SRS about Rescue App Idea

Software Requirement Engineering (COMSATS University Islamabad)

Software Requirement Engineering

SOFTWARE REQUIREMENT SPECIFICATION

Section: c

Group Members

Hashir Tariq **(FA17-BSE-133)**

Usama-Bin-Abid **(FA17-BSE-69)**

Table of Contents

1.1 Introduction

This document includes software requirement specification of Rescue 1122 mobile and web application. This app is basically designed to find the exact location of people in any calamity, so that the rescue van or ambulance can arrive there in short span of time and try their best to save people live and reduce further impairments. This document includes all the functions and specifications with their explanations to solve related problems as a project of Comsats Lahore.

5.1.1 Problem Definition

In this project, a system that will be used in Rescue 1122 operations will be designed and its software components will be implemented. Main components of the system are

- Mission Planning and Coordination Center
- Rescue Team Member Computers
- Hospital Computers
- Rescue Van Software module

Rescue System is used for planning and executing missions that are aimed to find and rescue people who are injured in any accident or other calamity such as fire rescue. It consists of some subsystems which are used in mission planning, tracking and execution of the rescue missions. A rescue mission is planned in computer at Mission Planning and Coordination Center.

During planning phase, last known position and estimated position of lost person, digital maps of city, rescue team's information etc. are used as an input. Mission will be planned in Planning & Coordination Center by using a computer having internet access and after planning has been completed, necessary information will be loaded to devices that is going to be used by rescue team.

When mission starts, everyone in the rescue team will be able to see their planned route on the mission computers and on the user devices. Also, they will be able to see their position and everyone else's position on their devices. Information on the computers and smart phones will be presented in two ways; by augmented reality technology over device's camera view and by placing information on digital maps. Everyone in the rescue team will be able to contact to each other by text messages or by talking using VoIP technology. They will also be able to transfer their device's video streams to each other's when they have been asked.

Everyone in the rescue team will be wearing pulse sensors.

Everyone's health information will be displayed on mission computers and planning center by

using these pulse sensors' information. This information will also be shared among team members' computers. Also, the actions (walking, sitting, running, etc..) of members during the mission will be shown on the mission computers. During the mission, execution of the rescue operation can be monitored by the mission center. All the positions of rescue members, health states and their actions will be shown on the Planning Software using Geographical Information System. If needed, a member will be requested to transfer his video image to the center and watched from there.
Then he is going to be able to mark his position and inform the other members and planning center.
When the person's location is found during the mission, rescue team will put a pulse sensor to the person and person's health status will also be shown in all mission computers and in mission planning center and also in the hospital module to show location of rescue van and estimated time to approach to the hospital and show that rescue operation is completed.

5.1.2 Purpose

This document aims to specify the requirements of our system, the path finding over maps for rescue 1122 teams to help them finding the people location in accidents, erupted Fire area's and other serious calamities.
There exist three main parameters for rescue 1122 teams that affects their performance, which are time limit, the area where lastly communicated with people in calamities and the possible ways to reach that field. Our system will help them with guessing the places where people may be and then, showing the possible fastest ways to reach the estimated lands over the map of that territory to rescue 1122 teams. Also, we will include a module HELP for which the persons can seek help by nearby people such as any vehicle failure or any serious calamity in a lonely place.
The preparation of this SRS will help consider all of the requirements before design begins, and reduce later redesign, recoding, and retesting. The review of SRS can reveal omissions, misunderstandings, and inconsistencies early in the development cycle when these problems are easier to correct.

5.1.3 Scope

The project name which is going to be presented in this software requirement specification is Rescue 1122. This application will be compatible with mobile devices & tablets using Android operating system.
Our system will have three phases. First, the program will indicate the area where the lost person may locate. According to the estimated point where that person lastly positioned and given time duration of his being out of contact, the program will draw the area which he may be in.
While the area is being drawn the things like whether that person is with a vehicle or on foot, if with a vehicle the average speed of it will be taken into account. For the second phase of the program the rescue team will move to mark point given by the user (a single point or more) to start searching.
Then, the program will give the fastest path to reach those point for land vehicle. For example, if there is any accident of one or many vehicles the app will show the location of accident and shortest route to reach at that path and any nearby hospital for emergency. In third phase the module installed on the hospital will show that any emergency case is arriving with (n) people and to make further arrangements for them.

5.1.4 Definitions, acronyms, and abbreviations

Terms	Definitions
SRS	Software Requirements Specification
AR	Augmented Reality
GPS	Global Positioning System
GUI	Graphical User Interface
IDE	Integrated Development Environment
Android SDK	Software Development Kit which is officially released for Android
Android Studio	Official IDE designed for Android
UML	Unified Modelling Language
Use Case Diagram	Diagram of interactions of users with the system
Class Diagram	Diagram that describes the structure of a system by showing its classes, attributes of these classes and method
GIS	Geographical Information System

5.1.5 References

- https://jersey.java.net/
- http://hibernate.org/orm/
- https://dev.mysql.com/doc/refman/8.0/en/
- https://www.jetbrains.com/idea/
- https://developer.android.com/studio/index.html
- https://en.wikipedia.org/wiki/Voice_over_IP
- https://en.wikipedia.org/wiki/Livestream
- Çakıcı, R. (n.d.) Software Requirements Specification Report. Retrieved from https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=3574
- Doğru, A. (2001). Software Engineering. Ankara : Atlas Publishing.
- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- KILIÇ, A. (2005). Coordinate Systems for A Naval Virtual Environment. Retrieved from Bites Defence Industry Research and Development Sources.

- Larman, C. (2005) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition). United States: Prentice Hall.

5.1.1 Overview

Project Rescue 1122 is web and mobile application. Rescue 1122 has three main parts. These parts are Mission Planning and Coordination Center and Rescue Team Member Mission Devices and hospital module. For Mission Planning and Coordination Center there will be a web application which provides creating, tracking and coordinating the missions. For Rescue Team Member Mission Devices there will be an android application that provides tracking other rescue team members, communicating with each other's and coordination center and getting important data from external sensors such as pulse and muscle motions sensor. Since the project has an augmented reality part, also the camera is used for getting the real-world data to be augmented. Also, there will be a hospital web module in which the computer will be showing the emergency cases arriving towards the certain hospital and alert the hospital to arrange certain arrangements for emergency case that is being arrived in certain time.

5.2 Overall description

This section will give information about product perspective, product functions, constraints, assumptions and dependencies. This section usually consists of subsections, as follows

a) Product perspective
b) Product functions
c) User characteristics

5.2.1 Product perspective

Rescue application is totally independent system that is not related to any other system and not a component of a larger system. This program has only one type user, so there are no functionality differences between users which means there exists only one type user interface. However, this one type interface is separated into sub-interfaces which the users will encounter.

First, the application welcomes the users with a list of needed information about the person in calamity one which should be filled in, that are respectively the coordinates of lost one lastly observed which is needed to be learned at related GSM operator, time of that signal sensed, the location of vehicle of one using app, if exists, and the weather condition. Then it comes to second sub-phase consisting of necessary information again, but related with the user this time, whose are the coordinate of the user and their vehicles by choosing them in the default list of application. In third phase, the application calculates the fastest path for each vehicle to reach the matched point and shows them in map in the last phase.

In terms of hardware, Rescue App will be compatible with mobile devices using having touch-operated control, i.e. tablets and smart phones. It will be controlled with finger movement of the users. Hence, it will ease showing details of map.

a) System interfaces
b) User interfaces
c) Hardware interfaces
d) Software interfaces
e) Communications interfaces

f) Memory
g) Operations
h) Site adaptation requirements

5.2.1.1 System interfaces

First of all, the application needs to have a Google map about the area. It must be loaded before usage because some maps are inaccessible for security. Therefore, Rescue will be utilizable for authorized institutions, not individuals. There will be an API to recognize and process Google maps in Java language. By this helper program, the system will be able to point out the area, create a vector map and compute the fastest path using the map. After this process, the program will be ready to use.

5.2.1.2 User interfaces

Since our project has three main parts the team member application part will be an Android application that will run on smartphones, there will be a graphical user interface. The interaction with the application will be through touch screen. The other part of the project is mission coordination center is web application that will work on computer. There will also be graphical user interface on web browser. User will be interacting through web browser. Also, the hospital module will be having a web application to show the ambulances approaching and their estimated time and number of injured persons.

5.2.1.3 Hardware interfaces

Only mobile devices which have touch-operated properties will be suitable for the application. They can be tablet or mobile phones. These devices should have some limit requirements to make the application run effectively. We expect 1GHz, 1MB for processor & 1GB RAM for tablets as well as the same limitation necessity for smart phones.it will be good if the phone has 4G or 4G LTE.

5.2.1.4 Software interfaces

First of all, mobile devices will be used for the application and they must have android to run the application. Moreover, application will be coded by Java.
Therefore, Java must be setup to the mobile devices. For java there must be a compiler it can be android SDK. To make the map visual OpenGL will be used. For maps usage, Java Supporting GDAL shall be used.
Software used in this project include DBMS, IntelliJ, Maven, Open Layer, Android Studio, Android operating system and an augmented reality library.
MySQL is used in the server as DBMS, Communication between database and rest server is operated by hibernate.
The communication with the operating system is done through standard Android API.

5.2.1.5 Communications interfaces

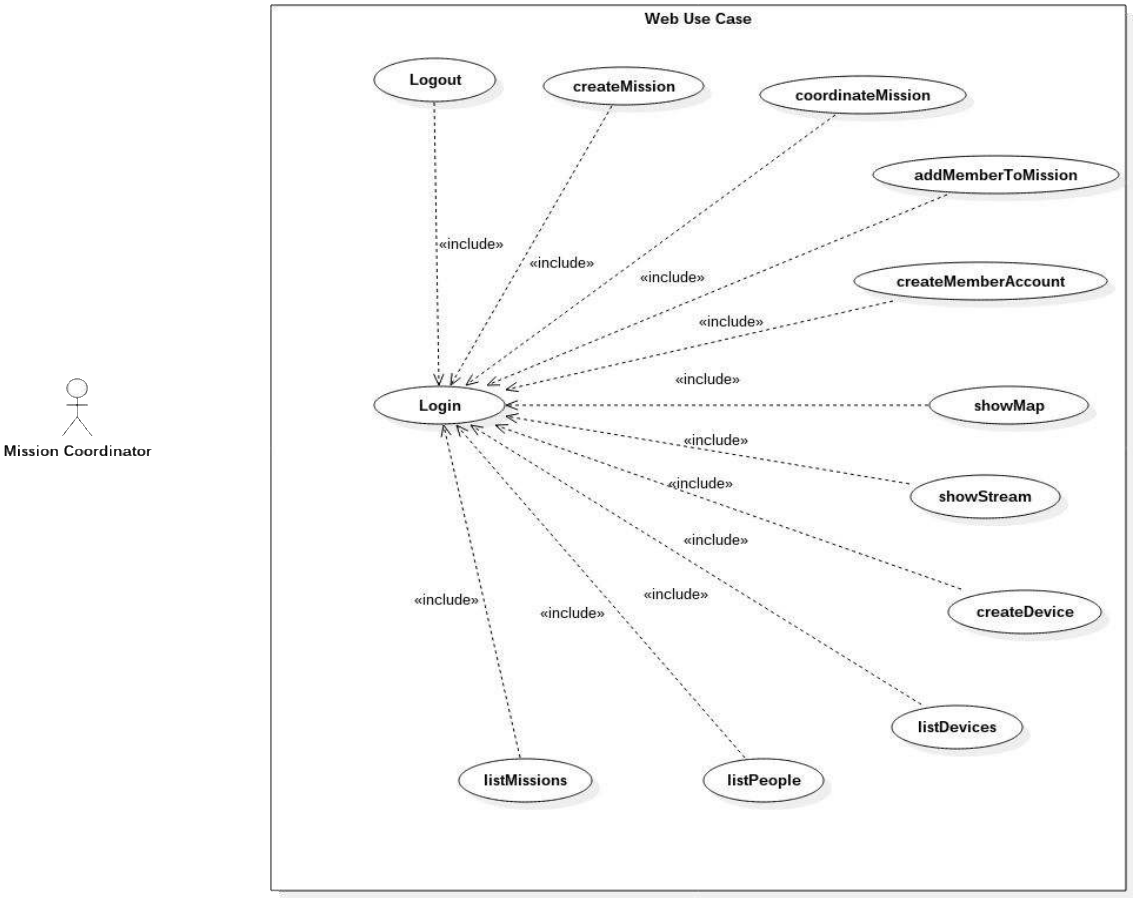
The application will communicate with the server via HTTP protocol over internet.

5.2.1.6 Memory constraints

The app will be consuming less memory so that it can load fast and all the data will be stored on online servers and data will be fetched from there.

5.2.2 Product functions

We have two actors which are Coordinator and Team Member. Their use cases are mentioned below.



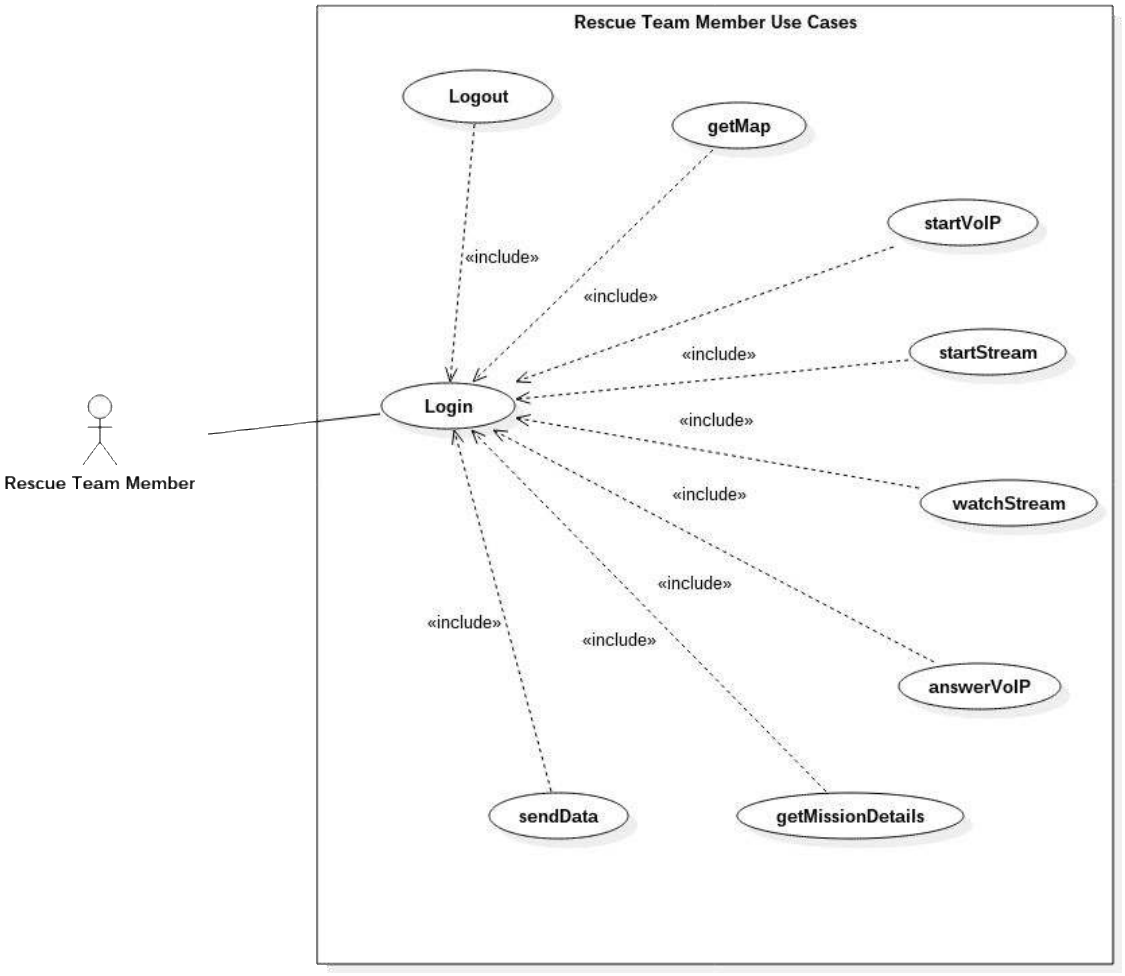
When actor is Mission Coordinator:

- Login: Coordinator will be logged in the system before executing any operation
- Create Mission: It creates a search and rescue operation
- Coordinate Mission: It coordinates the both mission and team members.
- AddMemberToMission: Coordinator can add team member to current mission
- createMemberAccount: Coordinator can create a new team member model on database.
- showMap: Coordinator can get all team member locations and show them on real search area map.

ShowStream: Coordinator can play current video streams on web site

- createDevice: Coordinator can create a new device model on database.

- ListDevices: Coordinator can list all devices to use them in the mission
- listPeople: Coordinator can list all persons to select mission members
- listMissions: Coordinator can list all missions and mission details for current mission
- Logout: Coordinator can logout



When actor is Team Member:

- Login: Team members will be logged in the system by using their android devices
- Logout: Team members can logout
- getMap: Team members can see other team members positions on the map view
- startVoIP: Any team member can start audio conversation
- startStream: Any team member can start live video stream
- watchStream: When someone is streaming any team member can watch this stream
- answerVoIP: When there is a started conservation user can join
- getMissionDetails: Any team member is able to see mission details
- sendData: Team members can send their information's to center

5.2.3 User characteristics

The user can use the app, we ensure of talking to higher authorities to make the app free so that people can fetch help in any calamity, or we can make it donation based. The app will be user friendly and a help module will also be added to ask for help with nearby people.

5.2.4 Constraints

- Email address should be valid.
- Username should be an email address.
- Android version must be at least 5.0.
- Mobile device has touch-operated property.
- Mobile device should be working on Android operating system.
- Java platform exists in the mobile device.
- Google map must be pre-installed before usage.

The application has to process and answer in real time, but, obviously, this application is limited by the performance of the portable device and the camera. The application does not have any safety and security concerns.

5.2.5 Assumptions and dependencies

The software will initially be designed for android but with advancement of time will launch it for other platforms and if the success rate grows, we will implement it in other cities and increase the servers and maintenance and updating will improve our app accordingly.

5.2.6 Apportioning of requirements

The release of app can be delayed for other versions and also due to government restrictions and policies and other licensing issues.

5.3 Specific requirements

This section will describe the software requirements in detail as subsections which are interface requirements, functional and non-functional requirements.

Interface Requirements: Since there is just one type of user, the application will have only the user interface which is diagrammed. Basically, the user interface will direct the user through steps and will display the calculated results as an output.
The steps of the user interface are the following;

Lost variables: The primary input to the system is the information about the lost which includes four parameters. First parameter is the time duration between the last signal received from the calamity and the beginning of the research. Second, the coordinates of the signal location will be kept. Third necessary data is the vehicle status of the calamity. Whether people with a vehicle or on foot, if with a vehicle the type of it would be necessary for further calculations. In this case possible vehicle types would be predefined and if necessary additional vehicle definition would be

allowed. The last parameter is the weather condition which affects both the user and the team vehicles. After providing these parameters the user passes to the forward step.

Vehicle selection: In this part of the user interface, predefined vehicle list would be visible according to the weather condition parameter given in first step. The user whom belongs to rescue team will select the vehicles which they have. As stated in the first part, new vehicle definition shall be done if necessary, providing three parameters basically. First parameter will be used for naming which ensures further recognition of the vehicle. Second parameter will be the average speed of it. When the vehicle selection is finished, next step comes.

Choosing points: With the data given in first part, the application will calculate the possible area in which the calamity may locate. The user will choose points over the map and match vehicles with these points. User may match one or more points with one or more vehicles.
In addition, user may indicate the order of the points to be reached or may allow automatic ordering which defines the fastest reachable point as the first, and the next fastest point as the second, and so on.

Resultant paths: According to the data set in former steps, this part will reveal the resultant paths. When user selects a vehicle or a group of vehicles to be sent, the resultant path will be visible over the map

5.3.1 External interfaces

This should be a detailed description of all inputs into and outputs from the software system. It should complement the interface descriptions in 5.2 and should not repeat information there.

It should include both content and format as follows:

a) Name of item;
b) Description of purpose;
c) Source of input or destination of output;
d) Valid range, accuracy, and/or tolerance;
e) Units of measure;
f) Timing;
g) Relationships to other inputs/outputs;
h) Screen formats/organization;
i) Window formats/organization;
j) Data formats;
k) Command formats;
l) End messages.

5.3.2 Functions

Functional requirements
In this section, we will explain the major functions of the Rescue 1122

Start
This function becomes active just after touching Rescue application icon from menu of the mobile device and gets the users to the step 1. As it is obvious, the input of “start” function is touching Rescue icon and the output is step 1 interface.

Target Area

This function tries to find an area for injured people. The last signal coordinates, elapsed time since the last signal came and the vehicle type are parameters of the function. After getting parameters function tries to compute the lost people area by considering the slope, vehicle's path, weather conditions, lakes or rivers. Output of the function will be a visual map and tagged area on it.

Fastest Path

Fastest path function tries to find the fastest path from the start point to target points. Target points will be inputs for the function. There is one more input which is the vehicle type for rescue team. Target point(s) can be a point or more. If there is one point, then function just compute the fastest path between these two points – start point and target point. If there is more than one point in the target area, program will compute the fastest path by combining the points. Either user will indicate the order of the points to be visited or the program will decide for priority of the points. If the order is not indicated manually, the function will choose the first point, second point, etc. by itself for fastest path. Fastest path function tries to find the way by using the properties of the vehicles. This function shall use Greedy algorithm which is a special case of A* search algorithm.

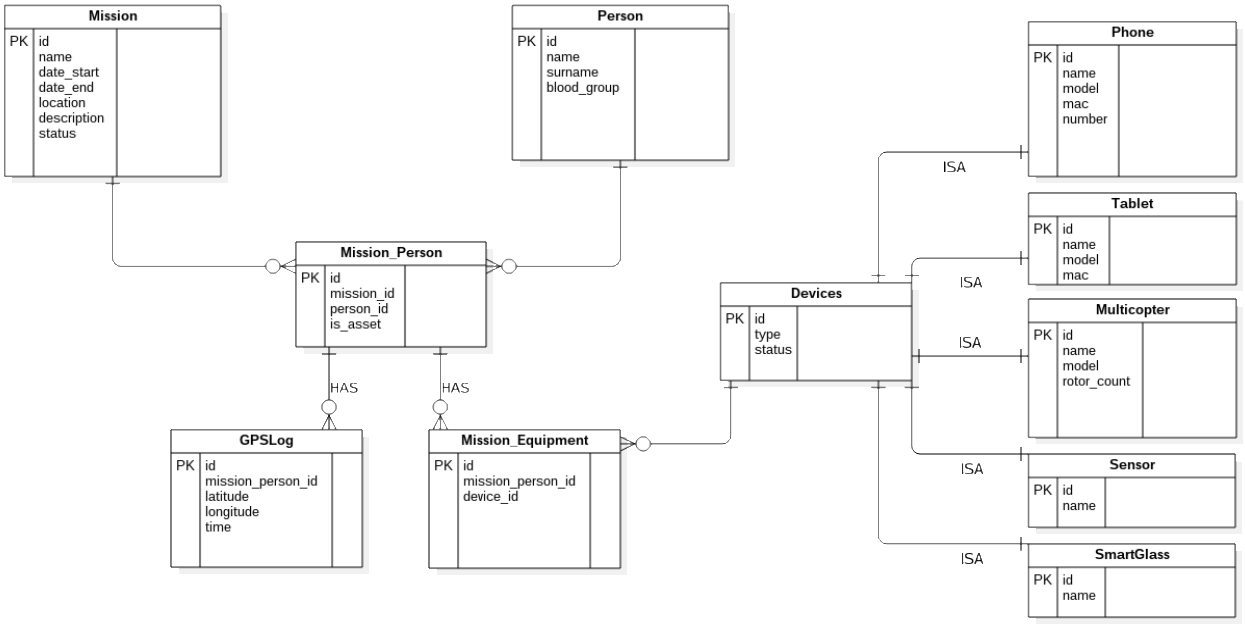
Hospital Path

The screen on hospital will show the van arriving towards it and tell the severity of accident and no of injured people. Also, it will tell ETA for the van arriving towards hospital.
The app with the rescue team shows nearest hospitals and there fastest path to approach towards it.

5.3.3 Performance requirements

As it is stated before, this application will be used by a single user. There will be no multiple users since the application runs on a single portable device and it would be requiring a good speed internet

5.3.1 Logical database requirements



5.3.2 Design constraints

Language: Rescue application is designed to work on portable devices.
For this purpose, a programming language that works on a portable platform is appropriate. Java programming language will be used for this aim.

Map processing: To handle maps, which Google maps, system will use Java language supporting GDAL. The program should have an ability to point out the area and compute the fastest path towards location on map.

Visuality: OpenGL will be used for visualization. What we will visualize is to cover physical conditions on the related map

Hardware Constraints: Since the system will be a mobile application, mobile devices will be required. They need to have touch-operated feature and Android applicable specifications. Moreover, we expect for devices having some limit features which is needed for the performance of application.

5.3.5.1 Standards compliance

5.3.6 Software system attributes

5.3.6.1 Reliability

- System should be up for 99% of the time excluding scheduled system maintenance.
- In case of a system crash, system can be brought up within four hours.
- The system should provide reliable results over the map.
- Height indication should be sensitive.

5.3.6.2 Availability

- should have fast responses as it is an emergency kind application.
- Team Members and coordinators will be trained before using this project so users can easily adapt on this project.
- Database hold past missions and records therefore which device or person problematic can easily be identified and replaced

5.3.6.3 Security

- The user and the target person/vehicle information will be out of concern as they will not be kept.
- The access permissions for system data can only be changed by the system's administrator.
- The communication between the system's data server and clients will be encrypted

5.3.6.4 Performance

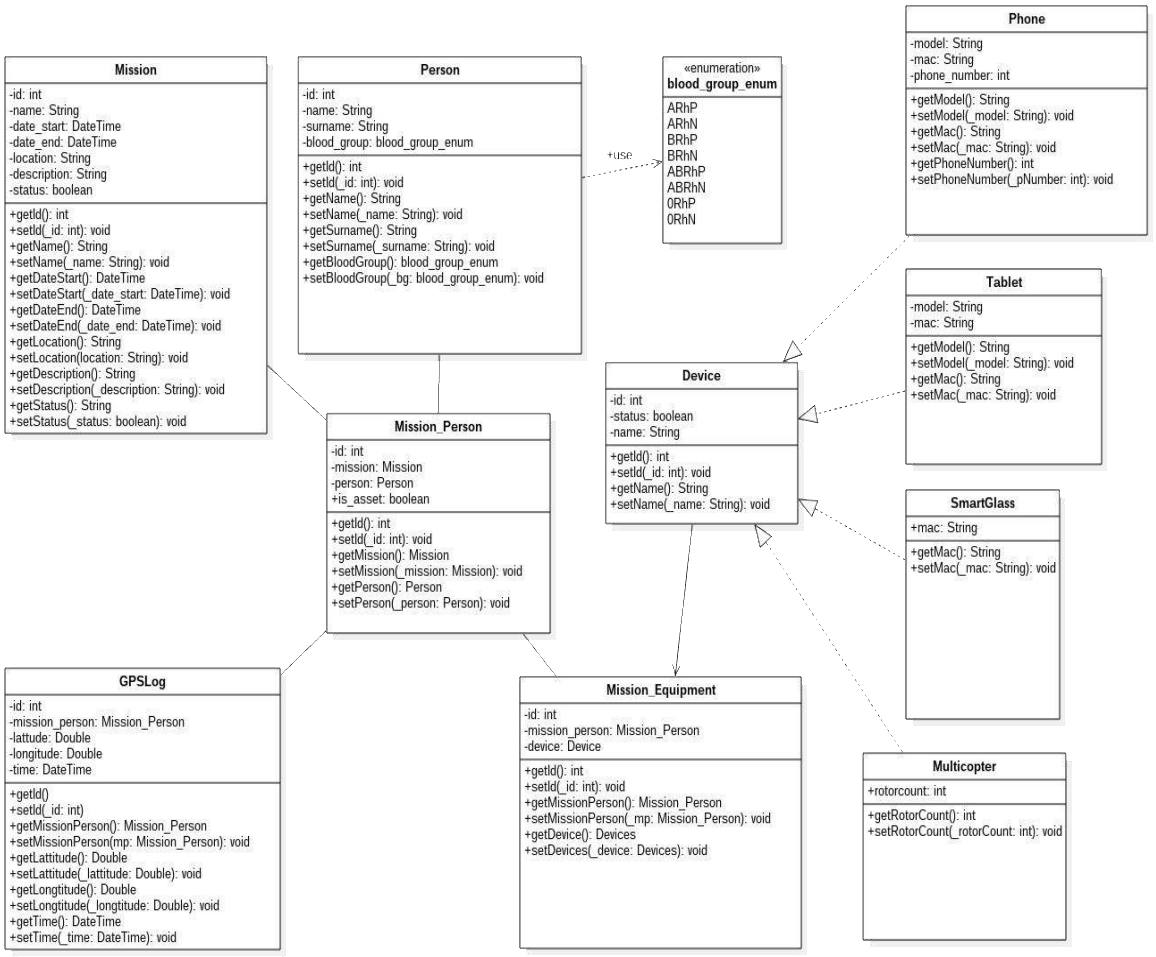
- The server should capable of 100 mission at same time.
- The server should respond in 0.2 seconds at maximum.

5.3.6.5 Portability

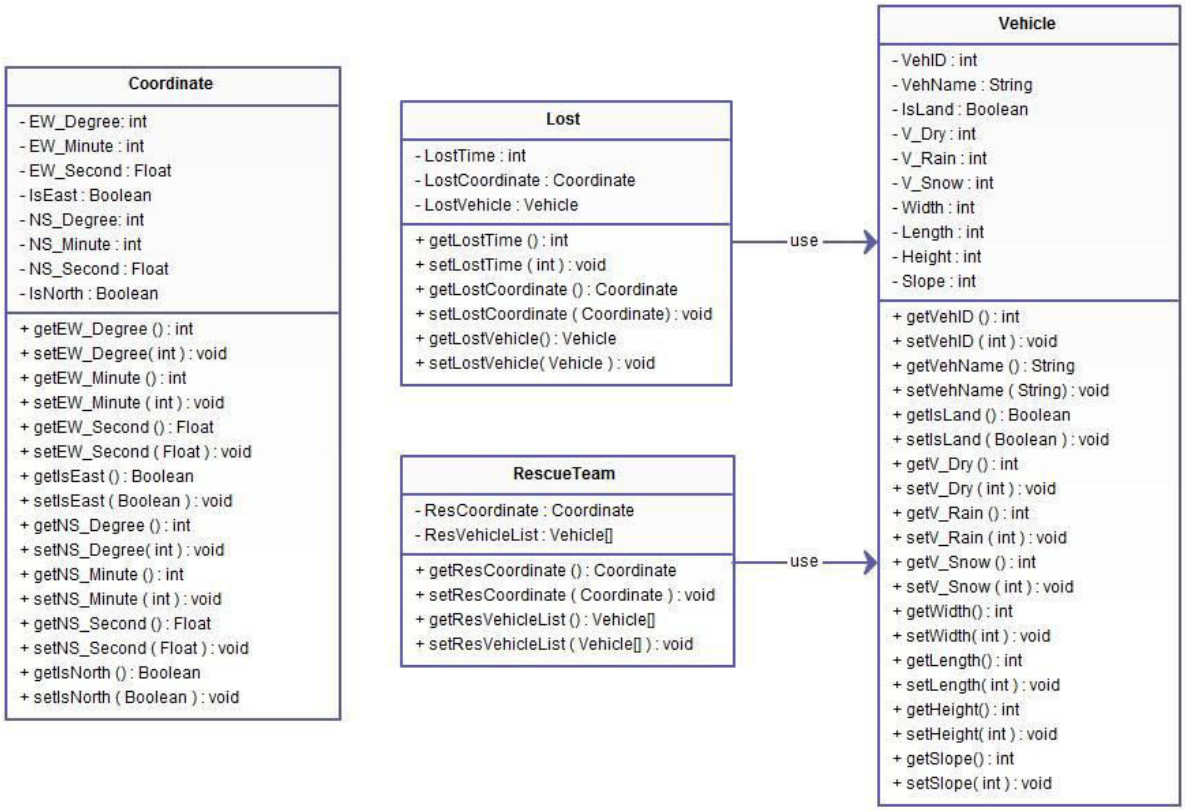
- The application will run on any mobile device that has Android version 5.0 and later versions.
- The web application will run on any web browser

5.3.6.6 Organizing the specific requirements

5.3.6.7 System mode



5.3.6.1 Functional hierarchy



5.3.7 Additional comments

This SRS has been intended to show a brief description of the system. The requirements of the project have been explained clearly. Developers interested in the project and users of the system can benefit from this document. Design details of the project will be explained in the Software Design Description document.