VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Jnana Sangama, Belgaum-590014, Karnataka, INDIA

PROJECT REPORT
on

"Disaster Management Using Crowdsourcing"

Submitted in partial fulfillment of the requirements for the VIII Semester

Bachelor of Engineering
IN
COMPUTER SCIENCE AND ENGINEERING

For the Academic year
2013-2014
BY

Arpith K	1PE10CS018
Kumar Vishal	1PE10CS047
Vinutha M V	1PE10CS115
Manigandan C	1PE11CS408

Under the Guidance of
Mrs. Bidisha Goswami
Assistant Professor, Dept. of CSE
PESIT-BSC, Bangalore-560100

Department of Computer Science and Engineering
PESIT BANGALORE SOUTH CAMPUS
Hosur Road, Bangalore -560100

PESIT BANGALORE SOUTH CAMPUS
Hosur Road, Bangalore -560100

Department of Computer Science and Engineering

CERTIFICATE

Certified that the project work entitled ***"Disaster Management Using Crowdsourcing"*** *is a bonafide work carried out by* ***Arpith K, Kumar Vishal, Vinutha M V*** *and* ***Manigandan C*** *bearing USN* ***1PE10CS018, 1PE10CS047, 1PE10CS115*** *and* ***1PE11CS408*** *respectively, students of* ***PESIT Bangalore South Campus*** *in partial fulfillment for the award of* ***Bachelor of Engineering*** *in* ***Computer Science and Engineering*** *of the* ***Visvesvaraya Technological University****, Belgaum during the year 2013-2014. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.*

Signatures:

Project Guide **Mrs. Bidisha Goswami** Asst. Professor, Dept. of CSE PESIT-BSC, Bangalore	Prof and Head Dept of CSE **Dr. Srikanta Murthy. K** HOD, Dept. of CSE, PESIT-BSC, Bangalore	Director/Principal **Dr. Surya Prasad J** PESIT-BSC, Bangalore

External Viva

Name of the Examiners	**Signature with date**
1. ______________________	______________________
2. ______________________	______________________

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned my effort with success.

We are indebted to our Guide, **Mrs. Bidisha Goswami,** Asst. Professor, Department of Computer Science and Engineering, PESIT Bangalore South Campus, who has not only coordinated our work but also given suggestions from time to time.

We are also extremely grateful to our Project Co-ordinator, **Mrs. Pooja Agarwal** Associate Professor, Department of Computer Science and Engineering, PESIT Bangalore South Campus, for her constant support and advice throughout the course of preparation of this document.

We are greatly thankful to **Dr. Srikanta Murthy K**, Prof and HOD, Department of Computer Science and Engineering, PESIT Bangalore South Campus, for his able guidance, regular source of encouragement and assistance throughout this project.

We would like to express our immense gratitude to **Dr. Surya Prasad J,** Director and Principal, PESIT Bangalore South Campus, for providing us with excellent infrastructure to complete our project work.

We gratefully acknowledge the help lent out to us by all faculty members of the Department of Computer Science and Engineering, PESIT Bangalore South Campus, at all difficult times. We would also take this opportunity to thank our college management for the facilities provided during the course of the project. Furthermore, we acknowledge the support and feedback of my parents and friends.

Arpith K
Kumar Vishal
Vinutha M V
Manigandan C

ABSTRACT

Crowdsourcing is a method of information collection which utilizes data received from volunteers. One of the approach to its use is in the production of volunteer-produced information before, during and after a disaster takes place. This approach is inherently in line with the requirements of disaster preparedness and recovery.

During a disaster, information is at a premium: there is an urgent need to know which areas are affected, how they are affected, what the priority problems are and so forth. This typically coincides with the disruption of traditional lines of communication. The advent of modern connected devices, such as mobile phones, has provided a new digital development infrastructure, which may be of significant value to disaster response.

When existing surveillance sensors used by a disaster warning and response system cannot provide adequate data for situation assessment purposes, crowdsourcing information collection can be an effective solution: People armed with wireless devices and social network services can be used as mobile human sensors. Eye-witness reports from them can complement data from in-situ physical sensors and provide the system with more extensive and detailed sensor coverage.

The crowdsourcing strategy used by the system can be random, relying solely on mobility of individuals for coverage of the threatened area, or crowd driven, with the system providing situation updates as feedbacks to aid the crowd; or system-driven with individuals moving in response to directives from the system. With such a system, emergency management professionals can also receive most of the information they need for preparing themselves to perform a timely and accurate treatment of their patients even before dispatching a response team to the event.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

The effectiveness of emergency response largely depends on having a precise, up-to-date situational picture. The aim of this project, is to collect massive amounts of citizen-generated content to get this situational picture and extract other relevant information to support the command staff in making better informed decisions and aid the crowd.

1.1 Purpose

During a disaster, information is at a premium: there is an urgent need to know which areas are affected, how they are affected, what the priority problems are and so forth. The advent of modern connected devices, such as mobile phones, has provided a new digital development infrastructure, which is of significant value to disaster response and recovery.

With the use of this system, emergency management professionals receive most of the information they need for preparing themselves to perform a timely and accurate treatment of their patients even before dispatching a response team to the event.

1.2 Scope

Smartphones are quickly becoming the primary computing and communication platform for people's daily tasks. This project proposes a crowdsourcing strategy to be used by the general public to fetch information about an incident or a disaster, relying solely on mobility of individuals for coverage of the threatened area, with the system providing situation updates as feedbacks to aid the crowd; or system-driven with individuals moving in response to directives from the system.

With such a system, emergency management professionals can also receive most of the information they need for preparing themselves to perform a timely and accurate treatment of their patients even before dispatching a response team to the event.

1.3 Acronyms and Abbreviation

The following section will discuss various terms that will be used in this report

1.3.1 Crowdsourcing

Crowdsourcing is the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, and especially from an online community, rather than from traditional employees or suppliers. This process is often used to subdivide tedious work or to fund-raise charities. It combines the efforts of numerous self-identified volunteers or part-time workers, where each contributor of their own initiative adds a small portion to the greater result.

1.3.2 Disaster

A disaster is a natural or man-made (or technological) hazard resulting in an event of substantial extent causing significant physical damage or destruction, loss of life, or drastic change to the environment. A disaster can be ostensive defined as any tragic event stemming from events such as earthquakes, floods, catastrophic accidents, fires, or explosions. It is a phenomenon that can cause damage to life and property and destroy the economic, social and cultural life of people.

1.3.3 Disaster Management

Disaster management can be defined as the organization and management of resources and responsibilities for dealing with all humanitarian aspects of emergencies, in particular preparedness, response and recovery in order to lessen the impact of disasters.

1.3.4 Social Media

Social media refers to interaction among people in which they create, share, and/or exchange information and ideas in virtual communities and networks.

1.3.5 Participatory Sensing

Participatory Sensing is the concept of communities (or other groups of people) contributing sensory information to form a body of knowledge.

A growth in mobile devices, such as the ones running Android, which has multiple sensors, has made participatory sensing viable in the large-scale. Participatory sensing can be used to retrieve information that collectively forms knowledge.

1.4 Literature Survey

Emergency resources are often insufficient to satisfy fully the demands for professional help and supplies after a public disaster. Furthermore, in a mass casualty situation, the emphasis shifts from ensuring the best possible outcome for each individual patient to ensuring the best possible outcome for the greatest number of patients.

Recent disasters have drawn attention to the vulnerability of human populations and infrastructure, and the extremely high cost of recovering from the damage they have caused. Examples range from minor incidents like fire, accidents to Wenchuan earthquake of May 2009, Hurricane Katrina in September 2005, and the Indian Ocean Tsunami of December 2004. In all of these cases impacts were severe, in damage, injury, and loss of life, and were spread over large areas.

In most of these cases modern technology has not been used for the full extent which otherwise, could have brought reports and images to the almost immediate attention of much of the world’s population. Also use of modern strategies such as crowdsourcing can make it possible for millions around the world to be aware of the events as they unfolded in near-real time.

Images captured from satellites have been used to create damage assessments, and digital maps have been used to direct supplies and to guide the recovery effort, in an increasingly important application of Digital Earth. Nevertheless it has been clear in all of these cases that the potential of such data, and of geospatial data and tools more generally, has not been realized, that the benefits of such technology have fallen far short of expectation, and that research is needed on several key issues if the situation is to improve.

1.5 Existing System

Ushahidi

Ushahidi, Inc. is a non-profit software company that develops free and open-source software (LGPL) for information collection, visualization, and interactive mapping. Ushahidi (Swahili for "testimony" or "witness") created a website (http://legacy.ushahidi.com) in the aftermath of Kenya's disputed 2007 presidential election that collected eyewitness reports of violence reported by email and text message and placed them on a Google Maps map.

The organization uses the concept of crowdsourcing for social activism and public accountability, serving as an initial model for what has been coined as "activist mapping"—the combination of social activism, citizen journalism and geospatial information. Ushahidi offers products that enable local observers to submit reports using their mobile phones or the internet, while simultaneously creating a temporal and geospatial archive of events.

1.5.1 Limitations of existing system

The following are the limitations of the existing system:

- No spam control
- Crowdfunding and reporting of missing people is not possible.
- Does not give alerts to the command staff
- Does not provide notification services to inform the crowd about the incident that has occurred near them.
- Does not support new trends such as that of wearable devices.
- Lack of encouragement to use their service defeats the concept of crowdsourcing.

1.6 Proposed System

The proposed system would use the concept of crowdsourcing for social activism and public accountability, serving as an initial model for what has been coined as "activist mapping”. It would use voting mechanism to verify the authenticity of an incident report. Also the proposed system would provide notification services which would alert the command staff, the relief workers and the citizens in the vicinity of the disaster about n incident.

The system under question would also have features to collect money for NGOs using the concept of crowd funding. Also, the vigilant nature of human beings will be used to expedite the recovery of missing people. Further, gamification enhances user engagement with the system.

1.7 Summary

In this chapter, the introduction, the problem statement of the project, the literature survey and the system to be generated are looked into. The application to be generated is discussed in the upcoming chapters. In the next chapter, system requirements and specifications is discussed.

CHAPTER 2
SYSTEM REQUIREMENTS SPECIFICATION

CHAPTER 2

SYSTEM REQUIREMENTS SPECIFICATION

Requirement specification is the activity of translating the information gathered during analysis into requirement document. It is structured collection of information that embodies the requirements of a system.

A Software Requirement Specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

The SRS establishes the basis for agreement between customers and developers on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It also provides a realistic basis for estimating product costs, risks, and schedules.

2.1 Operating Environment

2.1.1 Hardware Requirements

The following are the various hardware requirements of the mobile device:

- A mobile phone using Android OS version 4.2.2 onwards.
- GSM 850/900/1800/1900
- 1 GHz processor
- 512 MB of RAM

The following are the various hardware requirements of desktop:

- Desktop running Windows (XP onwards), Mac (10.6 or later) or Linux
- Intel Pentium 4 or later
- 100 MB of free disk space

- 128 MB of RAM

2.1.2 Software Requirements

The following are the software requirements of the project

- MySQL Cluster 7.2
- Web browser (IE, Chrome, Firefox and Safari is supported)
- Might work on other browsers, but the system will be untested and may not yield unexpected outputs.
- IDE: Eclipse with Android Developers Tools (ADT), Sublime Text

2.2 Functional Requirements

These are statement of services the system should provide, how the system should react for a particular inputs and how the system should behave in a particular situation.

2.2.1 Unlimited amount of users and content

The community would be able to scale to any size without extra fees or functional slowdowns as a result.

2.2.2 Quick and easy reporting of incidents

A user should be able to quickly and easily report an incident with the Android application. The User Interface (UI) should be intuitive to use.

2.2.3 Global Access

All reports and incident lists should be accessible from anywhere, be it from a mobile device or a desktop.

2.2.4 Single Login

It’s important to create a single login for various components including notification and providing information. It is essential to create a unified seamless user experience

2.2.5 Different user types

System shall be capable to create different user types with different features. Example- Doctors (to help victims around them immediately), relief workers, general public etc.

2.2.6 User moderation

Users are the best police of bad reports and will be able to flag inappropriate reports for removal. This will help the moderator's spot content that could be damaging to the credibility of the community.

2.2.7 Quick and easy registration

It will be easy for the user to register on to the service. The verification process will be short and he would be able to start using it at the earliest.

2.2.8 Voting system

A voting system will be integrated to give community members the opportunity to rate how helpful reports or answers were and whether or not an incident report helped them or not. This will also assist with seeding the community with quality information.

2.2.9 Search

A user will easily be able to find the information required by him. E.g.- He will be able to search for a particular incident within a specified radius of his current location.

2.3 Non-Functional requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

The following are the various non-functional requirement of the system

2.3.1 Reliability

The system shall be accessible at any time, with the exception of technology infrastructure failure.

2.3.2 Availability

For disaster response availability is the key. Best server design with the use of MySQL cluster for database, will ensure an availability of 99.999%.

2.3.3 Security

The system's security can be ensured by converting passwords to hashes by SHA algorithm. General users shall not have access to modify reports within the system.

2.3.4 Maintainability

The system shall be easily maintainable by the administrator. Also, other programmers shall be capable of easily modifying and updating code by using the documentation provided with the system.

2.4 User Characteristics

User requirements are abstract statement of the system requirements for the customer and end user of the system who do not have a detailed technical knowledge of the system.

- User should open the application when required.
- Alert messages should be displayed on the notification bar or on other wearable devices such as a smart watch, if present.
- He should have access to all menu items with ease.
- User should be able to report an incident or perform other actions with ease.
- Application must display accurate information to the user in a readable format.

2.5 Feasibility Study

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional e1nbarrassment can be averted if an illconceived system is recognized early in the defini6on phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case there are three primary areas of interest:-

- Technical Feasibility
- Economic Feasibility
- Social Feasibility

2.5.1 Technical Feasibility

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system. The proposed system does not have high demands and is completely built on resources, sensors and features already available on a mobile device.

2.5.2 Economic Feasibility

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

Most of the functions are implemented by the use of software modules and cost virtually nothing. The only area that require investment is to create a cluster. But however, considering the benefits that can be reaped, the costs incurred are justified.

2.5.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not be threatened by the system. His level of confidence must be increased so that the entire community can be benefited by his constructive reports. Also the concept of gamification is used to improve user engagement with the system.

2.6 Advantages of the proposed system:

- Helps the command staff and relief workers to make better informed decisions by providing real-time analytics and thus, aid the crowd.
- Makes mass populations more aware of incidents in their locality.
- With the concept of Crowd funding, even if an individual donates a small amount of money, the total revenue generates is usually sufficient for the cause.
- This system enables donations of items other than money (like used clothes) and hence aids those affected by a disaster.
- Enables a more expedite way to find missing people.
- The concept of gamification improves user engagement with the system.

2.7 Summary

In the above discussed chapter, the system requirements are discussed, with the hardware and software requirements, functional and non-functional requirements, user and system characteristics are some of the characteristics of the systems that are discussed. In the next chapter, high-level design of the application is designed.

CHAPTER 3
HIGH LEVEL DESIGN

CHAPTER 3

HIGH LEVEL DESIGN

The purpose of the design is to obtain the solution of a problem specified by the system requirements. This phase is the first step in moving from problem to the solution domain. In other words, starting with what is needed to how long it takes to design it, defines how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software and has a major impact on the later phases, particularly testing and maintenance.

3.1 Design Considerations

High level design covers specific assumption and dependencies for the project. It also covers data flow diagram which gives the flow of data for overall project, DFDs includes two levels - Level 0 and Level 1.

3.1.1 Assumptions and Dependencies

The product is entirely based on development of the Disaster Management Using Crowdsourcing. The basic assumptions and dependencies are as follows:

- Users are required to have an Android device with internet connectivity.
- All devices have Bluetooth, which helps in forming the scatternet for chatting during emergencies.
- The admin handles the maintenance of user information on the database and only admin can add, remove or update the clients.
- Once connected all features of the application are available to all users.

3.1.2 Goals and Constraints

The basic goals and constraints are as follows:

- The admin/user cannot login to application without correct username and password.
- The user cannot establish connection without entering the valid server IP address. However a default IP address is provided for his convenience.
- Admin on server side manages the user accounts.
- The users must have an Android device with the android version 4.2.2+.

3.2 System Architecture

Large systems are always decomposed into sub-system that provides some related set of services. The initial design process of identifying these sub-systems and establishing a framework for sub-system control and communication is called architecture design and output of this design process is a description of the software architecture.

The architectural design process is concerned with establishing a basic structural framework for a system. It involves identifying the major components of the system and communication between these components.

Our application would be accomplished on a Server-Client Architecture. The client–server model of computing is a distributed application that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. The Server Side would be a continuously running service listening to the different Clients asking its services. The application would be installed on every communicating client. A Database of users would be maintained by the Server. When a client logs into the application, the Server authenticates the user of the client system. Once the user is authenticated, it sends the list of online features and other relevant data to the Client. Similarly, multiple clients can connect simultaneously. When the user wishes to report an incident, his details along with his present coordinates are stored in the database. This new report is updated in all the devices accessing the application. Similarly other features are synchronised. The admin on the server side can see the consolidated report using features such as heat maps. The admin with the database can manage the user accounts.

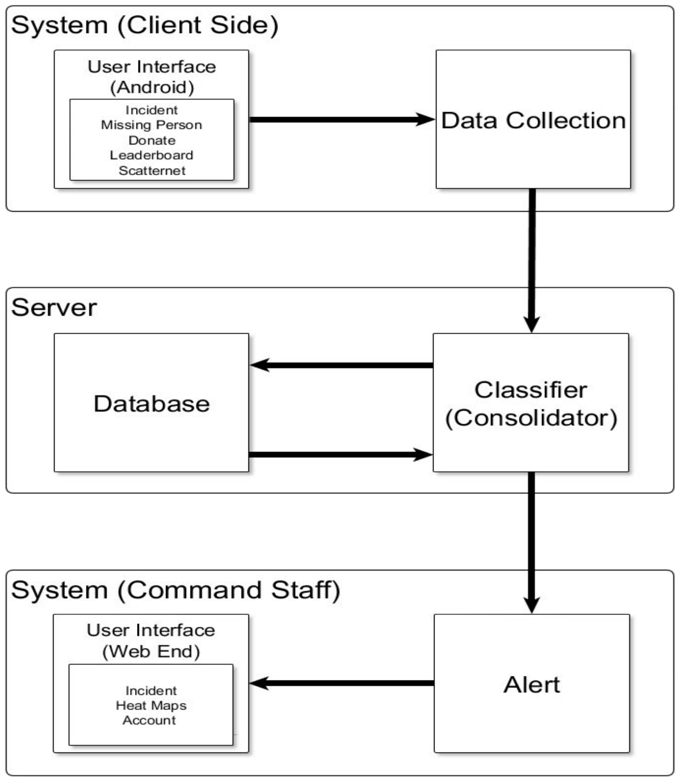


Fig 3.1: System Architecture Diagram

3.3 Data Flow Diagrams

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD which is then “exploded” to show more detail of the system being modelled.

With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish and how the system will be implemented. Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input, ultimately has an effect upon the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram. There are several common modelling rules to be followed while creating DFDs. They are listed below.

- All processes must have at least one data flow in and one data flow out.
- All processes should modify the incoming data, producing new forms of outgoing data.
- Each data store must be involved with at least one data flow.
- Each external entity must be involved with at least one data flow.
- A data flow must be attached to at least one process.

For this project of Disaster Management using Crowdsourcing, data flow diagram have been elaborated in two levels such as level 0 and level 1 separately for server & client.

3.3.1 Data Flow Diagram - Level 0

Level 0 DFDs show the basic data flow included in the overall process. Level 0 highlights the main windows of the application. It contains the first login page for both the user and admin. The android application takes the user to the home page, from where the user can navigate to any option of his/her choice. The web application takes the admin to the report page, from where he/she can navigate to any option. The level 0 DFDs are shown below:

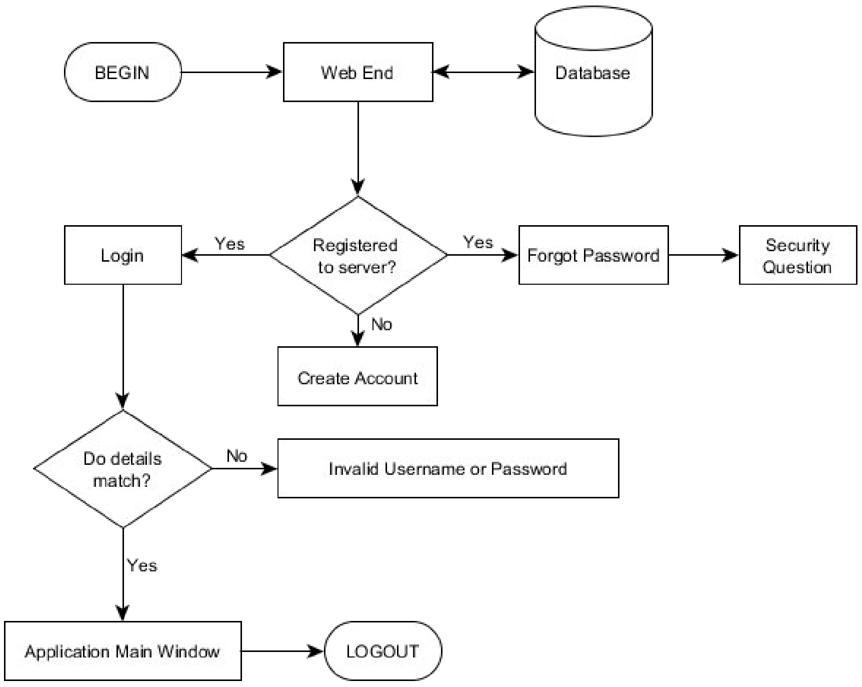


Fig 3.2: Level 0 DFD for Web End

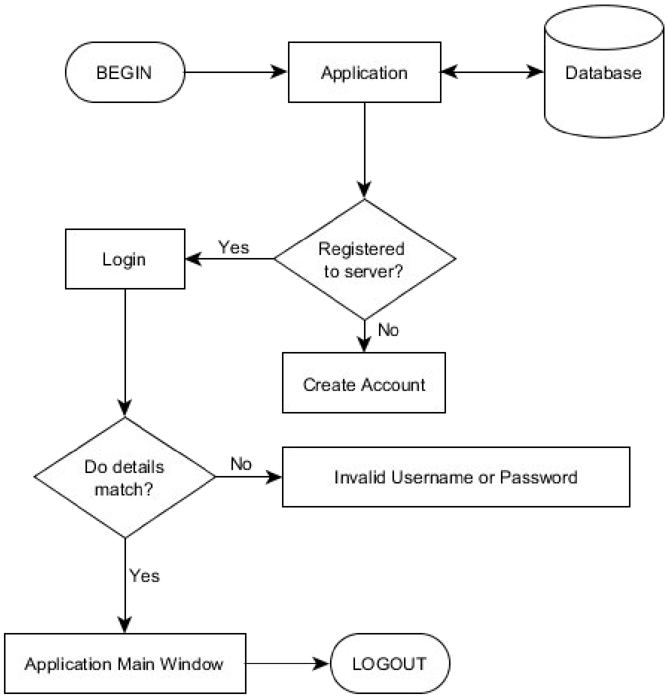


Fig 3.3: Level 0 DFD for User End

3.3.3 Data Flow Diagram - Level 1

Level 1 DFDs shows a more elaborate system design. Level 1 DFDs contain all the different options that a user can navigate to. The android application user can choose between the different options shown in the figure. Similarly the admin can also look at the consolidated report by looking at the heat maps. Level 1 DFDs are shown below:

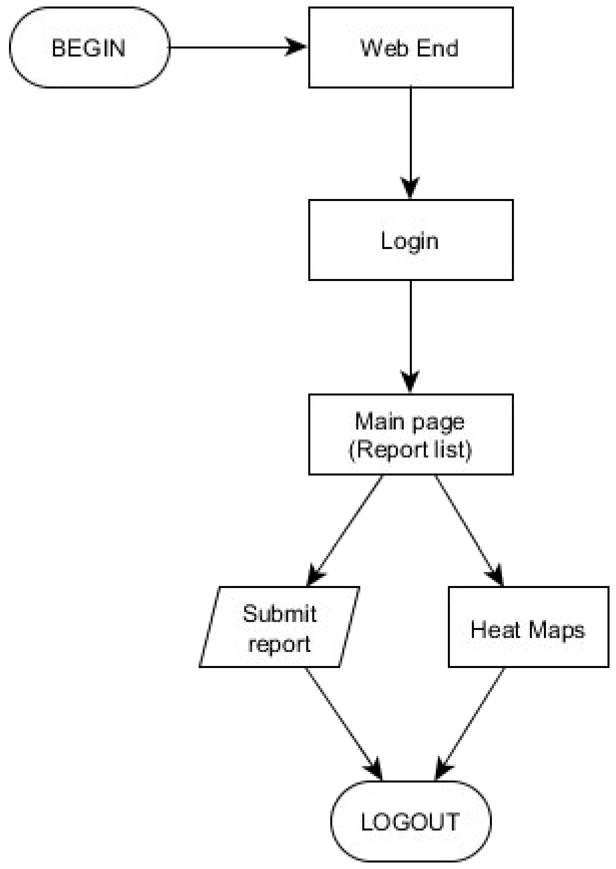


Fig 3.4: Level 1 DFD for Web End

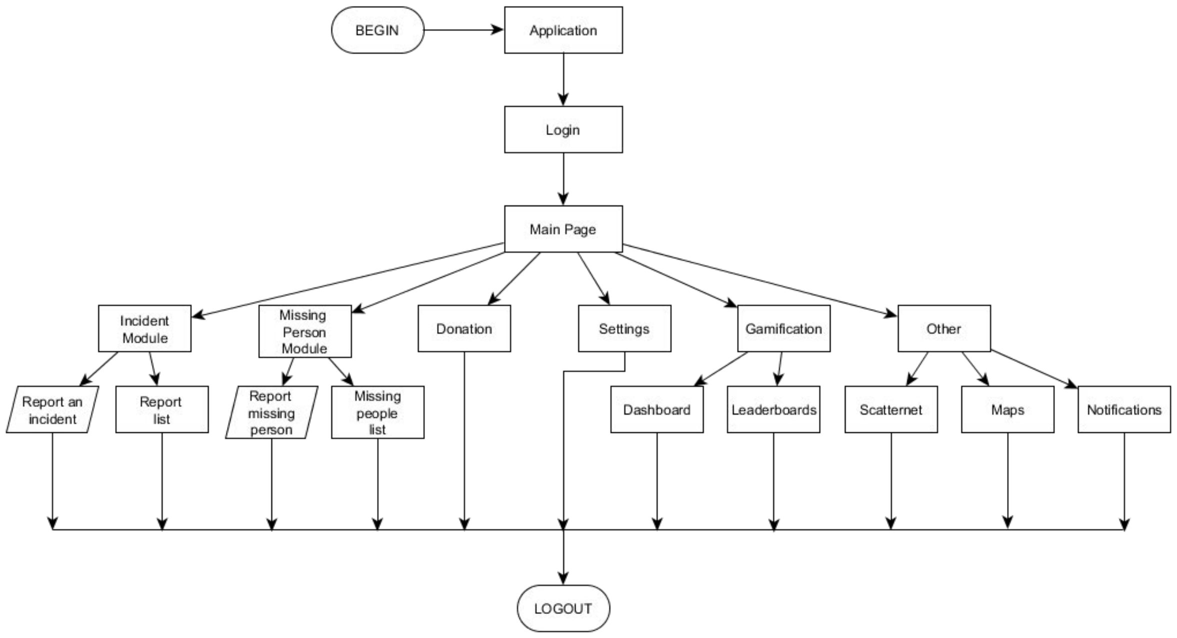


Fig 3.5: Level 1 DFD for User End

3.4 Functionality of the DMUCS System

The main functionalities of the system are as follows:

- Multiple users can report simultaneously.
- Allows user to communicate and solve situations through scatternet.
- Users and admin have access to consolidated report.
- Heat maps give a better estimation of the damage.

3.5 Activity Diagram

The activity diagram for the web end is shown below:

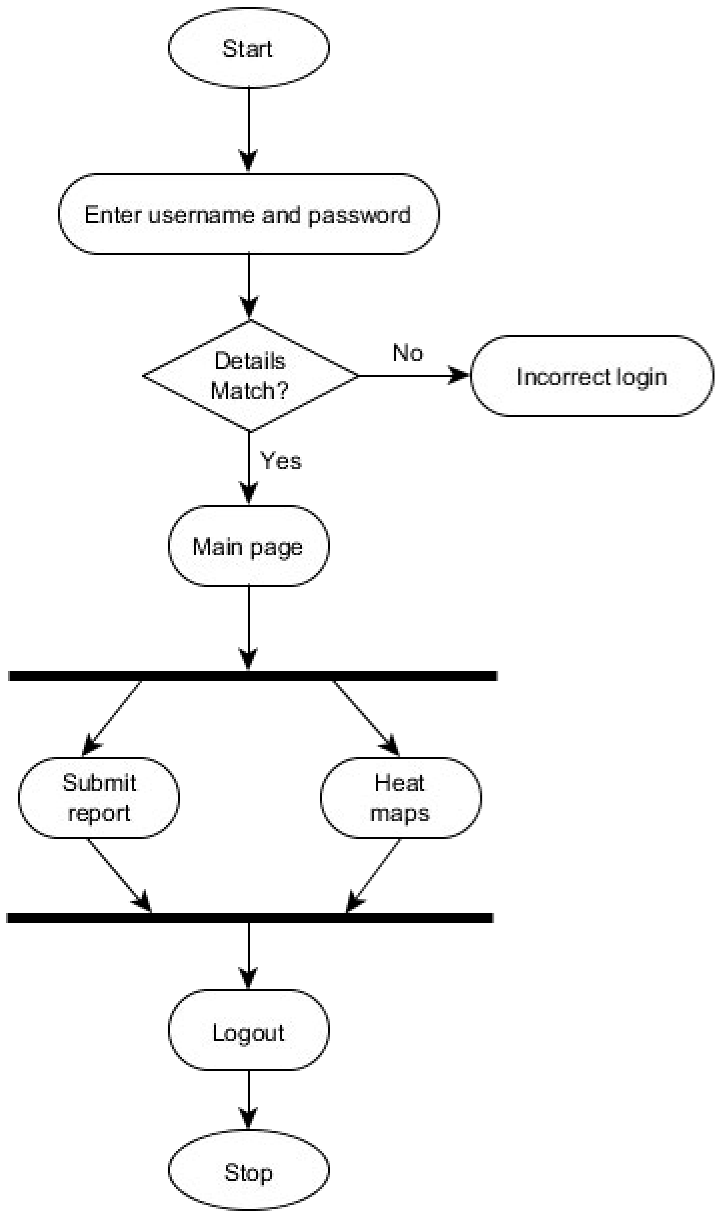


Fig 3.6: Activity Diagram for Web End

The activity diagram shows that the admin begins by logging into the system. The systems checks whether the details entered are consistent with the values entered into the database. If the details are correct, he/she is redirected to the main page, else incorrect login message is displayed. From the main page the admin has options to either submit a report or view the heat maps.

The activity diagram for the user end is shown below:

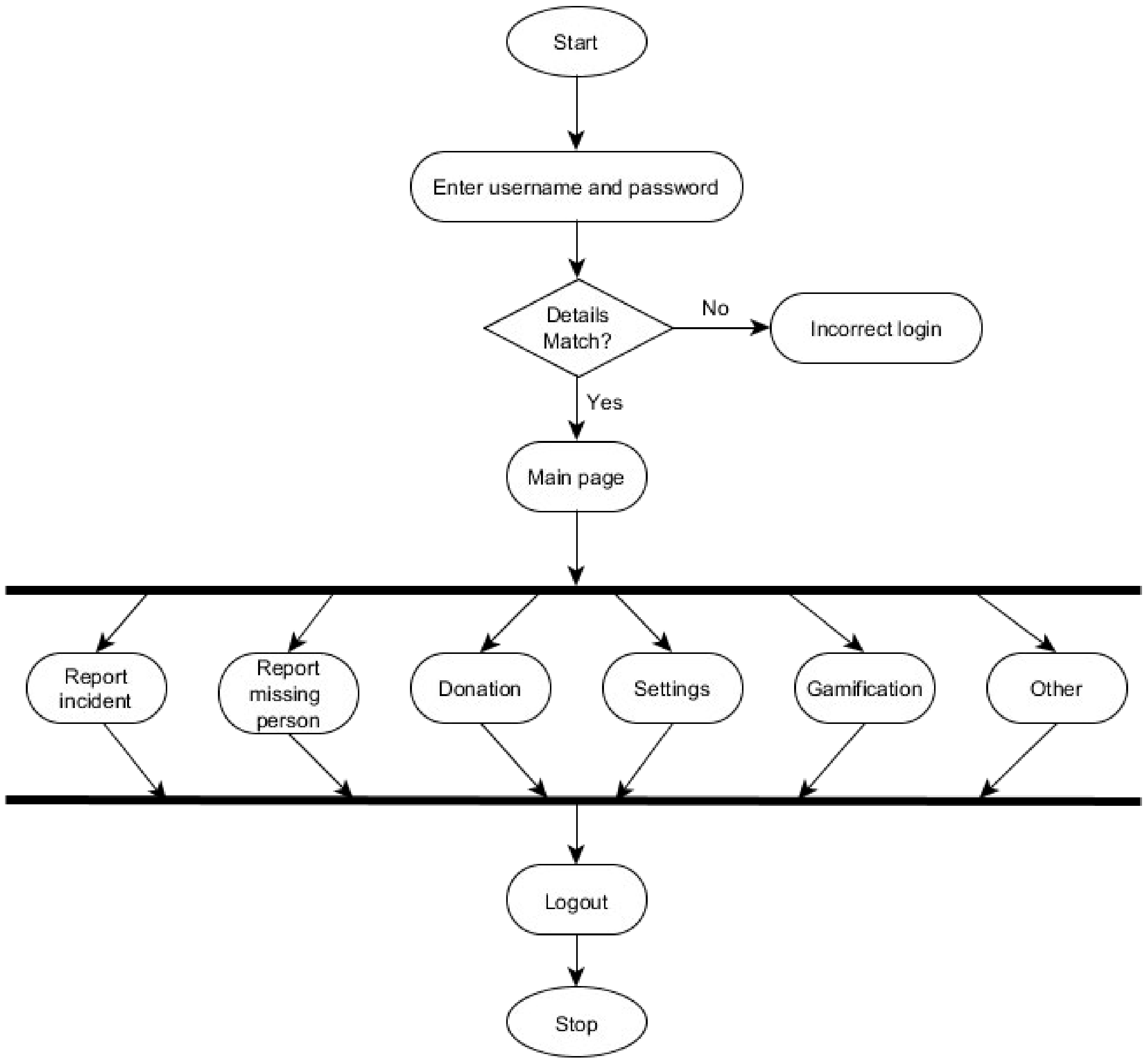


Fig 3.7: Activity Diagram for User End

This activity diagram shows that the user is given a login page. If the details entered are correct he/she is redirected to the main page. Here the user has many options such as report an incident or report a missing person. The user ends this interaction by logging out of the system.

3.6 Sequence Diagram

The sequence diagram of the system is shown below:

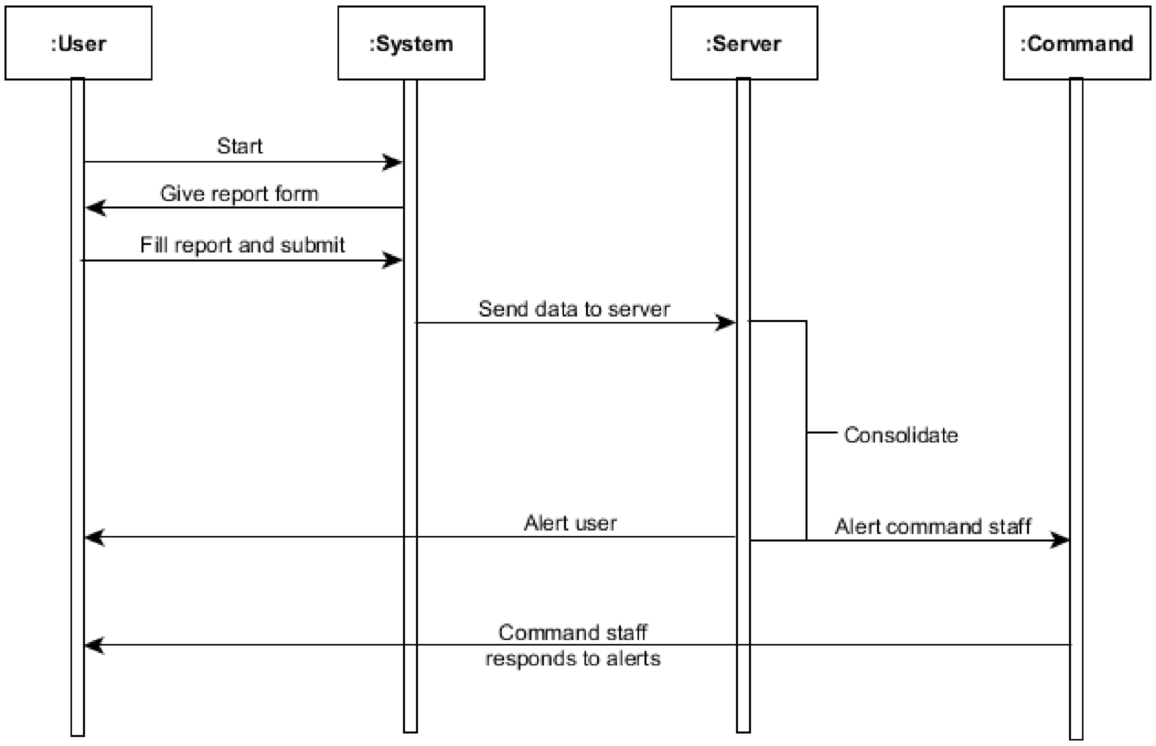


Fig 3.8: Sequence Diagram

3.7 Use Case Diagram

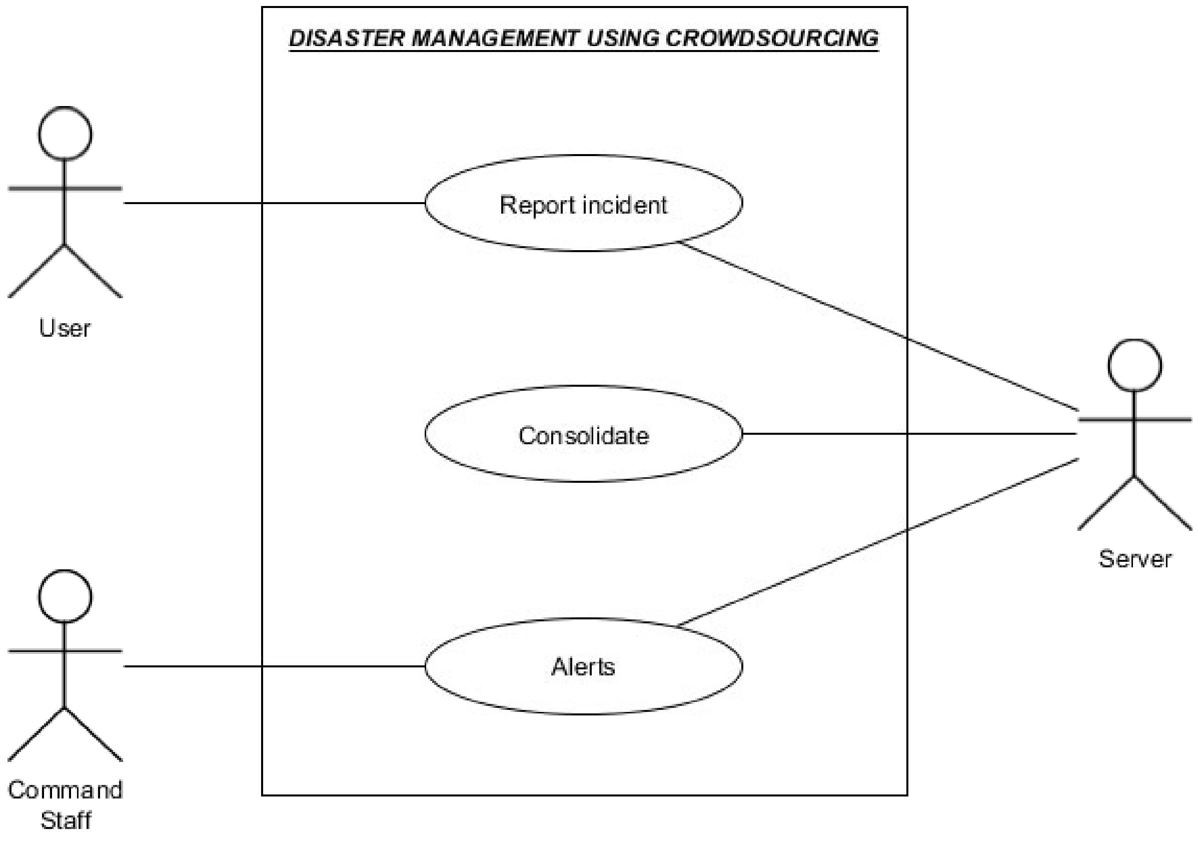


Fig 3.9: Use Case Diagram

The use case diagram mainly consists of actors, functionalities and relationships. This is described through the above diagram. Here there are three Actors – User, Command Staff and Server. The various functionalities of these actors are shown within the box and the relationships between the actors and the functionalities are shown by the lines which connect them.

3.8 Summary

This chapter describes the High Level Design of the Disaster Management Using Crowdsourcing. It describes the main design considerations of the system which includes describing briefly the assumptions and dependencies and the goals and constraints of the system. It also describes the system architecture which gives an idea as to the working of the system. It then shows the data flow diagrams of the system. After this, we discussed the functionality of the system and described the activity diagram, the use case diagram and the sequence diagram of the system.

CHAPTER 4
DETAILED DESIGN

CHAPTER 4

DETAILED DESIGN

Design is concerned with identifying software components specifying relationships among components. It specifies the software structure and provides the blueprint the blue print for the document phase.

Modularity is one of the desirable properties of a large system. It implies that the system is divided into several parts. In such a manner, the interaction between parts is clearly specified.

Design will explain software components in detail. This will help in the implementation of the system. Moreover, this will guide the future changes in the system to specify the future requirements.

4.1 Modules

The following are the various modules in this project:

- Report Incident
- View Incidents
- Missing Person Finder
- Donation and Crowd Funding
- Heat maps
- Gamification
- API Support

4.2 Module 1: Report Incident

Incident reporting module is designed to quickly obtain information about the incidents and reports in the vicinity of the user.

This activity consists of an AutoCompleteTextView which gives the list of incidents that a user can choose from.

```
AutoCompleteTextView incident;

incident = (AutoCompleteTextView) findViewById(R.id.autocomplete_incident);

String[] incidents = getResources().getStringArray(R.array.incidents);

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, incidents);

incident.setAdapter(adapter);
```

This activity also consists of radio buttons used to obtain various information from the user such as the number of casualties. The following is the code written for this purpose.

```
SegmentedRadioGroup segmentText_cas;

segmentText_cas = (SegmentedRadioGroup) findViewById(R.id.id_casuality);

@Override

public void onCheckedChanged(RadioGroup group, int checkedId) {

if (group == segmentText_cas) {

        if (checkedId == R.id.b1) {

                no_casualty = "1";

        } else if (checkedId == R.id.b2) {
                no_casualty = "50";

        } else if (checkedId == R.id.b3) {

                no_casualty = "200";

        } else if (checkedId == R.id.b4) {

                no_casualty = "-1";

        }

}
```

This activity also has a map which allows the user to accurately specify the location where the disaster has occurred. Google Maps API v2 has been used for this purpose. The maps automatically zooms to a location near your current location using this code

```
LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

Criteria criteria = new Criteria();

String provider = locationManager.getBestProvider(criteria, true);

Location lastKnownLocation = locationManager.getLastKnownLocation(provider);

currentLocation = new LatLng(lastKnownLocation.getLatitude(),
lastKnownLocation.getLongitude());

map = ((MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap();

map.setOnMapClickListener(this);

map.animateCamera(CameraUpdateFactory.newLatLngZoom(currentLocation, 14), 2000,
null);
```

To obtain the point where the user clicks onMapClick function is used as follows:

```
@Override

public void onMapClick(LatLng POINT) {

map.clear();

lng = String.valueOf(POINT.longitude);

lat = String.valueOf(POINT.latitude);

Marker incident = map.addMarker (new MarkerOptions ().position(POINT)
.icon(BitmapDescriptorFactory.fromResource(R.drawable.incidentmarker)));

map.animateCamera (CameraUpdateFactory.newLatLngZoom (POINT, 15), 2000, null);

}
```

4.3 Module 2: List Incidents

This module in an activity in android which lists all incidents near your current location. The radius within which incidents are displayed is specified by the user in a TextBox.

```
List<NameValuePair> params = new ArrayList<NameValuePair>();

JSONObject json = jParser.makeHttpRequest(url_all_reports, "GET", params);

String uid = c.getString("uid");

String incident = c.getString("incident");

String lat = c.getString("lat");

String lng = c.getString("lng");

if (Integer.parseInt(loc) <= Integer.parseInt(within)) {

        HashMap<String, String> map = new HashMap<String, String>();
        map.put("uid", uid);
        map.put("incident", incident);

        map.put("loc", loc);

        reportsList.add(map);

}

adapter = new SimpleAdapter ( ReportList.this, reportsList, R.layout.report_view, new
String[] { "uid","incident", "loc", "report_time", "comments" }, new int[] { R.id.uid,
R.id.incident,   R.id.location, R.id.time, R.id.comments });

setListAdapter(adapter);
```

One of the things that this activity does is check if an incident is within the radius specified by the user. To do this, the distance of an incident from the user's current location should be calculated. The following Java function is used to find this distance:

```
private String distance(String lat1s, String lon1s, double lat2,double lon2, String unit) {

                double lat1 = Double.parseDouble(lat1s);

                double lon1 = Double.parseDouble(lon1s);

                double theta = lon1 - lon2;

                double dist = Math.sin(deg2rad(lat1)) * Math.sin(deg2rad(lat2))

                                + Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2))

                                * Math.cos(deg2rad(theta));

                dist = Math.acos(dist);

                dist = rad2deg(dist);

                dist = dist * 60 * 1.1515;

                if (unit == "K") {
                        dist = dist * 1.609344;
                } else if (unit == "N") {
                        dist = dist * 0.8684;
                }
                return ((int) dist + "");
        }
        private double deg2rad(double deg) {
                return (deg * Math.PI / 180.0);
        }
        private double rad2deg(double rad) {
                return (rad * 180.0 / Math.PI);
        }
}
```

4.4 Module 3: Missing Person Finder

Missing Person finder helps people reconnect with friends and loved ones in the aftermath of natural and humanitarian disasters.

A user whose friend is missing, reports him by initiating an Android activity which helps him do so. Here he fills in various information pertaining to the person who is identified as missing. Other users in the vicinity can see who is missing in their locality.

When a user finds that person all that he has to do is click a ‘person found’ and an SMS is sent to the user who reported him as missing.

```
private void sendSMS(String to) {

                SmsManager smsManager = SmsManager.getDefault();

                smsManager.sendTextMessage (to, null, "The person you reported as missing
has been found. Please contact "+found_by+" or open the app for more details", null, null);

}
```

This activity also records the location where the person in question went missing and where he has been found.

4.5 Module 4: Donation and Crowd Funding

Crowd funding is the collection of resources from backers—the "crowd"—to fund an initiative. Such a resource can be in form of money or other items such as old clothes, rice etc.

Within the donate activity, two fragments have been created, one used to donate money and the other, to donate items such as old clothes.

The first, donate money fragment, reads a QR code to determine to whom the money has to be donated.

```
@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == 0) {

                if (resultCode == -1) {

                        String contents = data.getStringExtra("SCAN_RESULT");

                        id.setText(contents);

                } else {
                        Toast.makeText(getActivity(), "QR code not detected\nEnter
UniqueID manually",
                        Toast.LENGTH_SHORT).show();
                }
        }
}
```

Next, the amount to be donated is chosen by the user and upon pressing the submit button, the money can be transferred to the respective NGO.

The Relief Fragment helps locate the nearest 'dropbox' and using this application, he can navigate to that location and donate old clothes or other items to the needy.

```
Button gmaps = (Button) findViewById(R.id.r_map);
gmaps.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0) {
        String uri = String.format(Locale.ENGLISH,
"http://maps.google.com/maps?saddr="+c_lat+","+c_lng+"&daddr="+lat+","+lng+"");
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(uri));
        startActivity(intent);
}});
```

4.6 Module 5: Heatmaps

The heat map module is used to give a graphical representation of areas affected by a disaster. The areas most affected by a disaster is marked red and the areas which are less affected are marked green.

This web module gives a bird’s eye view of the areas that have been affected by a disaster, the most.

```
map = new google.maps.Map(document.getElementById('map_canvas'), mapOptions);

makeRequest('get.php', function(data) {

     var data = JSON.parse(data.responseText);

    for (var i = 0; i < data.length; i++) {

        if(data[i].incident == 'fire') {

            Data[j] = new
        google.maps.LatLng(parseFloat(data[i].lat),parseFloat(data[i].lng));

            j++;

        }

    }

});

var pointArray = new google.maps.MVCArray(Data);

heatmap = new google.maps.visualization.HeatmapLayer({data: pointArray

  });

heatmap.setMap(map);
```

4.6 Module 5: Gamification module

Gamification is the use of game thinking and game mechanics in non-game contexts to engage users to use this crowd sourcing service that our application provides.

For each report, donation or other activities that a user performs, he gets points and unlocks achievements. These points are stored Swarm servers. Further, Swarm APIs are used to implement leaderboards and achievements.

Leaderboards are a great way to get users to compete with their friends and the rest of the community.

It is of need to initialize Swarm at every entry point into your app, typically in the main activity's onCreate() method.

```
Swarm.init(this, SWARM_APP_ID, "SWARM_APP_KEY");
```

The following method is called to submit a score to the leaderboard where LEADERBOARD_ID is the Leaderboard ID number shown in the admin panel, and points is the score being submitted.

```
SwarmLeaderboard.submitScore(LEADERBOARD_ID, points);
```

Leaderboards can be seen using the following

```
Swarm.showLeaderboards();
```

Achievements are an easy way to entice users to reach for interesting goals. The following method is called to unlock an achievement where ACEHIEVEMENT_ID is the Achievement ID shown in the admin panel.

```
if (objectiveCompleted) {

   SwarmAchievement.unlock(ACHIEVEMENT_ID);

}
```

4.7 Module 6: API Support

In computer programming, an application programming interface (API) specifies how some software components should interact with each other.

Our APIs gives a third party developers access to the databases. This enables him to develop his own applications using our data.

To access the database, a third party developer needs to call a PHP script which returns the data to him in JSON format.

JavaScript Object Notation (JSON), is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. This format is used because the data represented in JSON format is easily parable by a multitude of programming and scripting languages. An example of the data returned to the developer is shown below.

The following is a small code segment that encodes the data in database in JSON format.

```
$result = mysql_query("SELECT *FROM report") or die(mysql_error());

if (mysql_num_rows($result) > 0) {

        $response["reports"] = array();

        while ($row = mysql_fetch_array($result)) {

                        $rep            = array();

                        $rep["uid"]        = $row["uid"];

                        $rep["pid"]        = $row["phone"];  //phoneID

                        $rep["incident"]     = $row["incident"];

                        $rep["lat"]        = $row["lat"];

                        $rep["lng"]        = $row["lng"];
```

```
                $rep["report_time"]    = $row["report_time"];

                $rep["modified_time"]  = $row["modified_time"];

                // push single product into final response array

                array_push($response["reports"], $rep);

        }

        $response["success"] = 1;

        echo json_encode($response);

}
```

4.8 Summary

In this chapter, six most important modules in the application that is developed has been explained in depth. The next chapter gives details of the implementation.

CHAPTER 5
IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

Implementation of any software is always preceded by important decisions regarding selection of platform, the language used, etc. These decisions are often influenced by several factors such as the real environment in which the system works, the speed that is required, the security concerns, other implementation specific details etc.

Eclipse Integrated Development Environment (IDE) is used for the development of the application. Java is used as a programming language. The strengths of java as a programming tool for the coders and its facilities offered to users to minimize the logic flow by using standard libraries have been discussed in this chapter.

Android Software Development Kit (SDK) is used for the development of android application. Android SDK is license free SDK which is provided by Google Company. The Android SDK is added as a plug-in to Eclipse IDE which allows development of the native application for android.

5.1 Programming Language Selection

5.1.1 Java

Java is a programming language used in this project, originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. Java derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications.

Java consists of following features:

- **Productivity**: The top reason Java has become popular is because of the increased productivity of Java programmers.
- **GUI**: Java a good, portable library for Graphical User Interface. Most programming languages supply only a text mode interface.
- **Internet**: Java lets you easily use internet. Most languages were designed before the internet was born.
- **Portability**: Java programs can run on many different machines (Intel, Sparc, PowerPC) and many different operating systems (Windows, UNIX, Macintosh). You can move a C program if it is written very carefully, but it is usually difficult or impossible. In contrast, it is easy to move most Java programs.
- **Reliability**: Java programs are more reliable because Java doesn't have very dangerous things like C, C++ pointer arithmetic. Java also checks array bounds and other error-prone operations. Memory management is much safer because Java does automatic garbage collection.
- **Libraries**: Java has a very large number of packages which extend the language. Therefore it is unnecessary to call the operating system directly.
- **OOP**: Object-Oriented programming features (inheritance, encapsulation and polymorphism) make many programs, especially large programs, easier to write.
- **Large Programs**: Java supports large programming projects with object-oriented programming, packages and components.

5.1.2 Android

Android is a combination of three components:

- A free, open source operating system for mobile devices.
- An open-source development platform for creating mobile applications.
- Devices, particularly mobile phones, that run the Android operating system and the applications created for it.

More specifically, Android is made up of several necessary and dependent parts, including the following:

- A hardware reference design that describes the capabilities required for a mobile device to support the software stack.
- A Linux operating system kernel that provides low-level interface with the hardware, memory management and process control, all optimized for mobile devices.
- Open-source libraries for application development, including SQLite, Web Kit, OpenGL and a media manager.
- A run time used to execute and host Android applications, and the core libraries that provide Android specific functionality. The run time is designed to be small and efficient for use on mobile devices.
- An application framework that agnostically exposes system services to the application layer, including the windows manager and the location manager, content providers, telephony and sensors.
- A user interface framework used to host and launch applications.
- Preinstalled applications shipped as part of the stack.
- A software development kit used to create applications, including tools, plug-ins and documentation.

What really makes android compelling is its open philosophy, which ensures that you can fix any deficiencies in user interface or native application design by writing an extension or replacement. Android Native Applications provide you, as a developer, with the opportunity to create mobile phone interfaces and applications designed to look, feel and function exactly as you imagine them.

5.1.3 Android SDK features

The true appeal of Android as a development environment lies in the APIs it provides. As an application-neutral platform, Android gives you the opportunity to create applications that are as much a part of the phone as anything provided out of the box. The following list highlights some of the most noteworthy Android features:

- No licensing, distribution or development fees or release approval processes.
- Wi-Fi hardware access.
- GSM, EDGE and 3G networks for telephony or data transfer, enabling you to make or receive calls or SMS messages, or to send and retrieve data across mobile networks.
- Comprehensive APIs for location-based services such as GPS.
- Full multimedia hardware control, including playback and recording with the camera and microphone.
- APIs for using sensor hardware, including accelerometers and the compass.
- Libraries for using Bluetooth for peer-to-peer data transfer.
- IPC message passing.
- Shared data stores.
- Background applications and processes.
- Home-screen widgets, Live folders and Live wallpapers.
- The ability to integrate application search results into the system search.
- Full support for applications that integrate map controls as part of their user interface.
- Mobile-optimized hardware-accelerated graphics, including a path-based 2D graphics library and support for 3D graphics using OpenGL ES 2.0
- Media libraries for playing and recording a variety of audio/video or still image formats.
- Localization through a dynamic resource framework.
- An application framework that encourages reuse of application components and the replacement of native applications.
- Access to hardware, including camera, GPS and accelerometer.

Android includes API libraries to simplify development involving the device hardware. The Android SDK includes APIs for location-based hardware (such as GPS), the camera, audio, network connections, Wi-Fi, Bluetooth, accelerometers, touchscreen and power management. Android SDK features Native Google Maps, Geocoding and Location-Based Services Native Map support allow to create a range of map-based applications that leverage the mobility of Android devices. Android allows to create activities that include interactive Google Maps as part of user interface, with full access to maps that can control programmatically and annotate using Android's rich graphics library. Android's location-based services manage technologies like GPS and Google's GSM cell-based location technology to determine the device's current position. These services enforce an abstraction from specific location-detecting technology and let you specify minimum requirements (e.g., accuracy and cost) rather than choosing a particular technology. They also mean that location-based applications will work no matter what technology the host handset supports. To combine maps with locations, Android includes an API for forward and reverse geocoding that lets to find map coordinates for an address, and the address of a map position.

Android's process and memory management is a little unusual. Like Java and .NET, Android uses its own run time and virtual machine to manage application memory. Unlike with either of these other frameworks, Android run time also manages the process lifetimes. Android ensures application responsiveness by stopping and killing processes as necessary to free resources for higher-priority applications.

5.2 Platform Selection

Linux platform is chosen to implement the modules faster using java technologies and speed up to the markets. The reason for choosing Linux operating system is mainly due to user friendliness and familiarity in use and Java version compatible to windows is used for developing the application.

5.3 Code Conventions

Coding standards for Java are important because they lead to greater consistency within code of all developers. Consistency leads to code that is easier to understand, which in turn results in a code, which is easier to develop and maintain. Code that is difficult to understand and maintain runs the risk of being scrapped and rewritten.

5.3.1 Naming Convention

The following conventions are followed for naming:

- Full English descriptors that accurately describe the variable/field/class/interface are used.

  For example, names like firstName, grandTotal, or CorporateCustomer have been used.
- Terminology applicable to the domain have been used.
- Mixed case is used to make names readable.
- Abbreviations are sparingly used, in cases where they are used, it is done so intelligently and have been documented.

  For example, to use a short form for the word "number" like nbr, no or num have been avoided.
- Long names have been avoided (<15 characters is a good tradeoff)
- Names that are similar or differ only in case have been avoided.

5.3.2 Java Comments

The following standard have been used for comments in this project:

Comment Type	Usage	Example
Documentation Starts with /** and ends with */	Used before declarations of interfaces, classes, member functions, and fields to document them.	/** * Customer – a person or * organization */
C style Starts with /* and ends with */	Used to document out lines of code that are no longer applicable. It is helpful in debugging.	/* This code was commented out by Arpith K */
Single line Starts with // and go until the end of the line	Used internally within member functions to document business logic, sections of code, and declarations of temporary variables.	// If the amount is greater // than 10 multiply by 100

5.3.3 Naming member functions

Member functions have been named using a full English description, using mixed case with the first letter of any non-initial word capitalized. The first word of the member function is a verb.

Examples

openTask()

save()

delete()

This results in member functions whose purpose can be determined just by looking at its name.

5.4 Other Programming Utilities

This section discusses about the database software used in the project.

5.4.1 MySQL Cluster

MySQL Cluster is a technology providing shared-nothing clustering and auto-sharding for the MySQL database management system. It is designed to provide high availability and high throughput with low latency, while allowing for near linear scalability. MySQL Cluster is implemented through the NDB or NDBCLUSTER storage engine for MySQL ("NDB" stands for Network Database).

MySQL Cluster provides you with the following benefits:

- **Auto-sharding for Write-scalability**
  MySQL Cluster automatically shards (partitions) tables across nodes, enabling databases to scale horizontally on low cost, commodity hardware while maintaining complete application transparency.

- **99.999% Availability**
  With its distributed, shared-nothing architecture, MySQL Cluster has been designed to deliver 99.999% availability ensuring resilience to failures and the ability to perform scheduled maintenance without downtime.

- **Real-time Performance**
  MySQL Cluster provides real-time response time and throughput meet the needs of the most demanding web, telecommunications and enterprise applications.

- **Multi-site Clusters with Geographical Replication**
  Geographic replication enables multiple clusters to be distributed geographically for disaster recovery and the scalability of global web services.

- **Online Scaling & Schema Upgrades**
  To support continuous operation, MySQL Cluster allows on-line addition of nodes and updates to live database schema to support rapidly evolving and highly dynamic workloads

- **MySQL Cluster Auto-Installer**
  Get MySQL Cluster up and running in minutes! Graphically configure and provision a production-grade cluster, automatically tuned for your workload and environment.

5.5 Summary

This chapter discusses the implementation details of the platform that is used, why Java id chosen for the application coding and the various coding conventions. The testing of the modules developed and the application as a whole is discussed in the next chapter.

CHAPTER 6
TESTING AND RESULT ANALYSIS

CHAPTER 6

TESTING AND RESULT ANALYSIS

System testing is the stage before system implementation where the system is made error tree and all the needed modifications are made. The system was tested with test data and necessary corrections to the system were carried out. All the reports were checked by the admin and approved.

6.1 Software Testing

A test plan is a general document for the entire project, which defines the scope, approach to be taken, and schedule of testing, as well as identifying the test item for the entire testing process, and the personal responsible for the different activities of testing. This document describes the plan for testing, the knowledge management tool.

Major testing activities are:

1. Test units
2. Features to be tested
3. Approach for testing
4. Test deliverables
5. Schedule
6. Personal allocation

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort traditionally occurs after the requirements have been defined and the coding process has been completed having been shown that fixing a bug is less expensive when found earlier in the development process.

Different software development models will focus the test effort at different points in the development process. Newer development models such as Agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed.

6.2 Testing Process

An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing. Software Testing may be compared to tasting the food before being served to the guests. The best purpose of both is to ensure a defect free delivery.

Testing Objectives

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet-undiscovered error.
- A successful test case is one that uncovers an as-yet-undiscovered error.

These objectives imply a dramatic change in the viewpoint. They move opposite to the commonly held view that a successful test is one in which no errors are found. Our objective is to design test cases such that they systematically uncover different classes of errors with minimum time and effort.

Testing is mainly used to determine the quality of the product according to the user needs.

6.2.1 Levels of Testing

It is impractical and often impossible to find all errors in a program. This fundamental problem in testing thus throws open the question as to what would be the strategy that you would adopt to test. Two of the most prevalent strategies include black-box testing and white-box testing.

System Testing

Taking various kinds of data plays a vital role in system testing. After preparing the test data, the system under steady is tested using the test data. While testing errors are again uncovered and corrected by using the above steps and corrections are also noted for future use. The system has been verified and validated by running the test data and live data. First the system is tested with some sample test data that are generated with knowledge of the possible range of values that are required to hold by the field`s the system runs successfully for the given test data and for the live data.

6.2.2 Types of Testing

- **Unit Testing**

  Here we test each module individually and integrate the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is also known as “module testing”. The modules of the system are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module there are some validation checks for the fields.

- **Integration Testing**

  Data can be lost across an interface, one module can have an adverse effect on the other sub functions, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors with an interface. This testing is done with simple data and developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance.

6.3 System Environment

There are three environment in case of java applications.

- Production
- Development
- Test

The production is the actual deployment of the code, whereas the development is the development of the application where the code will be added and modified frequently. The Test environment is to test the application during the development process.

6.4 Unit Testing of the Modules

In the unit testing of the modules each modules of the application is testing under testing environment and deployed on the production servers and application. The unit testing of each module includes testing out all possible inputs and comparing it with the expected output. The ruby provides the command line testing by writing the test cases and executing it. The process is automated and it require the code to be written to test the modules and also expects the expected output to be feed before executing the test cases.

6.4.1 Unit testing of session Module

In the unit testing of the session module, session create and delete action is performed.

The module is tested whether the session is created on the testing server. In the delete action of the module, it is testing whether session is destroyed on the testing server. The expected output would be successful creation of the session and redirecting the application to the startup window.

Table 6.1 Unit Test Case 1

Test Case ID	Unit Test case1
Description	To test if the session is getting created on the server.
Input	The user email id and the password is passed as input.
Expected Output	Based on the operation results, it should throw the error Message if it is unable to create the session or if the session Is created, it should redirect the user to the startup window.

Actual Output	Session was created and user page was redirected.
Remarks	Passed

6.4.2 Unit Testing of Report Module

Table 6.2 Unit Test Case 2

Test Case ID	Unit Test case 2
Description	To test if all reports are successfully being reported on to the server
Input	Incident reports. This includes various details like the type of disaster and location of disaster.
Expected Output	The incident should be successfully reported in the server database.
Actual Output	The database had recorded the incident correctly.
Remarks	Passed

6.4.3 Unit Testing of Report list Module

Table 6.3 Unit Test Case 3

Test Case ID	Unit Test case 3
Description	To test if all reports are successfully being listed on to the android and the desktop devices.
Input	Radius within which incident should be listed

Expected Output	List of all incident within the radius as selected by the user.
Actual Output	List of all incident within the radius as selected by the user.
Remarks	Passed

6.4.4 Unit Testing of Donation Module

Table 6.4 Unit Test Case 4

Test Case ID	Unit Test case 4
Description	This crowd funding module scans a QR code and submits the donation to the respective NGO
Input	QR Code and the amount to be donated
Expected Output	The amount of money donated by the user should be recorded by the database.
Actual Output	The amount of money donated by the user is recorded by the database.
Remarks	Passed

6.5 Integration testing

Once the unit testing was done, we performed the integration testing of modules together to check if there were any errors based on integration of modules or dependencies of modules on each other. Since unit testing rectified the basic errors, integration testing was more effective in ensuring the effective working of all modules of the application.

Table 6.5 Integration testing

Test Case ID	Integration Testing
Description	Testing for information flow and integration between the Each module of the application.
Input	Information flow, data flow diagram
Expected Output	The application module as to interact according to the data Flow diagram.
Actual Output	The application satisfied the flow diagram conditions.
Remarks	Passed

6.6 System Testing

In system testing, all the module are implemented on the system, mobile phones and tested for working.

Table 6.6 System Test Case

Test Case ID	System Test Case
Description	Application runs on the Mobile phone.
Input	Application was opened on mobile phone, android.
Expected Output	The application as to run all features.
Actual Output	The application was running smoothly on all the mobile Platforms passed to test case.
Remarks	Passed

6.7 Result Snapshots

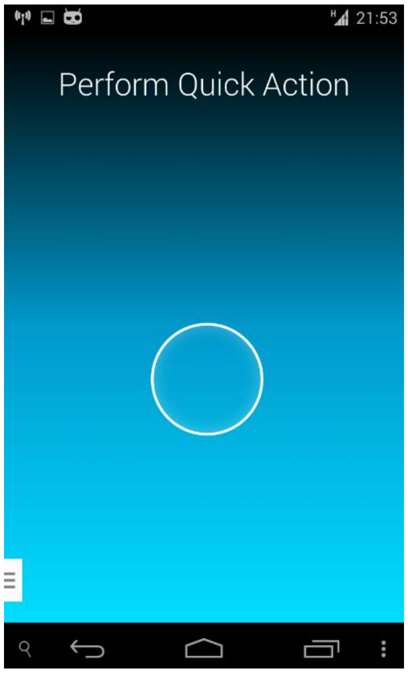


Fig 6.1: Main Activity page of the application

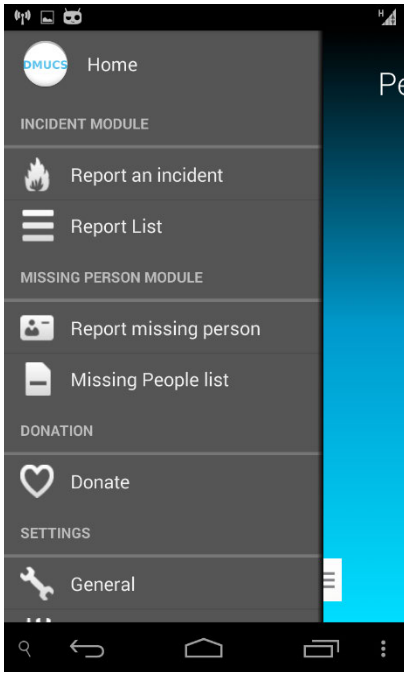
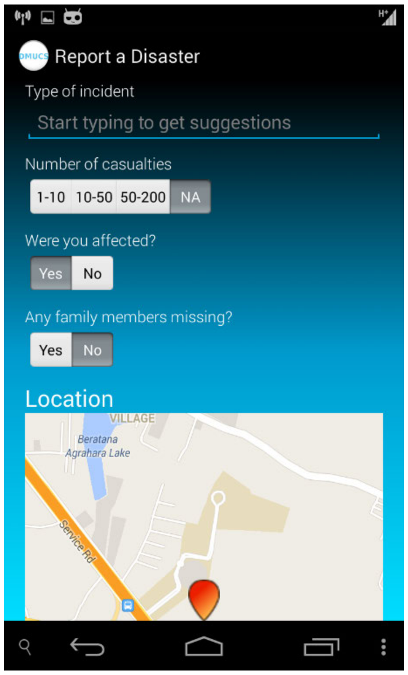


Fig 6.2: Menu Drawer giving access to various available functions


Report a Disaster

Type of incident

Start typing to get suggestions

Number of casualties

1-10 10-50 50-200 [✓ NA]

Were you affected?

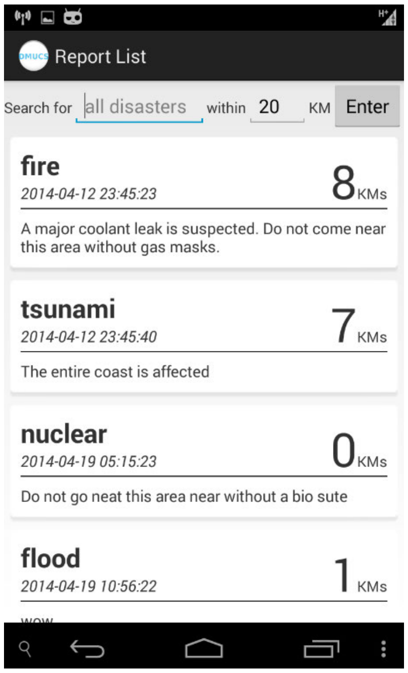
[✓ Yes] No

Any family members missing?

Yes [✓ No]

Location

Fig 6.3: Activity to report an incident


Report List

Search for all disasters within 20 KM Enter

fire
2014-04-12 23:45:23 8 KMs
A major coolant leak is suspected. Do not come near this area without gas masks.

tsunami
2014-04-12 23:45:40 7 KMs
The entire coast is affected

nuclear
2014-04-19 05:15:23 0 KMs
Do not go neat this area near without a bio sute

flood
2014-04-19 10:56:22 1 KMs

Fig 6.4: List of all the incidents

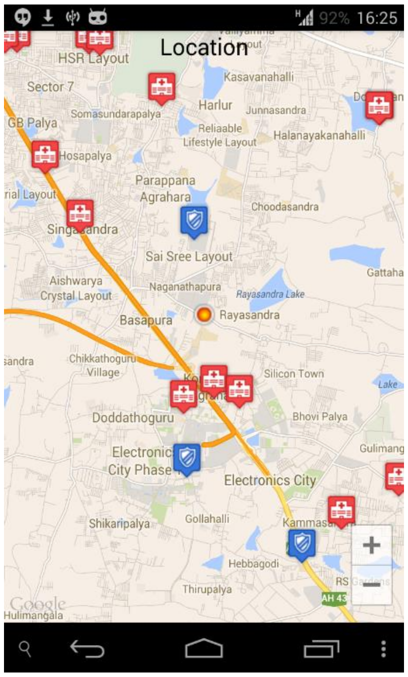
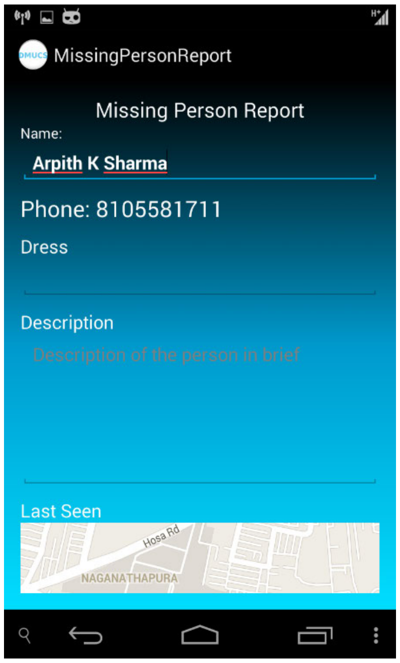


Fig 6.5: Nearby hospitals and police stations


MissingPersonReport

Missing Person Report

Name:

Arpith K Sharma

Phone: 8105581711

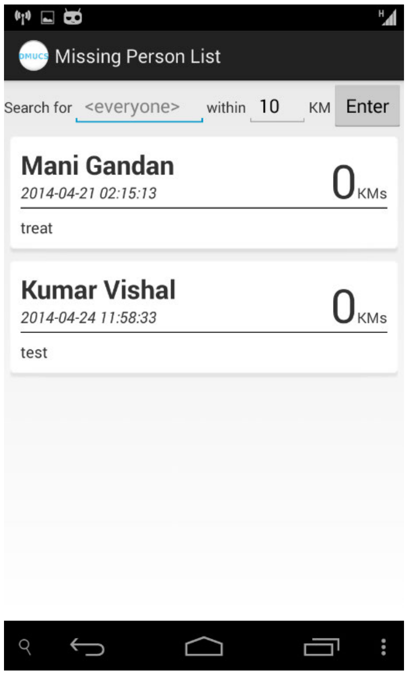
Dress

Description

Description of the person in brief

Last Seen

Fig 6.6: An activity to report a missing person

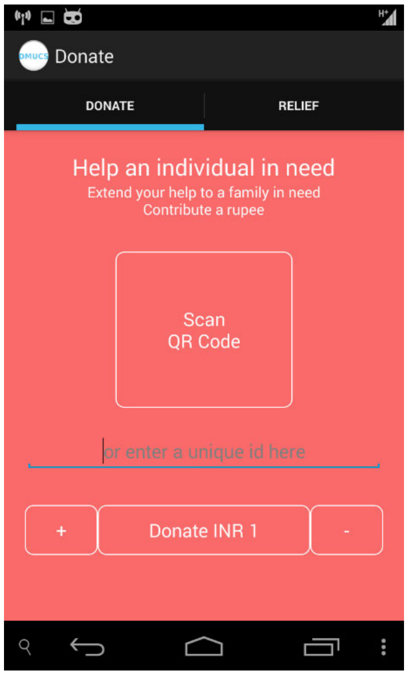

Missing Person List

Search for <everyone> within 10 KM Enter

Mani Gandan
2014-04-21 02:15:13
0 KMs
treat

Kumar Vishal
2014-04-24 11:58:33
0 KMs
test

Fig 6.7: List of all missing people


Donate

DONATE RELIEF

Help an individual in need
Extend your help to a family in need
Contribute a rupee

Scan QR Code

or enter a unique id here

+ Donate INR 1 -

Fig 6.8: A module to donate money to an NGO

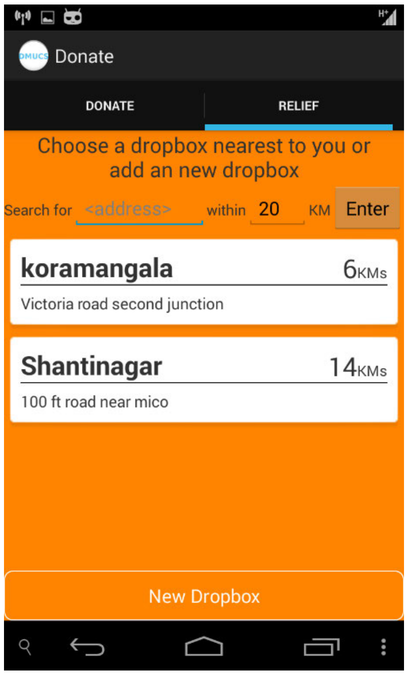

Donate
DONATE
RELIEF
Choose a dropbox nearest to you or add an new dropbox
Search for <address> within 20 KM Enter
koramangala 6KMs
Victoria road second junction
Shantinagar 14KMs
100 ft road near mico
New Dropbox

Fig 6.9: List of all drop boxes near the user's current location

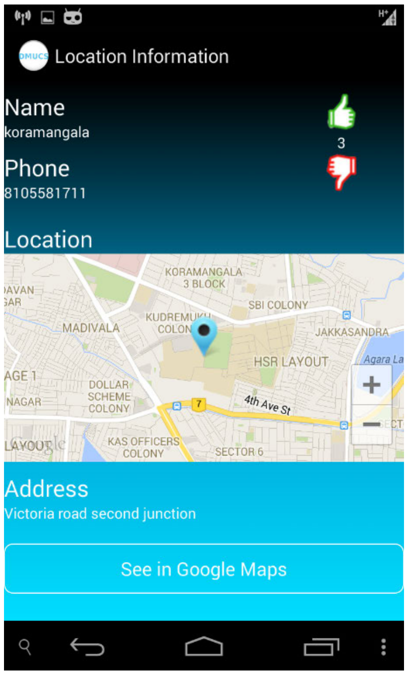


Fig 6.10: More information pertaining to the drop box

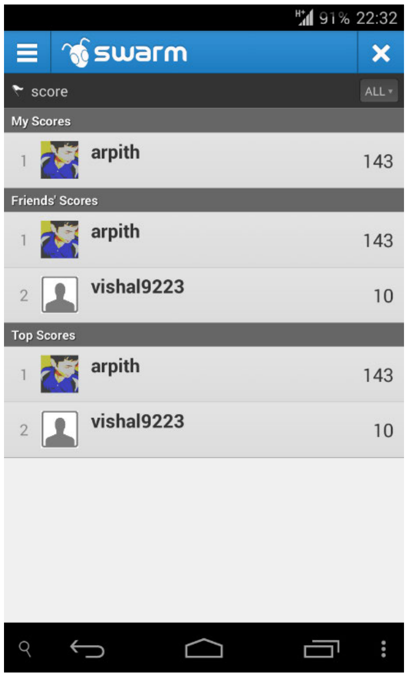
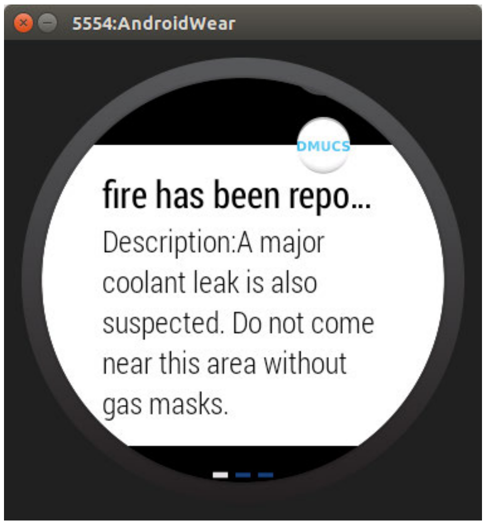
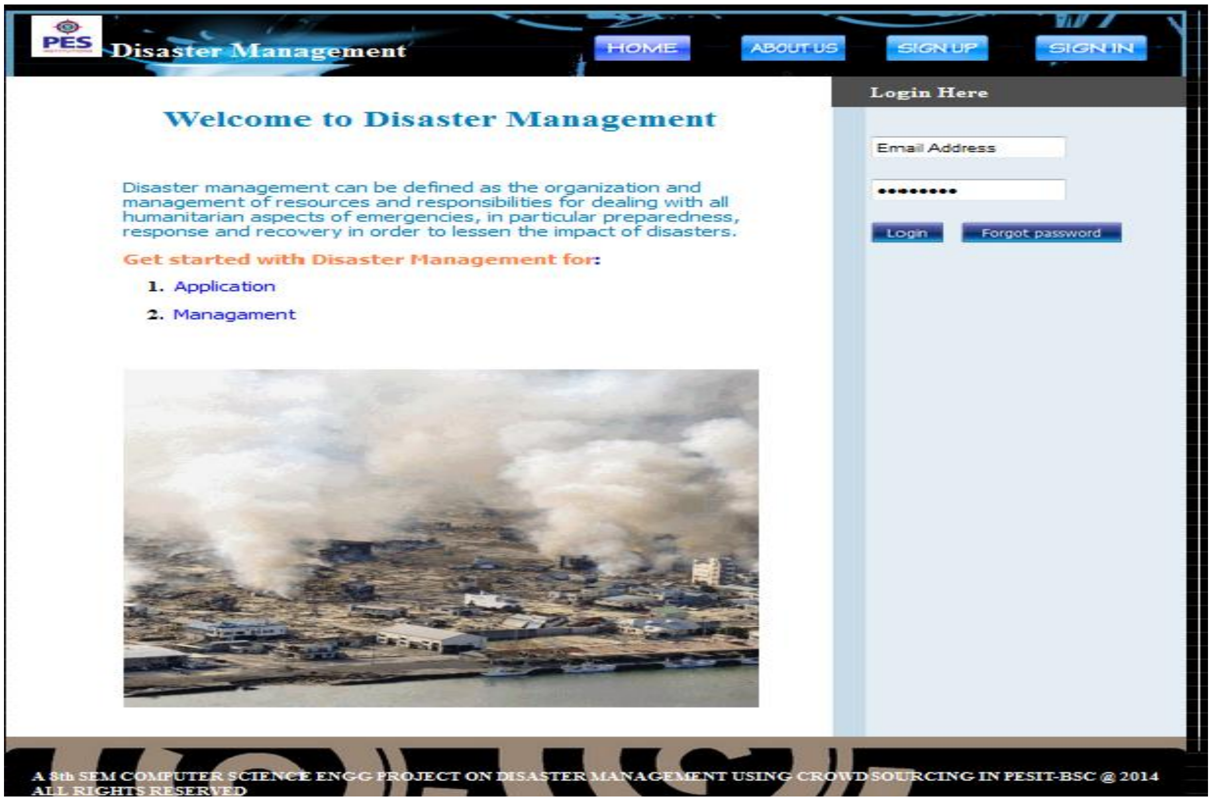


Fig 6.11: Leaderboards



fire has been repo...
Description:A major coolant leak is also suspected. Do not come near this area without gas masks.

Fig 6.12: Notification as seen on a wearable device. Smart watch, in this case.



Welcome to Disaster Management

Disaster management can be defined as the organization and management of resources and responsibilities for dealing with all humanitarian aspects of emergencies, in particular preparedness, response and recovery in order to lessen the impact of disasters.

Get started with Disaster Management for:

1. Application
2. Managament

A 8th SEM COMPUTER SCIENCE ENGG PROJECT ON DISASTER MANAGEMENT USING CROWDSOURCING IN PESIT-BSC @ 2014 ALL RIGHTS RESERVED

Fig 6.13: Login page as seen on a desktop

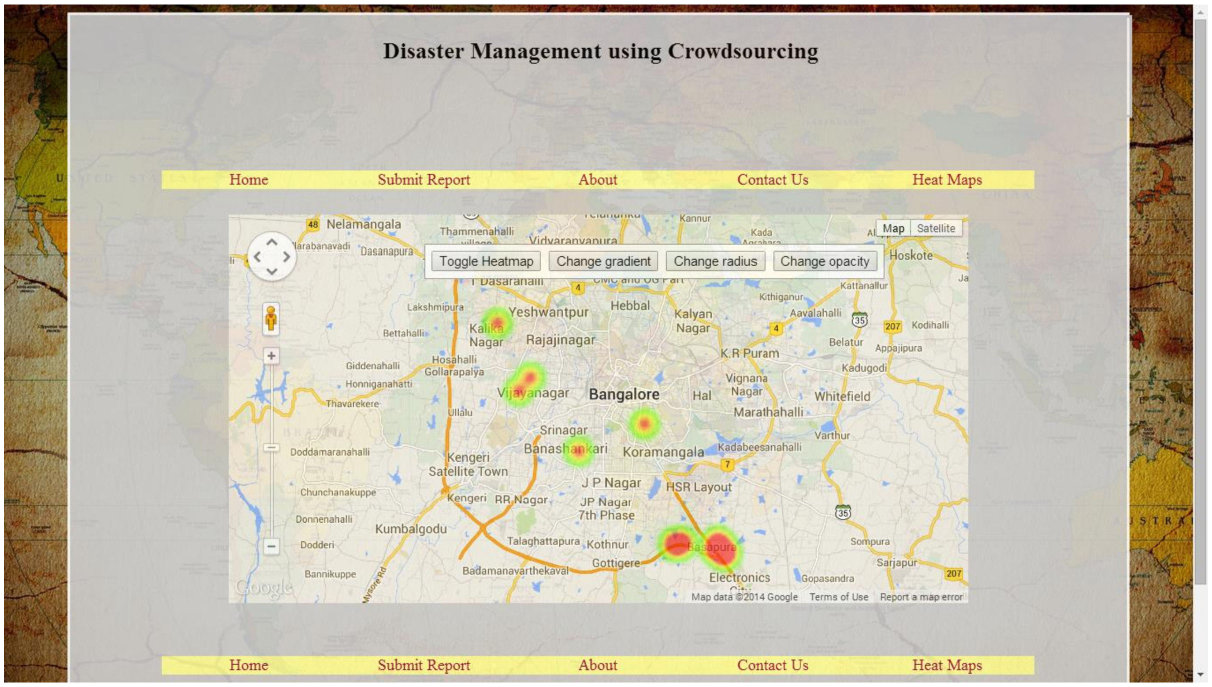


Fig 6.14: Heatmaps, to highlight the affected areas

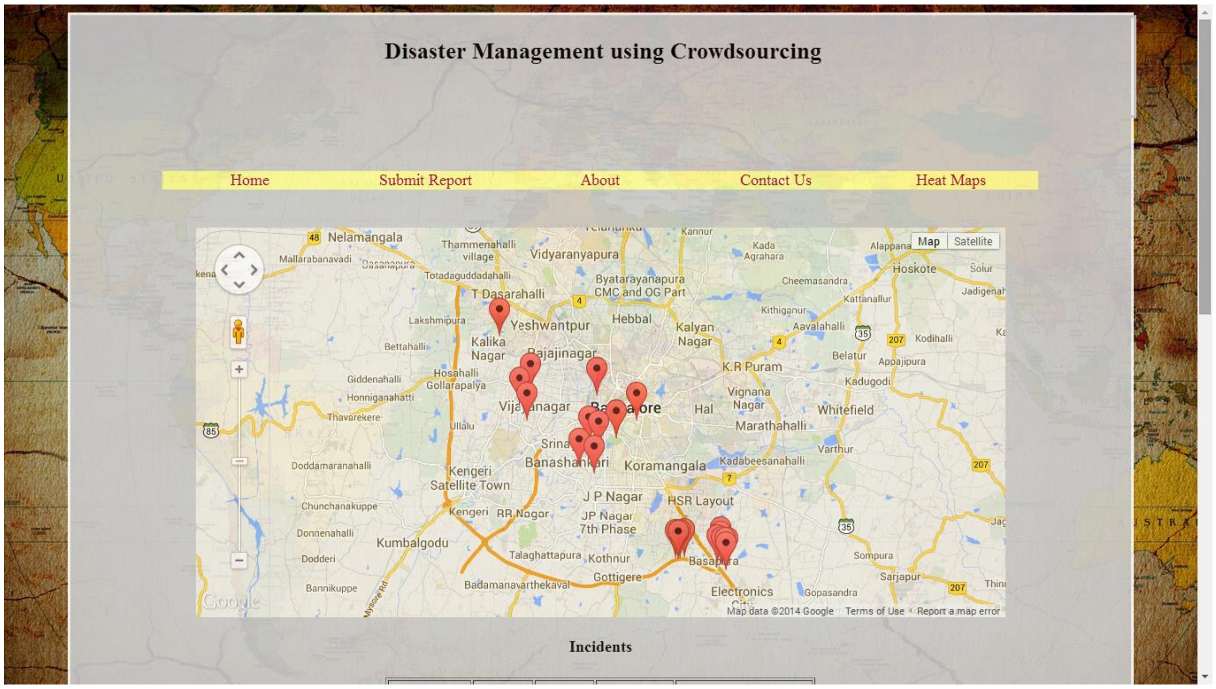


Fig 6.15: List of incidents as seen by the command staff

6.8 Summary

The various modules in the project are tested and integrated together to check for the successful implementation of the problem statement. Further snapshots of the project show its working. It can now be concluded that the system works as intended as a whole.

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENTS

CHAPTER 7

CONCLUSION

7.1 Conclusion

This project presented an approach and a design for a new way of crisis information management, combining mobile applications and the Web. This project shows how the potential information overload of user-generated content can be efficiently reduced in such a way that it produces a set of relevant information to help the command staff direct relief operations to the required regions. In this approach, we used crowdsourcing to enhance, classify, and filter the information at hand.

The result of this approach is a dataset, which enhances the situational picture for the command staff, resulting in an increased situational awareness which in turn helps in disaster management.

7.2 Limitation of the project

The main limitation of this project is that many of its modules relies on the presence of an internet connection. However, appropriate methods have been taken so that the application works in a Local Area Network with the server present in that network. However full functionality such as displaying Google Maps is not possible. It must be noted that in such a case, the current location of the user is assumed to be the location where an incident has occurred.

7.3 Future Enhancements

In very specific types of disasters, such as earthquake or tsunami, there might be a complete loss of communication mediums. Also, it might not be feasible for the relief workers to set up a local area network. A solution to this problem is the use of MANETs. A mobile ad hoc network (MANET) is a self-configuring infrastructureless network of mobile devices connected by wireless ad-hoc means.

Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet. At the time of writing this report, Android officially, does not support ad-hoc networking.

This approach would make it possible to communicate and transmit incident reports even in complete absence of cellular or any other pre-established networks.

REFERENCES

REFERENCES

[1] Strategies for Crowdsourcing for Disaster Situation Information by E. T.-H Chu, Y.-L. Chen, J. W. S. Liu and J. K. Zao

[2] Crisis Information Management in the Web 3.0 Age by Axel Schulz

[3] CrowdHelp: Application for Improved Emergency Response through Crowdsourced Information by Liliya I. Besaleva

[4] CrisisTracker: Crowdsourced social media curation for disaster awareness by J. Rogstadius, M. Vukovic, C. A. Teixeira, V. Kostakos and J. A. Laredo

[5] Crowdsourcing Linked Open Data for Disaster Management by Jens Ortmann, Minu Limbu, Dong Wang, and Tomi Kauppinen

TEAM INFORMATION

TEAM INFORMATION

NAME: Arpith K USN: 1PE10CS018 CONTACT NUMBER: +91 8105581711 EMAIL ID: arpith@live.com ADDRESS: P-3/336 Jalvayu Vihar Sector-21, NOIDA, 201301	NAME: Kumar Vishal USN: 1PE10CS047 CONTACT NUMBER: +91 9448729259 EMAIL ID: vishal.pesit@gmail.com ADDRESS: DS-2, Ittina Sarva Appts Begur Road, Bangalore- 560068
NAME: Vinutha M V USN: 1PE10CS115 CONTACT NUMBER: +91 9449444490 EMAIL ID: vinuthareddy92@gmail.com ADDRESS: #340,muthanallur(village and post), bommasandra via, anekal taluk, Bangalore-560099	NAME: Manigandan C USN: 1PE11CS408 CONTACT NUMBER: +91 9686037730 EMAIL ID: himanichandraa@gmail.com ADDRESS: #43,10th main 4th cross, vinayaka layout, Begur main road Bangalore- 560068