MadadGaar – Emergency App

By

Ishtiaq Ismail

2014-2018

Supervised By

Dr. Onaiza Maqbool

Department of Computer Sciences

Quaid-i-Azam University

Islamabad

Acknowledgment

First of all I would like to express my sincere and humble gratitude to Almighty Allah whose blessings and guidance has been a real source of all the achievements in my life. I am thankful to my parents and brothers for their prayers and moral & financial support during my educational career.

I offer my gratitude to my supervisor, **Dr. Oniaza Maqbool**, Assistant professor, Department of computer Sciences, Quaid-I-Azam University, Islamabad, who has supported me throughout my project with his patience and knowledge whilst allowing me to work in my own way.

Special thanks to my teachers **Dr. M Afzal Bhatti, Dr. Khalid Saleem, Dr. Ghazanfar Farooq, Dr. Shuaib Karim, Dr. Mubashar Mushtaq, Dr. Asim Raafique, Mr. Umer Rashid, Mrs. Memoona Afsheen, Ms Iffrah Farrukh and Mr. S.M. Naqi** for their kind help and guidance.

I am also thankful to all my friends specially Muhammad Usman for his help and company throughout my studies.

Ishtiaq Ismail

2014-2018

Abstract

MadadGaar is an android based mobile application to help people in emergency sitautions. It provides two facilities to users; one is emergency rescue service and other is blood donation.

In case of emergency people have to call Emergency Service Providers like Rescue 1122. People might not be able to give accurate address of a destination to ambulance so it is confusing. Current system relies on calls from the user who gives information about any kind of emergency. All the important information required such as caller's name, location and emergency type is collected through phone calls which waste time.

To avoid these problems, we introduced **MadadGaar**. In case of emergency user just request Rescue 1122 for rescue by selecting type of emergency. User request along with his location will be provided to rescue team. Rescue team will reach user by using GPS quickly. Moreover, those people who want to donate their blood can register as blood donor on this application so in case of emergency blood could be managed easily.

Table Of Contents

List of Figures

List of Table

1 Software Project Management Plan

1.1 Introduction

This chapter is Software Project Management Plan for MadadGaar – Emergency Application.

1.1.1 Project Overview

MadadGaar is an Android application for Emergency Service Providers. It is based on Rescue 1122 Emergency services. User can request Rescue service in case of emergency. Rescue team reaches accident's place by tracking user's location. Application also provides blood donation facility. People who want to donate blood can register as blood donors. So, in case of emergency blood could be found easily.

1.1.2 Project Deliverables

1. Software Project Management Plan (SPMP).
2. Software requirement Specifications (SRS).
3. Software Design Description (SDD).
4. Software Test Documentation (STD).

1.2 Project Organization

Project organization provides information about which software process model is followed, what are the roles of different team members and which tools and techniques are used in the project.

1.2.1 Software Process Model

Waterfall Model is used for the development of this project due to the following reasons:

- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- Lowers initial delivery cost.
- Generates working software quickly and early during the software life cycle [1].

1.2.2 Roles and Responsibilities

Being a single member of this project, all roles and responsibilities are on my side.

1.2.3 Tools and Techniques

I have used the following tools for this project.

Table 1.1 Tools and Techniques

MS word	For the Document Purpose
Draw.io	Online Tool for UML diagrams
Marvel App	For Interfaces
MS Visio	It is used for to develop the UML diagrams
Project Libre	For Project Plan
Android Studio	For Development

1.3 Project Management Plan

Tasks are divided into two parts Analysis Phase and Design Phase.

1.3.1 Analysis Phase Tasks

	Name	Duration	Start	Finish
1	Understanding Problem	2 days	3/3/18 8:00 AM	3/6/18 5:00 PM
2	Making SPMP Document	6 days	3/7/18 8:00 AM	3/14/18 5:00 PM
3	**⊟Analysis Phase**	**46 days**	**3/15/18 8:00 AM**	**5/17/18 5:00 PM**
4	Gathering Requirement	2 days	3/15/18 8:00 AM	3/16/18 5:00 PM
5	Refine Requirement	2 days	3/19/18 8:00 AM	3/20/18 5:00 PM
6	Making of Document V1	1 day	3/21/18 8:00 AM	3/21/18 5:00 PM
7	**⊟Identify Specific Requirement**	**41 days**	**3/22/18 8:00 AM**	**5/17/18 5:00 PM**
8	**⊟External Interface Requirement**	**5 days**	**3/22/18 8:00 AM**	**3/28/18 5:00 PM**
9	User Interface	1 day	3/22/18 8:00 AM	3/22/18 5:00 PM
10	Hardware Interface	1 day	3/23/18 8:00 AM	3/23/18 5:00 PM
11	Software Interface	1 day	3/24/18 8:00 AM	3/26/18 5:00 PM
12	Communication Interface	1 day	3/27/18 8:00 AM	3/27/18 5:00 PM
13	Making of Document V2	1 day	3/28/18 8:00 AM	3/28/18 5:00 PM
14	**⊟Software Product Features**	**16 days**	**3/29/18 8:00 AM**	**4/19/18 5:00 PM**
15	Idenfity Usecases	5 days	3/29/18 8:00 AM	4/4/18 5:00 PM
16	Refine Usecase	8 days	4/5/18 8:00 AM	4/16/18 5:00 PM
17	Making Document V3	3 days	4/17/18 8:00 AM	4/19/18 5:00 PM
18	**⊟Software System Functions**	**9 days**	**4/20/18 8:00 AM**	**5/2/18 5:00 PM**
19	Idenfy System Functions	2 days	4/20/18 8:00 AM	4/23/18 5:00 PM
20	Refine System Funtions	1 day	4/24/18 8:00 AM	4/24/18 5:00 PM
21	Identify Entities and Their Relationship	1 day	4/25/18 8:00 AM	4/25/18 5:00 PM
22	Develop Domain Model	3 days	4/26/18 8:00 AM	4/30/18 5:00 PM
23	Database Requirements	3 days	4/27/18 8:00 AM	5/1/18 5:00 PM
24	Making Document V4	1 day	5/2/18 8:00 AM	5/2/18 5:00 PM
25	**⊟Idenfity Software Attributes**	**5 days**	**5/3/18 8:00 AM**	**5/9/18 5:00 PM**
26	Availability	1 day	5/3/18 8:00 AM	5/3/18 5:00 PM
27	Security	1 day	5/4/18 8:00 AM	5/4/18 5:00 PM
28	Maintainability	1 day	5/5/18 8:00 AM	5/7/18 5:00 PM
29	Portability	1 day	5/8/18 8:00 AM	5/8/18 5:00 PM
30	Making Document V5	1 day	5/9/18 8:00 AM	5/9/18 5:00 PM
31	**⊟Making Final Document**	**6 days**	**5/10/18 8:00 AM**	**5/17/18 5:00 PM**
32	Refine Final Document	6 days	5/10/18 8:00 AM	5/17/18 5:00 PM

Fig (1.1) Analysis Phase Tasks

1.3.2 Design Phase Tasks

		Name	Duration	Start	Finish
33		**Design Phase**	34 days	**5/18/18 8:00 AM**	**7/4/18 5:00 PM**
34		**Design Phase**	21 days	**5/18/18 8:00 AM**	**6/15/18 5:00 PM**
35		Develop Architectural Design	2 days	5/18/18 8:00 AM	5/21/18 5:00 PM
36		Refine Architectural Design	2 days	5/22/18 8:00 AM	5/23/18 5:00 PM
37		Develop Data Design	3 days	5/24/18 8:00 AM	5/28/18 5:00 PM
38		Develop Interface Design	2 days	5/29/18 8:00 AM	5/30/18 5:00 PM
39		Refine Interface Design	2 days	5/31/18 8:00 AM	6/1/18 5:00 PM
40		Develop Sequence Diagram	2 days	6/2/18 8:00 AM	6/5/18 5:00 PM
41		Refine Sequecne Diagram	3 days	6/6/18 8:00 AM	6/8/18 5:00 PM
42		Develop Class Diagram	3 days	6/9/18 8:00 AM	6/13/18 5:00 PM
43		Refine Class Diafgram	2 days	6/14/18 8:00 AM	6/15/18 5:00 PM
44		**Develop Algorithm**	12 days	**6/18/18 8:00 AM**	**7/3/18 5:00 PM**
45		Develop Pseudo Code	10 days	6/18/18 8:00 AM	6/29/18 5:00 PM
46		Review Pseudo Code	2 days	7/2/18 8:00 AM	7/3/18 5:00 PM
47		**Evaluate Design**	11 days	**5/18/18 8:00 AM**	**6/1/18 5:00 PM**
48		Validate Design	3 days	5/18/18 8:00 AM	5/22/18 5:00 PM
49		Verify Design	3 days	5/23/18 8:00 AM	5/25/18 5:00 PM
50		Review and Refine Design	5 days	5/28/18 8:00 AM	6/1/18 5:00 PM
51		Finalize Document	1 day?	7/4/18 8:00 AM	7/4/18 5:00 PM

Fig (1.2) Design Phase Tasks

1.3.3 Analysis Phase Tasks Description

1.3.3.1 Description

Initial requirements have been gathered and analyzed. After which the functional and non-functional requirements are extracted.

1.3.3.2 Deliverables and Milestones

1. Software Project Management Plan
2. Software Requirement Specification

1.3.3.3 Resources Needed

1. MS Office
2. Project Libre
3. Snipping Tool
4. Internet

1.3.3.4 Dependencies and Constraints

This task depends upon Software Project Management Plan.

1.3.3.5 Risks and Contingencies

Not Applicable

1.3.4 Design Phase Tasks

1.3.4.1 Description

Develop Architecture Design, Interface design, Develop Sequence Diagrams and Class diagram.

1.3.4.2 Deliverables and Milestones

Software Design Document

1.3.4.3 Resources Needed

1. MS Word
2. Snipping Tool
3. ArgoUML
4. MS Visio
5. Draw.io
6. Internet

1.3.4.4 Dependencies and Constraints

This task depends upon SRS.

1.3.4.5 Risks and Contingencies

Not Applicable

2 Background of Emergency Rescue Service Providers in Pakistan

2.1 What are Rescue Services

Rescue comprises responsive operations that usually involve the saving of life, or prevention of injury during an incident or dangerous situation. Rescue services are helping people in different emergency situations. These services rush in emergency so that they might aid needy people on time.

2.2 Why we need Rescue Services

Threats, danger and accidents are the global concern. Nobody can omit these situations from its life. Our life is full of hard and tough experiences. People often face some critical situations which can never be denied. Keeping in view these realities man is always concerned about his safety security and protection.

Accidents are part of our life. In some situations, they can never be avoided. In such case where some emergency arises in some sudden situations, need of rescue service rose up. There may be danger to anybody's life at home and may require ambulance. Someone might be injured at the road through accident. In these situations, people need fast rescue services to be provided to tackle such difficulties. Rescue 1122 network working in Punjab is the best example of such services.

2.3 Emergency Service Providers in Pakistan

In Pakistan different governmental and non-governmental organizations are providing assistance to the people in emergency situations.

Some most famous and most important rescue services working in Pakistan are as follows:

- Rescue 1122
- Edhi Ambulance (115)
- Fire Brigade Centre (16)
- Pakistan Red Crescent Society
- Chhipa Welfare Association
- Al-Khidmat Foundation Pakistan

2.4 Why Rescue 1122

Rescue 1122 is the most reliable and trusted emergency service provider in Pakistan because:

- It is free and open for all.
- Provide much more emergency services as compared to others.
- Very fast response i-e an average response time of 7 minutes.
- It is government-based organization.
- Have Large number of Rescue Vehicles.
- Have soft image.

Due to these reasons my Application MadadGaar is based on Rescue 1122 Emergency services.

2.5 Rescue 1122

Rescue 1122 is the leading emergency humanitarian service of Pakistan with infrastructure in all 36 districts of Punjab and is providing technical assistance to other provinces. The service is accessed by calling *1122* from any phone. It was established under the 2006 Punjab Emergency Service Act to provide management of emergencies such as fire, rescue and emergency medical services. Rescue 1122 has rescued millions of victims of emergencies through its Emergency Ambulance, Rescue & Fire services and Community Emergency Response Teams while maintaining its average response time of 7 minutes and standards in all districts of Punjab province [2].

2.5.1 Mission of Rescue 1122

Establishment of an effective system for emergency preparedness, response, protection and prevention; while contributing towards building socially responsible. healthy, resilient and safer communities.

2.5.2 Objectives of Rescue 1122

1) Provision of the right to timely emergency care by providing quality emergency services as per international standards.
2) Undertake research to recommend evidence-based measures to related organizations for prevention of emergencies.
3) Contribute towards establishment of socially responsible community emergency response teams through awareness, enrollment, training and organizing volunteers for emergency preparedness, response and prevention.

2.5.3 Services Provided by Rescue 1122

Rescue 1122 provides following services:

- Ambulance Service
- Fire Service
- Rescue Service
- Community Safety

2.5.4 Performance of Rescue 1122 Since Establishment

Table 2.1 Emergency Data

Road Accidents	Medical Emergencies	Fire Emergencies	Building Collapse	Drowning
1884512	2540868	114909	7439	9755

3 Software Requirement Specification

3.1 Introduction

This chapter describes all the functional and non-functional requirements for MadadGaar – Emergency App.

3.1.1 Problem Statement

In case of emergency people have to call Emergency Service Providers like Rescue 1122. People might not be able to give accurate address of a destination to ambulance so it is confusing. Current system relies on calls from the user who gives information about any kind of emergency. All the important information required such as caller's name, location and emergency type are collected through phone calls which waste time.

3.1.2 Proposed Solution

To avoid these problems, an Android application for Emergency services providers with built-in GPS technology is proposed. In case of emergency user just request Rescue 1122 for rescue by selecting type of emergency. User request along with his location will be provided to nearest rescue team. Rescue team will reach user by using GPS quickly. Moreover, those people who want to donate their blood can register as blood donor on this application so in case of emergency blood could be managed easily.

3.1.3 Scope

This project is about creating an android application for rescuing people in case of emergency like road accident, fire, building collapse or medical emergency.

With this Application:

- User request Emergency service by selecting type of emergency.
- Nearest Rescue team respond to user by tracking user's location.
- People can register as blood donor.
- User can check available blood donors and can request for blood.

3.1.4 Objectives

The objectives of this application are:

- To help people to rescue themselves in case of emergency by simplifying the process of calling 1122.
- To save time taken in collecting accident related information.
- To enable Rescue team to reach easily at accident place.
- To help people in blood finding by simplifying the process.
- To save time in blood finding process.

3.1.5 Definitions, Acronyms and Abbreviations

Table 3.1 Acronyms and Abbreviations

SRS	Software Requirement Specification
SPMP	Software Project Management Plan
SDD	Software Design Document
SD	Sequence Diagram
UI	User Interface
Rescue 1122	Government Organization responsible for helping people in emergency situation.

3.2 Overall Description

This section of the SRS would describe the general factors that affect the application and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which we describe in the specific requirement section.

3.2.1 Product perspective

MadadGaar is standalone application. This is the Android based application for facilitating users in case of emergency situations.

3.2.2 Product Functions

The product functions are:

1) Request Emergency Service
2) Respond to user request
3) Request Blood
4) Register as blood donor

3.2.3 Users and Characteristics

Table 3.2 Users and Characteristics

Emergency User	Emergency User is the person which request emergency service.
Blood Donor	The person who donates blood.
Rescuer/Rescue Team	The person(s) responsible for rescuing people in emergency.

3.2.4 Constraint

- Application is for Android platform
- GPS is must for this application
- Internet is required for application.

3.2.5 Assumptions

- Internet is available 24 hours.

3.3 Specific Requirements

3.3.1 External Interface Requirements

It contains the following interfaces requirements:

3.3.1.1 User Interface

It is android application so all android devices are compatible with this application. Application has easy and user-friendly interface. GUI is provided only in English but a Non-English person can easily use application because icons used in application are very simple and well understood.

Application provides:

- Interface for Emergency User
- Interface for Rescue Team
- Interface for Blood donor.

3.3.1.2 Hardware Interfaces

This application works using GPS so GPS is essential tool for this application. An android phone having 512MB RAM and 1.3 GHZ processor will be required for this application.

3.3.1.3 Software Interfaces

Android platform will be used for the development of this application. Therefore, android phone users from **Jellybean (4.1-4.3.1)** to **Oreo (8.0-8.1)** can use this application.

3.3.1.4 Communication Protocol

It will use Http and TCP/IP protocol for communication.

3.3.2 Functional Requirements

Functional requirements are those requirements which focus on the functionality of the system.

3.3.2.1 List of Use Cases

1) Request Emergency Service
2) Respond to user request
3) Request Blood
4) Register as blood donor
5) Accept Blood Request

3.3.2.2 Use Case Diagram

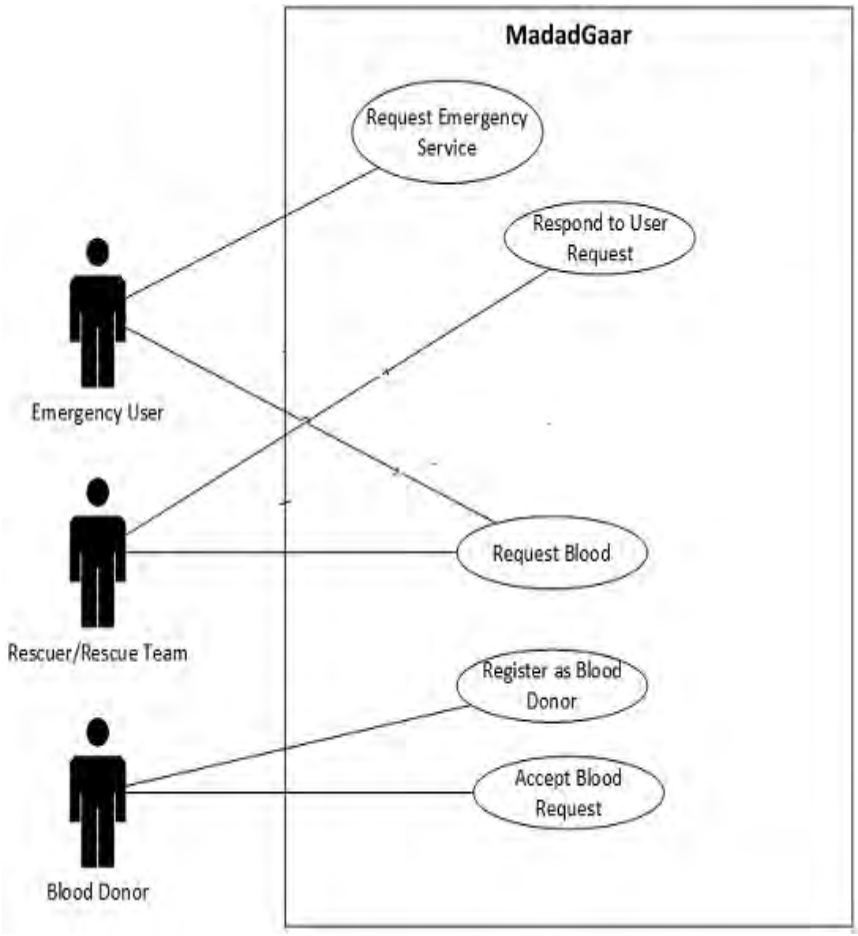


Fig (3.1) Use case Diagram

3.3.2.3 Use Case Details

UC-1: Request Emergency Service
Primary Actor: Emergency User

Pre-Condition: User is Login.

Post Condition: Rescue Team has got user request for emergency service. User location is identified.

Main Success Scenario:

1. User Select Request Emergency Service.
2. System displays different types of emergencies.
3. User selects an emergency type.
4. Use upload an image of incident and send request.
5. System sends request along with user's location to nearest rescue team.
6. System display message regarding successful submission of request.

Alternative Scenario:

1. User may select wrong category of emergency then user have to select again emergency service.
2. System may not accept user request then user will retry.

UC-2: Respond to User Request
Primary Actor: Rescuer/Rescue Team

Pre-Condition: Rescue team have got user request of Emergency.

Post-Condition: Requesting user receive notification that user request is accepted.

Main Success Scenario:

1. Rescue team view emergency request.
2. System display user location, distance and type of emergency.
3. Rescue team accept request to respond.
4. System displays distance and expected arrival time of Rescue team.

Alternative flow:

1) Rescue team may not accept user request due to busy in rescuing another victim then some other team will accept request.

UC-3: Request Blood
Primary actor: Emergency User, Rescuer.

Precondition: User is Login

Post-Condition: Registered Blood donors have received notification regarding blood request of user

Main Success Scenario:

1. User Select Find Blood option.
2. System ask for blood group and Hospital name.
3. User enter blood group and hospital name.
4. System send notification to nearest available blood donors.

Alternate Scenario:

1. System may fail to send blood alert to registered users. Then user will try again

UC-4: Register as Blood donor
Primary actor: Blood donor and Emergency User

Post-Condition:

User record is saved. User has been placed in category of blood donors.

Main Success Scenario:

1. User select Blood donor option.
2. System Prompt for registration.
3. User select registration option.
4. System display a registration form.
5. User fills all mandatory information and submit the form.
6. System asks for confirmation.
7. User confirms.
8. System display a message regarding successful registration.

Alternative Scenario:

1. System may not display registration form. User again tries for registration.
2. User may quit registration process.

3.3.3 Domain Model

A domain model is a representation of real world conceptual classes, not of software components. A domain model is a visual representation of conceptual classes or real world objects in a domain of interest. Domain model describes conceptual classes, association between conceptual classes and attributes of conceptual classes [3].

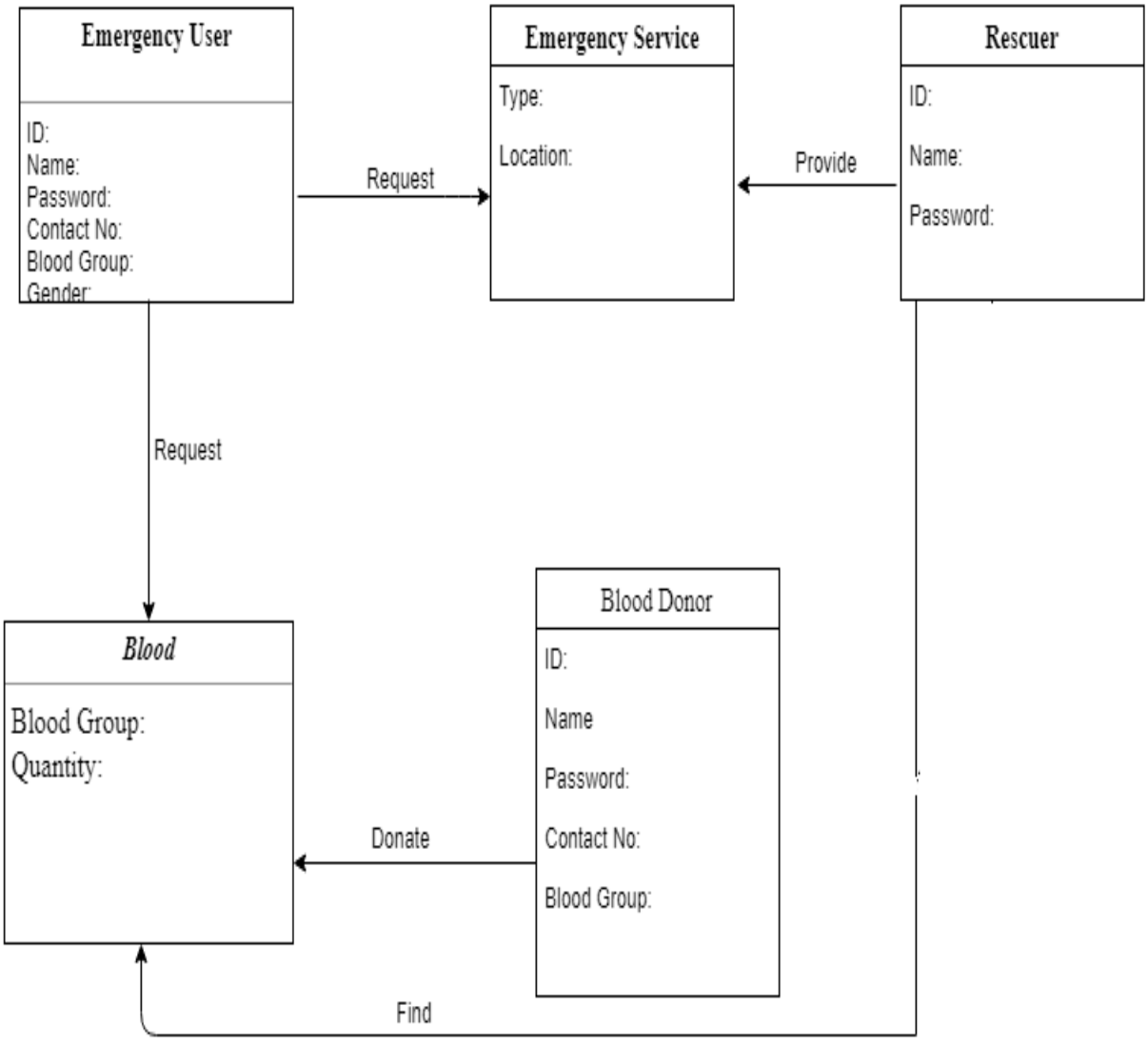


Fig (3.2) Domain Model

3.4 Software System attributes

3.4.1 Availability

The application must be available 24 hours.

3.4.2 Maintainability

During the development period, all the things will be properly documented so that we can easily make changes and upgrade our application.

3.4.3 Portability

Application is portable to other android devices.

3.5 Database Requirements

MadadGaar is an android-based application in which different record are saved. NoSQL database will use for this application to store all records. A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations (ERD) used in relational databases. Relational databases rely on tables, columns, rows, or schemas to organize and retrieve data. In contrast, NoSQL databases do not rely on these structures and use more flexible data models [4].

I am going to use Firebase database which is of document datatype. In firebase Data is stored as JSON.

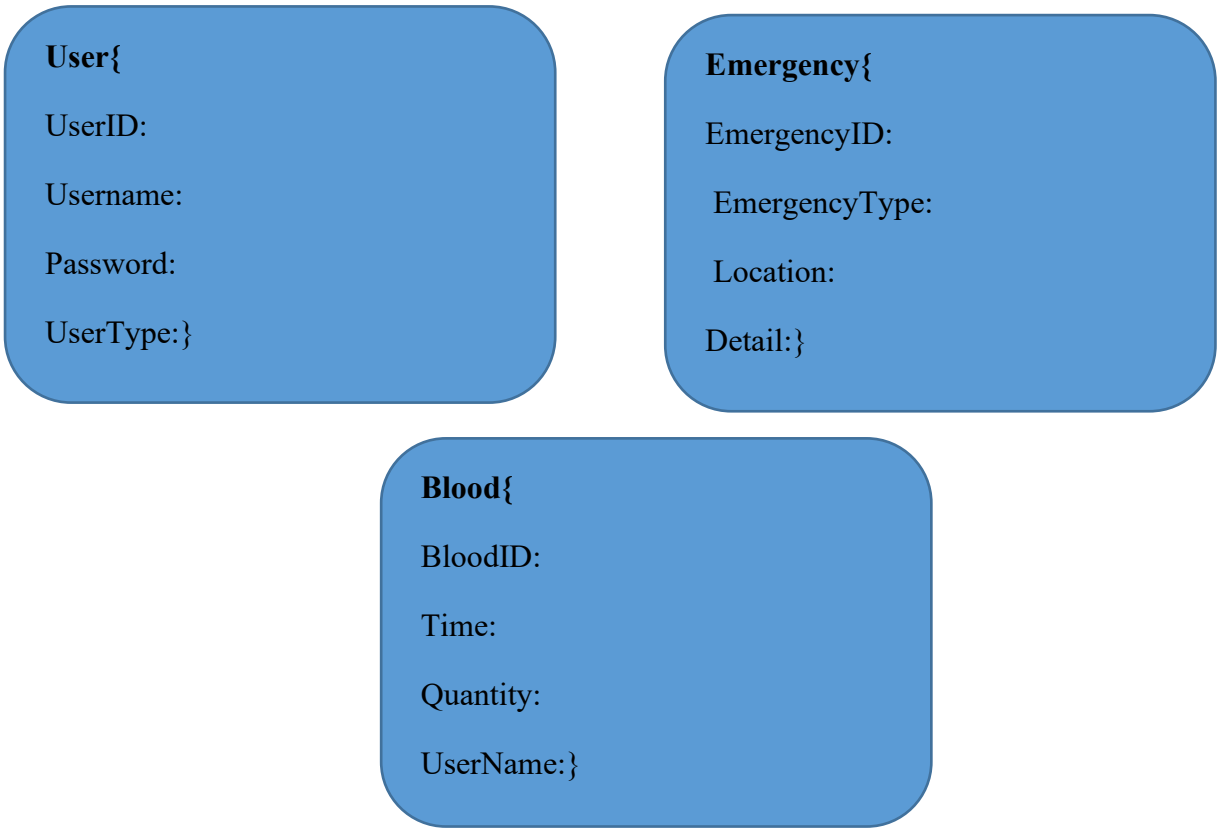


Fig (3.3) Document Object Model

4 Software Design Description

4.1 Introduction

Software Design Description (SDD) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. The SDD document describes the system architecture design in detail and also provides the complete description of the different component. It also describes that how the different components will communicate with each other. The SDD document also contains the interface design, sequence and Class diagram.

4.1.1 Requirement Traceability Matrix

Requirement traceability matrix is the matrix which keeps track the entire requirement throughout the development of the software. The requirement traceability matrix is required during the validation process to check either we have fulfilled the entire requirements or not.

Table 4.1 Traceability matrix

Requirement ID	Requirement Name	Interface Design	Test Case	Sequence Diagram	Class Diagram
UC-1	Request Emergency Service	UI:2	TC-1	SD-1	Fig 4.13
UC-2	Respond to User Request	UI:2	TC-2	SD-2	Fig 4.13
UC-3	Find Hospital	UI:3	TC-3	SD-3	Fig 4.13
UC-4	Request Blood	UI:4	TC-4	SD-4	Fig 4.13
UC-5	Register as Blood donor	UI:5	TC-5	SD-5	Fig 4.13

4.2 Software Architecture Design

An architecture is the set of significant decisions about the organization of a software system, the selection of the structural elements and their interfaces by which the system is composed, together with their behavior as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style. Architectural design is the resolution of the requirements in the design of the software, the hardware and networking, operations, policies, and so forth.

4.2.1 Chosen System Architecture

3-Tier architecture is used to make this project. Basically, Three-tier (layer) is a client-server architecture in which the user interface, business process (business rules) and data storage and data access are developed and maintained as independent modules. Basically, there are 3 layers, tier 1 (presentation tier, GUI tier), tier 2 (business objects, business logic tier) and tier 3 (data access tier). These tiers can be developed and tested separately.

Three-Tier Architecture provides the following benefits.

- Scalability—Each tier can scale horizontally. We can load-balance the Presentation tier among three servers to satisfy more Web requests without adding servers to the Application and Data tiers.
- Performance—Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers
- Availability—If the Application tier server is down and caching is sufficient, the Presentation tier can process Web requests using the cache.

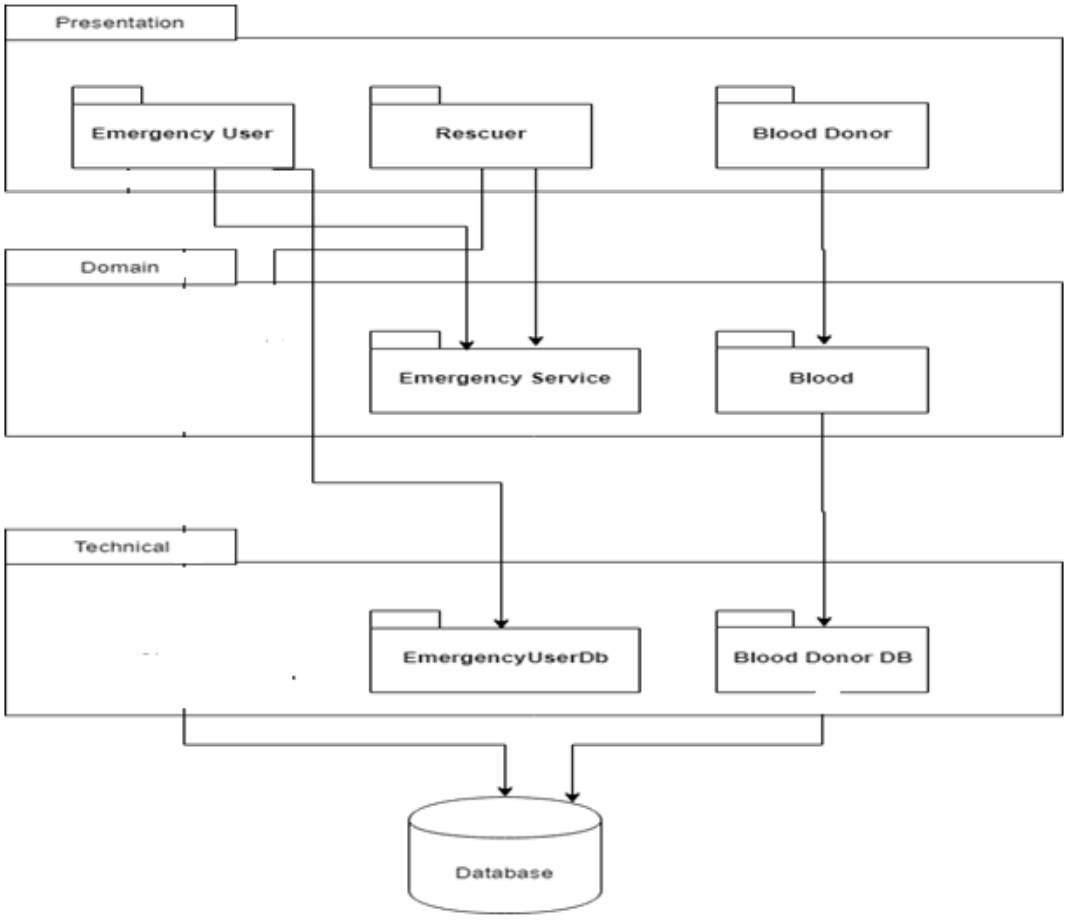


Fig (4.1) Architecture Diagram

4.2.2 Discussion of Alternative Design

An alternative design was to use two-tier architecture, based on Client Server. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server. Because of tight coupling a 2-tiered application will run faster. But the main problem of two tier architecture is the server cannot respond multiple request same time, as a result it causes a data integrity issue. Communication is easy but performance will be degraded upon increasing the users, So I did not use this design due to the performance issues and continued the above one.

4.3 User Interface Design

It is smartphone's application so all android phones are compatible with this application. Application provides different interfaces for users. Application has user interface for following user:

- Emergency User
- Rescuer
- Blood Donors

4.4 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

4.4.1 SD-1: Request Emergency Service

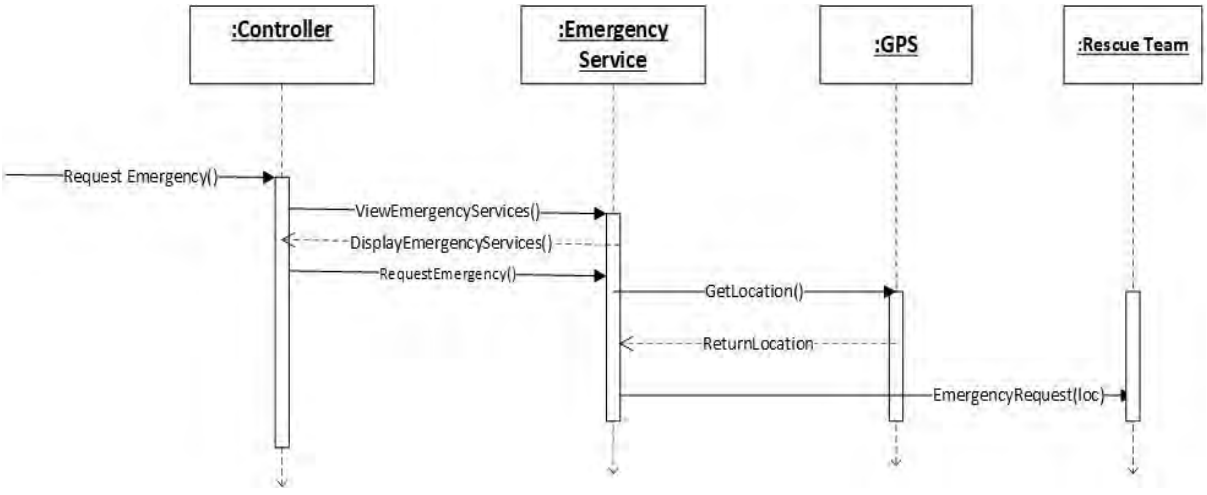


Fig (4.9) SD-1 Request Emergency

4.4.2 SD-2: Respond to User Request

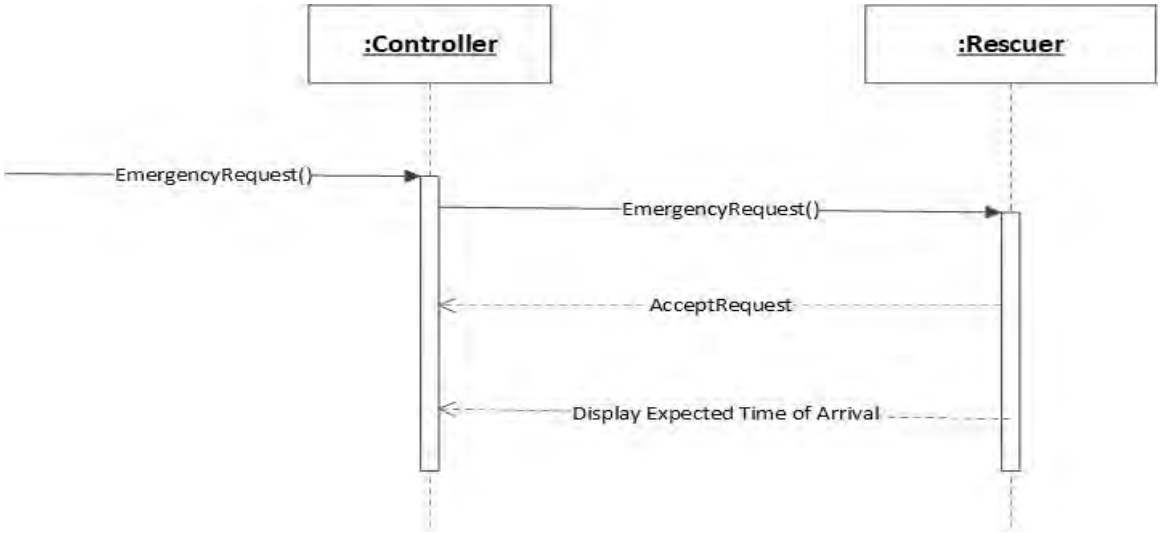


Fig (4.10) SD-2 Respond to user request

4.4.3 SD-3: Register as Blood Donor

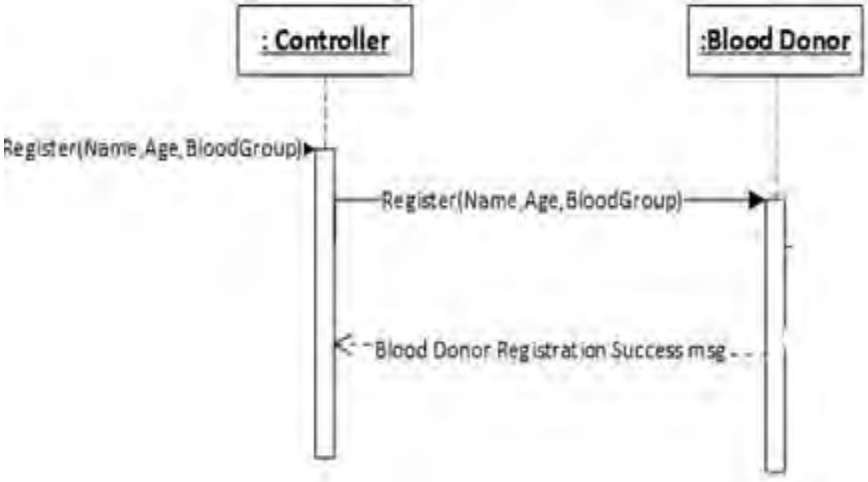


Fig (4.11) SD-3 Register as Blood Donor

4.4.4 SD-4: Request Blood

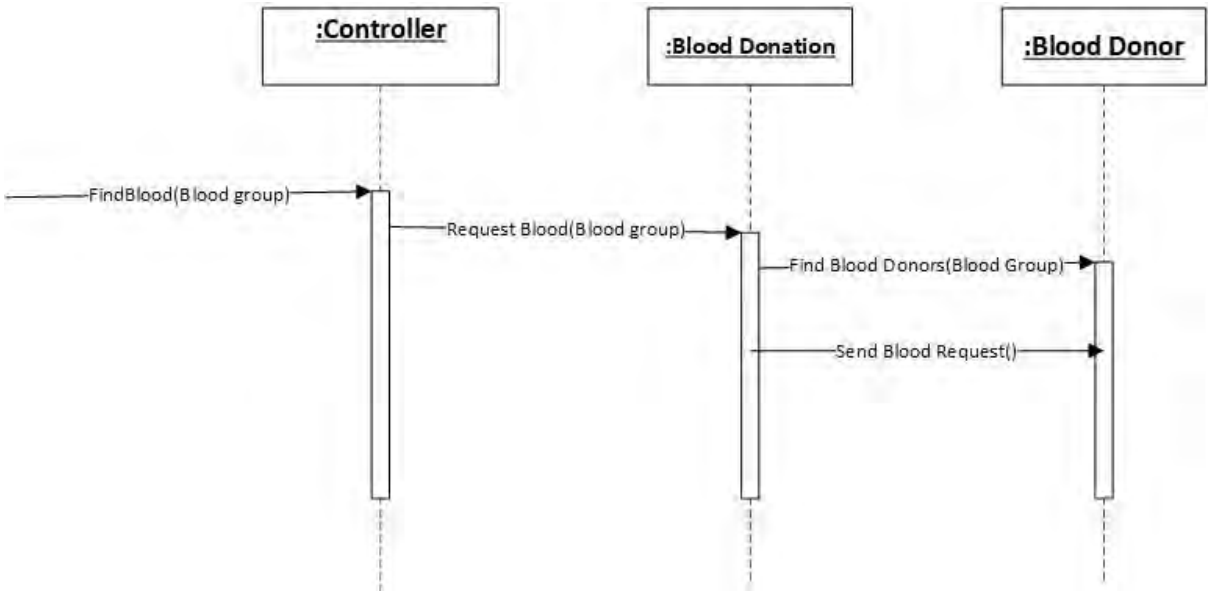


Fig (4.12) SD-4 Request Blood

4.5 Class Diagram

Class diagrams depict the software classes and their relationships. This diagram defines individual classes along with their attributes, types of the attributes, and operations, associations between classes and navigability (direction of association) that define attribute visibility, and dependency relationships that define non-attribute visibility. Conceptual models (Domain Model) and interaction diagrams are very useful to identify the software classes (classes in class diagram) and their methods.

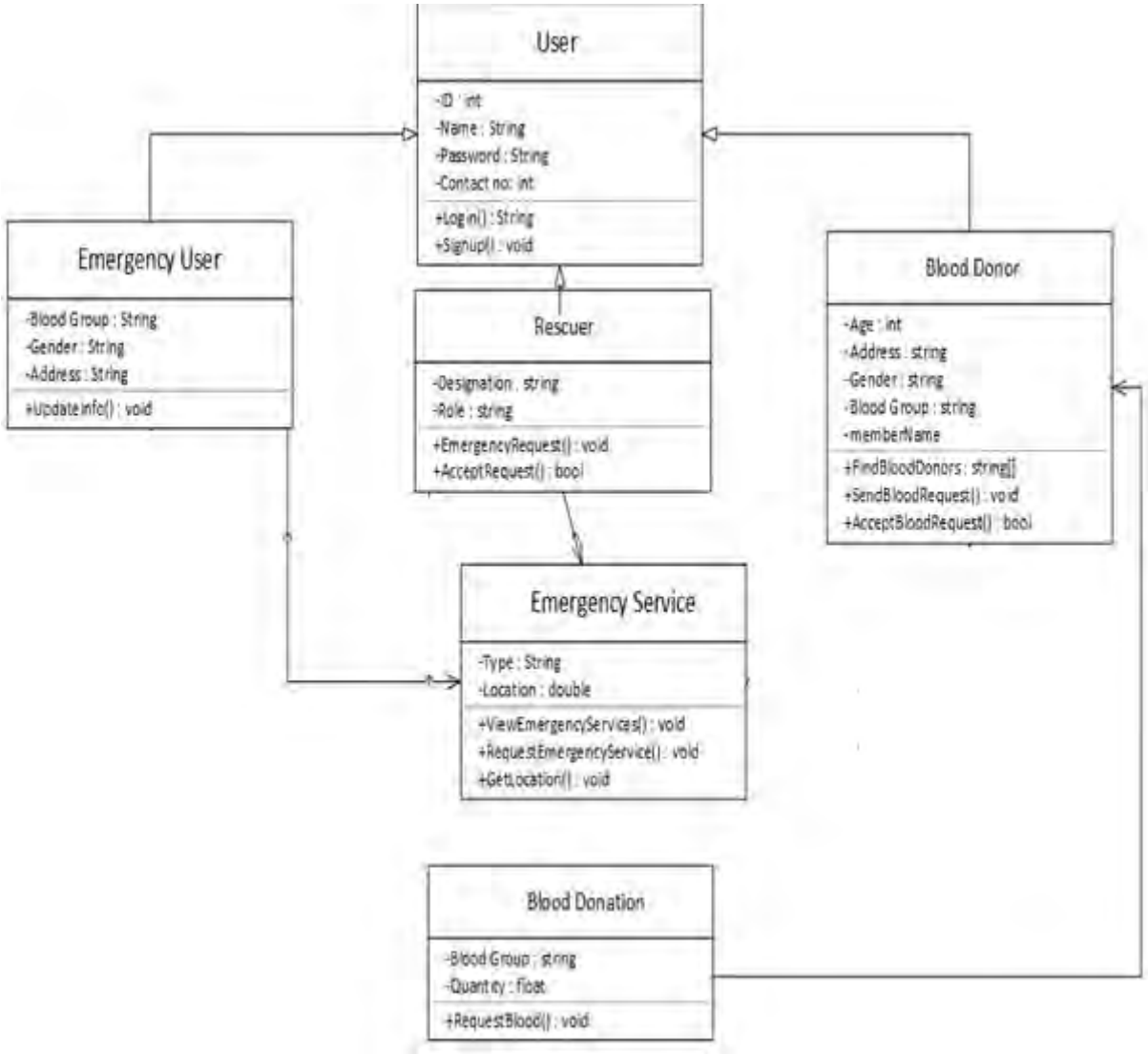


Fig (4.14) Class Diagram

5 Software Implementation

5.1 Introduction

In this chapter, the framework and language selection for the project is provided. It also includes screenshots for the implementation of the application.

5.2 Framework Selection

The framework selected for this project is Android Development Kit. Android development kit support development of applications that run on variety of Android devices. Android development is free to use kit includes an IDE i.e. Android Studio. It supports development and testing of application. Android studio uses Gradle as its build to convert source and resource files into Android application package.

5.3 Language Selection

I have use Java language to develop the Application logic of the Android application. And use XML to design the UI of the android Application.

5.4 Database Selection

Firebase cloud hosted database is used for data storage.

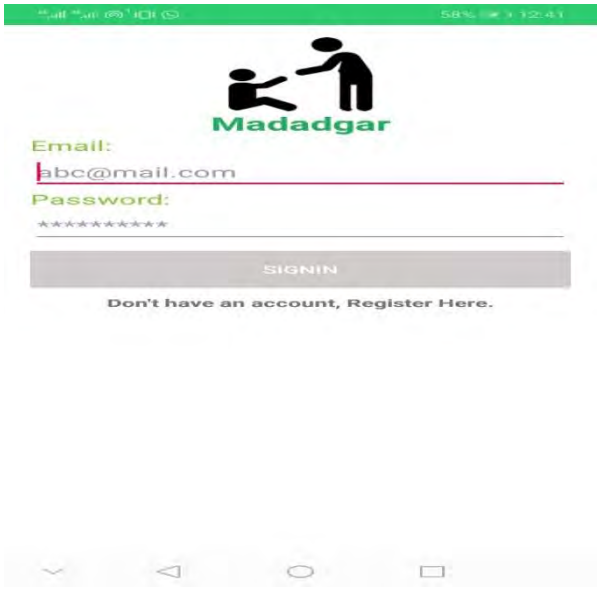
The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in Realtime.

5.5 Software Used

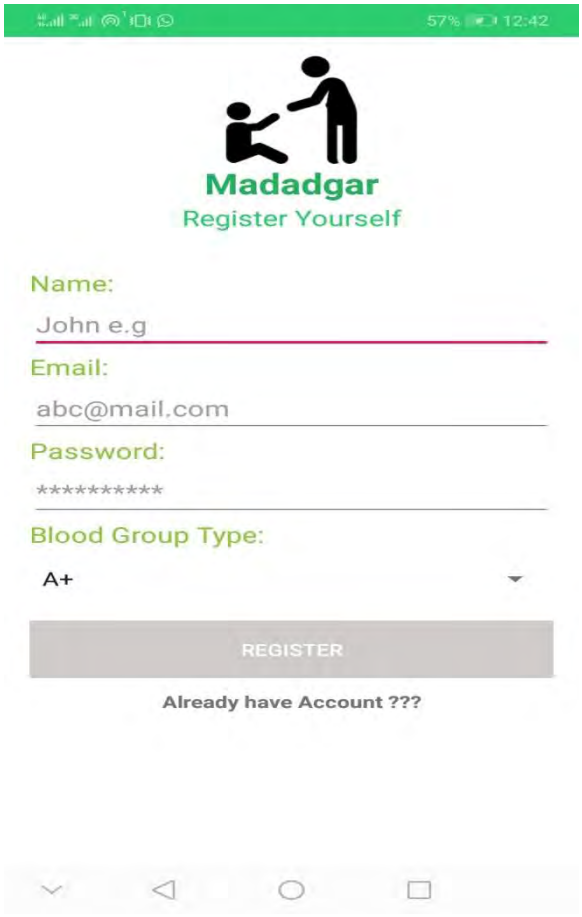
- Android Studio 3.3.1.
- Firebase Server

5.6 Application Screenshots

5.6.1 User Login Screen

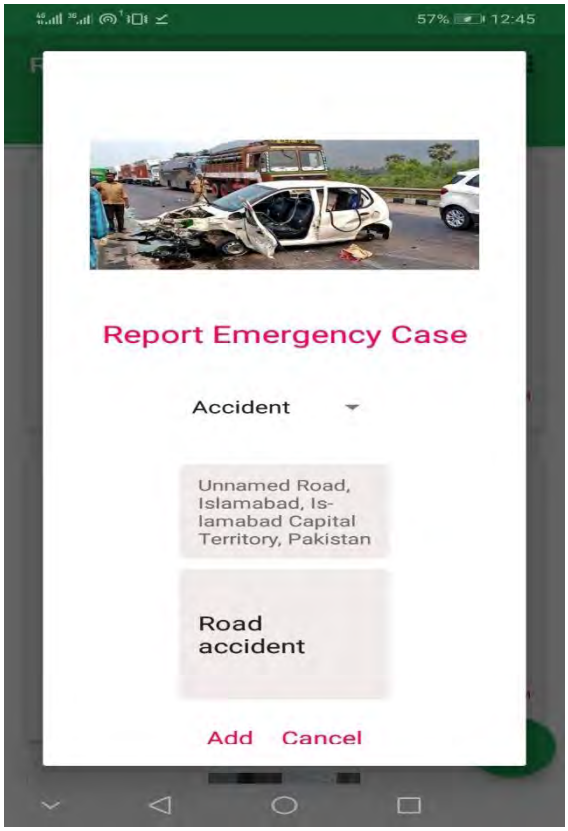


5.6.2 Signup Screen

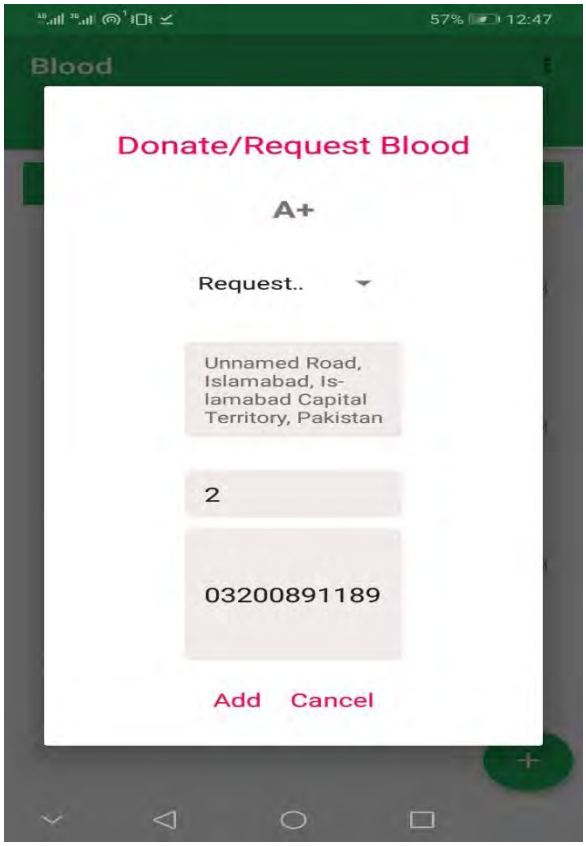


5.6.3 User Dashboard

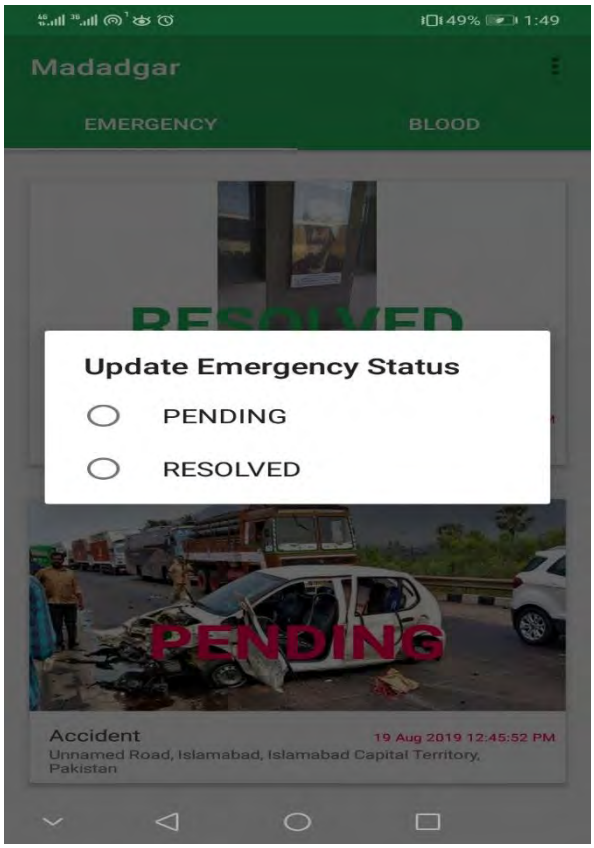
5.6.4 Request Emergency



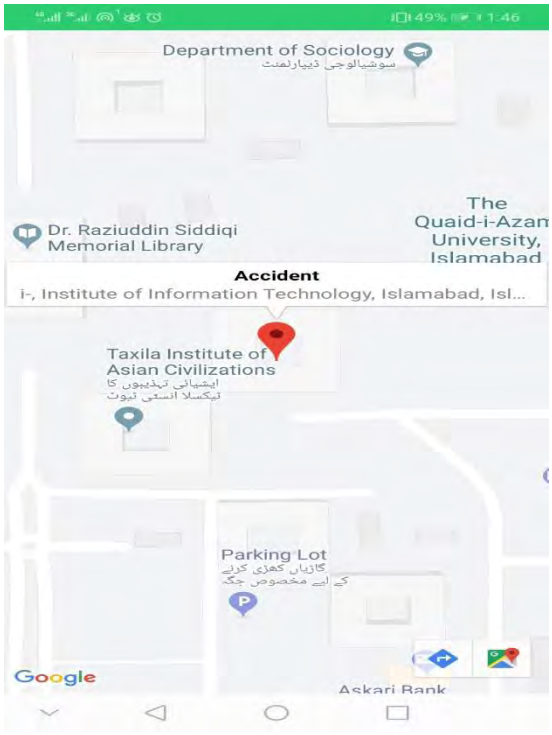
5.6.5 Request Blood



5.6.6 Service Provider Dashboard



5.6.7 Emergency Tracking



6 Software Test Document

6.1 Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. This chapter specifying the test approach, test plan which include features to be tested, testing tools and techniques and test cases that involve a set of steps that can be used while performing testing tasks.

6.2 Test Strategy

Test approaches or test strategy is one of the most powerful factor in the success of the test effort and the accuracy of the test plans and estimates. I will use **Acceptance test** approach for testing the NAA. Acceptance test is a test conducted to determine if the requirements of a specification are met. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

6.3 Test Plan

A test plan describes how testing will be accomplished on a software product, together with the resources and schedule needed. It is the most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. It also defines the size of the test effort.

It also describes that which features to be tested and which features are not to be tested.

6.3.1 Features to be Tested

All the major functions (functional requirements) of MadadGaar Application are to be tested. Following are the list of features to be tested:

- Request Emergency Service
- Respond to User Request
- Register as Blood Donor
- Request Blood

6.3.2 Testing Tool and Environment

Testing environment consists on those tools which will be used to conduct the actual testing. As this System is an Android application therefore, we will use the Android studio for the testing purpose because the Android studio also provides the separate environment to test the Application.

6.4 Test Cases

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. It is good practice to make your test cases atomic.

6.4.1 TC-1 Request Emergency Service

Table 5.1 TC-1 Request Emergency

ID	TC-1 Request Emergency Service
Description	Check that user can request emergency service in emergency situation.
Actor	Emergency User.
Setup	Display Request Emergency option to user.
Instruction	1. Select Request Emergency option. 3. Select type of emergency. 4. Upload captured image of incident. 5. Click submit button. 6.Enter Age.
Expected Result	Rescue Team has got user request and location.
Observed Result	Rescue Team has got user request and location

6.4.2 TC-2 Respond to User Request

Table 5.2 TC-2 Respond to User Request

ID	TC-2 Respond to user request
Description	Check that Rescue Team can accept user request
Actor	Rescuer/Rescue Team
Setup	Display Emergency Request and location of Emergency.
Instruction	1. View Emergency location and other details related to emergency. 2. Accept request.

	3. Update status.
Expected Result	User received request approval message.
Observed Result	User received approval message

6.4.3 TC-3 Register as Blood Donor

Table 5.3 TC-3 Register as Blood Donor

ID	TC-2
Description	This test will check the registration feature of system.
Actor	User
Setup	1.Display Signup screen.
Instruction	2. Enter name “Ali”. 3. Enter ID “1234567891234”. 4. Enter password “xxxxxx”. 5. Enter email “ali123@gmail.com”. 6.Enter Age. 7.Enter Blood group. 8. Enter contact number “03001238933”. 9. Click the submit button.
Expected Result	“Signup Successful” message is displayed.
Observed Results	User registered successfully.

6.4.4 TC-4 Request Blood

Table 5.4 TC-4 Request Blood

ID	TC-6
Description	Find Blood.
Actor	Emergency user, Rescuer
Setup	Display blood donation screen to user.
Instruction	1. Select Find Blood option. 2. Enter Blood group. 3. Enter quantity required.
Expected Result	1. Nearest blood donors got blood request. 2. User received message regarding successful submission of their request.
Observed Results	User received success message of blood request.

7 Conclusion and Future Enhancements

7.1 Introduction

This document describes the project conclusions and future enhancements.

7.2 Conclusion

MadadGaar will allow user to send emergency request in case of emergency. User upload incident picture, select emergency category and send request. Rescue team view emergency request and track user location. After processing user request, Rescue team update emergency status. MadadGaar also allow user to donate and request blood.

7.3 Future Enhancements

In future application can be enhanced by:

- Real time tracking will be included in application.
- Application will be available for web users.
- Application will contain messaging service.

References:

[1] Roger S. Pressman, Software Engineering - A Practitioner's Approach, Seventh Edition, 2010.

[2] Rescue 1122 official website; www.rescue.gov.pk.

[3] Craig Larman, Applying UML and Patterns, Second Edition, 2001.

[4] https://firebase.google.com/docs/firestore/data-model.