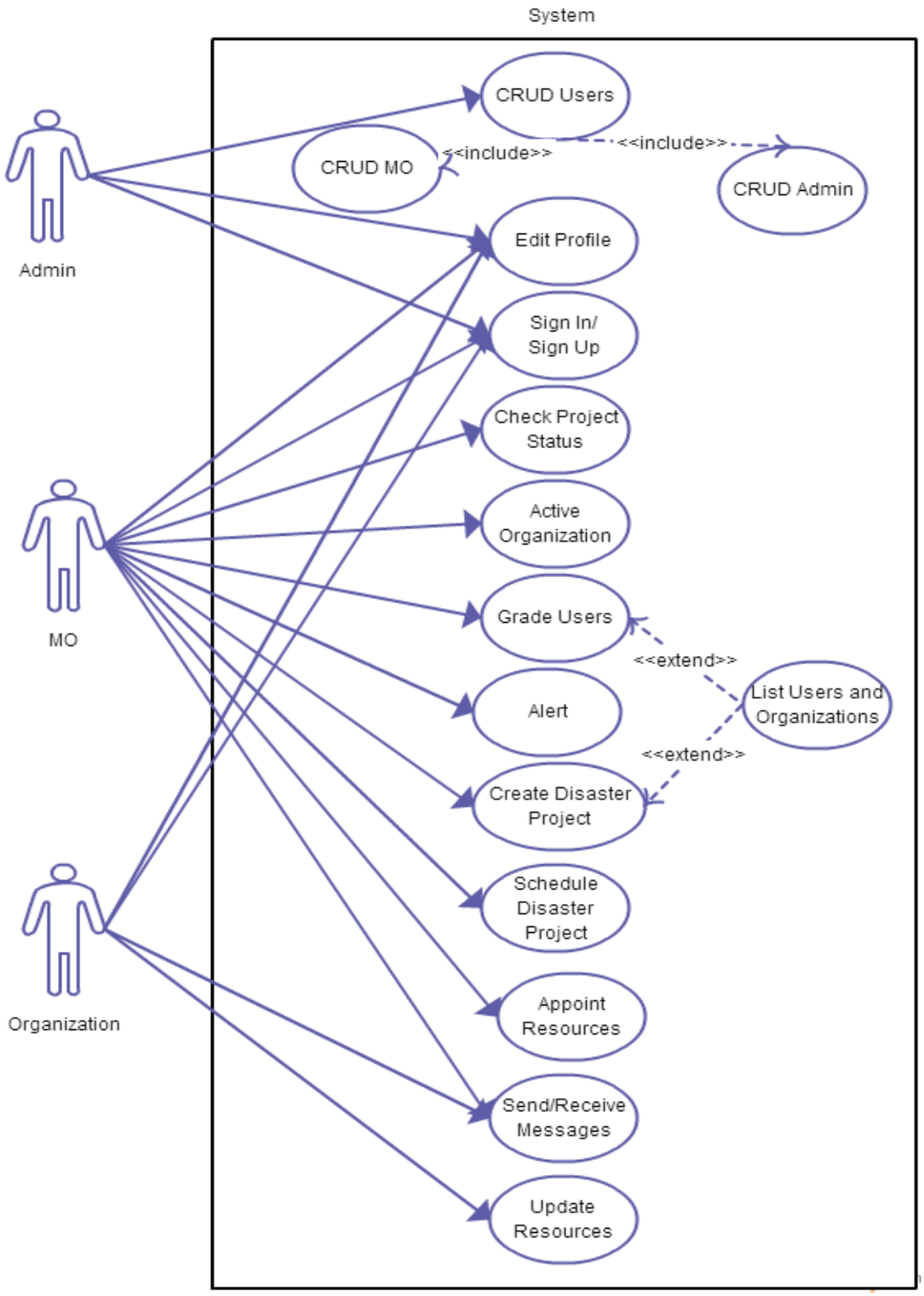

Figure 4 - 3 Use Case Diagram

4.8.2 Post Conditions

System has performed the action which was requested by user through web interface.

4.8.3 Alternate Scenarios

No alternate scenarios are defined because the objective of the system is described in chapter 1 and the software is being developed by following the given requirements.

4.9 Class Diagram

Figure 4-4 shows the class diagram of IRS. All the classes shown in the class diagram are further described in the sub sections.

4.9.1 Disaster Support Service

This is the main service to provide system access to users and allow them to access different system functions based on their roles. This service acts as an interface between web application's front end and back end Java classes.

4.9.2 Disaster DB Access

This is the main class to access database objects. All the data operations are performed through this class. All other classes send or retrieve data from database through this class. It acts as an interface between all software classes and database system.

4.9.3 Login System

This class contains data and functions for users and organizations to login. This class simply authenticates user information and checks which role is assigned to user based on which user is given access to different areas and functions of the software.

4.9.4 Organization System

This class provides data related to an organization. It also provides functions for MO to see list of organizations and narrowing search by different parameter inputs. Also this class has the important function for MO to register organizations. This class is inherited from User base class.

4.9.5 System Rights

This class has information roles and responsibilities for each user type. So all the roles and their details and functionalities offered for different roles are in this class.

4.9.6 User

This class has basic information about user and organization. This is the base class which is extended by user system and organization system classes which have details specific to a user or organization.

4.9.7 User System

This class is extended from user and adds additional detail which is more specific to a user which, in this case, can be the Web application administrator or manager operations.

4.9.8 Disaster Manager Service

This is the second web service providing various Web methods which are used to provide functionality regarding a disaster. All functionalities of the system related to disaster management are exported as web methods by this service.

4.9.9 Disaster Project

This class contains data about a disaster project and functions to view existing project or create a new one.

4.9.10 Task

This class has all the data about a task which is related to some disaster project. It has functions to create new tasks and set up priorities or set up a workflow of different tasks which may or may not be interrelated.

4.9.11 Message board

This class keeps track of all messages, it provides functions to create new message, post message to desired authority and view messages.

4.9.12 Organization Resource

This class keeps a record of organization's resources. Whether a new resource is added, a resource is busy or free all information is maintained through this class. This class has a high importance in project because efficient resource management will lead to better performance in different tasks or activities on a project.

4.9.13 Task resource

This class keeps a track of resources which are currently allocated to a task. This class is extended from organization resource class to add the extra functionality if a task has acquired a resource.

4.9.14 Task Status

This class monitors status of an individual task and provides functions to set status, update status, and view status to concerned users.

4.9.15 Grading

This class is used for grading different tasks so to track performance level of different organizations and help shaping future decisions like task appointment and resource appointment to improve overall disaster management strategies.

4.9.16 Disaster Grading Service

This is the service which exposes grading service functions to authorized users.

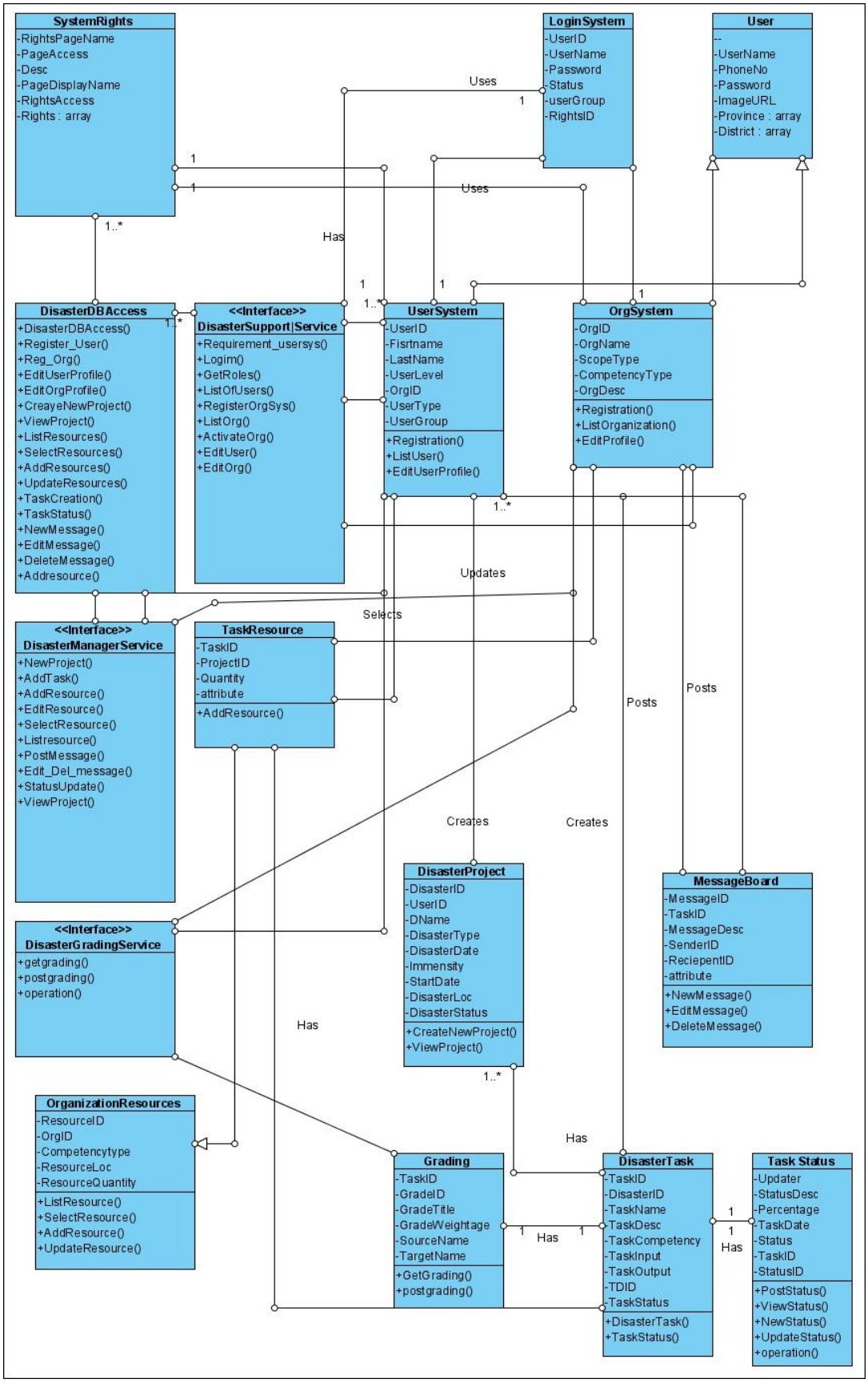
SystemRights
-RightsPageName
-PageAccess
-Desc
-PageDisplayName
-RightsAccess
-Rights : array
LoginSystem
-UserID
-UserName
-Password
-Status
-userGroup
-RightsID
User
--
-UserName
-PhoneNo
-Password
-ImageURL
-Province : array
-District : array
Uses
Uses
Has
DisasterDBAccess
+DisasterDBAccess()
+Register_User()
+Reg_Org()
+EditUserProfile()
+EditOrgProfile()
+CreayeNewProject()
+ViewProject()
+ListResources()
+SelectResources()
+AddResources()
+UpdateResources()
+TaskCreation()
+TaskStatus()
+NewMessage()
+EditMessage()
+DeleteMessage()
+Addresource()
<<Interface>>
DisasterSupport|Service
+Requirement_usersys()
+Logim()
+GetRoles()
+ListOfUsers()
+RegisterOrgSys()
+ListOrg()
+ActivateOrg()
+EditUser()
+EditOrg()
UserSystem
-UserID
-Fisrtname
-LastName
-UserLevel
-OrgID
-UserType
-UserGroup
+Registration()
+ListUser()
+EditUserProfile()
OrgSystem
-OrgID
-OrgName
-ScopeType
-CompetencyType
-OrgDesc
+Registration()
+ListOrganization()
+EditProfile()
Updates
Selects
<<Interface>>
DisasterManagerService
+NewProject()
+AddTask()
+AddResource()
+EditResource()
+SelectResource()
+Listresource()
+PostMessage()
+Edit_Del_message()
+StatusUpdate()
+ViewProject()
TaskResource
-TaskID
-ProjectID
-Quantity
-attribute
+AddResource()
Posts
Posts
Creates
Creates
<<Interface>>
DisasterGradingService
+getgrading()
+postgrading()
+operation()
DisasterProject
-DisasterID
-UserID
-DName
-DisasterType
-DisasterDate
-Immensity
-StartDate
-DisasterLoc
-DisasterStatus
+CreateNewProject()
+ViewProject()
MessageBoard
-MessageID
-TaskID
-MessageDesc
-SenderID
-RecipientID
-attribute
+NewMessage()
+EditMessage()
+DeleteMessage()
Has
OrganizationResources
-ResourceID
-OrgID
-Competencytype
-ResourceLoc
-ResourceQuantity
+ListResource()
+SelectResource()
+AddResource()
+UpdateResource()
Grading
-TaskID
-GradeID
-GradeTitle
-GradeWeightage
-SourceName
-TargetName
+GetGrading()
+postgrading()
Has
Has
DisasterTask
-TaskID
-DisasterID
-TaskName
-TaskDesc
-TaskCompetency
-TaskInput
-TaskOutput
-TDID
-TaskStatus
+DisasterTask()
+TaskStatus()
Has
Task Status
-Updater
-StatusDesc
-Percentage
-TaskDate
-Status
-TaskID
-StatusID
+PostStatus()
+ViewStatus()
+NewStatus()
+UpdateStatus()
+operation()

Figure 4 - 4 Class Diagram

4.10 Entity Relationship Diagram

IRS has a database repository to maintain all the records. ER Diagram shows how data is modeled and relationship between different data are identified and stored in a database management system. ER Diagram is shown in Figure4-5.

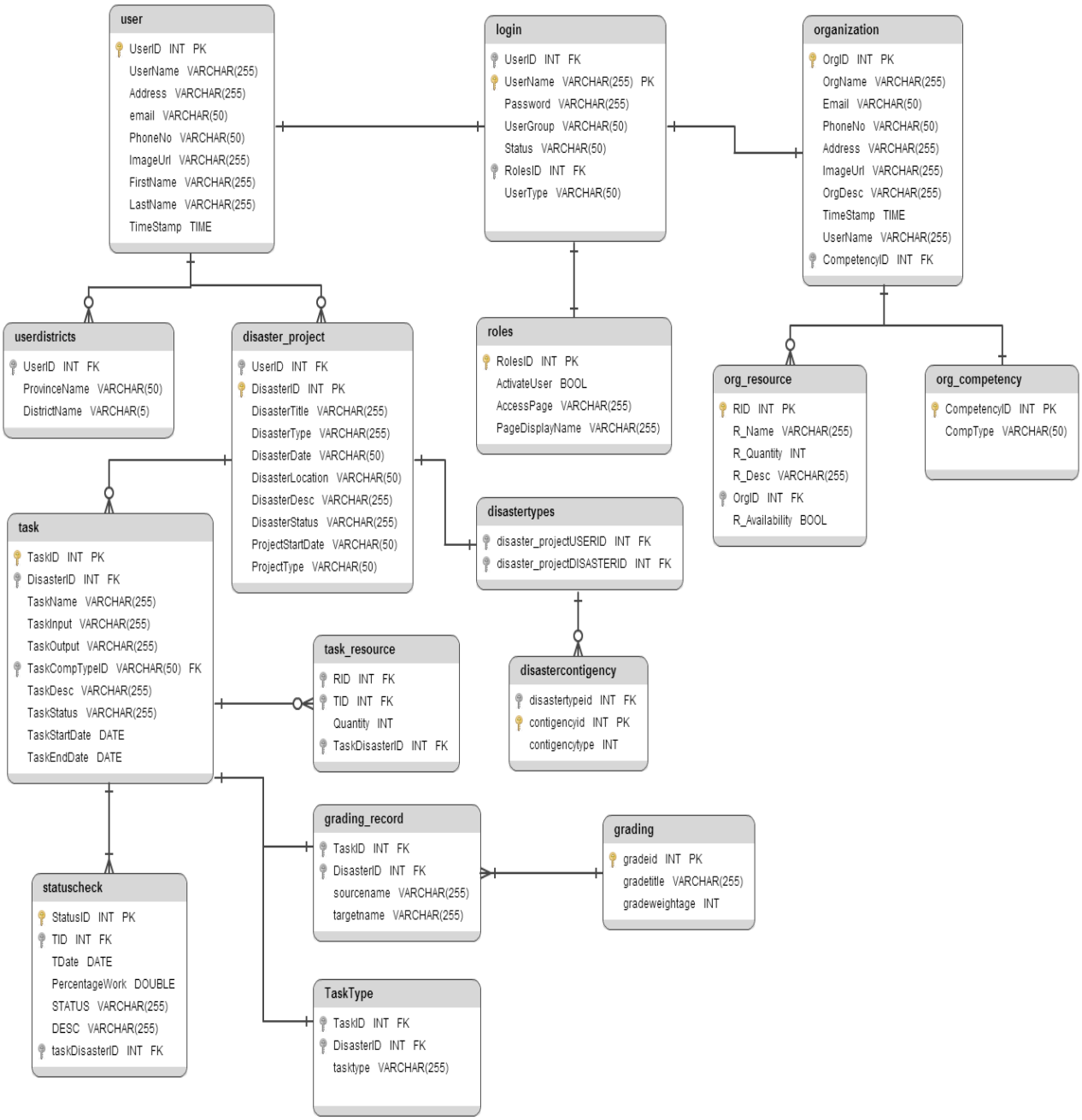

Figure 4 - 5 ER Diagram

4.11 Detailed Design

In order to ensure that the correct software is being build according to requirements defined in chapter 3; low level design of IRS is build using requirements, component diagram and class diagram. In this section a few

of these diagrams related to a particular system scenario are described. The rest of system diagrams are attached as Appendix A.

4.11.1 Activity Diagram: Task creation

The Figure 4-6 shows how the activity of task creation is performed. When a new project is created a list of tasks is created. Once a task is created it can be assigned to an organization and multiple resources can be assigned to it. After a task is appointed its progress is tracked. The exit can be by two ways, task is completed or aborted.

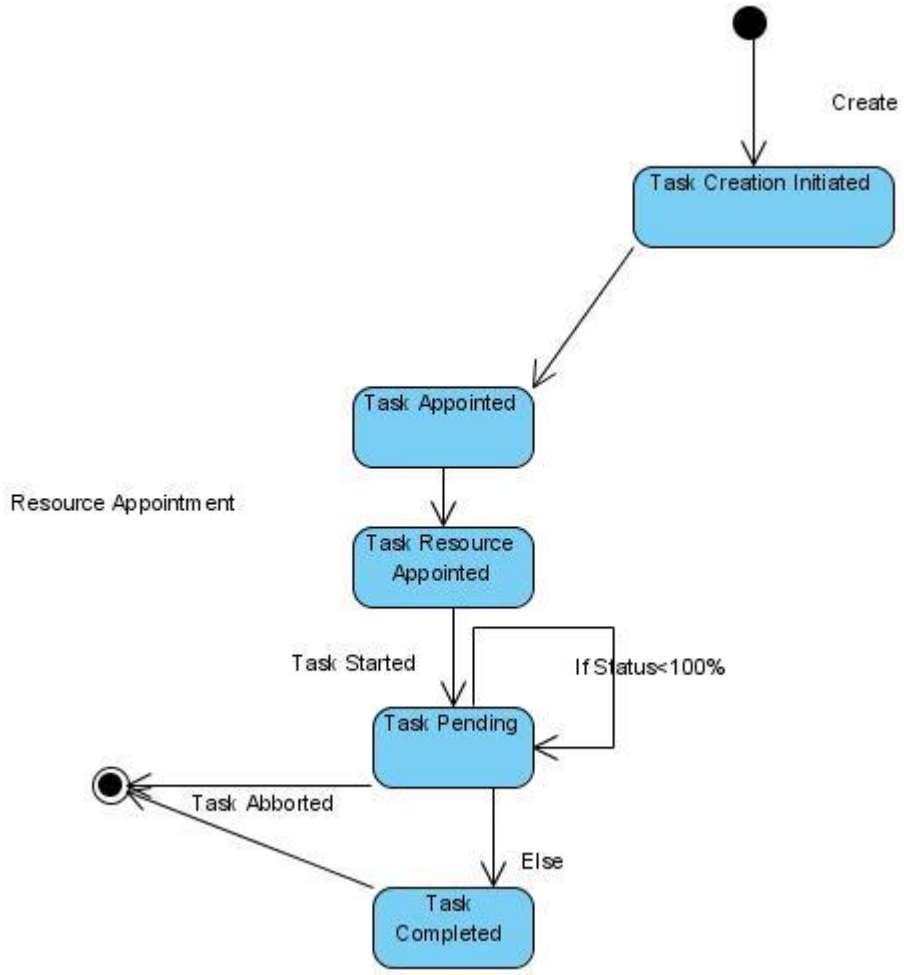

Figure 4 - 6 Activity Diagram of Creating Task Process

4.11.2 Sequence Diagram: Adding resource to task

Figure 4-7 shows the sequence of steps and timeline of different objects involved in adding resource to a task. First resource selection method is

invoked that shows a list of available resources. Then the resources selected and database is updated. Once a resource is selected it's assigned to a resource as shown in the timeline. Also the task creation process is completed and the respective organization is notified.

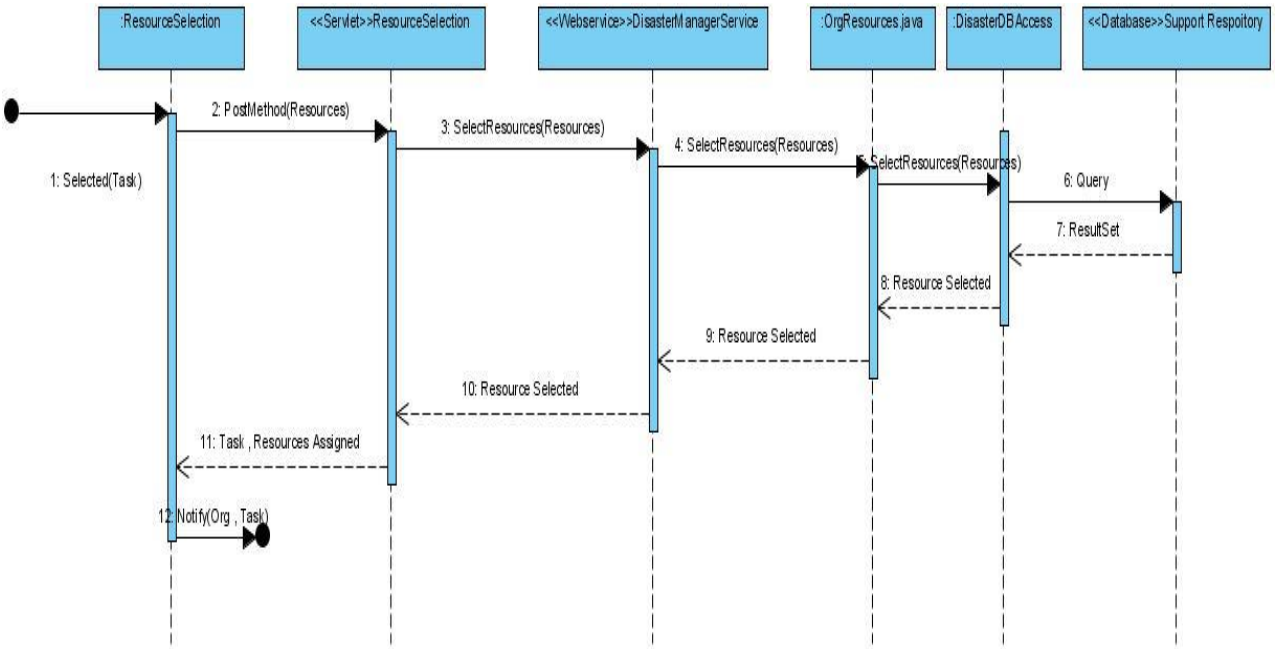

Figure 4 - 7 Sequence Diagram of adding resource to task

The above diagram shows the sequence of steps and timeline of different objects involved in adding resource to a task.

4.11.3 State Machine Diagram: Organization

State machine diagram shows the states an object can move between when it's active in a software system. The diagram below shows the different states an organization object can acquire in its lifetime.

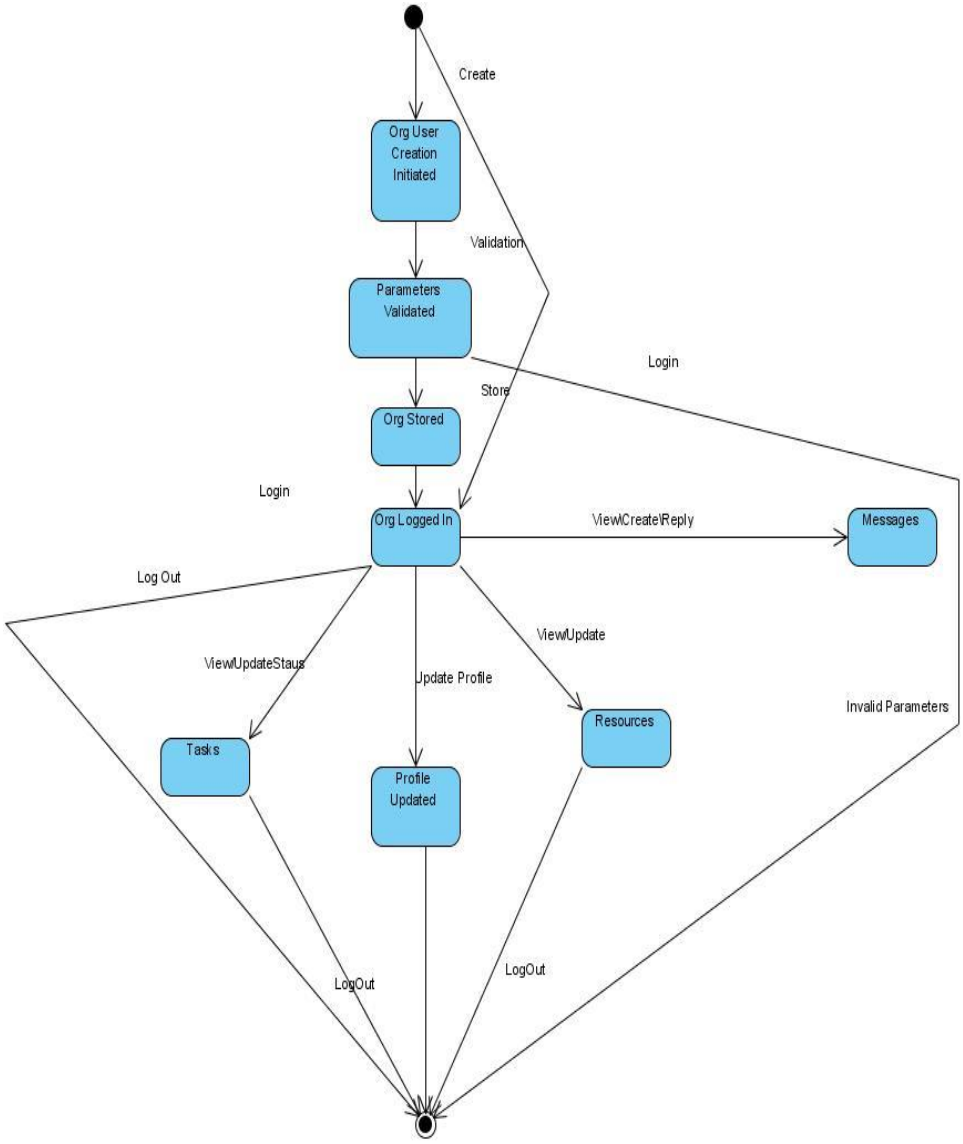

Figure 4 - 8 State Machine Diagram: Organization

4.12 Web Modeling

This section provides design details of web application interface. It includes navigation model and presentation model.

4.12.1 Presentation Model

Below figure 4-9 shows the presentation of different objects and navigation links on dashboard page of an admin.

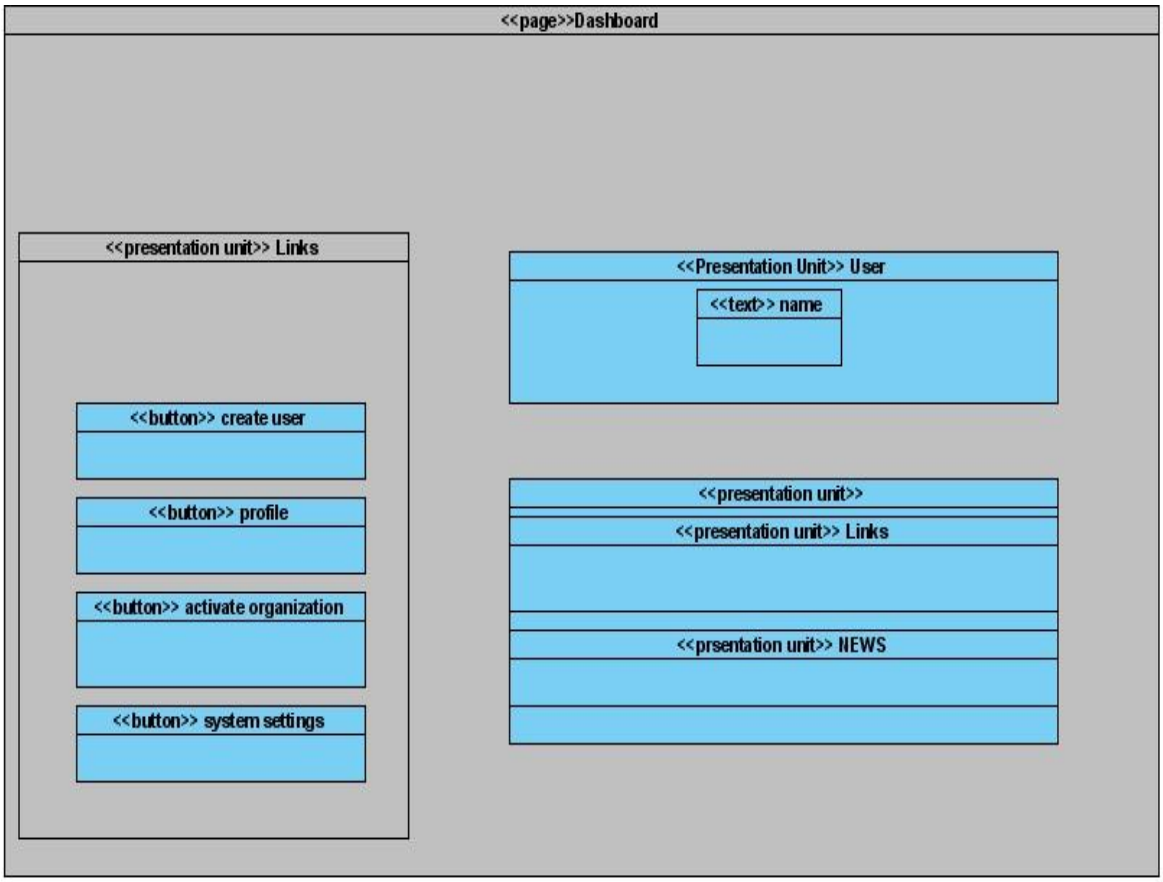

Figure 4 - 9 Presentation Model: Admin Dashboard

4.12.2 Hypertext Structure Model

Figure 4-10 explains the hypertext structure model for an organization user. The figure shows different classes associated with an organization user and organization user can navigate through in hypertext model. The organization user can access only the projects it's currently working on through projects navigation class. The resource class basically keeps a record of all the resources an organization has and allows an organization to update resources while profile class allows it to change different attributes of its profile like business address, telephone number and so on.

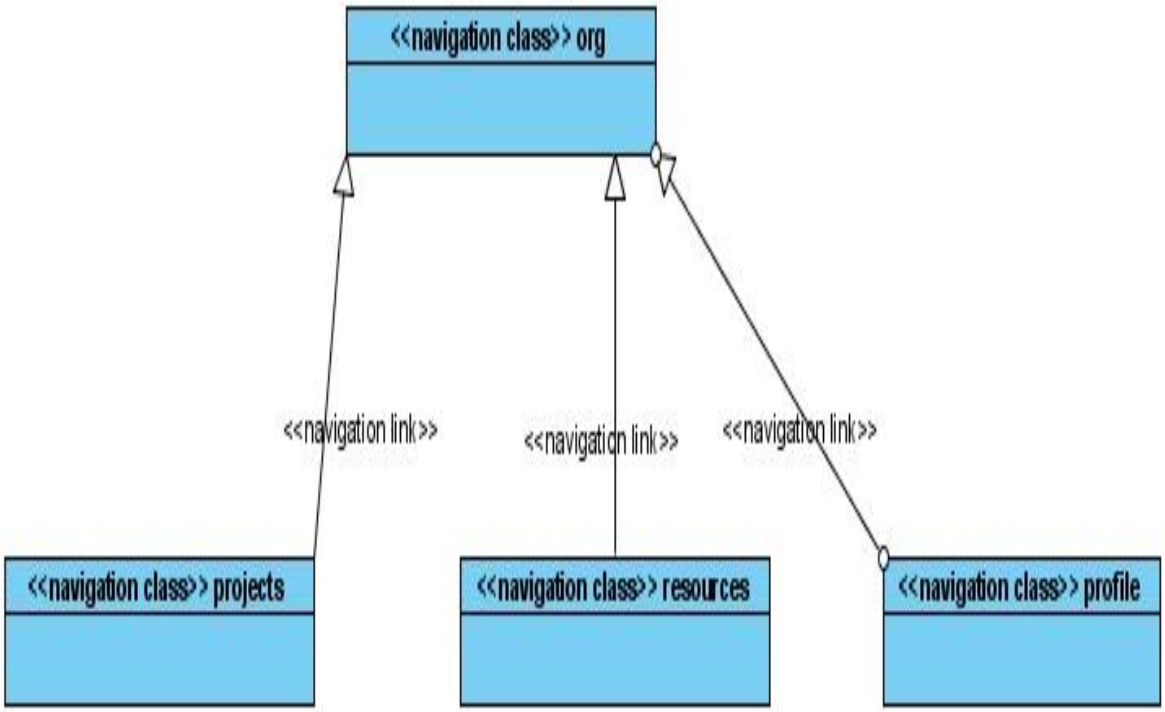

Figure 4 - 10 Hypertext Model: Organization

4.13 Summary:

IRS has to perform in real time environment and has to be available 24x7 to its users. This chapter described the design of the software taking into consideration different assumptions and constraints that applied on the system because of its goals and requirements. Component Diagram, Class Diagram and Use case diagram have been added to explain system functionalities. ER Diagram has been added to explain database design and relationships between different database objects. A few low level design diagrams are shown to elaborate how system behaves internally and how objects change states when the system is put into operational environment.

Chapter 5

Implementation Details

5.1 Introduction

Detailed design of the Incident Reporting System (IRS) are discussed in the previous chapter. This design is transformed into an application by using various technologies. The implementation details are discussed in the following sections giving details of the system's internal working.

5.2 Tools and Technologies

Web applications can be implemented using various technologies. These include server side technologies (JSP, Servlets, PHP, ASP.NET) and different client side technologies (JavaScript). However, the selection of these technologies depends on the system architecture and its detailed design. Figure 5-1 shows an overview of different tools and technologies used in the IRS along with their interaction. ASP provides the user interface of the application and handles the data exchange made by the user to the application. This data exchange is in the form of parameters passed to the system. The proxy classes' object then initiates a call to a required function of the Web Service. The call is handled by the Web Service End Point. The core functionality of the system is handled by Classes at the Service End Point. Classes then communicates with the database to perform different transactions. The Web Service End Point returns result to the Proxy Object.

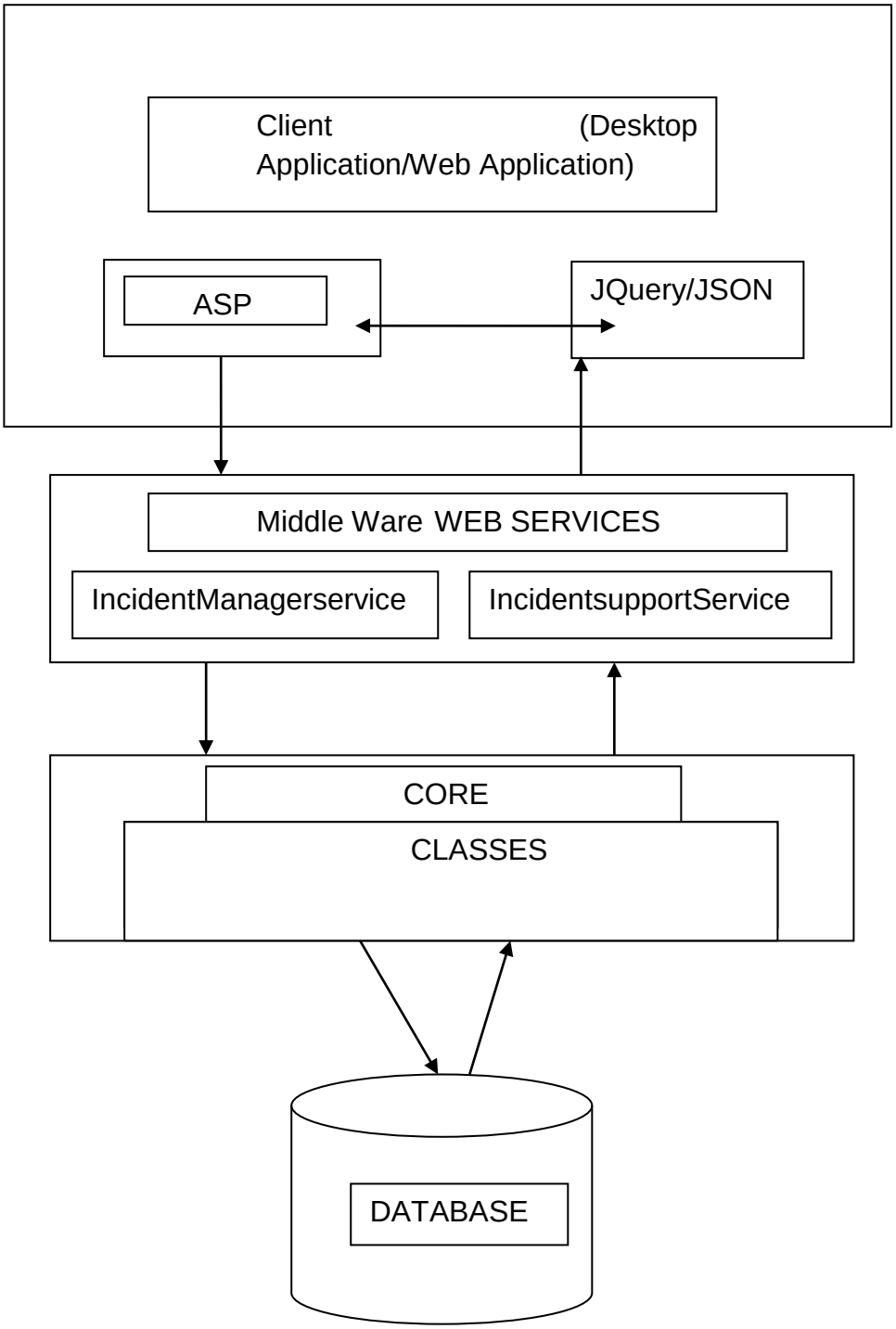

Figure 5 - 1 Overall System Diagram

The functionalities provided by the main system are exposed as Web services.

5.2.1 Web Services

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or

frameworks [6]. Web services have an interface and a client can first register with web service and then access the functionalities through its interface.

5.2.2 WSDL

Web Service Directory Language (WSDL) is Web Service's language which contains information about a specific web service. WSDL contains information like transport, protocol, service location, operations or functions available and payloads transferring to and from web service.

A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the operations listed in the WSDL file using XML or HTTP. A WSDL document uses the following elements in the definition of network services:

Types– a container for data type definitions using some type system (such as XSD).

Message– an abstract, typed definition of the data being communicated.

Operation– an abstract description of an action supported by the service.

Port Type–an abstract set of operations supported by one or more endpoints.

Binding– a concrete protocol and data format specification for a particular port type.

Port– a single endpoint defined as a combination of a binding and a network address.

Service – a collection of related endpoints [7].

5.2.4 SOAP MESSAGE

Simple Object Access Protocol (SOAP) is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Remote Procedure Call (RPC) and Hypertext Transfer Protocol (HTTP), for message negotiation and transmission. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built. This XML based protocol consists of three parts: an envelope, which defines what is in the message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing procedure calls and responses [8].

It requires profuse attribute specification tags, namespaces, and other complexities, to describe exactly what is being sent. This has its advantages and disadvantages. SOAP involves significantly more overhead but adds much more information about what is being sent. If you require complex user defined data types and the ability to have each message define how it should be processed then SOAP is a better solution than XML-RPC.

5.2.4.1 HTTP GET AND POST MESSAGE

GET Requests a representation of the specified resource. Requests using GET (and a few other HTTP methods) "SHOULD NOT have the significance of taking an action other than retrieval". [9]

POST Submits data to be processed (e.g., from an HTML form) to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both [10].

5.2.5 jQuery

jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. jQuery is free, open source software. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications [12].

5.2.6 JSON

JSON (an acronym for JavaScript Object Notation) is a lightweight text-based open standard designed for human-readable data interchange. The JSON format is often used for serializing and transmitting structured data over a network connection. [4]

5.2.10 AJAX

Asynchronous JavaScript and XML (Ajax) is a group of interrelated web development methods used on the client-side to create interactive web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. [15]

5.3 User Interface

Since IRS is a web application so user interface is an important part of it and it is treated as a separate module. The issues addressed in the user

interface design are that no functionality is hidden from user and data should be presented in a clear way to end user so nothing is missed by him.

The interface is developed using ASP, JQuery and AJAX. The interface is kept separate from business layer by making a new web service project. CSS 3.0 is used so that most old browsers are compatible with the web application but still a few features won't show up correctly on browsers older than Internet Explorer 6.0.

JQuery is an open source JavaScript library used for client side scripting (Writing the program of web application which will run on client side). JQuery is used for form validation in user interface, so when a user is creating a new project for a disaster, registering into the system or using any other form for entering data he shouldn't enter a wrong value or type. This will ensure data is in its correct format.

Ajax standards for asynchronous JavaScript and XML. It is also used to write client side code of a web application and main purpose is to make web applications interactive. Ajax was used in development of user interface for IRS. The reason to use Ajax was it sends data to and retrieves from server asynchronously without affecting the web page or redirecting user to a new page.

In IRS JSON (JavaScript Object Notation) is used in conjunction with JavaScript to generate dynamic data. This data is then send or received from server using AJAX (POST and GET HTML methods) to avoid redirection of user on other page.

5.4 User Management Module

This module covers all functionalities associated with a user. It provide functionalities to create a new user , edit user , authorize user to login into the system and display system features to a user based on his role in the organization. The user management module essentially consists of a web

service which acts as an interface between client side code and server side code. This web service methods than interacts with classes which further interact with database through a single database access class to make up the whole hierarchy.

5.5 Disaster Management Module

This module provides all functionalities that will help user in disaster management and manage different tasks and resources related to a specific disaster. It provides functions to create a new disaster, define a workflow for different tasks related to a disaster, and appoint tasks to resources. Monitor project and task status and collaboration in form of a messaging system are also a part of this module. Also a grading system has been implemented to keep a track of performance of different organizations on different tasks.

5.6 Disaster Support Repository

This is the database system for IRS which contains all the records related to different users, user type, project, tasks, messages, task status. The database has been designed focusing on data integrity and confidentiality.

The database has been developed using SQL Server 12.

5.7 Grading Service

This service basically provides an interface for trust engine which Is used to grade how different organizations have performed rescue and relief activities.

5.8 Disaster Alert Service

This service basically is a messaging service which provides alerts to organizations that have subscribed to IRS. The alerts include the information about a new disaster along with the location and scale of the disaster.

5.9 Summary

Implementation details of IRS are discussed in this chapter. Different functionalities and strategies to develop the system are also discussed. A brief introduction to different tools and technologies employed is also given.

Chapter 6

Testing

6.1 Introduction

To ensure quality of the product, testing is conducted. Accuracy of functions performed by IRS has to be tested and maintained to improve quality of software. Software testing techniques and results obtained are discussed in the coming sections.

6.2 Testing Levels

Separate modules are developed to provide different functionalities of IRS. All of these modules are tested at different levels in their development and after integration. Different levels at which IRS has been tested and results obtained are described in this section.

6.2.1 Unit Testing

Each module is developed and tested individually. Different sets of sample data are used to test all functionalities. The module for user login was developed first. This module is tested for both user login and organization login. Setting up different organizations and user's and admin the login module was tested to see if each type of system user gets has the correct responsibilities assigned and each user type should only be able to view system functionalities he has been assigned to. Also no user should be able to view data on another user. All the tests confirmed that data integrity and confidentiality is maintained and user(s) only see information intended for them. Another aspect is that user shouldn't be able to access any page and session handling was done at all levels. Different tests confirmed that accessing a restricted page without authentication is not possible and user is redirected to login page.

Project creation is the second module developed which has sub modules of workflow management; appoint tasks, communication system, and

resource management. All these sub modules are developed and tested using relevant sample data and then tested at integration level to complete the whole project module. Regression testing is used so that at any level of project creation no wrong data is fed into the system. The expected output was that once all steps of project creation are followed, the system should show all data related to a project and shouldn't miss any relevant data. Also system should be able to generate reports regarding a project by a number of ways; for example if we want system to display resources it should be able to do that by city, district or province level. All the tests were successful showing that data entry and output from system was correct in all fashion.

6.2.2 Integration Testing

IRS's different modules which were developed and tested independently were also tested during integration to ensure system stability. Integration testing helped in ensuring that different modules when combined give complete functionality and nothing is missed or some functionality doesn't give error when integrated with other modules. Integration testing gave us more than 90% results ensuring that most modules were integrated with others as well as compatible. This shows that errors were minimized during integration testing.

6.2.3 System Testing

System testing was performed at the end of development and integration of IRS. Complete system was tested using sample data. User registration, user roles and responsibilities, creating new project and monitoring project all sub modules were tested as a whole using sample data. Almost 90% of test cases were successful ensuring that most of errors and bugs in the system were removed and system was stable enough to perform optimally.

6.3 Box Approach

To test whether IRS functionality is in accordance with the code written box approach testing was done. The two box testing approaches used to test software were black box testing and white box testing. Black box testing is done at various stages of testing by inserting sample data in various components, checking outputs and removing errors. White box testing is done when a separate module is developed (unit level), integration of different modules and sub-modules and at system level. Once the system is developed white box testing was the most performed testing technique.

6.4 Test Cases

The system is thoroughly checked for consistency and for errors using different test cases. Overall system is checked for the following attributes; parameters and resources are displayed correctly in given scenarios, the forms are validated correctly, session handling is done correctly, user interface is easy to use and users get features and options defined by their roles.

6.4.1 Test Case 1

Each form is checked to ensure that for a given a set of inputs the expected output was received. The related test case is shown below:

Scenario: In the Create New MO page if a user selects a province then the options in the district dropdown are loaded based on that.

Given Input: Punjab

Expected Output: All districts filed under Punjab come up in the districts dropdown

Actual Output: All districts filed under Punjab come up in the districts dropdown.

Test status: PASS

6.4.2 Test Case 2

Each form has been validated using the JQuery validation plugin. If a required field is missed or a wrong input is filled in then the form is not submitted and an appropriate error message is displayed. The related test case for one of the scenarios is given below:

Scenario: In the Create New MO page if a field is missed or text is filled in the phone number field which should be all digits an error message should be displayed.

Given Input: Text input in phone number field and form submitted.

Expected Output: Error message is displayed and form is not submitted.

Actual Output: Error message displayed and form not submitted.

Test status: PASS.

6.4.3 Test Case 3

Whenever a user logs in a new session is created. If that session times out or is corrupted the user is logged out of the system and has to sign in again. Also without a valid session ID a user cannot access any page.

Given Input: Username and Password submitted.

Expected Output: Session ID and Cookie are created in browser's temporary file.

Actual Output: Cookie is shown in browser.

Test status: PASS.

6.4.4 Test Case 4

Five students who had not used the system before were given different scenarios and told to execute them. They successfully navigated the system and were able to accomplish their tasks with little or no assistance.

6.4.5 Test Case 5

There are three different types of users who can log into the system: Admin, MO and Organization. They each have their own roles and there are different features which are accessible only to them. To ensure that no user type got access to any page restricted for them rigorous testing was done.

Scenario: While logged in as MO the system settings page was tried to be accessed through the address bar.

Expected Output: Error displayed

Actual Output: Error displayed

Test status: PASS.

6.4.5 Test Case 6

User can send alert of disaster using mobile phone.

Scenario: While logged in from mobile app. After that user enter some detail of alert.

Expected Output: Alert Message successfully shown in website for all users.

Actual Output: Alert Message successfully shown in website for all users.

Test status: PASS.

6.5 Summary

Testing not only maintains the software quality but also improves over all usability of the project. At different stages of development suitable testing techniques were used to ensure product works accurately and efficiently. All errors detected during testing were removed.

Chapter 7

Results and Analysis

7.1 Introduction

IRS has been developed to work in real time environment. Since disaster can happen any second, IRS should always be available to help organizations to track emergencies and plan rescue and relief operations efficiently. The data integrity and confidentiality should be maintained at all levels.

7.2 Results

IRS has been developed to facilitate organizations in post emergency activities or rescue/ relief. The idea was to develop a web application which is at the center of disaster management activities of rescue and relief and helps organization to execute these activities efficiently and effectively. IRS performs all the functionalities defined in chapter 3 system requirements.

All the functionalities have been achieved using the most advanced tools and technologies for development of Web services.

7.3 Analysis

Since IRS is a web application performance, robustness and usability are important features.

IRS code is optimized so that page loading time is minimum and has been tested using multiple connections to server to test its load and stress and how system will perform. The results are more than 90% accurate showing it will perform fairly well with multiple client connections.

Usability is an important aspect of web application and in IRS from the design phase this issue was paid special attention by developing

navigation model for different web pages to show how information will be displayed and how different pages will link to each other.

Another important part of analysis is how IRS compares with existing system. One of the existing systems that exist is SAHANA FOSS Disaster Management System. The functionalities that IRS has and SAHANA doesn't are a central database of resources for all organizations. SAHANA also was developed for rehabilitation efforts and then extend to rescue and relief activities.

SAHANA isn't based on SOA, which means existing solutions of all independent organizations won't be able to fully integrate with it. With web services IRS will save crucial times of other organizations and they can easily integrate with our system. IRS also is focused on one single IT infrastructure for collaboration between all organizations an objective on which it was built and SAHANA wasn't.

7.4 Summary

IRS performs all the functionalities functional and nonfunctional provided in the system requirements. All the important nonfunctional requirements that are essential for a web application to perform effectively are present in the system.

Other systems that are similar to IRS are built on older web technologies and do not use SOA. Also the scope or the disaster management activities those systems perform are different from IRS.

Chapter 8

Conclusion and Future Work

8.1 Introduction

This chapter introduces the achievements of IRS. Suggestions have been presented in further sections for improvement of features and adding new features to IRS. IRS has been developed in such a way to make it easy to expand and reuse code. This will help in extension of IRS to cover further disciplines of incident/ emergency management and also how web services can be used to solve real life problems in an effective way.

8.2 Concept

IRS concept was initiated by the large amount of calamities humanity has faced in recent past and using web application for better management of resources to tackle such conditions. The existing solutions as explained in chapter 2 had limitations which were basically the technologies used for solutions of disaster management weren't compatible with each other. Using web services to develop IRS has helped to make different technologies.

8.3 Future Work

IRS is a start to a wide field of using disaster management IRS can achieve a lot of milestones that can help people manage post disaster activities and resources effectively.

Workflow plans to organizations, thus allowing organizations to develop their own plans for the tasks they have been assigned to. Implementing a Mailing and SMS service for better communication between involved parties. This can further include applications for mobile devices like tablets and smart phones. The scope of system can be extended to include rehabilitation tasks and maintaining a database of shelters. Live streaming can be another future work so the possibilities to extend this system are

limitless and without doubt will help in better management of post disaster activities of relief and rescue.

Dynamic team formation can also be included in the system where a system can suggest on its own the best possible team for a particular disaster. For this purpose rule based system can be developed to choose appropriate teams for a particular disaster.

8.4 Summary

Web services can be used conveniently to help us in real life problems like disaster management. There is a wide range of options in web services to allow organizations to collaborate and work towards the common goal of disaster management. This will not only help in better management of post disaster activities but also prevent another disaster of mismanagement which can lead to not only issues of wasting resource but the bigger picture of losing more human lives.

Appendix A

Low Level Design

Sequence Diagram

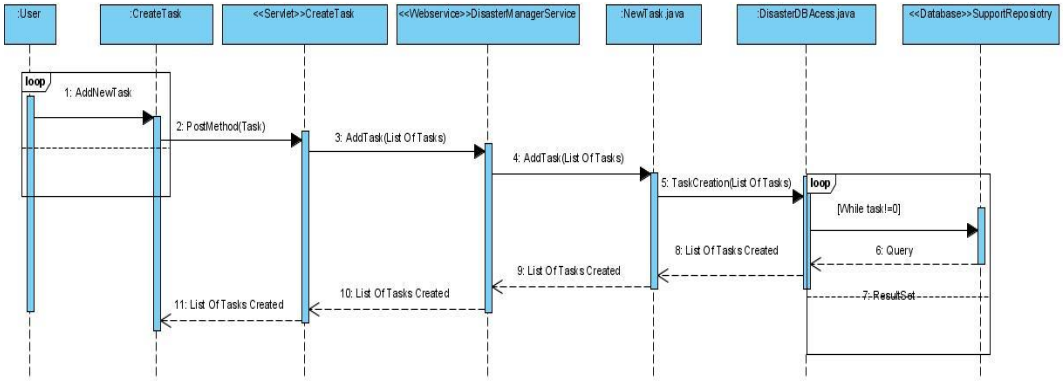

Figure A - 1 Sequence Diagram Add Resource

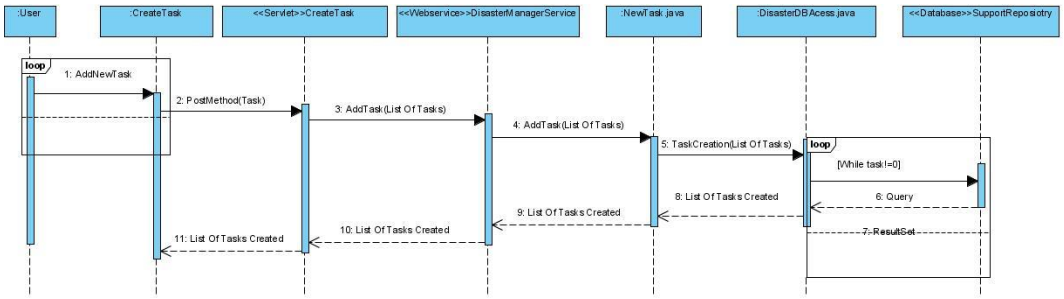

Figure A - 2 Sequence Diagram Add Task

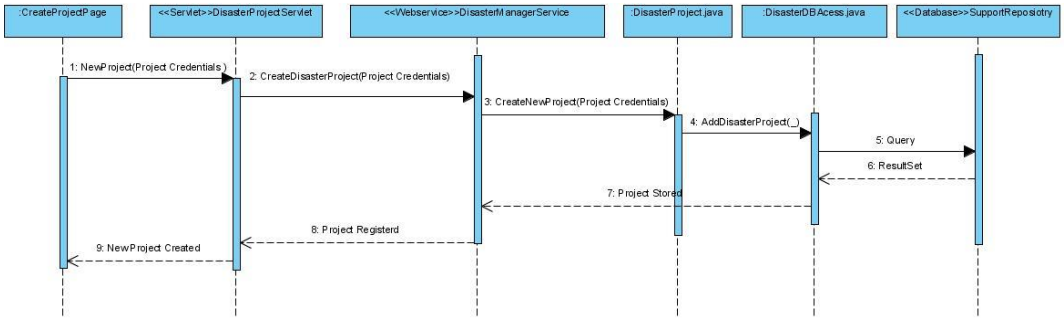

Figure A - 3 Sequence Diagram Create Disaster Project

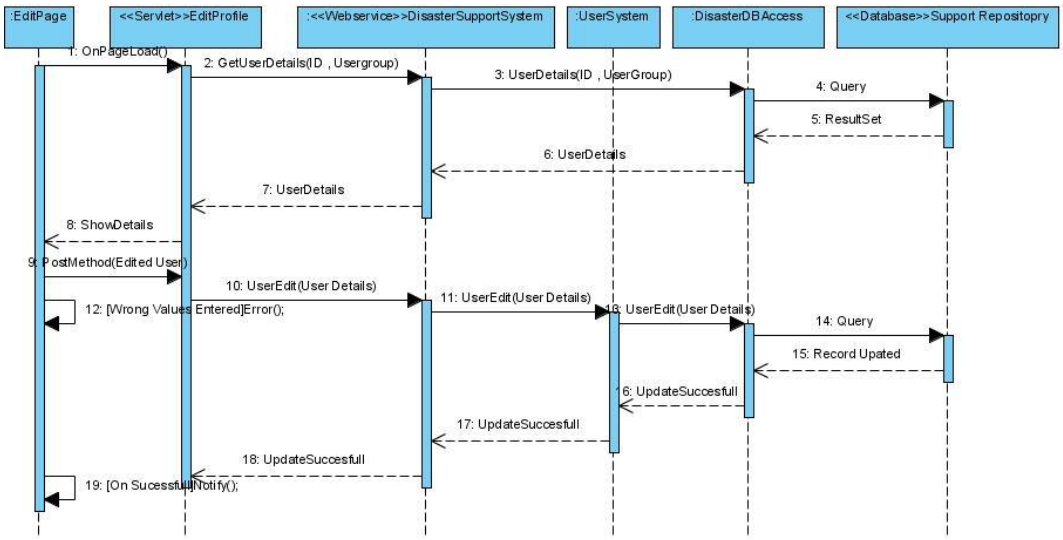

Figure A - 4 Sequence Diagram Edit User Profile

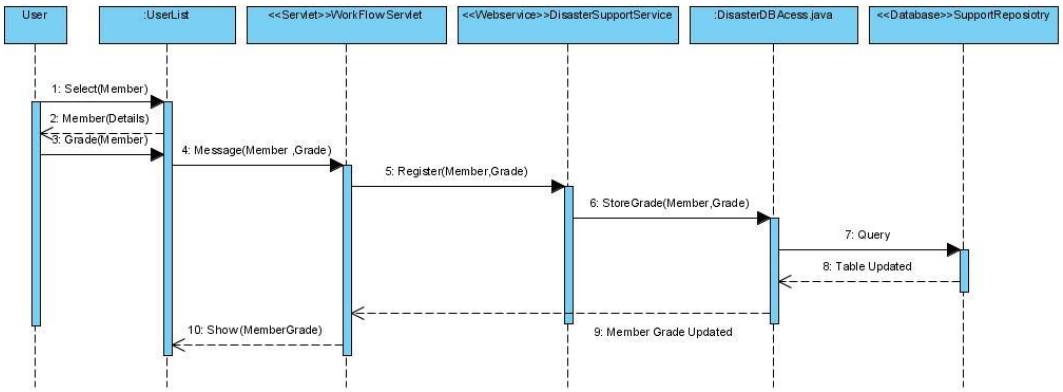

Figure A - 5 Sequence Diagram Grade Peers

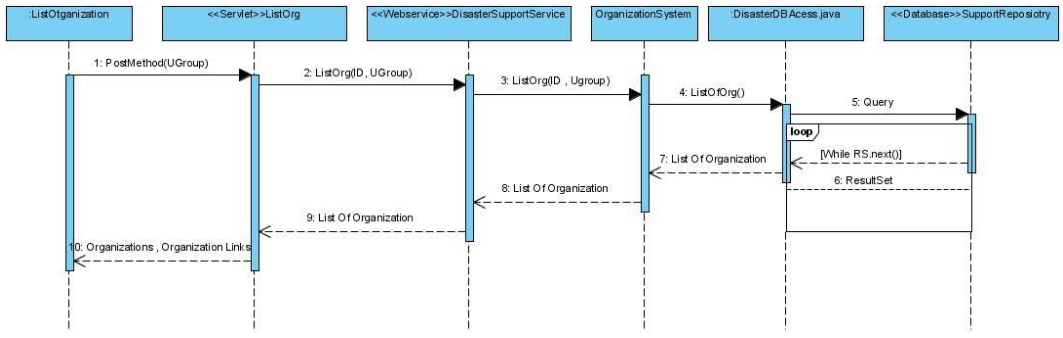

Figure A - 6 Sequence Diagram List Organizations

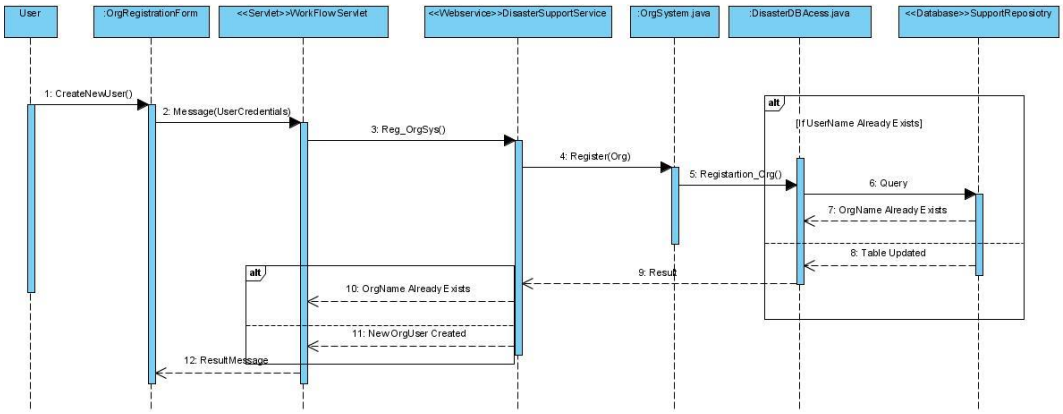

Figure A - 7 Sequence Diagram Register Organization

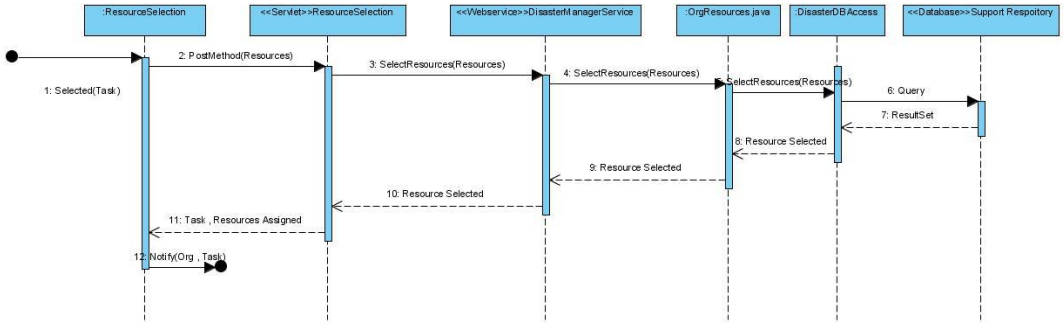

Figure A - 8 Sequence Diagram Resource Appointment

Communication Diagram

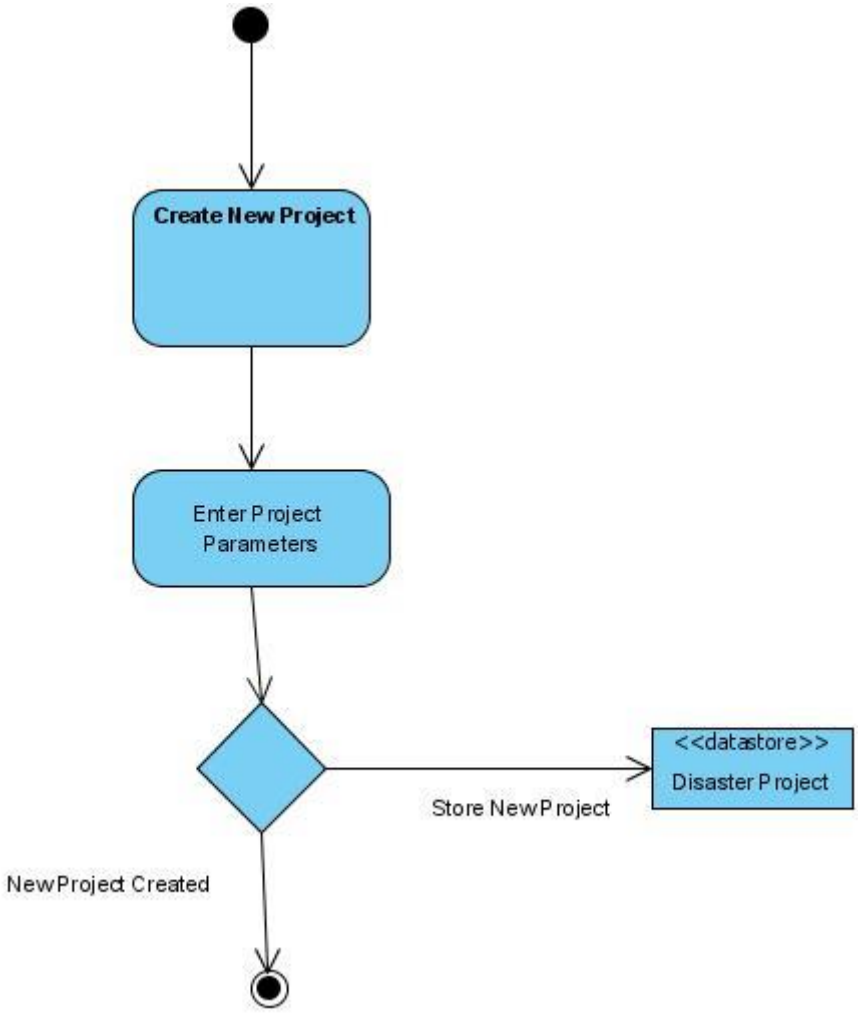

Figure A - 9 Communication Diagram Create New Project

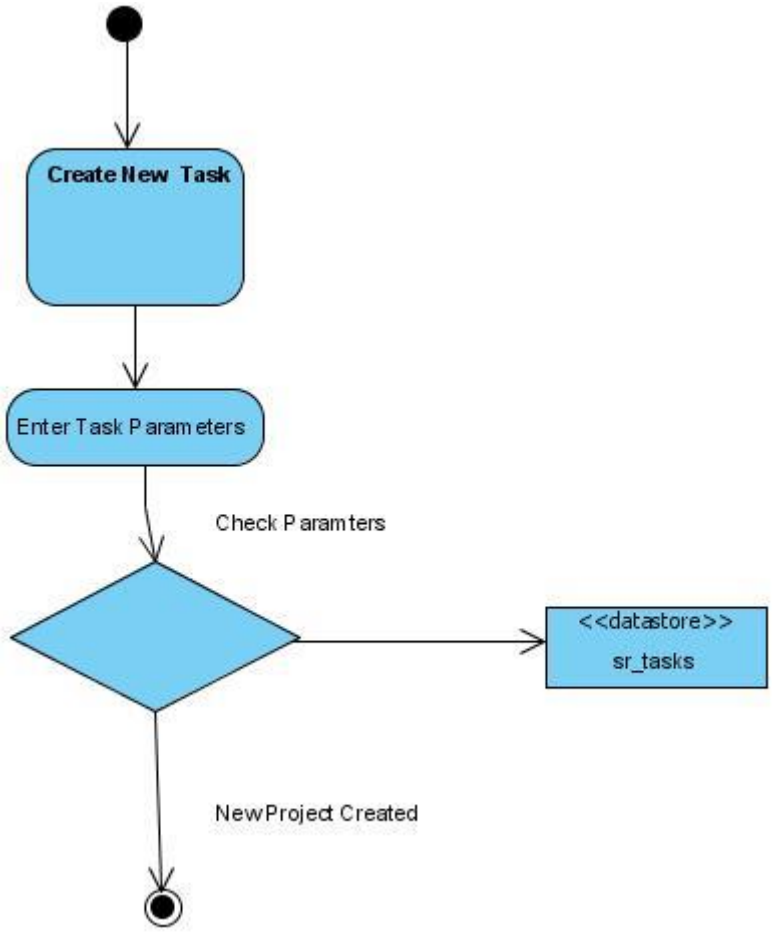

Figure A - 10 Communication Diagram Create New Task

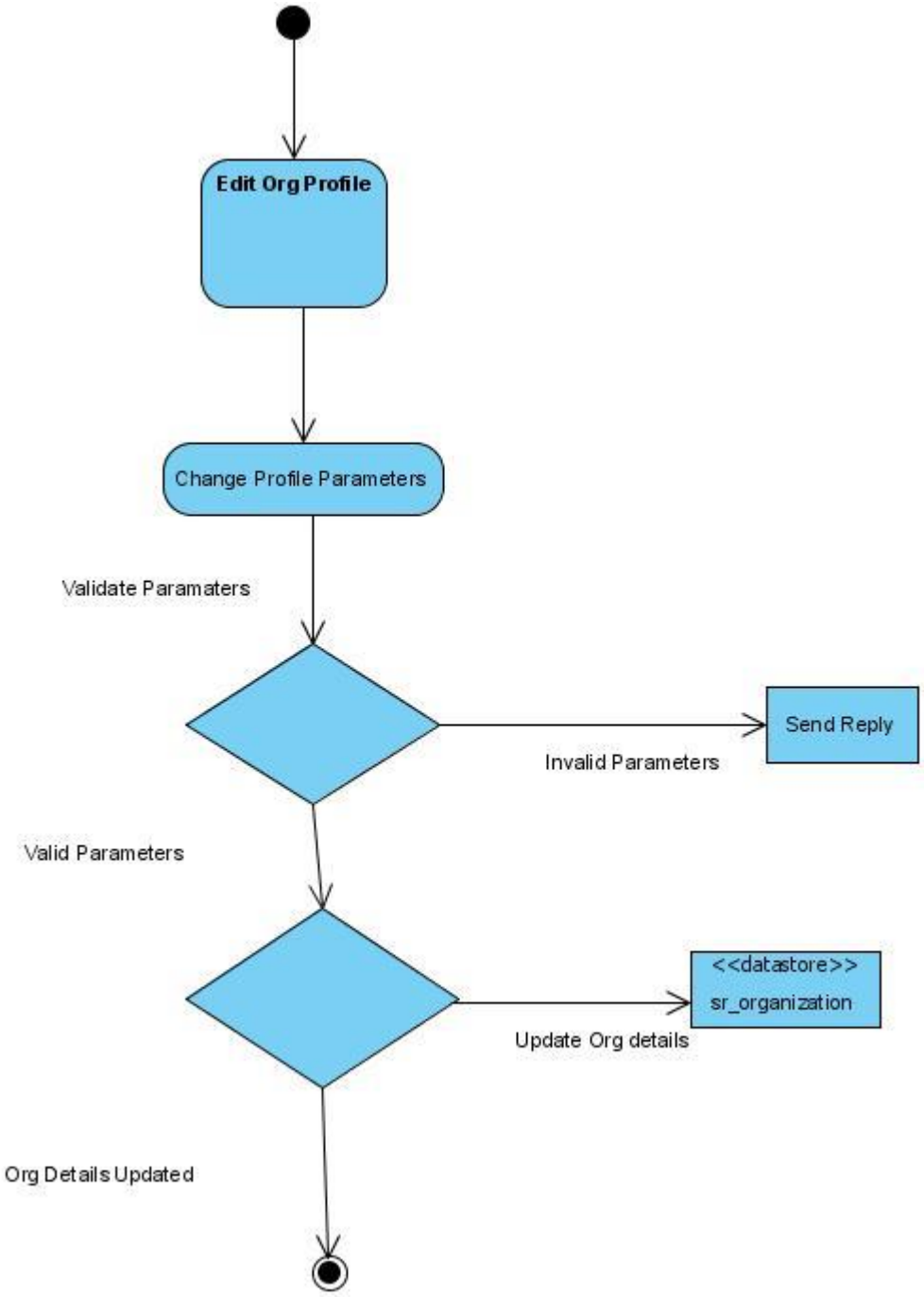

Figure A - 11 Communication Diagram Edit Organization User Profile

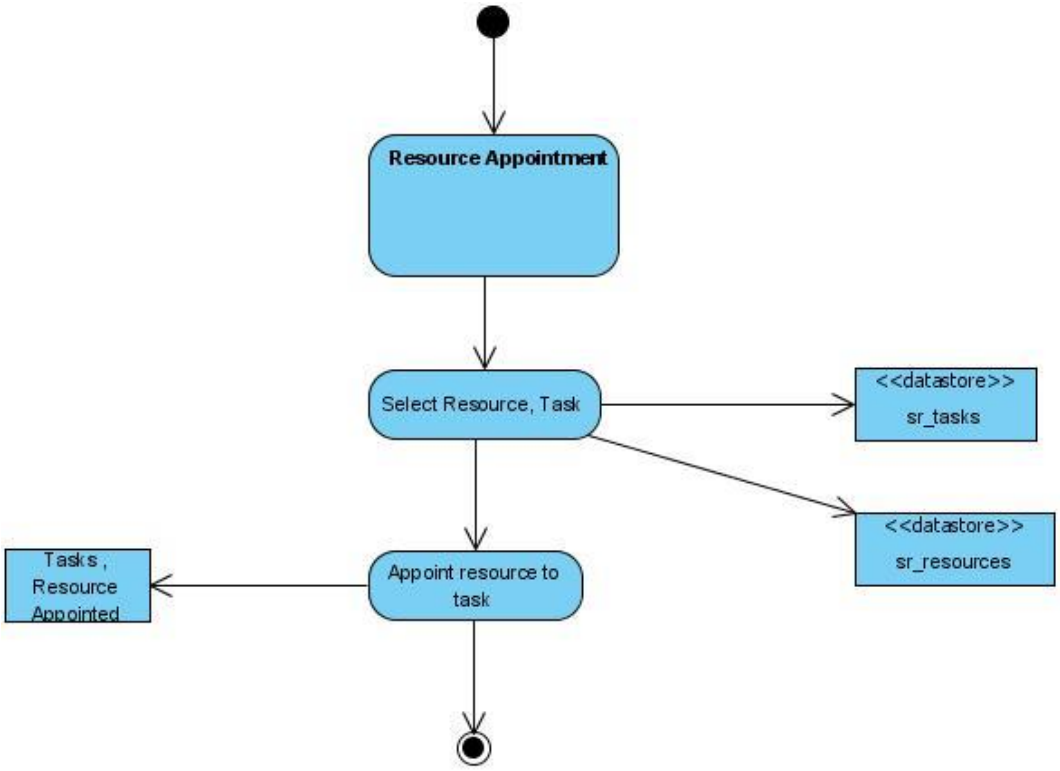

Figure A - 12 Communication Diagram Add Resource to Task

State Machine Diagram

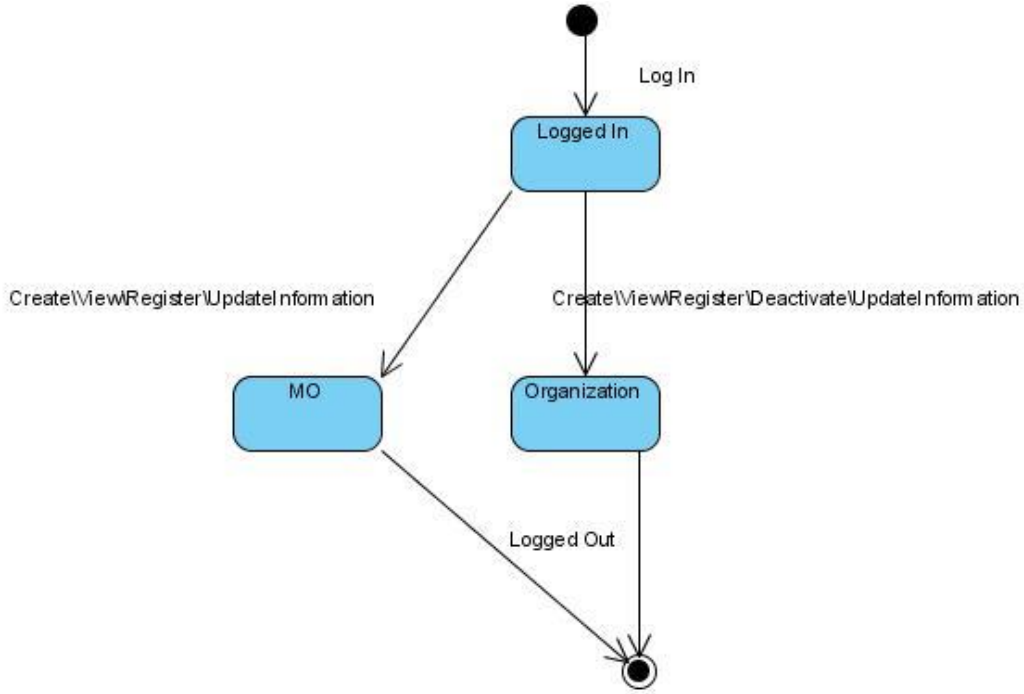

Figure A - 13 State Machine Diagram Manager Operation User

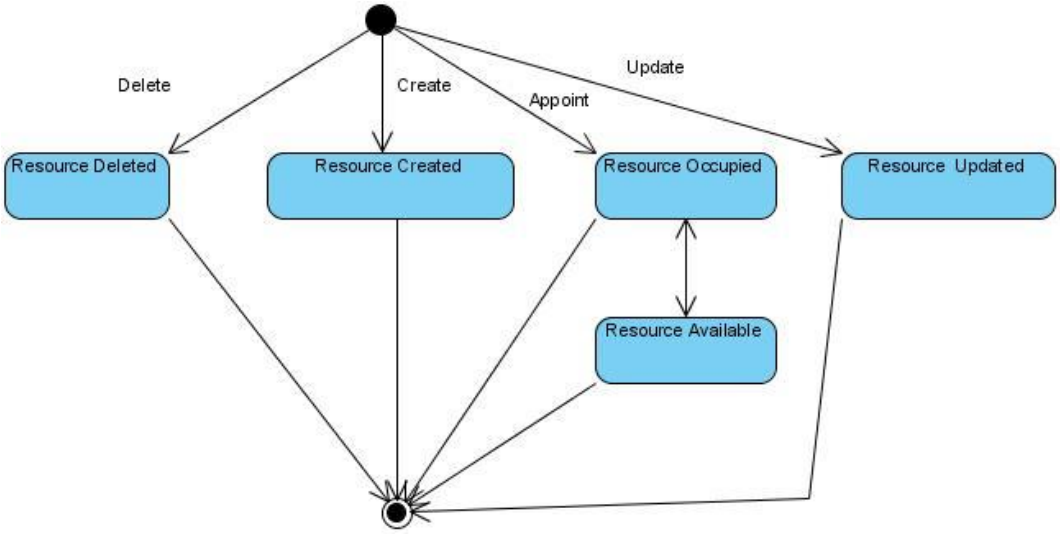

Figure A - 14 State Machine Diagram Resource

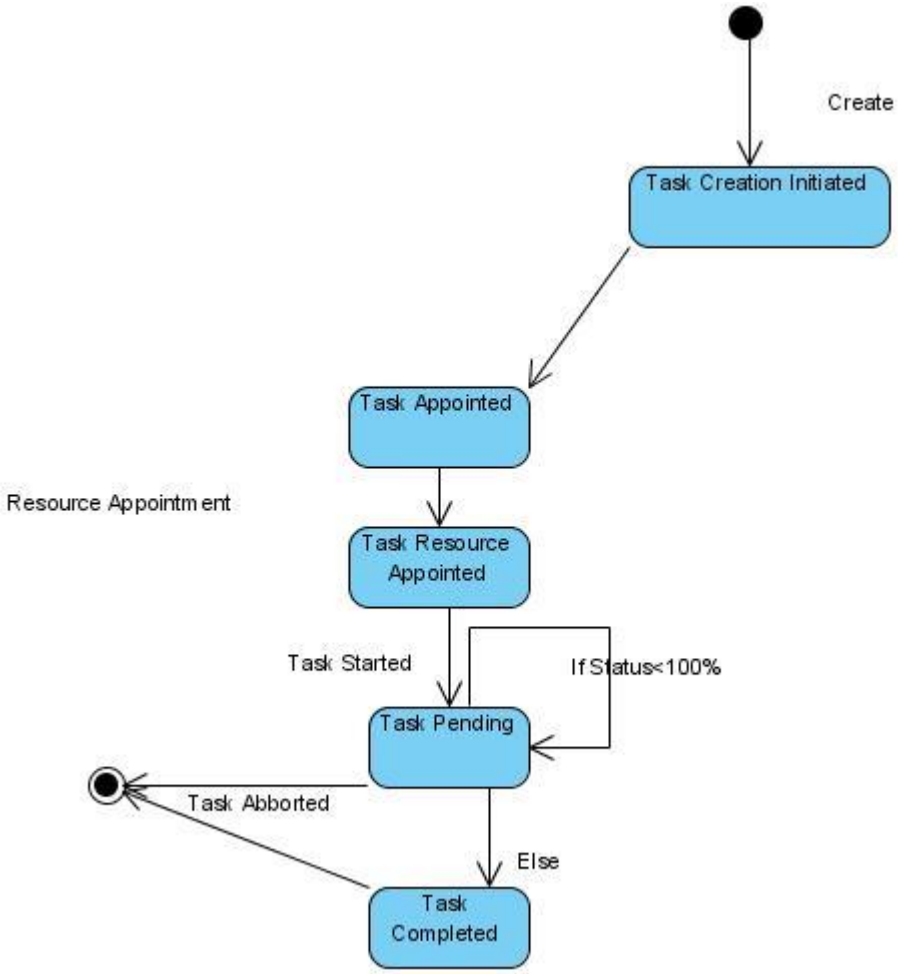

Figure A - 15 State Machine Diagram Task

Appendix B

User Manual

Overview of User Manual for Emergency Response System

This is the user manual for Emergency Response System. IRS is a web application having three different navigation pathways for three different user types: Admin, Manager Operation (MO) and Organization.

B.1 Navigation Path for Admin

B.1.1 Login

Figure 1 shows the login page which is same for all three user types. It has an additional link for organization to create a new user which is only active after Admin verifies the request.

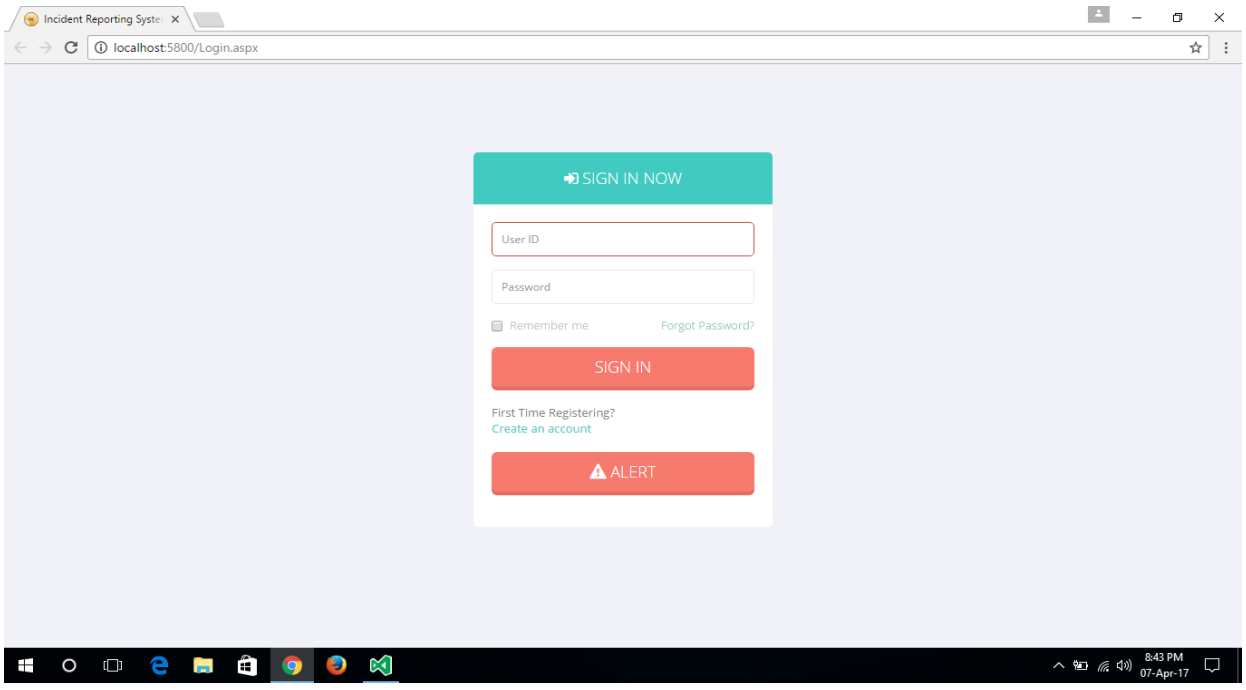

Figure B - 1 Main Login Page

B.1.2 Admin Dashboard

This is the dashboard for admin providing admin with different functionalities that only he/she can access. Figure 2 shows main dashboard for admin. Only admin can create new users and activate organizations and change system changes.

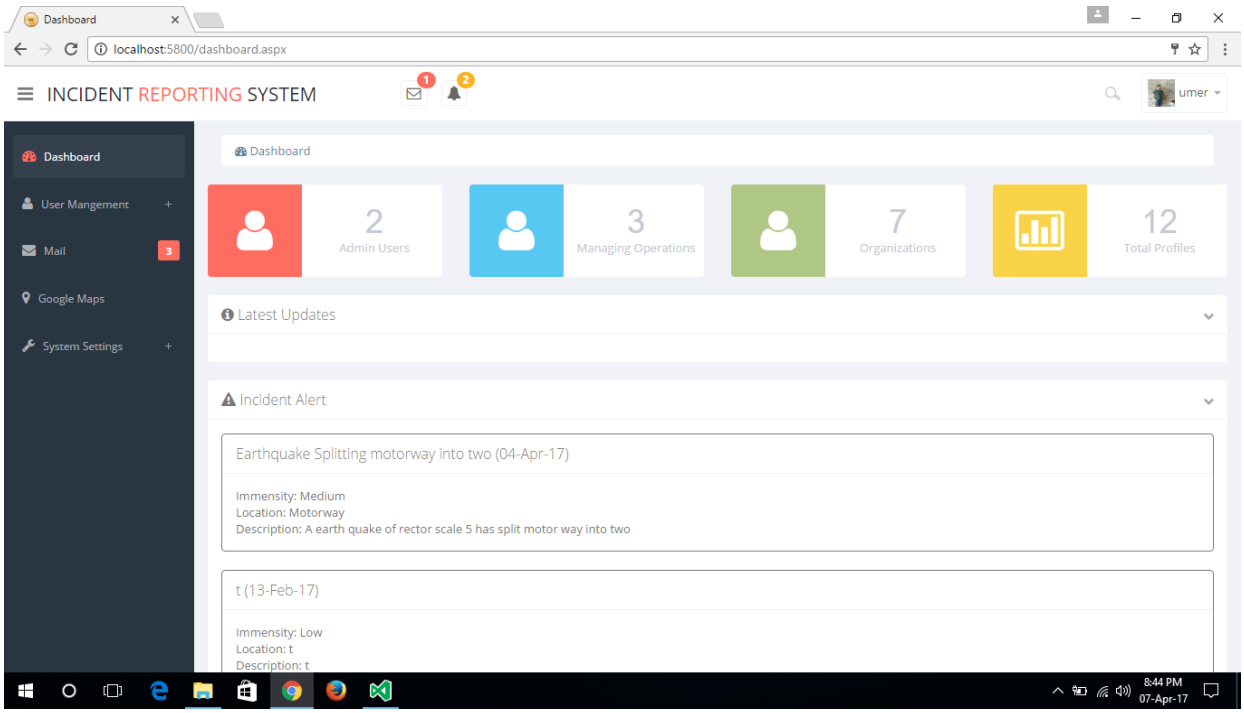

Figure B - 2 Admin Dashboard

B.1.3 System Settings

This is system settings panel only admin can view this page and change different system settings.

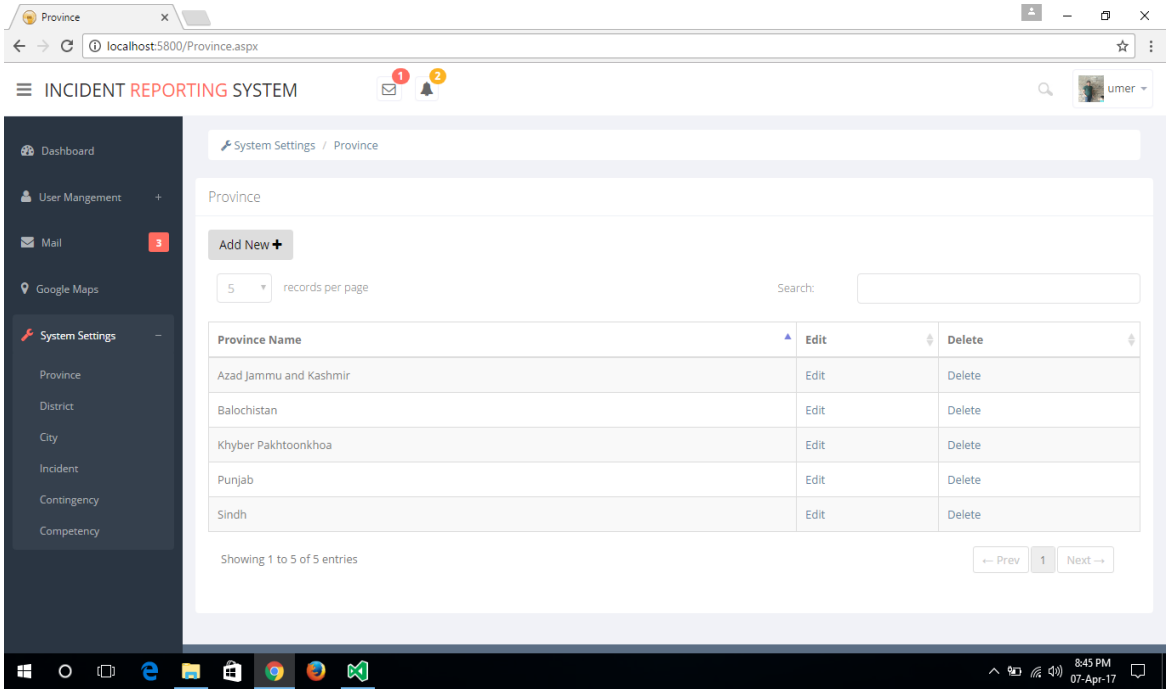

Figure B - 3 System Settings

B.1.4 Create MO

Only admin can create a new MO user through this page shown in Figure 4.

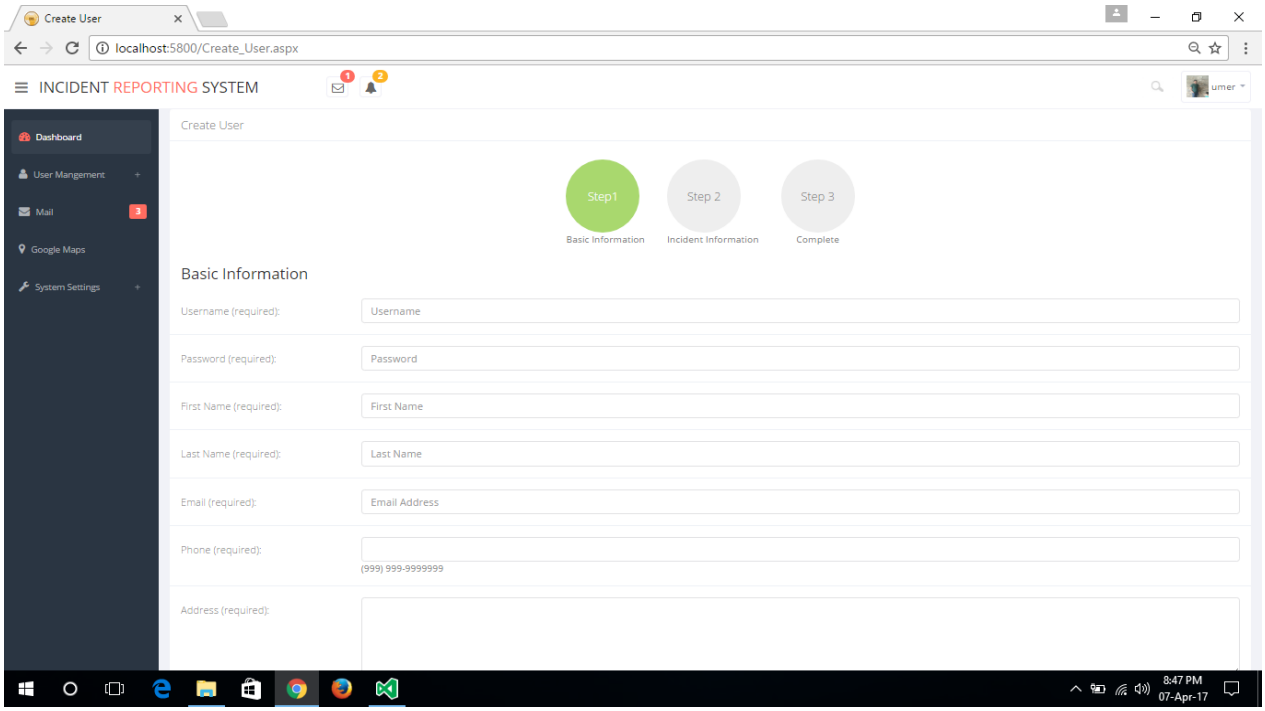

Figure B - 4 Create New Manager Operation

B.2 Navigation Path for Manager Operation (MO)

This sections shows navigation paths and different functions a manager operation user can perform.

B.2.2 Manager Operation Profile

The Figure 5 shows the profile options and editing page for MO. After entering data press submit to complete action.

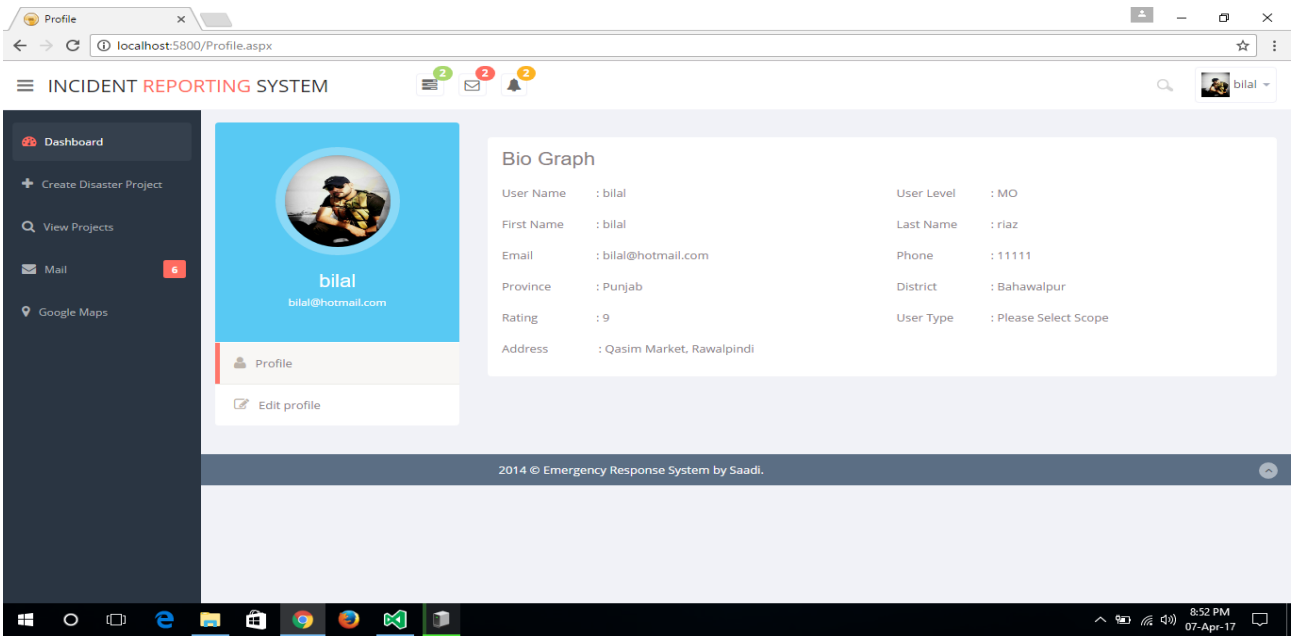

Figure B - 5 MO Profile

B.2.3 Create New Project

Only a manager operation user type can create a new project. Selecting the new project link will open a form to fill, the options help in developing a workflow for a new disaster. After entering data press submit to complete action.

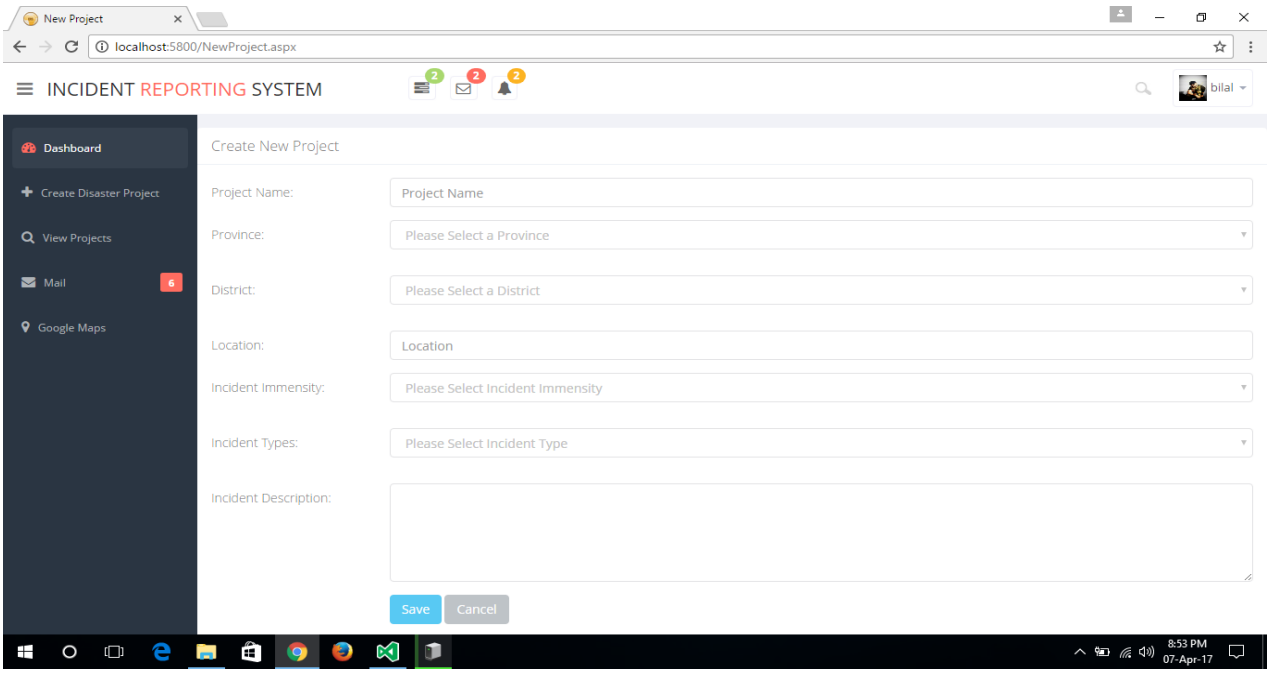

Figure B - 6 Create New Project

B.2.4 Contingency Plan (s)

After creating a new project for a disaster, the MO is directed to a new page where he/she can make a contingency plan to be followed to perform activities related to a disaster. The options can be selected from a predefined list or can add tasks by it. This page is shown in figure 7.

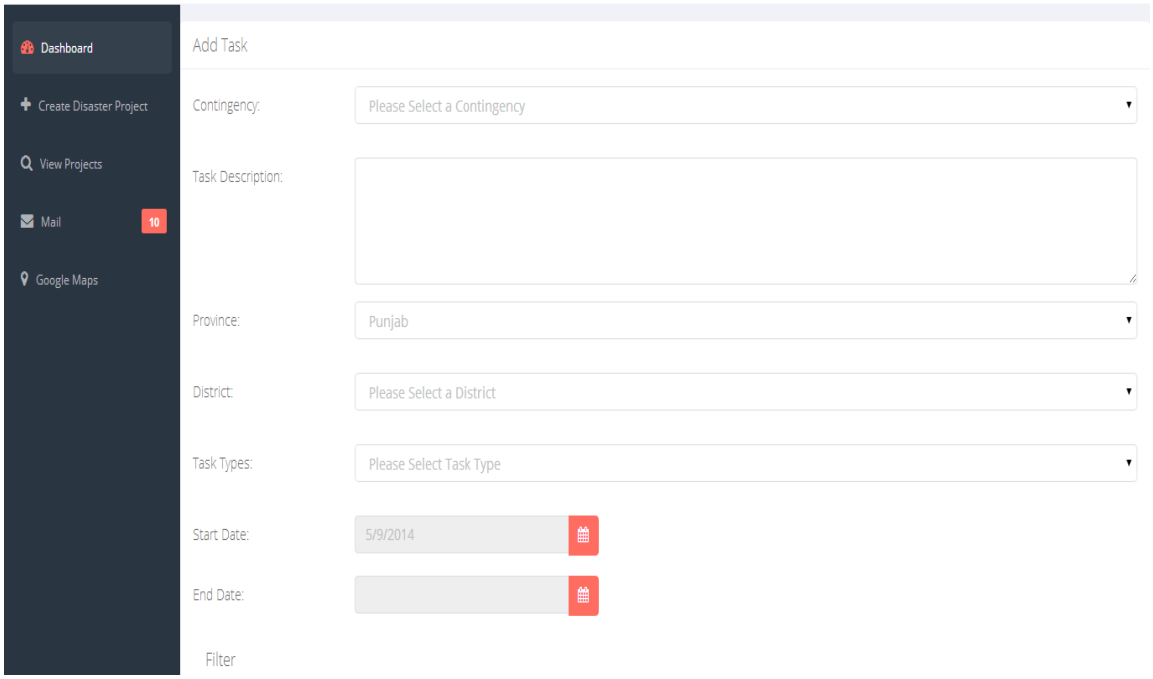

Figure B - 7 Contingency Plan

B.2.5 Add Task(s) Detail

The next navigation page is for adding details specific to a task(s). After entering data press submit to complete action.

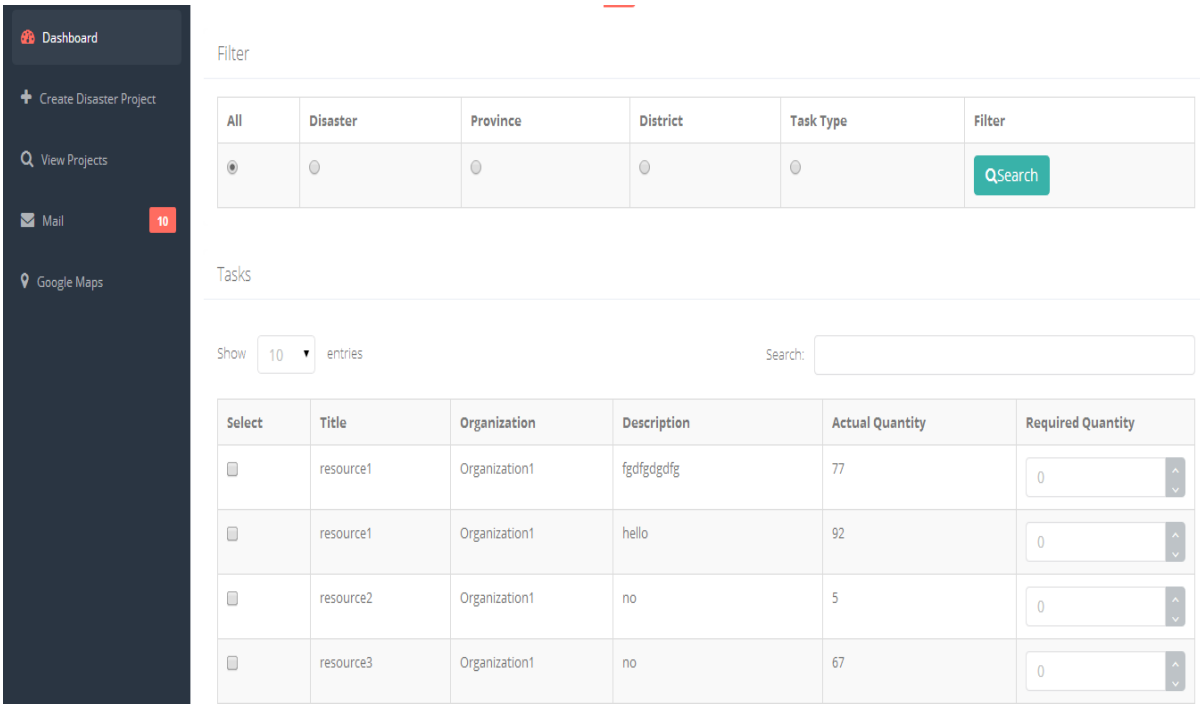

Figure B - 8 Add Task(s) Details

B.2.6 Add Resources to Tasks

Once MO has fed all details about all the tasks listed in contingency plan the next step is to add resources to those tasks. This webpage is shown in Figure 9. This is the last step of creating a task.

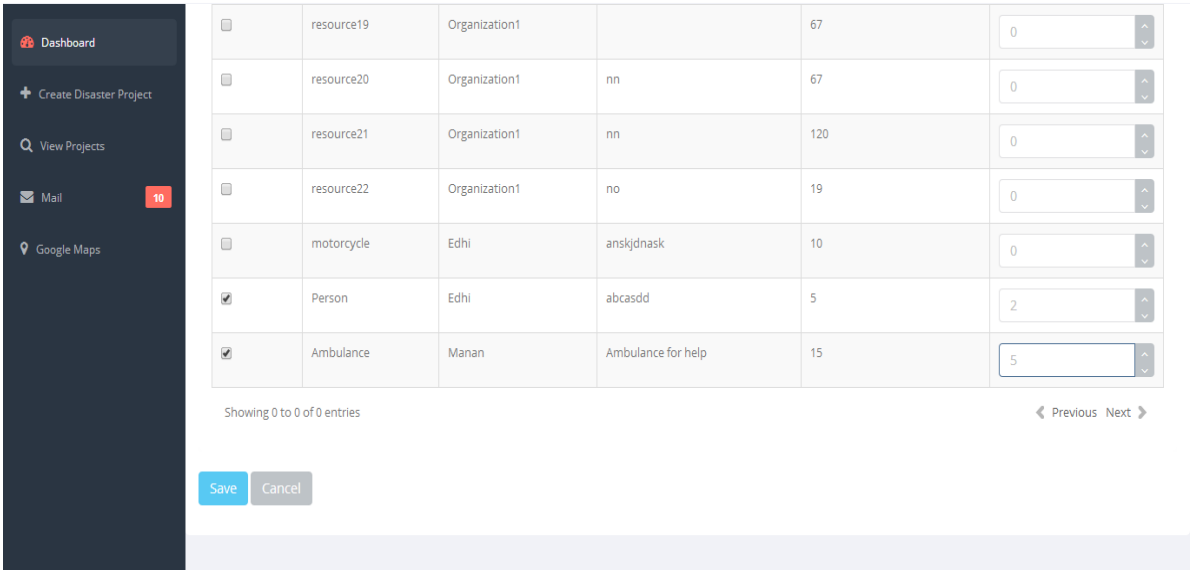

Figure B - 9 Add Tasks

An MO after creating a task can see different projects and their respective tasks and resources assigned to those tasks. To close a project he/she first has to free resources from tasks and close all tasks to close a project. Also an MO has option for status check to check status of different tasks and a message board for collaboration between different organizations.

B.3 Navigation Path for Organization

This sections shows navigation paths and different functionalities provided to an organization user type. The main dashboard is almost similar to admin and MO while the different navigation links available are described here

B.3.1 Current Projects

The Figure 10 shows the current project page visible to an organization, to help them see what project they are engaged on currently.

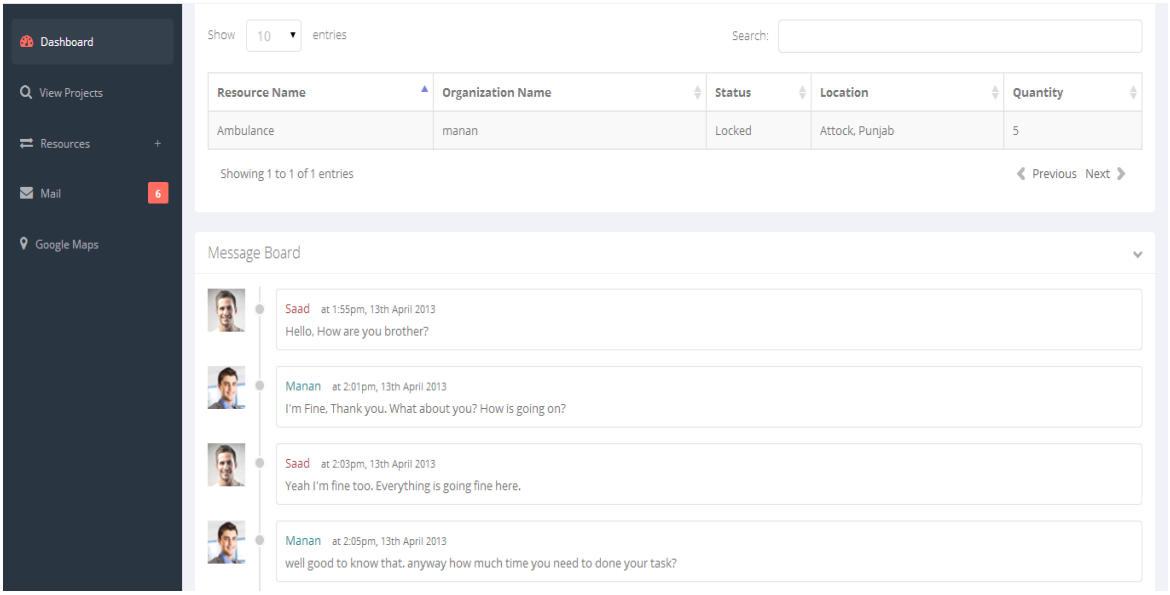

Figure B - 10 Current Projects for Organization

B.3.2 Add Resource

This web page adds resource shown in Figure 11 is used to update resources for an organization. After entering data press submit to complete action.

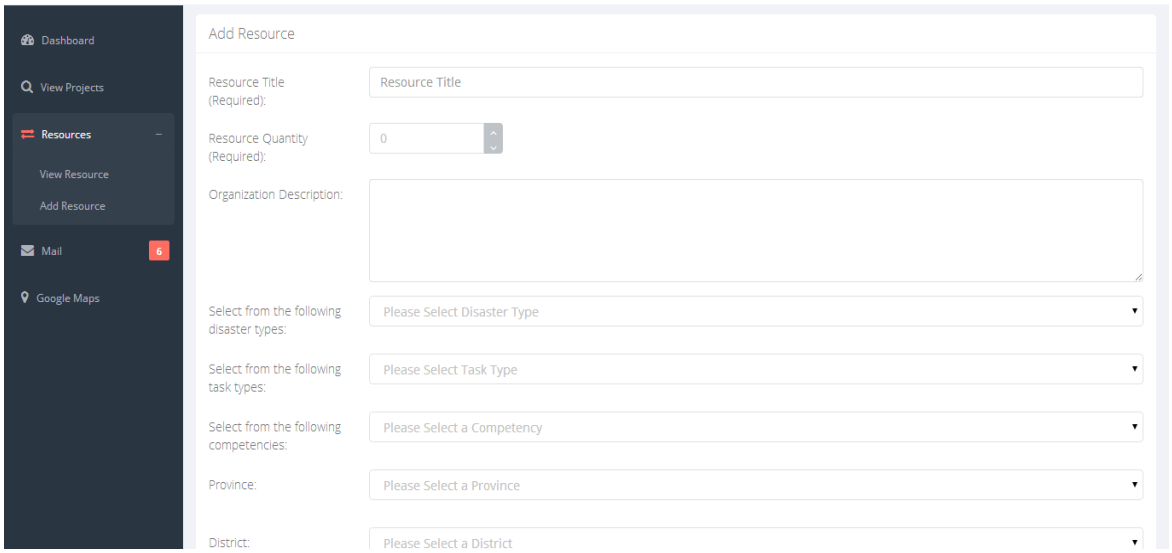

Figure B - 11 Add Resource

The other navigation links for organization helps in organization to see projects and tasks it's engaged on.

Map View:

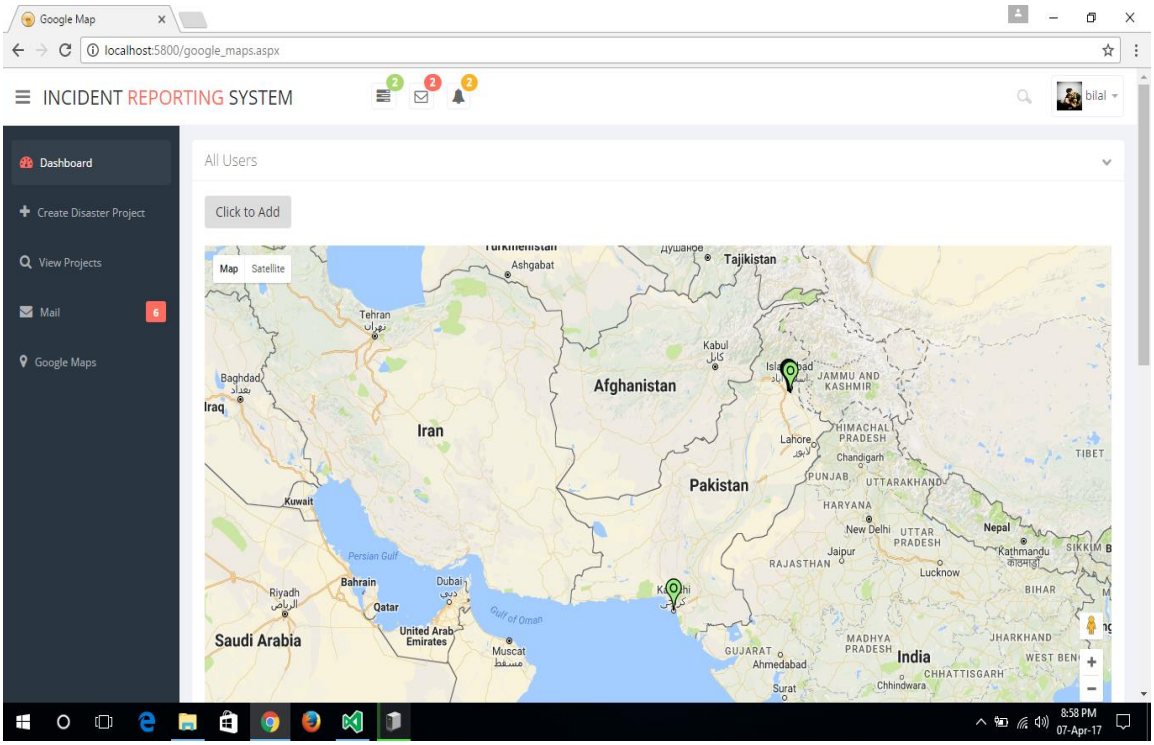

Figure B - 12 Map View

References

[1]W3C (2017), “Web Service architecture”, available from: http://www.w3.org/TR/ws-arch/#whatis [Accessed: 05-01-2017]

[2]Wikipedia (2017), “SAHANA FOSS” available from: http://en.wikipedia.org/wiki/Sahana_FOSS_Disaster_Management_System [Accessed: 07-01-2017]

[3] Secure-irfc (2017), “Disaster Management Information System” Available from: https://www-secure.ifrc.org/DMISII/Pages/00_Home/login.aspx [Accessed: 07-01-2017]

[4] Wikipedia (2017), “Java Script Object Notation” available from: http://en.wikipedia.org/wiki/JSON [Accessed: 10-02-2017]

[5] Wikipedia (2017), “Service Oriented Architecture” available from: http://en.wikipedia.org/wiki/Service_oriented_architecture [Accessed: 04-01-2017]

[6] W3C (2017), “Web Service” available from: http://www.w3.org/TR/ws-arch/ [Accessed: 04-01-2017]

[7] W3C (2017), “Web Services Directory Language” available from: http://www.w3.org/TR/ws-arch/ [Accessed: 13-02-2017]

[8] Wikipedia (2017), “SOAP Message” available from: http://en.wikipedia.org/wiki/SOAP [Accessed: 16-02-2017]

[9] Wikipedia (2017), “GET Message” available from: http://en.wikipedia.org/wiki/SOAP [Accessed: 16-02-2017]

[10] Wikipedia (2017), “POST Message” available from: http://en.wikipedia.org/wiki/POST [Accessed: 16-02-2017]

[11] W3C (2017), “XML Binding” available from: http://www.w3.org/TR/xbl/ [Accessed: 20-02-2017]

[12] Wikipedia (2017), “JQuery” available from:

http://en.wikipedia.org/wiki/JQuery [Accessed: 12-02-2017]

[13] Wikipedia (2017), “WCF Web Service” available from: http://en.wikipedia.org/wiki/Windows_Communication_Foundation [Accessed: 18-02-2017]

[14] Wikipedia (201), “AXP.NET XML Binding” available from: http://www.w3schools.com/aspnet/aspnet_xml.asp [Accessed: 18-02-2017]

[15] Wikipedia (2017), “AJAX” available from: http://en.wikipedia.org/wiki/AJAX_(Programming) [Accessed: 22-02-2017]

[16] Wikipedia (2017), “RESTful Web Services” available from: http://en.wikipedia.org/wiki/Representational_State_Transfer [Accessed: 10-02-2017]