Master Thesis

Submitted within the UNIGIS MSc. programme
at the Department of Geoinformatics - Z_GIS
University of Salzburg, Austria

Under the provisions of the UNIGIS India framework

Hajj Emergency Management System

Emergency Reporting, Webmapping and Response

By

UNIGIS MSc Amro Saber Abouzeid AHMED

GIS_104301

A project report submitted in partial fulfilment of the requirements of the degree of Master of Science (Geographical Information Science & Systems) – MSc (GISc)

Advisor (s):

Dr. Shahnawaz

University of Salzburg, Austria

Riyadh, 10-04-2019

Science Pledge

By my signature below, I certify that my project report is entirely the result of my work; I have cited all sources of information and data I have used in my project report and indicated their origin.

Riyadh 12 April 2019

Place and Date	Signature

Acknowledgements:

I am indebted to my wife for her support and patient with me while working at my studies and also my teachers of UNGIS answering any questions and giving full support especially Dr. Shahnawaz guiding us to success and sharing his ideas, finally thanks for the collages of my work for their advising and support also cannot forget my friend Ahmed Zanati designing application elegant logo.

I used https://www.draw.io to sketching models and charts of work this free site gives the opportunity to present brainstorm for meaningful graphs.

Abstract:

Millions of Muslims coming to Saudi Arabia yearly to do ritual of Hajj at Makkah and moving among Hajj sites, most of the Pilgrims coming from outside have no idea about Hajj sites that may lead to missing many people, family dispersion, or one of the groups at any time during Hajj duties and also cognition is high result of huge number of these visitors at the same time also elderly may needs to emergency help at any time, fast response and location accuracy will help to save many cases and lives, Hajj Management System will cover all these cases to provides all needs normal or emergency for Pilgrims during Hajj duties at these rush places following their emergency call locations using HajiPilot android mobile application, HajiRelief android mobile application and Emergency Calls Map Explorers.

HajiPilot tracking Pilgrims' locations in background and enabling family and friends to request current location to their member locating to assure movements among safe places, Pilgrim may could not use mobile to call or ask for help, HajiPilot lets family or friends to acquire about pilgrim no action needed by pilgrim to location request while HajiPilot will auto-response by the target location as SMS containing on location link, HajiRelief android mobile application used by Safety or medical teams showing them emergency calls on the map and listing calls ordered by call time and allowing to select assigning call for himself, directing to call's location and finally update call's status to be closed after treatment and Pilgrim now safe.

Emergency calls map explorer displaying emergency calls on map, Each call identified by unique number and colored marker relating to its status (New, In process or Closed), Status displayed by different Marker symbology as a feature on the map that will monitoring medical teams activities on the lands representing hot sites that maybe needs to additional medical teams from emergency centers.

Automation of emergency recovery process will decrease dependency on human factor avoiding the probability of deaths occurs resulting from process delay or call missed, Fast emergency response for emergency calls having highly location accuracy, HajiRelief mobile application receiving emergency calls at real-time and pushing to Google maps at the same moment map explorer invoked for appending this new call beside emergency calls including New, In process and Completed helping decision support to redirect efforts and resources to hot regions and recovering at shortest time response and monitoring teams' performance at different Hajj sites.

The primary goal optimizing available resources usability side by side designed mobile application and GPS technologies to improve and develop an emergency strategy to dealing with emergency needs of Pilgrims shortly avoiding missing anyone results of the weakness of emergency plan or rush sites.

Table of Content

List of Tables

List of Figures

Chapter-1: Introduction

A simple definition of emergency management is "a discipline that deals with risk and risk avoidance." Risk represents abroad range of issues and includes an equally diverse set of players. The range of situations that could possibly involve emergency management or the emergency management system is extensive. This supports the premise that emergency management is integral to the security of everyone's daily lives and should be integrated into daily decisions and not just called on during times of disaster.

Emergency management is an essential role of government. The constitution gives the states the responsibility for public health and safety-hence the responsibility for public risks-with the federal government in a secondary role. The federal role is to help when the state, local, or individual entity is overwhelmed, this fundamental philosophy continues to guide the government function of emergency management.

Based on this strong foundation, the validity of emergency management as a government function has never been in question. Entities and organizations fulfilling the emergency management function existed at the state and local levels long before the federal government becomes involved. But as events occurred, political philosophies changed, and as the nation developed, the federal role in emergency management steadily increased. (HADDOW, BULLOCK & COPPOLA 2017)

1.1 Background

1.1.1 What is Hajj?

An annual Muslim pilgrimage - and one of the most eagerly-anticipated religious events of the calendar year - is coming; The Hajj is a five-day pilgrimage that takes place from the 8th to 12th day of Dhu al-Hijjah, the 12th and final month in the Islamic calendar, All adult

Muslims who are healthy enough to make the journey, can afford to do so and able to support their family while they are away, must make the pilgrimage at least once. It is one of the five pillars of Islam, alongside Shahadah, Salat, Zakat, and Sawm.

During Hajj, Pilgrims join processions of hundreds of thousands of people, who simultaneously converge on Mecca for the week of the Hajj, and perform a series of rituals: Each person walks counter-clockwise seven times around the Ka'aba (the cube-shaped building and the direction of prayer for the Muslims), runs back and forth between the hills of Al-Safa and Al-Marwah, drinks from the Zamzam Well, goes to the plains of Mount Arafat to stand in vigil, spends a night in the plain of Muzdalifa, and performs symbolic stoning of the devil by throwing stones at three pillars.

Pilgrims then shave their heads, perform a ritual of animal sacrifice, and celebrate the three-day global festival of Eid al-Adha; Pilgrims can also go to Mecca to perform the rituals at other times of the year. This is sometimes called the "lesser pilgrimage", or Umrah. However, even if they choose to perform the Umrah, they are still obligated to perform the Hajj at some other point in their lifetime if they have the means to do so because Umrah is not a substitute for Hajj.

(LOMBARDO, MEDNIEKS, MEIKE & ROGERS 2009)

1.1.2 **Hajj facilities statistics** (HUDA 2018)

Public president of Hajj and Umra preparing yearly its facilities for housing and controlling Hajj occasion to help pilgrims carrying out Hajj duties smoothly here will browse different facilities provided

1.1.2.1 45,000Tents

Mina, about 3 miles outside Makkah, is known as the hajj tent city. The tents house pilgrims for a few days of the pilgrimage; at other times of the year, it lays bare and abandoned. The tents are neatly arranged in rows and grouped into areas labeled with numbers and colors according to nationality. Pilgrims each have badges with their assigned number and color to help find the way back if they get lost. To prevent fires, the tents are constructed of fiberglass coated with the nonstick coating and are fitted with sprinklers and fire extinguishers. The tents are air-conditioned and carpeted, with a hall of 12 bathroom stalls for every 100 pilgrims.

1.1.2.2 16,280 Officers

Civil defense and emergency personnel are visible throughout the pilgrimage sites. Their job is to direct the flow of pilgrims, assure their safety, and assist those who are lost or in need of medical assistance. In addition to more than 16,000 officers, there were many other support personnel at the 2017 hajj, Cision PR Wire notes, including:

- 51,700 staff members representing more than 20 governmental entities working around the clock
- 6,300 staff members from the General Presidency for the Affairs of the Grand Mosque and the Prophet's Mosque providing cold Zamzam water for pilgrims at prayer sites
- 4,480 staff members providing municipal services for pilgrims, citizens, and residents in Makkah and the Holy sites
- 4,470 staff members from the General Department of Passports to prepare and equip the entry ports to receive pilgrims
- 3,706 scouts providing a wide range of services such as directions to lost pilgrims

1.1.2.3 100 Ambulances

Pilgrim health needs are met at 150 permanent and seasonal health facilities throughout the holy sites, with over 5,000 hospital beds, staffed by over 22,000 doctors, paramedics, nurses, and administrative personnel. Emergency patients are immediately cared for and transported, if needed, by ambulance to one of several nearby hospitals. The Ministry of Health stores 16,000 units of blood to treat patients.

1.1.2.4 6,000 Security Cameras

The security and control command center for Hajj installed nearly 6,000 digital cameras to monitor the movement of pilgrims at the holy sites to ensure their safety at the 2017 hajj, said "Arab News." The high-tech command center for hajj security monitors security cameras throughout the holy sites, including 1,200 at the Grand Mosque itself.

1.1.2.5 Safety and Emergency response

Nearly two millions of people do the same steps at the same time and places that limit of fast response for emergency calls in addition different speaking languages and cultures among all of these Pilgrims leads to missing understanding among each of the resulting difficulty to support help in the right way from Pilgrim to others, so Saudi government during all of past years put policies, strategies, and facilities for serving Pilgrims for safety and fast response for any crises or special care at Hajj places, we have a look for the next section to get clear vision to the number of Pilgrims coming for Hajj during last years.

1.1.2.6 Saudi Red Crescent Authority

SRCA is the Core player in emergency situations at Hajj sites deals with different emergency Pilgrim health needs, yearly hundreds of imbalance cars is ready for this great

event at Makkah and medina for short arrival to emergency call location having high skilled medical stuff teams to serve Pilgrims at his\her location transferring to hospitals fast if needed, recently small imbalance cars used at different sites to reaching Pilgrim locations that achieves great success at rush sites and also have all medical tools that may be needed for treatment.

1.1.3 Pilgrim statistics

While implementing the domestic pilgrim's statistics program, General Authority for Statistics (GaStat) depends on the comprehensive counting method to count all arriving Pilgrims to Makkah. This can be done through the Hajj statistics centers which can be found at all entrances of Makkah. There are also some supporting centers in Jeddah, Taif, and Madina. The counting process starts from 1/12 till 9/12 at 6:00 PM (In Islamic calendar).

Table 1: Hajj statistics 2018-1439 H

Hajj statistics for 2018 – 1439 H		
Total pilgrims	2,371,675	
Domestic pilgrims		
Pilgrims by Nationality	**Number of Pilgrims**	
Saudi	211,736	
Non-Saudi	401,217	
Total	612,953	
Foreign pilgrims		
Males	**Females**	**Total**
931,450	827,272	1,758,722

(General Authority for Statistics 2018)

Controlling and monitoring huge number of these people by old strategies not enough to providing help at moment that has revolution of using mobility technologies and Smartphones integrating with GPS location technology that will decrease response time

and moving guiding direct to target area of emergency call specially at rush areas between mountains at Arafat, Mina, and Muzdalifa region of our context, not monitoring all people at Makkah Just will focus on Pilgrims in the Pilgrim period, During the years we have noticed an increasing number of Pilgrims till 2012-2013 and down sharp at the following years why?

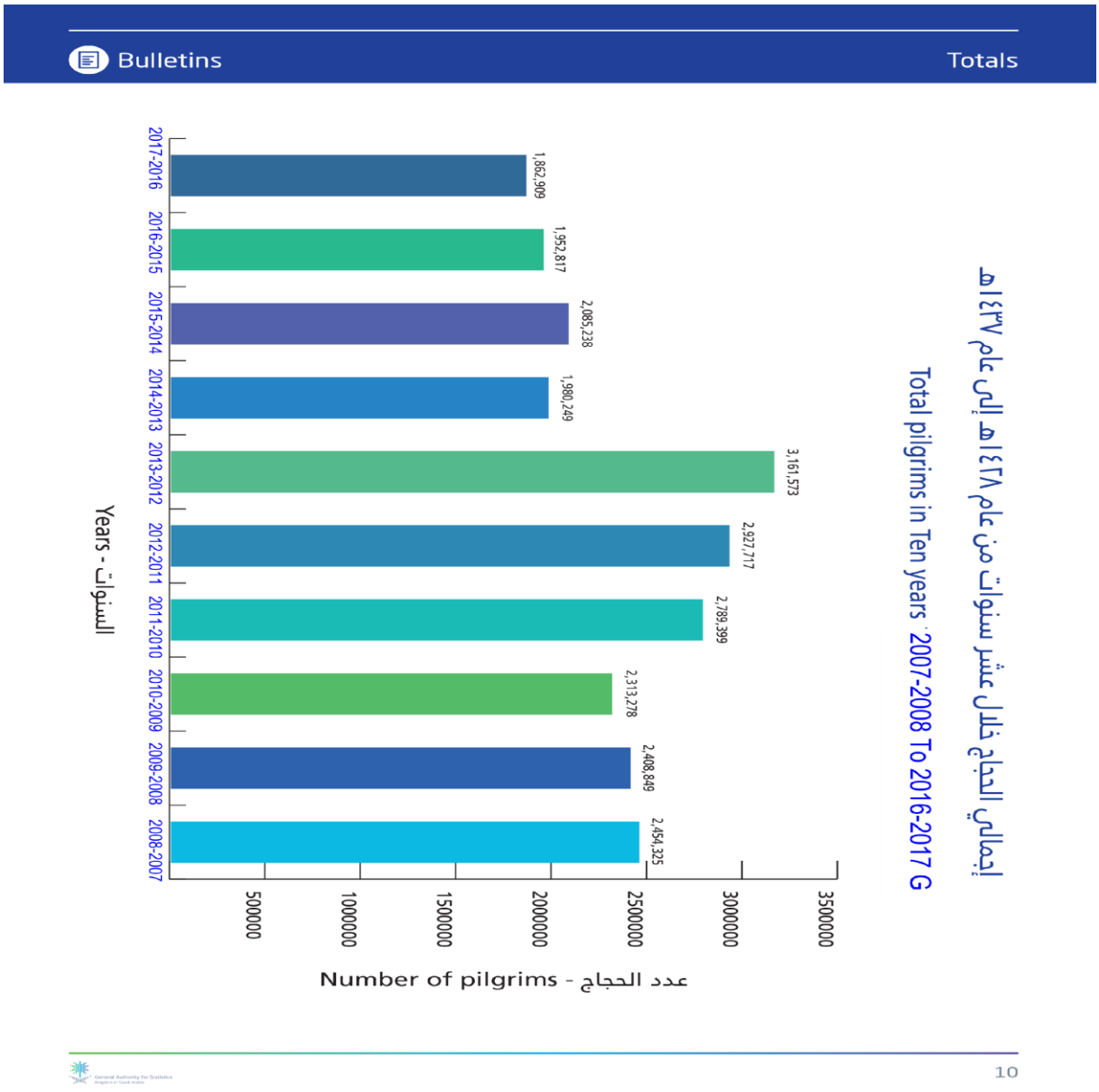

Figure 1: Total Pilgrims From 2006 to 2017

(General Authority for Statistics 2016)

Millions of Muslims in the world hope to come to Hajj, they will not stop but relating to limited resources of Hajj Management president and also government makes limitations for the number of Pilgrims that can be served and caring about this great ritual avoiding increasing risks at emergency cases and also saving their lives.

A critical question now raised is "Can we increase Pilgrims numbers and giving chance for the maximum number of visitors?" Removing hotels and historical buildings around Al masjid Al-haram at Makkah should not affect number of Pilgrims by decreasing so government working on finding alternative staying places and hotels, HajiPilot support Pilgrims and government to keep connected, location monitoring and support fast emergency call by providing Pilgrims the ability to send emergency call and sharing location using auto background messaging service between to families and friends this will avoid missing people and giving fast response for their emergency needs and also using the available resources to recover any crisis, that may encourage the government to increase again number of Visas for Pilgrims.

1.1.4 The changing media world

The internet and social media have radically and irreversibly transformed the communication landscape. We are living through a media revolution that rivals the effects of the earlier tectonic shifts-the invention of the printing press, the telephone, photograph, the radio, and television. The internet has created a "new" news landscape and changed forever the way and speed news is produced and consumed. Former New York Times columnist Frank Rich explained, "We didn't recognize we were up against change as sweeping as the building of transcontinental or the invention of electricity."(Rich 2013), the old communication is a conversation between the many-we are all news producers and consumers, consumers, content creators, and curators and the operating premise in this

new media culture is now, according to Mark Glaser, executing editor of PBS MediaShift, "the audience knows more collectively than the reporter alone." (Glaser 2006)

The emergence of Internet-based social media platforms such as Twitter and Facebook as news providers and the fact that four out of five (80%) of Internet users use their Smartphone to access the Internet and 47% use their tables (Chaffey 2016), means people can access, generate, influence, or share news where they are, any time of day. "In this new multi-platform media environment," according to the Pew Research Centre for the People and the Press," people's relationship to news is becoming portable, personalized, and participatory."(Pew 2010)

Saudi Health Ministry recording patients information and services provided by its distributed emergency centers and hospitals prepared for receiving and dealing with emergency cases especially old people that may be suffering from diseases as diabetes, Heart Diseases, Elderly diseases that needs to fast response and right treatment referencing to personal and patient history about Pilgrims will give the paramedic the chance to decides the suitable treatment and medicine.

Pilgrims' information saved and retrieved by emergency teams to support taking the right decision, medicine and treatment that will recover current case once emergency call received, Referring to the following statistical about provided services and number for each kind of service help government to redesign best strategies in the future to provides medicine amount and assets numbers having the ability to cover Hajj Sites for any emergency situation.

Figure 2: Provided services for Pilgrims 2016-2017

(Ministry of Health-Saudi Arabia 2017)

Interior ministry training and providing thousands of policeman and volunteers yearly that will distributed on the land among Pilgrims on all over Hajj sites for staying arrangements,

monitoring, directing, open roads, providing support and help at different critical situations especially at crowded areas if anything wrong happened, there are many other Saudi organizations integrating providing human resources to help to share at Hajj occasion success yearly.

1.2 Objectives

- Designing a database for storing data that should fit the required business for Hajj management System
- Designing a simple user Interface for usability
- Building WebAPI services for data transactions
- Developing HajiPilot android mobile application for Pilgrims providing the ability to send an emergency call to the system including current location
- Detecting Pilgrim's location by Pilgrim's family or friends and receiving SMS message including location link
- Pilgrim's identification card used and critical situation identifying Pilgrim and staying information
- An Arrangement of medical teams tasks using developed HajiRelief android mobile application on the map
- Following up emergency recovery lifecycle and medical teams' efforts to covering currently active emergency calls on web map explorer displaying all emergency calls identified by its status

1.3 Scope

Pilgrims moving between Hajj sites in ordered sequence starting by Ka'ba, Arafat, Mina, Muzdalifa, Mina end by revisiting Ka'ba gain before leaving at the end of Hajj duties in

specific days so our focus here to study these regions and its boundary during crowed period start by Arafat day and ends by the farewell tawaaf around Ka'ba, relating to time limitation I will developing Hajj Management Systems that will be controlling all emergency processes starting from Pilgrim emergency call, receiving and routing to all teams mobile devices.

Medical teams browsing new currently emergency calls on mobile map at HajiRelief application and assigning the actions to receive some emergency calls to themselves as a mission and moving directly to calls' the locations using Google maps routing, dealing with the situation and transferring to hospital if needed that may require private care, finally close this mission at the moment Pilgrim be safe at site or hospital, restart again to receive next emergency call.

Table 2: Hajj sites distance

	Distance	Mode	Time
Makkah - Mina	5 km/3 miles	Walking	1.5 h
		Taxi/Bus	15 m
		Train	15 m
Mina - Arafat	10 km/6 miles	Walking	2 h
		Taxi/Bus	15 m
		Train	15 m
Arafat - Muzdalifa	7 km/4 miles	Walking	2 h
		Taxi/Bus	15 m
		Train	15 m
Muzdalifa - Mina	2 km/1.2 miles	Walking	30 m
		Taxi/Bus	5 m
		Train	5 m

(SOLUTIONS 2017)

1.4 Area of Focus

Interesting areas are religion areas Madina and Hajj duties at Makkah city these two popular cities that include all Holy places that always visited by Pilgrims coming for Hajj that are interesting to going and staying at Madina city visiting tomb and Masjid Nabawi,

Makkah includes Masjid Haram, Mina, Muzdalifa and Arafat Mountain everyone coming to Hajj duty must pass all of these regions in specific order at specific time referencing to Hajj Duties.

"According to the Shariah, the arrival of pilgrims who are bound or separated from one another by virtue of their inhospitality to Mina on the day of al-Tarawiyyah and overnight in it is on their way to stand up with a sure Sunah of 'Arafah. The area of Mina is 16.8 km^2 It is located between Makkah and Muzdalifah, seven kilometers north-east of Al-Masjid al-Haram. It is a border of the Haram surrounded by mountains on the northern and southern sides. It is inhabited only by the duration of Hajj.

The remaining area of the tents, one of the largest projects implemented by the Government of the Custodian of the Two Holy Mosques to accommodate pilgrims with an estimated area of 2.5 million square meters according to the specifications of achieving more security and safety, with the absorption of 2.6 million pilgrims, and thus be the largest tends city in the world, Each tent was equipped with water sprayers that operated automatically as soon as they felt heat".

(Arab News 2018)

1.4.1 Arafat Location

Where pilgrims should exist on (9 Dul-Hija) Arafat at the same time it is the only holy site that outside Harm area, it is plane containing on Arafat mountain its length 300 m and height 7 m fare 22 km from Makkah and 10 km from Mina, no life or people there just Hajj days. (Saudi Arabian Scouts Association 2018)

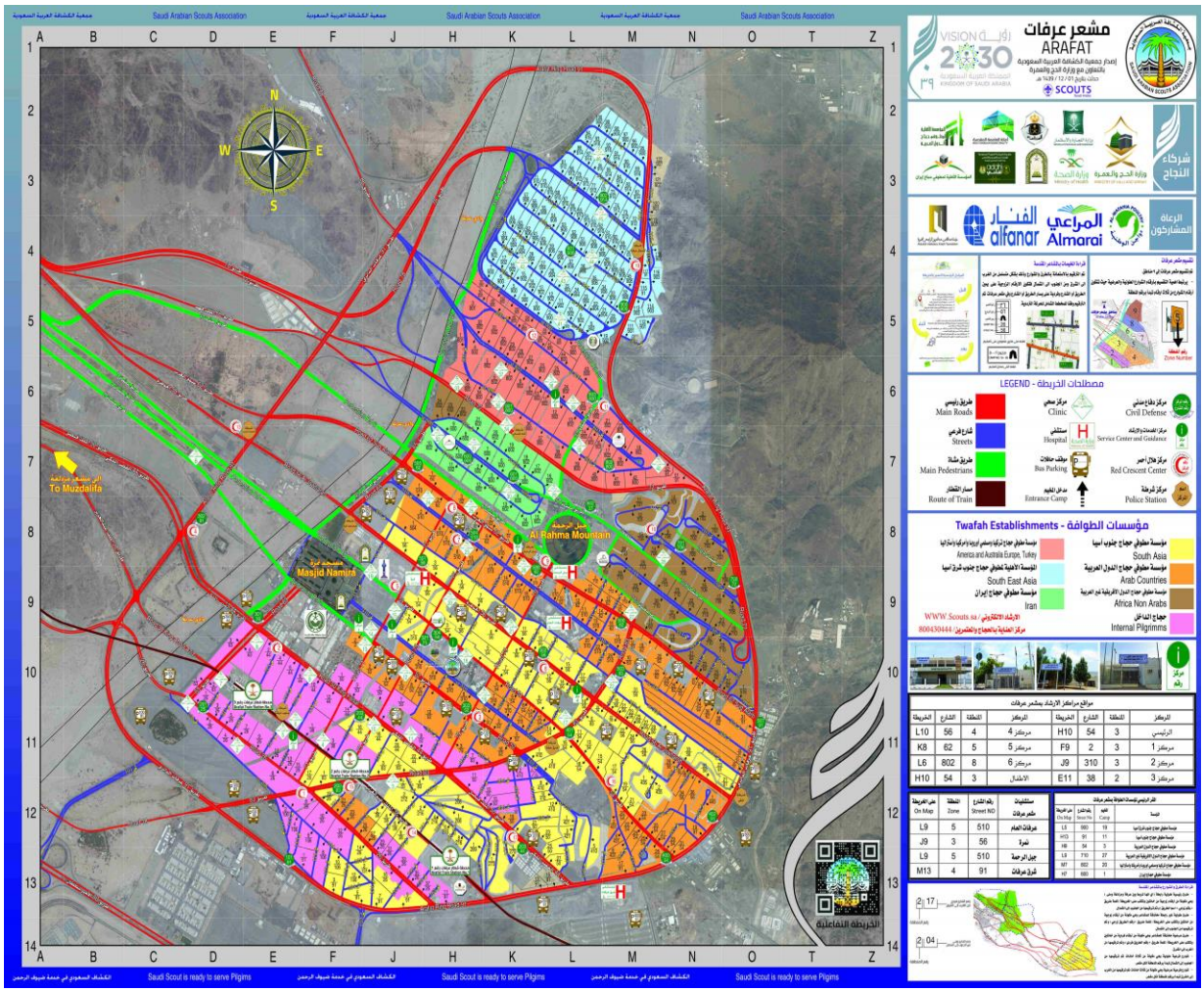

Figure 3: Arafat Location

No Hajj without stepping at Arafat on 9 Dhul-Hijjah that means more than 2 millions of people will meet there at the same time praying for God, it is rea 10.4 km^2.

1.4.2 Mina Location

Makkah - Mina is one of the largest holy sites in the Kingdom of Saudi Arabia. It is considered one of the largest holy sites for the government departments and service agencies working to facilitate the performance of the pilgrims.

(Saudi Arabian Scouts Association 2018)

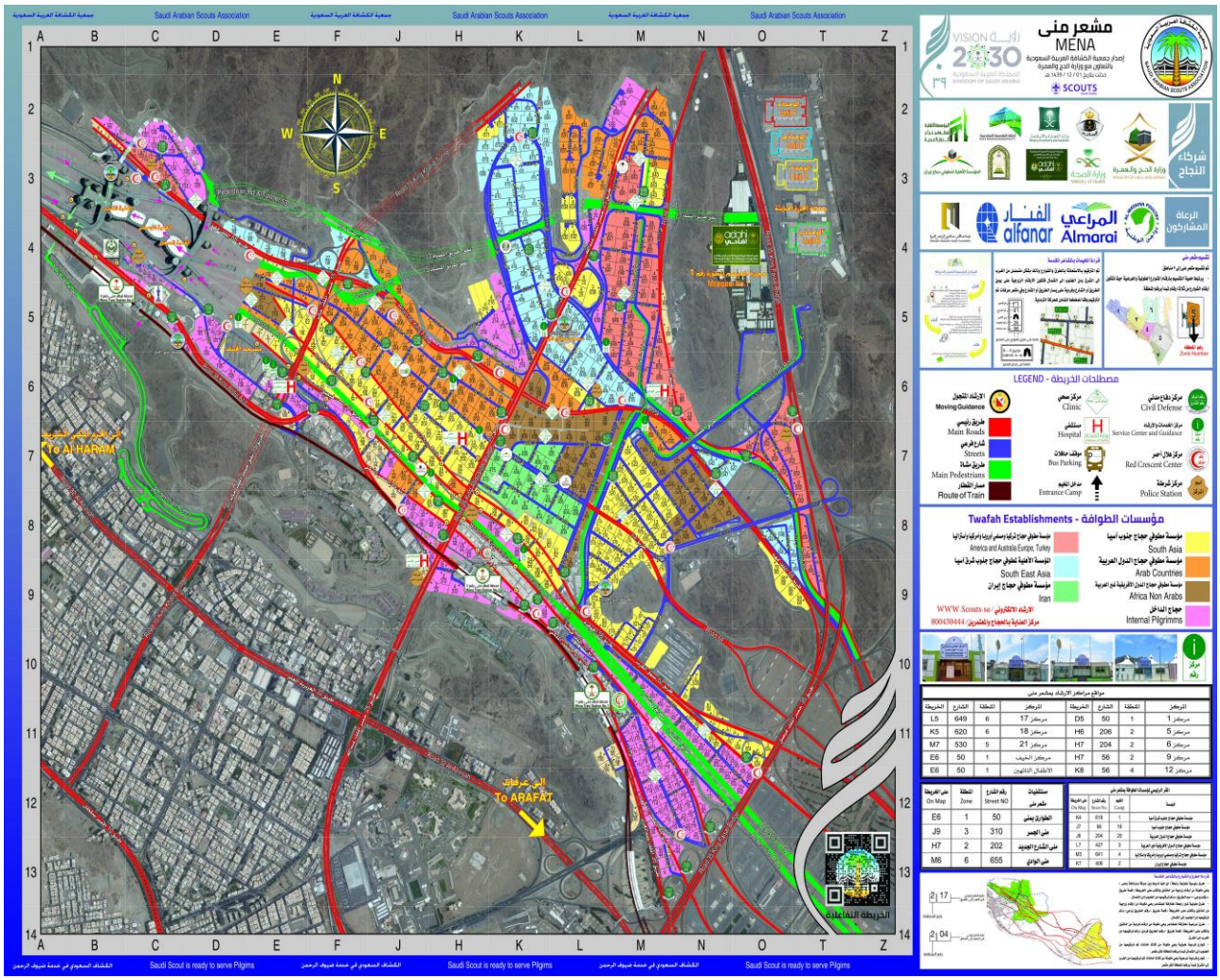

Figure 4: Mina Location

1.5 Time

Hajj occasion in between 8 - 12 Dhul-Hijjah in Islamic calendar but most of Pilgrims coming from outside starting by Medina and staying before 8 Dhul-Hijjah one or two weeks to visiting Madina visiting Tomb of Prophet Mohamed and many Islamic historical places, before Arafat day by one or two days Pilgrims starting to moving to Makkah for Hajj duties at the following order locations Arafat – Muzdalifa – Mina – Makkah.

There are 6-points different Miqat depends on the coming direction for each Pilgrim should starting their Hajj from these points by changing closes and decide again to do Hajj, on vice versa Umra available at any time exclude Hajj period while all Saudi government resources and efforts directed for great occasion of Hajj in the beginning of Saudi embassies distributed on all over the world that do the best to finishing process for huge number of Visa requests in a short time, preparing international ports to receiving incoming Pilgrims, distributing to available staying places on Hajj sites and hotels, putting emergency and medical teams and also hospital at the highest level of ready to dealing with any emergency situations.

"Short-term recovery restores vital services and systems. This may include temporary food, water, and shelter to citizens who have lost homes in a hurricane or large wildfire, assuring injured persons have medical care, and/or restoring electrical services through emergency generators, and so forth. The effects of the emergency may be continuous and ongoing, but the immediate threats are halted and basic services and vital needs are restored. A GIS can play an important role in short-term recovery efforts.

One of the most difficult jobs in a disaster is damage assessment. A GIS can work in concert with GPS to locate each damaged facility, identify the type and amount of damage, and begin to establish priorities for action (triage). Laptop computers can update the primary database from remote locations through a variety of methods. GIS can display (through the primary database) overall current damage assessment as it is conducted.

Emergency distribution centers' supplies (medical, food, water, clothing, etc.) can be assigned in appropriate amounts to shelters based on the amount and type of damage in each area. GIS can display the number of shelters needed and where they should be located for reasonable access. A GIS can display areas where services have been restored in order to quickly reallocate recovery work to prioritize tasks. Action plans with

maps can be printed, outlining work for each specific area. Shelters can update inventory databases allowing the primary command center to consolidate supply orders for all shelters. The immediate recovery efforts can be visually displayed and quickly updated until short term recovery is complete. This visual status map can be accessed and viewed from remote locations; this is particularly helpful for large emergencies or disasters where work is ongoing in different locations". (Johnson 2000)

Chapter-2: Methodology

2.1 Feature List

- Sending location request to Pilgrim
- Auto detect for the location request messages
- Auto-reply by sending locating SMS automatically
- Invoking real-time new emergency call to Hajj Management System
- Real-time web mapping for Pilgrim's emergency call location
- Staying and personal information identification screen
- Identifying real-time emergency call status on the map
- Ordered list for emergency calls on emergency calls explorer
- Receiving and storing emergency call by Hajj Management System server
- Real-time pushing received emergency calls to explorer map and HajiRelief mobile applications
- Emergency call's status updates in real-time mode
- Routing for emergency call location by Google Maps
- Managing emergency calls' life cycle to recovery until closing by medical teams

2.2 Literature Review

Review some cases and applications relating to emergency management and Hajj tracking

3.1.1 Pilgrims" Hajj" Tracking System (e-Mutawwif)

"A few years ago many systems providing tracking service using dedicated GPS devices as a trackers while most of these systems used for vehicles tracking, wide range of these systems developed used short message service(SMS) messages to send location date from GPS receivers to monitoring side, Other systems used general packet radio service (GPRS) for the same purpose, After the widespread of mobile phones with embedded GPS receivers; several searches proposed the usage of smartphone with GPS receiver for tracking purposes".

(AMRO & NIJEM 2012)

No personal and medical information about Pilgrims available to retrieve an emergency situation our proposed system consider of using mobile phones with embedded GPS receiver integrating with Pilgrim personal and stay information to receive and record emergency call at Hajj management system that will by default Invoked to all distributed medical centers and emergency call map for this new call to starting recovery process to Pilgrim.

System available all of the time of Hajj no need for phone calls or waiting call centers for call reply, accrued call location to each emergency call sent, shortly medical team moving for this location having a personal and sick description about caller supporting teams for being ready dealing with the situation at the moment call received.

3.1.2 RealOpt

"RealOpt allows public-health emergency coordinators to

- Determine locations for point-of-dispensing (POD) facility setup
- Design customized and efficient floor plans for PODs via an automatic graph-drawing tool

- Determine required labor resources and provide efficient placement of staff at individual stations within a POD
- Perform disease-propagation analysis, understand and monitor the intra-POD disease dilemma, and help to derive dynamic response strategies to mitigate casualties
- Assess resources and determine minimum needs to prepare for treating their regional populations in emergency situations
- Carry out large-scale virtual drills and performance analyses, and investigate alternative strategies
- Design a variety of dispensing scenarios that include emergency-event exercises to train personnel. These advanced and powerful computational strategies allow emergency coordinators to quickly analyze design decisions, generate feasible regional dispensing plans based on best estimates and analyses available, and reconfigure PODs as an event unfolds. The ability to analyze planning strategies, compare the various options, and determine the most cost-effective combination of dispensing strategies is critical to the ultimate success of any mass dispensing effort

In the past decade, a succession of public health emergencies has challenged preparedness and response capacities of government agencies, hospitals and clinics, public health agencies, and academic researchers, in the United States and abroad. The epidemic of the severe acute respiratory syndrome (SARS), the 9/11 terrorist attacks, and the anthrax mailings stand out as signal examples in the early years of the decade. In addition to natural disasters such as the 2010 earthquake in Haiti and the 2012 Super storm Sandy, other recent events — including the 2009 influenza A (H1N1) pandemic, the Deep water Horizon oil spill, and the Fukushima Daiichi nuclear reactor emergency in Japan — illustrate the diverse and complex forms that threats to public health can assume". (Nicole, Teri, Amy, Francis & Frieden 2013)

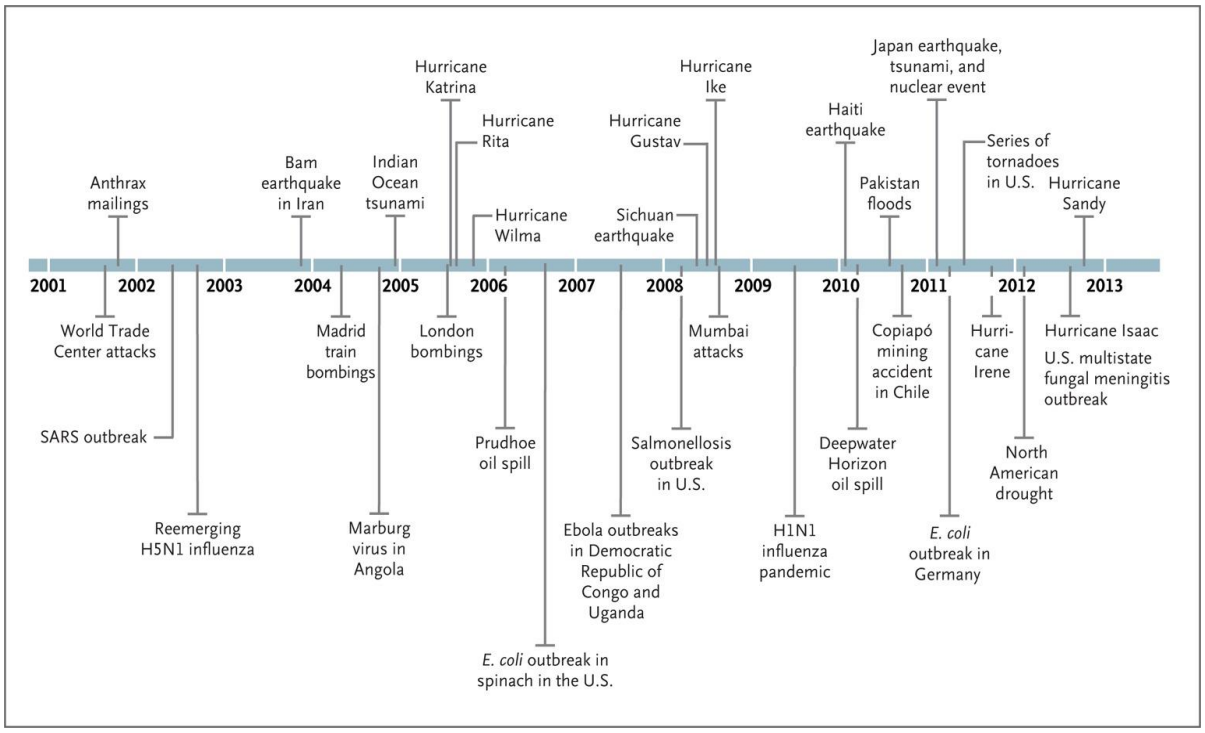

Figure 5: Popular International emergencies Timeline

(Nicole, Teri, Amy, Francis & Frieden 2013)

Figure 5 displays some examples over the past decade or so and highlights the diversity and frequency of events that can be expected to occur in the foreseeable future, Each of these emergencies has yielded important information and data that are essential to what is, by design and necessity, an ongoing effort to improve preparedness and response. But each has also underscored a persistent need to be better prepared to resolve important research questions in the context of a public health emergency.

The knowledge that is generated through well-designed, effectively executed research in anticipation of, in the midst of, and after an emergency is critical to our future capacity to better achieve the overarching goals of preparedness and response: preventing injury, illness, disability, and death and supporting recovery. We review challenges to the

conduct of research in recent public health emergencies to identify critical elements of an effective research response.

1.5.1 Japanese Earthquake and Tsunami

"Crowdsourcing websites were used for monitoring traffic patterns out of affected regions and for tracking radiation contamination of food in the affected region and beyond". (CDC 2012)

"Google's Crises Response site was one of the most visited social media sites used for sharing information on the crises. It provided access to the company's Person Finder search program, which helps people reconnect after a disaster, using both personal descriptions and photos. They could connect with missing person's phone lines and emergency voicemail message boarders. They could also receive alerts and statuses from world health agencies, Japanese utility companies, government agencies services, and real-time updates of RSS feed". (BULLOCK, COPPOLA & HADDOW 2017)

2.3 Risks and Opportunities

- Poor mobile Internet connection at high mountains that cover most sites of the interesting area
- Poor GPS signal while Makkah province contains many high mountains at Hajj sites poor signal may appear especially at cloudy weather
- Power absence at most of Hajj duties outside the building and electric networks does not exit mobiles needs for charging continuously depending on mobile battery size and application performance
- Android Mobile availability while my scope here to developing mobile application for Android system, Samsung depends on Android platform as an operating system for its

new mobiles that will be supporting the installation of HajiPilot and HajiRelief applications but IOS System that Apple depending on as the mean operating system not supporting for this application that means iPhone users cannot install the applications

- Application language HajiPilot and HajiRelief produced in the English language so it will be difficult for non-English users
- Spatial data unfortunately not available for mapping date of Hajj sites
- Skilled medical teams will support our goal of saving Pilgrims' lives
- Resources as poor communications, little cars or medicine may have a bad effect on time response for calls
- SMS Bulk for verifying registered phone number not available result of cost

2.4 Planning for system development

By identifying goal and objectives of my master thesis, interesting area identifying and scope to drawing practical steps enabling to achieving main goal of saving lives by decreasing crises effectiveness and giving high quality emergency response service for emergency calls depending on mobility technology integrated with GIS, I put my strategy to develop Hajj Management System consisting on three applications, first for sending emergency call the second to deal with received call and the third for exploring call emergency and monitoring progress of emergency call life cycle on web map.

2.4.1 Functional Requirements

Pilgrims register by phone number and Hajj visa number for HajiPilot to retrieving Pilgrim' personal and staying information by visa number, once registration succeed Pilgrim has the ability to making emergency call for any critical issue, System receiving this call,

storing call information and forwarding it to clients we have here HajiRelief mobile applications and emergency calls map explorer

Figure 6: HajiPilot functional requirements

Family members or friends can send location request for their Pilgrim, HajiPilot auto-detect this message request and auto-response by current coordinates by SMS containing on location link and finally Pilgrim has information icon to retrieve this staying information

any time and displaying information screen at any time to remember his staying information.

Figure 7: HajiRelief functional requirements

Emergency teams receiving emergency call on time for new tasks, browsing call's location and Pilgrim's information, receive tasks at his missions list and removing from other teams' applications and also real-time updates of call status at Map explorer displaying it as in progress task, Routing to call's location identifying the lowest cost pass to call's

location, dealing with the emergency and change task status to completed that will be displayed as completed at the same moment at emergency calls map explorer.

Figure 8: Map Explorer functional requirements

Emergency call map explorer acting as supervisor for the whole system processes by locating and browsing emergency calls identifying each one by different marker colour relating to call's status while red for new calls, yellow for in progress and green to served and closed, ordered list by time of call displayed beside the map clickable to zooming in to call extent on map.

2.4.2 Nonfunctional requirements

Describing nonfunctional requirements that should be provided to working the system

Table 3: Nonfunctional requirements

Requirement	Description
Performance	Fast response for emergency calls, Power saving for battery High bandwidth communication among server and emergency teams' mobile New mobiles in a good specification Good condition Microsoft server or host providing Microsoft IIS and SQL Server
Availability	Server availability in all day Should have a domain name Android mobiles for application support GPS location services and Internet
Security	Depending on web APIs security basic authentication to any request for the server User name and password for each one of the emergency teams
User Interface	The simple and nice interface as possible to be suitable for calls in a simple step Colored marker on the map to identify each call's status Google maps today one of the most used services so no much training needed to teams while experience usability available
Responsive Design	Should compatible with any screen size that varies according to phone
Reliability	The platform should have the ability to all 7 days of the week and 24 hours in a day Stable server and communication to teams' devices
Data backups / distribution	The system should have a backup plan and depend on database clustering be assure system working at any critical technical situation as server down or unavailable
Real-Time application development	The main goal of my study avoiding missing pilgrims and saving their lives by providing fast response for any emergency call so time hare has a great effect to achieving wanted success so it is necessary to depend on the real-time developing structure to send/receive calls and also updating the progress status for active tasks

2.4.3 Restful Web APIs Services

Representational State Transfer (REST) has become widespread acceptance across the web while it is replacing SOAP and Web Service Description Language (WSDL), the main key for this shifting is the adoption of REST by mainstream Web 2.0 service providers including Yahoo, Google, and Facebook that deprecated or passed SOAP and WSDL –

based interfaces in favor of an easier-to-use, resource-oriented model to expose their services, implementation of a REST Web Service follows four basic design principles

- Use HTTP methods explicitly
- Be stateless
- Expose directory structure-like URIs
- Transfer XML, JavaScript Object Notation (JSON), or both

REST is not always the right choice but referencing to my plane to developing three different applications tow Android application and one web mapping application so I need fast server response and high performance and also independent standard resources to integrate with any development platform and stranded data transfer format JSON.

2.4.4 Microsoft SignalR

It is a new Microsoft technology depending on network sockets that open channel to send and receive packets of data on real-time by proving a library for ASP.NET developers simplify of adding real-time functionality for applications giving the ability to pushing content from the server to clients on time no action from the client needed, SignalR handles connection management automatically, and lets you broadcast messages to all connected clients simultaneously, like a chat room. You can also send messages to specific clients. The connection between the client and server is persistent, unlike a classic HTTP connection, which is re-established for each communication.

SignalR supports "server push" functionality, in which server code can call out to client code in the browser using Remote Procedure Calls (RPC), rather than the request-response model common on the web today, The SignalR API contains two models for communicating between clients and servers Persistent Connections and Hubs.

A Connection represents a simple endpoint for sending single-recipient, grouped, or broadcast messages. The Persistent Connection API (represented in .NET code by the Persistent Connection class) gives the developer direct access to the low-level communication protocol that SignalR exposes. Using the Connections communication model will be familiar to developers who have used connection-based APIs such as Windows Communication Foundation.

A Hub is a more high-level pipeline built upon the Connection API that allows your client and server to call methods on each other directly. SignalR handles the dispatching across machine boundaries as if by magic, allowing clients to call methods on the server as easily as local methods, and vice versa. Using the Hubs communication model will be familiar to developers who have used remote invocation APIs such as .NET Remoting. Using a Hub also allows you to pass strongly typed parameters to methods, enabling model binding.

2.4.5 Native Android mobile application development

Many platforms and tools today developed to producing applications that support the Android operating system, Java language is the primary language for developing Android applications used for many years as the only language for this purpose but result of technology revolution of mobility services and a mobile manufactories wide scale of people moving to smartphones have many advanced services like GPS, Wi-Fi, 3G, 4G, So producing application for android mobiles required java platforms but for IOS should using another platforms and language such as Swift, double efforts required and cost.

To avoid this cost and effort a great platform appeared known as cross-platform no needs for Java, Swift just slandered HTML5, CSS3 and JavaScript's that building independent platform applications that will work for any mobile platform no need to focus where will run

your application that gives us the chance to focus on business but there are some limitation of mobile resources access such as background services that I will depend on to access pilgrims' locating even if application not active on the screen and working at system background

So that I looked for developing the platform that supporting background services, so I decided to depend on Xamarin platform writing C# programming language and also give the power of using all available mobile resources to developing tow android applications and depending on Microsoft ASP.Net for web mapping explorer integrated with OpenLayers JavaScript library.

Table 4: System requirements

Operating System	**Microsoft Windows 10 Pro**
IDE	Microsoft Visual Studio 2017 Xamarin Visual Studio Emulator Android SDK Java JDK 8.1 Microsoft Server 2016
Database	Microsoft SQL Server 2014 Microsoft SQL Management 2014
Platforms	OpenLayers JavaScript's library Microsoft ASP.Net MVC
Tools	Microsoft IIS 7.0 Godaddy Host Domain Name

2.5 Users

Pilgrim: Who coming for Hajj and has Visa information at Hajj management system

Observer: Maybe a family member or friend who will need to have pilgrim location anytime to assure he is safe in specify Hajj regions

Relief Team: Teams that are ready have medical tools and medicines at the moment to receiving emergency calls and moving directory for call's location

Hajj Management: Followers for all processes and progress of recovery emergency calls on the map explorer and can focus or give instructions for teams to moving for hot points on reference to the volume of calls distributed on the map.

2.6 Use Cases

Talking about the implemented use case for the three application in the required Hajj Management System

2.6.1 HajiPilot Use Cases

Table 5: Mobile registration

Mobile registration	**UC101**
Users	Pilgrim and Anonymous users
Description	Registering by the application using mobile
Requirements	Users should provide his phone number that will receive location requests from followers or send an emergency call by Pilgrims
Conditions	Phone number mandatory for using HajiPilot application
Integrations	Integrate with published Restful Web APIs Service to register mobile number
Steps	Typing the mobile number and click register just for one time

Table 6: Visa information

Visa information	**UC102**
Users	Pilgrims
Description	Providing visa number related to the Hajj process
Requirements	Each Pilgrim should have his visa on passport identified by visa number should provide to application to retrieve Hajj personal and stay information
Conditions	Visa number mandatory to provided
Integrations	Integrate with published Restful Web APIs Service to check visa number validity and retrieving personal and stay information
Steps	Typing visa number and click get information

Table 7: Contact importing

Contact importing	UC103
Users	Pilgrim and Anonymous User
Description	Importing contacts from contact phone list to watch list
Requirements	Registration by phone number
Conditions	A phone should have a contact list at least one contact
Outputs	Imported contact now added to watch
Integrations	
Steps	Click Add Contact Button Select contact from the list Clicking on target contact that will be adding to the application contact list

Table 8: Location request

Location request	UC104
Users	Pilgrim and Anonymous User
Description	Send a location request message to the target contact at watch list auto-message response from Pilgrims' phone received containing his location Google maps link
Requirements	Phone number registration Import contacts to watch list
Conditions	Contact added to watch list
Outputs	Receiving SMS contains Pilgrim's Google map location link
Integrations	Integrated to Google Maps location service at the Android system to get the best location accuracy
Steps	Clicking imported contact at watch list will send location request, background broadcast service filter messages, and responses by a message containing on the location link

Table 9: Stay information

Staying information	UC105
Users	Pilgrim User
Description	Displaying personal and staying information for hotels and camps at Hajj sites
Requirements	Having Visa number and register it to the application
Conditions	Visa number mandatory
Outputs	Displaying personal and staying information screen
Integrations	Integrate to developed WebAPI Service to retrieve data
Steps	Click on Information button retrieve data for the first time from the server

Table 10: Emergency call

Emergency call	UC106
Users	Pilgrim User
Description	Sending emergency call to Hajj management system acquiring for fast recovery
Requirements	Having Visa number and register it to the application
Conditions	Phone and Visa number mandatory
Outputs	Receiving emergency call, Storing call information at system's database and real-time pushing to medical teams' devices in addition to emergency calls map explorer
Integrations	Integrate to developed WebAPI Service recording received emergency calls to the server
Steps	Just clicking the emergency call button

Table 11: Fatwa Phone call

Fatwa Phone call	UC107
Users	Pilgrim User
Description	Fatwa center has a local free number to receive a phone call from anyone has a question about Islam or Hajj ritual 24/day so no need to saving this number just using HajiPilot application that will invoke phone call to fatwa center
Requirements	Phone number registered
Conditions	Phone number mandatory
Outputs	A phone call to fatwa center
Integrations	
Steps	Just clicking Fatwa call button

2.6.2 HajiRelief Use Cases

Table 12: Employee Login

Employee login	UC201
Users	Medical teams
Description	Logging by received user name and password
Requirements	Each one should receive user name and password from Hajj management to using logging in to the application
Conditions	Phone number mandatory for using HajiPilot application
Integrations	Integrate with published Restful Web APIs Service to register mobile number
Steps	Typing the mobile number and click register just for one time

Table 13: Call locating

Call locating	UC202
Users	Medical Team
Description	Google map viewer for all new emergency calls, when received by a member, will be hidden from other stuff' devices
Requirements	
Conditions	Signing in
Integrations	Google maps services and Google Maps APIs
Steps	The main screen of the application has a full screen of the map displaying new and received calls for each one at the location of the call Stuff can preview calls and select one to see its information and select it to assign for himself by clicking its marker Map real-time invoked by new call sent from Pilgrims

Table 14: In Progress Calls

In Progress Calls	UC203
Users	Medical Team
Description	Assigning new emergency calls to himself
Requirements	
Conditions	Signing in
Integrations	Google maps services and Google Maps APIs
Steps	Navigating between the call from the right list of new tasks Clicking new calls marker confirmation dialog popups clicking to accept this mission for his tasks New task pushes to the left list of my tasks in progress list Real-time hiding for other teams' devices

Table 15: Closing Calls

Closing calls	UC204
Users	Medical Team
Description	Closes my assigned in progress task
Requirements	
Conditions	Signing in
Integrations	Google maps services and Google Maps APIs
Steps	Once pilgrim critical situation recovered selecting his call from in progress left list Double-clicking closing conformation popups accept and close this call and real-time removing from map and tasks' list

2.8.3 Emergency Call Real-time Map Explorer Use Cases

Table 16: Calls Map Explorer

Calls' map explorer	**UC301**
Users	Hajj Management
Description	Displaying all emergency calls on map locating for its coordinates
Requirements	
Conditions	
Integrations	Open Layer maps and Open Street Map
Steps	No action here required where map displaying distributed emergency calls by location at different Hajj sites classified by different colored symbology while reading for new calls, yellow for in-progress calls, green for closed Real-time updates for calls on the map based on call status changing by medical teams on the land Following and supervision for efforts and resources that may be required to moving for hot points New calls invoking map at real-time

2.8 Ability to Apply

To discovering urgent need Hajj Management System integrated to GPS location service it is enough to searching online about the number of deaths and Injuries at Hajj yearly you will get endless numbers of reports and news talking about deaths and Injures yearly from different countries result of

- Crushes and failure of crowd control
- Fires
- Protest and violence
- Diseases
- Construction failures

Depending on traditional emergency management that has become not enough to decreasing and avoiding bad effect of crises depending on available human resources and assets that needs to emerging to new technologies and technology revolution specially that's related to network analysis and location service result of satellite launching and competition of location data accuracy between American location system GPS and other systems accuracy so I planned to merge all available resources and rebuild it to be compatible to new technology producing at Hajj Management System integrating with maps, location services and Android systems to build a complete system able to decreases response time required for emergency situation recovery and also best way to use available resources avoiding wasted time and efforts and decreasing the response time at critical situations.

As described new system still needs to available resources of medical teams, transportation, Imbalances, and hospitals to manage planned strategy depending on using android applications helping of monitoring calls and routing using maps
Pilgrim has a family mostly outside of Saudi Arabia hope to keep touch with Pilgrim any time anywhere to assured he/she is fine and not suffering from any emergencies so providing this application will achieve this Goal.

HajiPilot and HajiRelief Android mobile applications don't need to high-level qualifications or platforms to work so it is applicable to Pilgrim just download and install from Play Store and installing, registering by phone number and visa number already have, simple interface and usability to ask for help, retrieve staying information or calling Fatwa center, mobiles using android systems increasing between users from day to day and also using mobiles to finish most daily functions so there are critical needs to migrate from traditional ways of emergency needs' calls and also response.

We also cannot ignore Pilgrims that coming from inside and did not take the kick point of registration to take Hajj License from Saudi government especially who already living at Makkah, HajiPilot will ask them to register mobile number and relating already living at Makkah no need for housing at gust hotels or Camps at Muzdalifa, Mina or Arafat, but no information available for these Pilgrims but they will have the other same functionality to share or request location with others and also has the ability to ask for emergency call at any time.

Roaming service available while all communication companies at Saudi providing roaming service for any visitor having SIM card for most outside Communication Company any Pilgrim will have the option to use his mobile number from mother country or buying a new local SIM card of Saudi companies any way using roaming service or buying local SIM card pilgrim will have access to call emergency numbers or fatwa and also receiving or sending SMS.

Medical teams will play the same role among people on the land but they will be distributed and centralized on all Hajj sites ready for receiving real-time emergency calls directly on their mobiles using HajiRelief Application, no needs to carry any other thing just mobile connected to internet by receiving new task and assigning for himself will routing to call's location in moving in the shortest pass to arrive at shortest possible time as we see simple processes or steps starting by receiving emergency call till closing this task.

Chapter-3: Processes

3.1 Environment setup

3.1.3 Database design

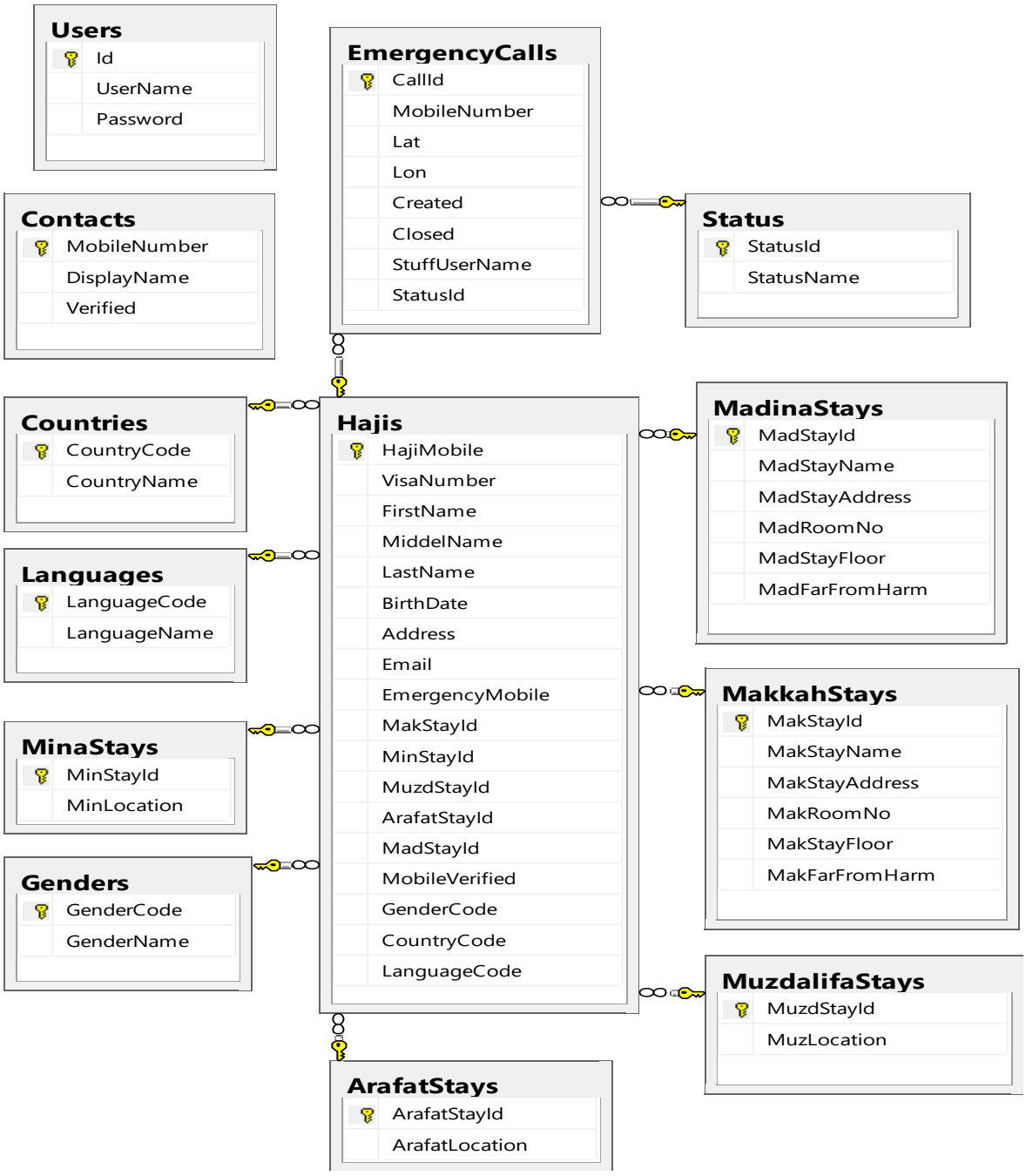

Figure 9: Hajj Management System Database

Referring to the case study and requirements identified there are two choices for database design relational database or spatial database, there are urgent questions should be answered here what is data types we have? What is the process and analysis will be required? I am choosing Google maps as spatial data for HajiPilot and HajiRelief and open street maps for calls' map explorer while my focus is emergency located calls no need to start from scratch by data collection, classification, and analysis Hajj sites location available at most popular map services

So the database will be designed as a relational database for storing Pilgrim's information and calls' location database designed by Microsoft SQL Server 2014 no need for spatial extension implementing 13 tables and identifying its relationships and also fields data type that consistent to data that will be stored, retrieved and modified, database diagram design appeared at the fowling section giving a clear vision about tables and its relationships, database hosted to Godaddy Microsoft Windows server and implemented to Microsoft SQL Server 2014

3.1.4 Web APIs Restful service

Developing Restful Web API that will be used as a layer for data management has the ability to database date modifications and serialize data between server and Hajj Management system applications, as we give a brief discussion about different types of services I decided to use Restful Web APIs service that comes popular on all online application and mobiles ESRI Itself shifting all its mapping services to Restful result of its advantages of performance, HTTP Protocol and URL routing features and hosted at Godaddy windows server, Emergency call service code in C# as the following

```
using HajiPilotAPI.Hubs;
using HajiPilotService.Models;

using System.Data;
using System.Data.Entity;
```

```
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;

using System.Threading.Tasks;
using System.Web.Http;
using System.Web.Http.Description;


namespace HajiPilotAPI.Controllers
{
    public class EmergencyCallsController : ApiController
    {
        private HajiPilotEntities db = new HajiPilotEntities();

        // GET: api/EmergencyCalls
        //[Route("api/EmergencyCalls/GetEmergencyCalls")]

        public IHttpActionResult GetEmergencyCalls()
        {

                var list = db.EmergencyCalls.ToList();
            if (list.Count>0)
                return Ok(list);
            return Ok();

        }
        [Route("api/EmergencyCalls/OpenedEmergencyCalls")]
        public IHttpActionResult GetOpenedEmergencyCalls()
        {
            var list = db.EmergencyCalls.Where(c=>c.StatusId==1).ToList();
            return Ok(list);
        }

        [Route("api/EmergencyCalls/GetMyEmergencyCalls")]
        public IHttpActionResult GetMyEmergencyCalls(string StuffUserName)
        {
            var   list   =   db.EmergencyCalls.Where(c   =>   (c.StuffUserName   ==
StuffUserName&&c.StatusId==2)).ToList();
            return Ok(list);
        }
        // GET: api/EmergencyCalls/5
        [ResponseType(typeof(EmergencyCalls))]
        public async Task<IHttpActionResult> GetEmergencyCalls(long id)
        {
            EmergencyCalls emergencyCalls = await db.EmergencyCalls.FindAsync(id);
            if (emergencyCalls == null)
            {
                return NotFound();
            }

            return Ok(emergencyCalls);
        }

        // PUT: api/EmergencyCalls/5
        [ResponseType(typeof(void))]
        public    async    Task<IHttpActionResult>    PutEmergencyCalls(long    id,
EmergencyCallsVM vm)
        {
            if (!ModelState.IsValid)
            {
```

```
            return BadRequest(ModelState);
        }

        if (id != vm.CallId)
        {
            return BadRequest();
        }
        var model = EmergencyCallsExists(id);
        if (model == null)
        {
            return NotFound();
        }
        if (vm.StatusId==model.StatusId)
        {
            return Conflict();
        }
        CopyToModel(vm, model);
        db.Entry(model).State = EntityState.Modified;

        try
        {
            int saved = await db.SaveChangesAsync();
            if (saved>0)
            {
                if (model.StatusId==2)
                    CallHub.AssignCall(id);

                else if (model.StatusId == 3)
                    CallHub.CloseCall(vm);
                return Ok();
            }
        }
        catch (DbUpdateConcurrencyException)
        {
            return BadRequest();
        }

        return StatusCode(HttpStatusCode.NoContent);
    }

    // POST: api/EmergencyCalls
    [ResponseType(typeof(EmergencyCalls))]
    public async Task<IHttpActionResult> PostEmergencyCalls(EmergencyCallsVM vm)
    {
        if (!ModelState.IsValid)
        {
            return BadRequest(ModelState);
        }
        EmergencyCalls model = new EmergencyCalls();
        CopyToModel(vm,model);
        db.EmergencyCalls.Add(model);
        int success = await db.SaveChangesAsync();

        if (success > 0)
        {
            vm.CallId = model.CallId;
            //string obj = JsonConvert.SerializeObject(call);
            CallHub.AddCall(vm);
        }
        return CreatedAtRoute("DefaultApi", new { id = model.CallId }, model);
    }
```

```
        private void CopyToModel(EmergencyCallsVM src, EmergencyCalls dest)
        {
            if (src.Closed.HasValue)
                dest.Closed = src.Closed;
            if (src.Created.HasValue)
                dest.Created = src.Created;
            if (!string.IsNullOrEmpty(src.Lat))
                dest.Lat = src.Lat;
            if (!string.IsNullOrEmpty(src.Lon))
                dest.Lon = src.Lon;

                dest.StuffUserName = src.StuffUserName;
            if (!string.IsNullOrEmpty(src.MobileNumber))
                dest.MobileNumber = src.MobileNumber;
            dest.StatusId = src.StatusId;
        }

        // DELETE: api/EmergencyCalls/5
        [ResponseType(typeof(EmergencyCalls))]
        public async Task<IHttpActionResult> DeleteEmergencyCalls(long id)
        {
            EmergencyCalls emergencyCalls = await db.EmergencyCalls.FindAsync(id);
            if (emergencyCalls == null)
            {
                return NotFound();
            }

            db.EmergencyCalls.Remove(emergencyCalls);
            await db.SaveChangesAsync();

            return Ok(emergencyCalls);
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }

        private EmergencyCalls EmergencyCallsExists(long id)
        {
            return db.EmergencyCalls.Where(e => e.CallId == id).FirstOrDefault();
        }
    }
}
```

3.1.5 System Development

We had talked about the case study, system infrastructure required; tools, functionalities, and required services, here will move deep translate plan and business to real working

system covers required functionalities, three applications required one for Pilgrim, the second for the medical team and third for Hajj Management.

There are many IDEs and programming languages used now for mobile application development that produces native mobile applications such as Java, C, C++ and etc... but each mobile system should have independent version of application and need longer time to develop application for each platform On the other hand fast revolution of HTML at HTML5 lead to new concept Cross-platform applications that have the compatibility to run on multiple platforms by the same version and shorter time needs for developing but have some limitations to using mobile resources such as background service.

Referring to previous discussion and requirements application should be native for my focus platform android so will using Xamarin integrated with Microsoft Visual Studio 2017 and programming language C#.

3.1.6 HajiPilot

Java programming language is the main programming language for the Android mobile operating system that depends on Linux and Mono framework best choice using Java that will give the developer full access for all mobile resources but recently Xamarin framework enabling to write C# language and providing the ability to have full access for android mobile resources while I have strong experience using C# language at Microsoft Visual Studio IDE, Xamarin bought by Microsoft and has become part of Visual Studio so it is my choice developing Native Android using Microsoft Visual Studio writing C#.

Code of the main HajiPilot screen in C# using Xamarin as the following

```
using System;
using Android.App;
using Android.Widget;
using Android.OS;
```

```
using Android.Util;

using System.Collections.Generic;
using Android.Content;
using System.Linq;
using Android.Telephony;
using Android.Locations;
using Location.Droid.Services;

using HajiPilotService.Models;
using HajiPilotService;
using HajPilot.Droid;


namespace HajiPilot.Droid
{

    [Activity(Label = "HajiPilot")]
    public class ActivityUnderObserver : Activity
    {
        readonly string logTag = "HajiDetailsActivity";
        Button btnAddContact, btnInformation, btnEmergncyCall, btnFatwaCall;
        ListView listView;
        List<ContactVM> lstUnderObservation;
        CustomListAdapter customListAdapter;
        ContactVM localContact = new ContactVM();
        TextView txtLat, txtLon;
        IHajiPilotService service;
        string MobileNumber;
        string lat, lon;
        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);
            Log.Debug("MainActivity", "OnCreate: Location app is becoming active");

            service = new HajiPilotService.HajiPilotService();
            lstUnderObservation = new List<ContactVM>();

            var localStoredContact =
            Application.Context.GetSharedPreferences("HajiContact",
            FileCreationMode.Private);
            int count = localStoredContact.All.Count;
            if (count != 0)
            {
              localContact.MobileNumber = localStoredContact.GetString("Mobile",
              "");
                 localContact.DisplayName =
                 localStoredContact.GetString("DisplayName", "");
                 MobileNumber = localContact.MobileNumber;
              var underObserevation =
       ContactStore.GetUnderObservation(localContact.MobileNumber);
                 if (underObserevation != null)
                 {
                     lstUnderObservation = underObserevation;
                 }

                 SetContentView(Resource.Layout.UnderObserever);

                 listView = FindViewById<ListView>(Resource.Id.LstViewContacts);
                customListAdapter = new CustomListAdapter(this, lstUnderObservation);
```

```
            listView.Adapter = customListAdapter;
            listView.ItemClick += OnListItemClick;

            btnAddContact = FindViewById<Button>(Resource.Id.btnAddContact);
            btnAddContact.Click += btnAddContact_Click;

            btnInformation = FindViewById<Button>(Resource.Id.btnInformation);
            btnInformation.Click += btnInformation_Click;

            btnEmergncyCall = FindViewById<Button>(Resource.Id.btnEmergncyCall);
            btnEmergncyCall.Click += BtnEmergncyCall_Click;

            btnFatwaCall = FindViewById<Button>(Resource.Id.btnFatwaCall);
            btnFatwaCall.Click += BtnFatwaCall_Click;
            //Initializing listview

            txtLat = FindViewById<TextView>(Resource.Id.txtLat);
            txtLon = FindViewById<TextView>(Resource.Id.txtLon);
                        // This event fires when the ServiceConnection lets
          the client (our App class) know that
            // the Service is connected. We use this event to start updating the
          UI with location
            // updates from the Service
            App.StartLocationService();
          App.Current.LocationServiceConnected += (object sender,
          ServiceConnectedEventArgs e) =>
            {
                Log.Debug(logTag, "ServiceConnected Event Raised");
                // notifies us of location changes from the system
                   App.Current.LocationService.LocationChanged +=
          HandleLocationChanged;
                //notifies us of user changes to the location provider (ie the
user disables or enables GPS)
          App.Current.LocationService.ProviderDisabled += HandleProviderDisabled;
          App.Current.LocationService.ProviderEnabled += HandleProviderEnabled;
          // notifies us of the changing status of a provider (ie GPS no longer
available)
          App.Current.LocationService.StatusChanged += HandleStatusChanged;
              };
          }
          else
              StartActivity(typeof(ActivityRegister));
      }


     private void BtnFatwaCall_Click(object sender, EventArgs e)
      {
          Call("8002451000");
      }

      private async void BtnEmergncyCall_Click(object sender, EventArgs e)
      {
          if (string.IsNullOrEmpty(lat) || string.IsNullOrEmpty(lon))
          {
Toast.MakeText(this, "No Location data Kindly enable location",
ToastLength.Long).Show();
              return;
          }
          ProgressDialog progress = new ProgressDialog(this);
```

```
            progress.Indeterminate = true;
            progress.SetProgressStyle(ProgressDialogStyle.Spinner);
            progress.SetMessage("Loading is Progress...");
            progress.SetCancelable(false);
            progress.Show();
            bool added = await service.AddEmergencyCall(new EmergencyCallsVM()
            {
                Created = DateTime.Now,
                Lat = lat,
                Lon = lon,
                StatusId = 1,
                MobileNumber = MobileNumber
            });
            progress.Hide();
            if (added)
           Toast.MakeText(this, "You emergency call sent succefully",
       ToastLength.Long).Show();
            else
             Toast.MakeText(this, "Sorry Your call not sent",
       ToastLength.Long).Show();
            //Call("911");
        }
        private void Call(string number)
        {
            var uri = Android.Net.Uri.Parse("tel:" + number);
            var intent = new Intent(Intent.ActionDial, uri);
            StartActivity(intent);
        }
        private void btnInformation_Click(object sender, EventArgs e)
        {
            string visaNumber = string.Empty;
var localHaji = Application.Context.GetSharedPreferences("HajiData",
FileCreationMode.Private);
            int countlocalHaj = localHaji.All.Count;
            if (countlocalHaj < 2)
            {
                visaNumber = localHaji.GetString("VisaNumber", string.Empty);
                if (string.IsNullOrEmpty(visaNumber))
                {
                    //GetHajiData("966583148535");
                    EditText txtVisaNumber = new EditText(this);
                    txtVisaNumber.InputType = Android.Text.InputTypes.ClassNumber;
                    AlertDialog.Builder alertDialog = new AlertDialog.Builder(this);
                    //alertDialog.SetCancelable(false);
                    alertDialog.SetTitle("Visa Number");
                    alertDialog.SetView(txtVisaNumber);
                    alertDialog.SetPositiveButton("OK", delegate {
GetHajiData(txtVisaNumber.Text); });
                    alertDialog.Show();
                }
            }

            else
                StartActivity(typeof(ActivityHajiData));

        }


private void btnAddContact_Click(object sender, EventArgs e)
        {
```

```
            var contactPickerIntent = new Intent(Intent.ActionPick,
Android.Provider.ContactsContract.Contacts.ContentUri);
            StartActivityForResult(contactPickerIntent, 101);

        }
        protected override void OnActivityResult(int requestCode, Result resultCode,
Intent data)
        {
            if (requestCode == 101 && resultCode == Result.Ok)
            {
                //Ensure we have data returned
                if (data == null || data.Data == null)
                    return;
                var addressBook = new Xamarin.Contacts.AddressBook(this);
                addressBook.PreferContactAggregation = false;
                //Load the contact via the android contact id
                // in the last segment of the Uri returned by the
                // android contact picker
                var contact = addressBook.Load(data.Data.LastPathSegment);
                //Use linq to find a mobile number
                var mobile = (from p in contact.Phones
                              where
                              p.Type == Xamarin.Contacts.PhoneType.Mobile
                              select p.Number).FirstOrDefault();
                //See if the contact has a mobile number
                if (string.IsNullOrEmpty(contact.Phones.FirstOrDefault().Number))
                {
                    Toast.MakeText(this, "No Mobile Number for contact!",
ToastLength.Long).Show();
                    return;
                }
                ContactVM contactToSave = new ContactVM() { MobileNumber = mobile,
DisplayName = contact.DisplayName };
                if (lstUnderObservation.Any(str =>
str.MobileNumber.Contains(mobile)))
                {
                    Toast.MakeText(this, "This Mobile number added before",
ToastLength.Long).Show();
                    return;
                }

                lstUnderObservation.Add(contactToSave);
                if (ContactStore.StoreUnderObservation(lstUnderObservation))
                {
                    customListAdapter.refreshEvents();
                    Toast.MakeText(this, "Contact " + contact.DisplayName + " added
Succefully", ToastLength.Long).Show();
                }
            }
        }
        private void OnListItemClick(object sender, AdapterView.ItemClickEventArgs
e)
        {
            ContactVM item = lstUnderObservation[e.Position];
            // Do whatever you like here
            Toast.MakeText(this, "Sending SMS... Please wait...",
ToastLength.Long).Show();
            //Send SMS!
            var smsMgr = SmsManager.Default;
            smsMgr.SendTextMessage(item.MobileNumber, null, "Get Location from
HajiPilot", null, null);
```

```
            Toast.MakeText(this, "SMS successfully sent", ToastLength.Long).Show();
        }

        private async void GetHajiData(string visaNumber)
        {
            if (visaNumber.Length < 10)
            {
                Toast.MakeText(this, "Kindly enter valid 10 Visa Numbers",
ToastLength.Long).Show();
                return;
            }
            long visaNo;
            long.TryParse(visaNumber, out visaNo);
            if (visaNo == -1)
            {
                Toast.MakeText(this, "Kindly enter valid 10 Visa Numbers",
ToastLength.Long).Show();
                return;
            }
            ProgressDialog progress = new ProgressDialog(this);
            progress.Indeterminate = true;
            progress.SetProgressStyle(ProgressDialogStyle.Spinner);
            progress.SetMessage("Loading is Progress...");
            progress.SetCancelable(false);
            progress.Show();

            try
            {

                HajiVM haji = await service.GetHajiByVisaNumber(visaNo);
                if (haji == null)
                {
                    progress.Hide();
                    Toast.MakeText(this, "No data availabale for your Visa at Haji
System", ToastLength.Long).Show();
                    StartActivity(typeof(ActivityUnderObserver));
                    return;
                }
                var localHaji = Application.Context.GetSharedPreferences("HajiData",
FileCreationMode.Private);
                var HajiEdit = localHaji.Edit();

                HajiEdit.PutString("VisaNumber", haji.VisaNumber);
                HajiEdit.PutString("FirstName", haji.FirstName);
                HajiEdit.PutString("LastName", haji.LastName);
                HajiEdit.PutString("MakkahStay", haji.MakkahStay);
                HajiEdit.PutString("MinaStay", haji.MinaStay);
                HajiEdit.PutString("MuzdalifaStay", haji.MuzdalifaStay);
                HajiEdit.PutString("ArafatStay", haji.ArafatStay);
                HajiEdit.PutString("MadinaStay", haji.MadinaStay);

                HajiEdit.Commit();
                progress.Hide();
                StartActivity(typeof(ActivityHajiData));
            }
            catch (Exception ex)
            {
                progress.Hide();
                Toast.MakeText(this, ex.Message + " Failed to Register your
information", ToastLength.Long).Show();
```

```
            }


        }
        #region Lifecycle

        protected override void OnPause()
        {
            Log.Debug(logTag, "OnPause: Location app is moving to background");
            base.OnPause();
        }

        protected override void OnResume()
        {
            Log.Debug(logTag, "OnResume: Location app is moving into foreground");
            base.OnResume();
        }

        protected override void OnDestroy()
        {
            Log.Debug(logTag, "OnDestroy: Location app is becoming inactive");
            base.OnDestroy();
            try
            {
                // Stop the location service:
                App.StopLocationService();
            }
            catch
            {


            }

        }
        #endregion
        #region Android Location Service methods

        ///<summary>
        /// Updates UI with location data
        /// </summary>
        public void HandleLocationChanged(object sender, LocationChangedEventArgs e)
        {
            //Android.Locations.Location location = e.Location;
            App.currentLocation = e.Location;
            Log.Debug(logTag, "Foreground updating");
            lat = App.currentLocation.Latitude.ToString();
            lon = App.currentLocation.Longitude.ToString();
            // these events are on a background thread, need to update on the UI
thread
            RunOnUiThread(() =>
            {
                txtLat.Text = String.Format("Lat: {0}",
App.currentLocation.Latitude);
                txtLon.Text = String.Format("Lon: {0}",
App.currentLocation.Longitude);

            });

        }
```

```
        public void HandleProviderDisabled(object sender, ProviderDisabledEventArgs
e)
        {
            Log.Debug(logTag, "Location provider disabled event raised");
        }

        public void HandleProviderEnabled(object sender, ProviderEnabledEventArgs e)
        {
            Log.Debug(logTag, "Location provider enabled event raised");
        }

        public void HandleStatusChanged(object sender, StatusChangedEventArgs e)
        {
            Log.Debug(logTag, "Location status changed, event raised");
        }

        #endregion
    }
}
```

To manage the GPS location service of mobile I develop the following code

```
using System;
using System.Threading;
using System.Threading.Tasks;

using Android.Content;
using Android.Util;

using Location.Droid.Services;

/// <summary>
/// Singleton class for Application wide objects.
/// </summary>
namespace HajiPilot.Droid
{
    public class App
    {
        // events
        public event EventHandler<ServiceConnectedEventArgs>
LocationServiceConnected = delegate { };

        // declarations
        protected readonly string logTag = "App";
        protected static LocationServiceConnection locationServiceConnection;
        public static Android.Locations.Location currentLocation;
        // properties

        public static App Current
        {
            get { return current; }
        }
        private static App current;

        public LocationService LocationService
        {
            get
            {
```

```
                if (locationServiceConnection.Binder == null)
                    throw new Exception("Service not bound yet");
                // note that we use the ServiceConnection to get the Binder, and the
Binder to get the Service here
                return locationServiceConnection.Binder.Service;
            }
        }

        #region Application context

        static App()
        {
            current = new App();
        }

        protected App()
        {
            // create a new service connection so we can get a binder to the service
            locationServiceConnection = new LocationServiceConnection(null);

            // this event will fire when the Service connectin in the
OnServiceConnected call
            locationServiceConnection.ServiceConnected += (object sender,
ServiceConnectedEventArgs e) =>
            {

                Log.Debug(logTag, "Service Connected");
                // we will use this event to notify MainActivity when to start
updating the UI
                this.LocationServiceConnected(this, e);
            };
        }

        public static void StartLocationService()
        {
            // Starting a service like this is blocking, so we want to do it on a
background thread
            new Task(() =>
            {

                // Start our main service
                Log.Debug("App", "Calling StartService");
                Android.App.Application.Context.StartService(new
Intent(Android.App.Application.Context, typeof(LocationService)));

                // bind our service (Android goes and finds the running service by
type, and puts a reference
                // on the binder to that service)
                // The Intent tells the OS where to find our Service (the Context)
and the Type of Service
                // we're looking for (LocationService)
                Intent locationServiceIntent = new
Intent(Android.App.Application.Context, typeof(LocationService));
                Log.Debug("App", "Calling service binding");

                // Finally, we can bind to the Service using our Intent and the
ServiceConnection we
                // created in a previous step.
                Android.App.Application.Context.BindService(locationServiceIntent,
locationServiceConnection, Bind.AutoCreate);
            }).Start();
```

```
        }

        public static void StopLocationService()
        {
            // Check for nulls in case StartLocationService task has not yet
completed.
            Log.Debug("App", "StopLocationService");

            // Unbind from the LocationService; otherwise, StopSelf (below) will not
work:
            if (locationServiceConnection != null)
            {
                Log.Debug("App", "Unbinding from LocationService");

Android.App.Application.Context.UnbindService(locationServiceConnection);
            }

            // Stop the LocationService:
            if (Current.LocationService != null)
            {
                Log.Debug("App", "Stopping the LocationService");
                Current.LocationService.StopSelf();
            }
        }

        #endregion

    }
}
```

HajiPilot Android mobile application requires some permission for working in full functionality as

- Access Location
- Access Fine Location
- Access Network Stat
- Internet Access
- Broadcast SMS Access
- Send SMS
- Receive SMS

To using mobile resources HajiPilot should ask for some permission access to locating pilgrim Access location and access fine location needed, to using mobile internet for data transfer access network state and internet access permission needed, BroadCast SMS

integrated with Receive SMS Access will give access for listening to coming SMS and filter if the sender HajiPilot Application for location request replaying by SMS containing current pilgrim location automatically using Send SMS permission.

3.1.7 HajiRelief

It is android application designed for using my medical teams' members on the land to catching emergency calls and browsing by location on the map and identifying each call and caller information, selecting any emergency call marker and assigning to updating its status to be in progress, routing to call's location and dealing with the situation relating to his skills and finishing this task and closing it, HajiPilot Require the following permission

- Access Location
- Access Fine Location
- Access Network Stat
- Internet Access
- Maps Receive

All these permission discussed at HajiPilot new one Maps Receive that is required for Google Maps Services to working and integrated with the application.

```
    [Activity(Label = "HajiRelief", Icon = "@drawable/icon", Theme =
"@style/MyTheme")]
#pragma warning disable CS0618 // Type or member is obsolete
    public class MainActivity : ActionBarActivity, IOnMapReadyCallback,
IInfoWindowAdapter, IOnInfoWindowClickListener
#pragma warning restore CS0618 // Type or member is obsolete
    {
        private SupportToolbar mToolbar;
        private MyActionBarDrawerToggle mDrawerToggle;
        private DrawerLayout mDrawerLayout;
        private ListView mLeftDrawer;
        private ListView mRightDrawer;
        private ArrayAdapter mLeftAdapter;
        private ArrayAdapter mRightAdapter;
        private List<string> mLeftDataSet;
        private List<string> mRightDataSet;
        private List<Marker> lstMarkers;
        //private static readonly LatLng Madinah = new LatLng(24.4687913,
39.6022902);
```

```
        private static readonly LatLng Makkah = new LatLng(21.4226812, 39.8256239);
        private GoogleMap _map;
        private MapFragment _mapFragment;
        HajiPilotService.IHajiPilotService _HajiPilotService = null;
        List<EmergencyCallsVM> emergencyCallsRight;
        List<EmergencyCallsVM> emergencyCallsLeft;
        private string StuffUserName;
        protected async override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);

            // Set our view from the "main" layout resource
            SetContentView(Resource.Layout.Main);

            lstMarkers = new List<Marker>();

            mToolbar = FindViewById<SupportToolbar>(Resource.Id.toolbar);
            mDrawerLayout = FindViewById<DrawerLayout>(Resource.Id.drawer_layout);
            mLeftDrawer = FindViewById<ListView>(Resource.Id.left_drawer);
            mRightDrawer = FindViewById<ListView>(Resource.Id.right_drawer);

            mLeftDrawer.Tag = 0;
            mRightDrawer.Tag = 1;
            mRightDrawer.ItemClick += MRightDrawer_ItemClick;
            mLeftDrawer.ItemClick += MLeftDrawer_ItemClick;
            mLeftDrawer.ItemLongClick += MLeftDrawer_ItemLongClickAsync;
            SetSupportActionBar(mToolbar);

            mLeftDataSet = new List<string>();

            mLeftAdapter = new ArrayAdapter<string>(this,
Android.Resource.Layout.SimpleListItem1, mLeftDataSet);
            mLeftDrawer.Adapter = mLeftAdapter;

            mRightDataSet = new List<string>();
            mDrawerToggle = new MyActionBarDrawerToggle(
                this,                           //Host Activity
                mDrawerLayout,                  //DrawerLayout
                Resource.String.openDrawer,     //Opened Message
                Resource.String.closeDrawer     //Closed Message
            );


#pragma warning disable CS0618 // Type or member is obsolete
            mDrawerLayout.SetDrawerListener(mDrawerToggle);
#pragma warning restore CS0618 // Type or member is obsolete
            SupportActionBar.SetHomeButtonEnabled(true);
            SupportActionBar.SetDisplayShowTitleEnabled(true);
            mDrawerToggle.SyncState();

            if (bundle != null)
            {
                if (bundle.GetString("DrawerState") == "Opened")
                {
                    SupportActionBar.SetTitle(Resource.String.openDrawer);
                }

                else
                {
                    SupportActionBar.SetTitle(Resource.String.closeDrawer);
                }
```

```
            }

            else
            {
                //This is the first the time the activity is ran
                SupportActionBar.SetTitle(Resource.String.closeDrawer);
            }
            try
            {
                _HajiPilotService = new HajiPilotService.HajiPilotService();

                emergencyCallsRight = await
_HajiPilotService.GetOpenedEmergencyCalls();

                var localStoredUser =
Application.Context.GetSharedPreferences("HajiRelief", FileCreationMode.Private);
                int count = localStoredUser.All.Count;
                if (count != 0)
                    StuffUserName = localStoredUser.GetString("HajiReliefUser", "");
                else
                    StuffUserName = Intent.GetStringExtra("UserName") ?? "";

                emergencyCallsLeft = await
_HajiPilotService.GetMyEmergencyCalls(StuffUserName);

                InitMapFragment();

                var hubConnection = new HubConnection("https://HajiPilot.com/");
                IHubProxy emergencyCallHubProxy =
hubConnection.CreateHubProxy("CallHub");
                emergencyCallHubProxy.On<EmergencyCallsVM>("addCall", call =>
AddRealTimeCall(call));
                emergencyCallHubProxy.On<int>("assignCall", callId =>
AssignCall(callId));
                await hubConnection.Start();
            }
            catch
            {


            }
        }

private void AddMarker(EmergencyCallsVM vm)
        {
            MarkerOptions markerCall = new MarkerOptions();
            LatLng point = new LatLng(double.Parse(vm.Lat), double.Parse(vm.Lon));
            markerCall.SetPosition(point);
            markerCall.SetTitle(vm.CallId.ToString());
            try
            {
                RunOnUiThread(() =>
                { lstMarkers.Add(_map.AddMarker(markerCall)); });
            }
            catch
            { }
        }
        private void AddRealTimeCall(EmergencyCallsVM vm)
        {
            AddMarker(vm);
            mRightDataSet.Add(vm.CallId.ToString());
```

```
            emergencyCallsRight.Add(vm);
            RefershCallsListRight();
        }
        private void AssignCall(int callId)
        {
            var call = emergencyCallsRight.Find(m => m.CallId == callId);
            if (call != null)
            {
                mRightDataSet.Remove(callId.ToString());
                emergencyCallsRight.Remove(call);
                RefershCallsListRight();
            }

        }
        #endregion
        private async void RefreshAsync()
        {
            emergencyCallsRight = await _HajiPilotService.GetOpenedEmergencyCalls();
            RefershCallsListRight();
            emergencyCallsLeft = await
_HajiPilotService.GetMyEmergencyCalls(StuffUserName);
            RefershCallsListLeft();

        }

}
```

3.1.8 Calls Map explorer

Developing by standard HTML5, CSS3 and JavaScript's library supported at Microsoft MVC platform and for mapping, I used Open layers as a GIS JavaScript library to controlling Open Map Street open source mapping data here widescreen to following each call located on the displayed map and identified by different colored markers presents call' status.

3.2 Data Resource

Data availability resources will be used for mapping and personal pilgrims' information

3.2.1 Pilgrim Data

Decision of Hajj for millions of Muslims out and inside of Kingdom of Saudi Arabia should followed by routing steps starting by preparing some papers and passport to acquires Hajj

Visa form Saudi embassies, embassies starting processing by checking given documents and inserting all of these information to their system such as full name, birthday, passport number .. etc., storing these row data for each one in addition to personal photo building a huge data storing on the other hand there is another process for inside Pilgrims while permission should be getting to going to Hajj but no Visa required.

3.2.2 Mapping Data

Unfortunately no spatial data available for Hajj landmarks to mapping it using ESRI ArcGIS Server or other spatial servers so I will depend on Google maps data to locating Pilgrims to providing any location request by observers or family member, Google maps has become the most accurate resource for mapping at the latest years using WGS 1984 as worldwide coordinate system for mobile applications and using Open map streets for emergency calls explorer.

3.3 HajiPilot Screens

3.3.1 Registration

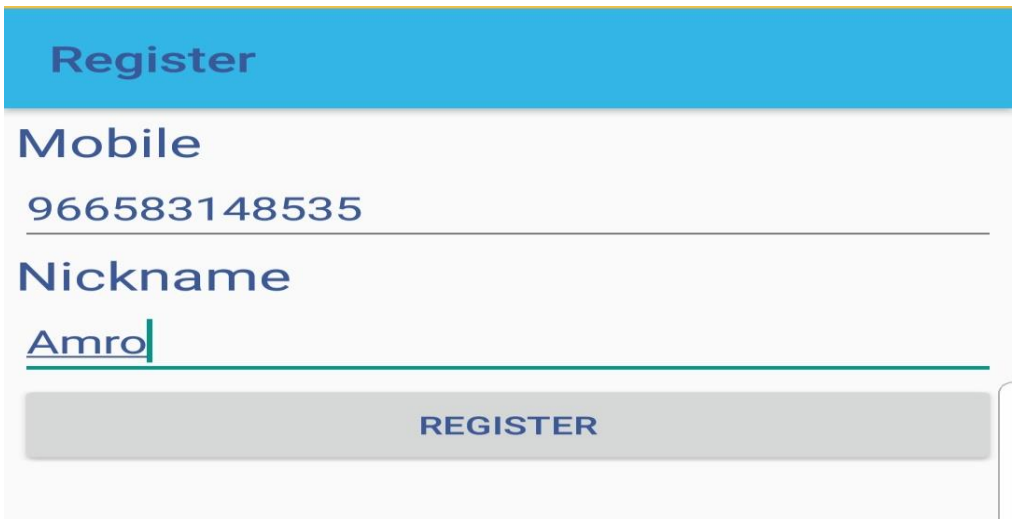

Figure 10: HajiPilot phone number registration

Register screen asks pilgrim to provide the phone number to identify him for his family and friend on HajiPilot application, HajiPilot use sends location request for this number and auto-message response generated containing current Pilgrim's location link.

3.3.2 Watching List

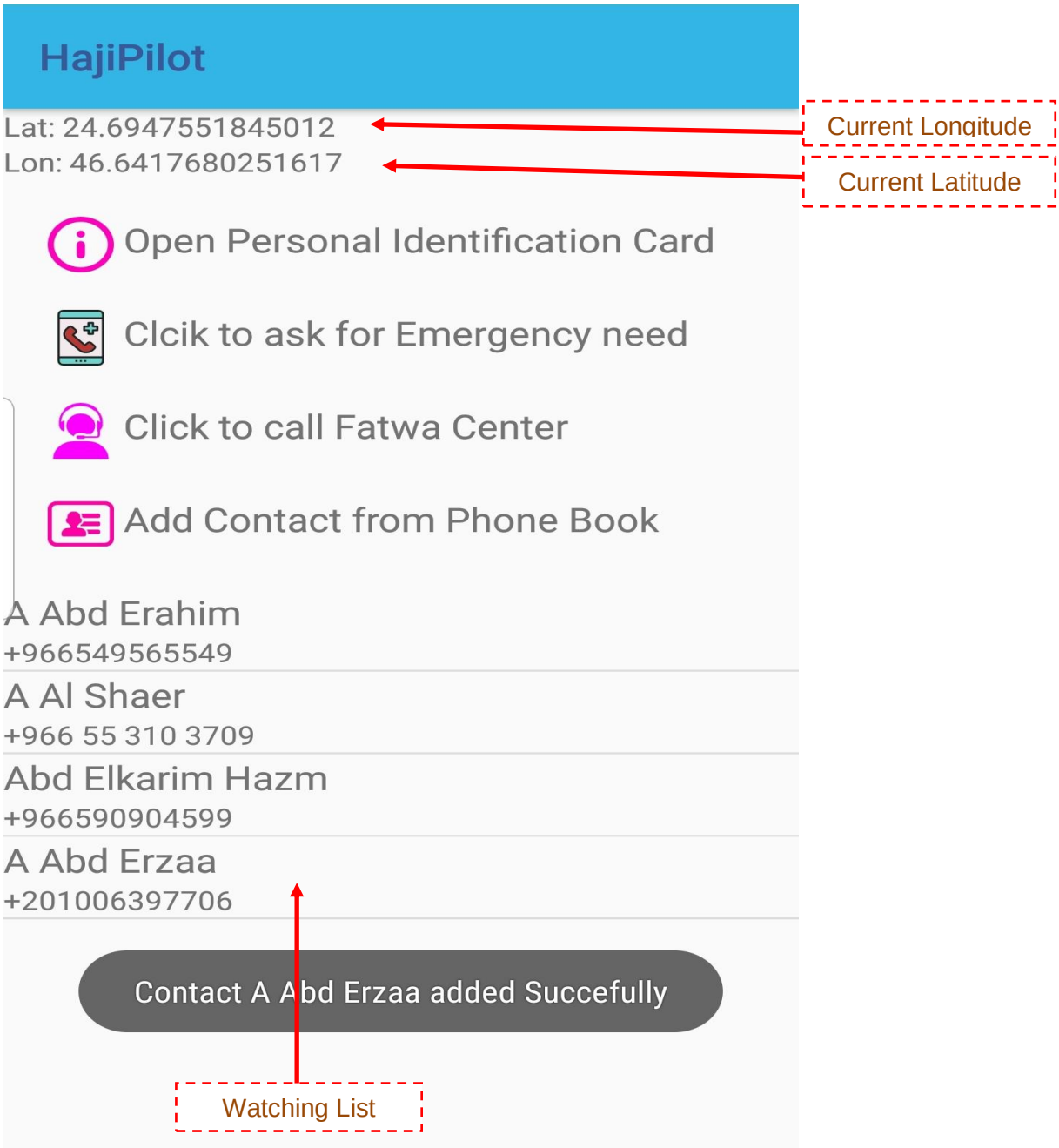

Figure 11: Current pilgrim location

List containing on selected contacts choose to monitor and following their movement among Hajj sites, arranged by their contact names on your phone, to adding new contact clicking contact button that will open phone address book choosing target contact will be listed on watching list to asking his current location at any time no need to getting or searching at all phone contacts.

Adding preferred contacts to following by clicking contacts button that will display phone address book selecting contact appending to watching list by name and phone number, current latitude and longitude displayed at the upper side of the screen if location service enabled that will be sent to location requester automatically by massage even if HajiPilot running at android background

3.3.3 Fatwa Call

Know also by Fatwa on Phone Service it is covered by highly qualified muftis and scholars answer queries via the toll-free phone no. Delivered through the most sophisticated communication tools all of the Fatwas are classified and archived, Pilgrim may need for asking some queries related to Islam or Hajj duties so theses service provided for 24 hours a day to provide accredited answers and helping for carrying out Hajj duties no mistakes.

3.3.4 Emergency Call

It is the core player and case study that I investigated in here and design the system to managing the recovery process starting from emergency call receiving, monitoring call's life cycle passing receiving calls, location identifying, real-time map displaying for mobile devices and web mapping explorer, real-time assigning, recovery, monitoring, and closing.

3.3.5 Pilgrim's Information

HajiPilot provides Pilgrim's information retrieving by Visa number for each pilgrim that will give medical teams a short brief about the caller that will help to have an idea about the situation, on the other side it is considered as an identity card for pilgrim if lost the way for staying or forget staying information.

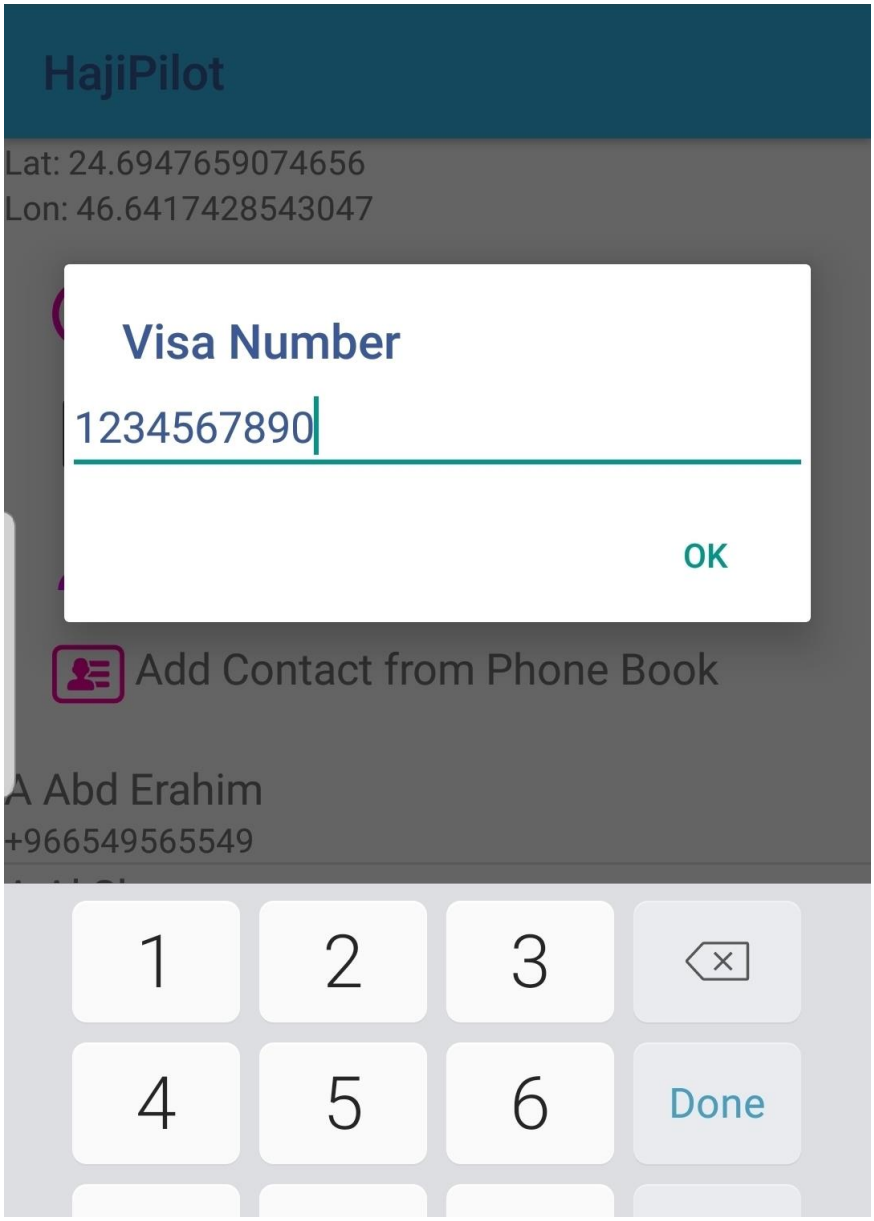

Figure 12: Retrieve information by Visa Number

Pilgrim information screen giving individual information about staying places on all over Hajj sites if lost anywhere or forget the name of hotels or staying places anyone can guide Pilgrim to his staying easily depending on displayed information on this screen may be acting as an identification card.

Figure 13: Pilgrim Staying Information Screen

3.4 HajiRelief Screens

3.4.1 Login Screen

It is a popular screen to most applications and websites that are required to get authorize permission for its private services, HajiRelief application will use only by medical teams at Hajj Management System.

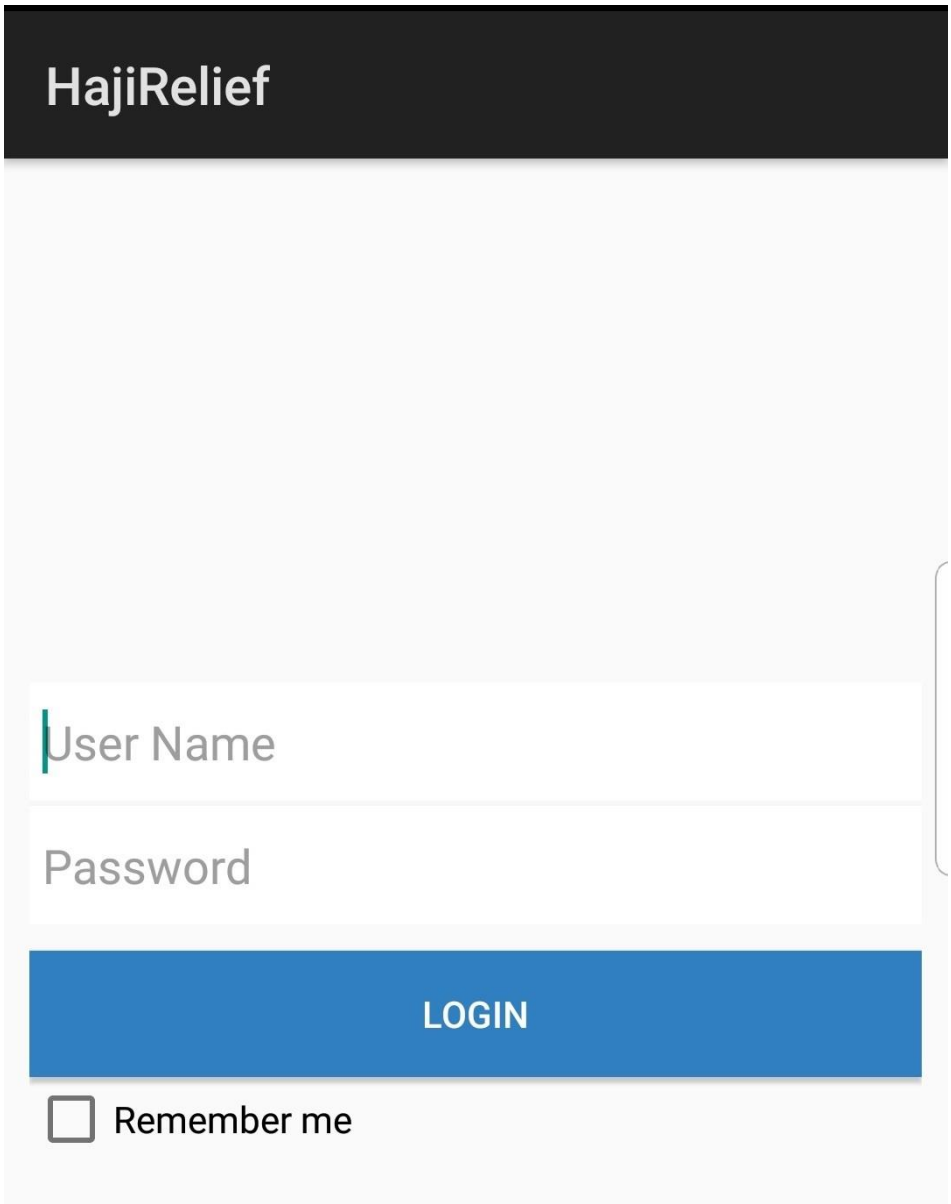

Figure 14: HajiRelief Login screen

Everyone in medical teams will have single user name and password that will be used for logging to HajiRelief application to access application services and functionalities, these users created and stored at database hosted on Godaddy, User credential will be saved permanently to android if Remember me checked and login screen will not appear each time application opened.

3.4.2 Emergency calls' Map

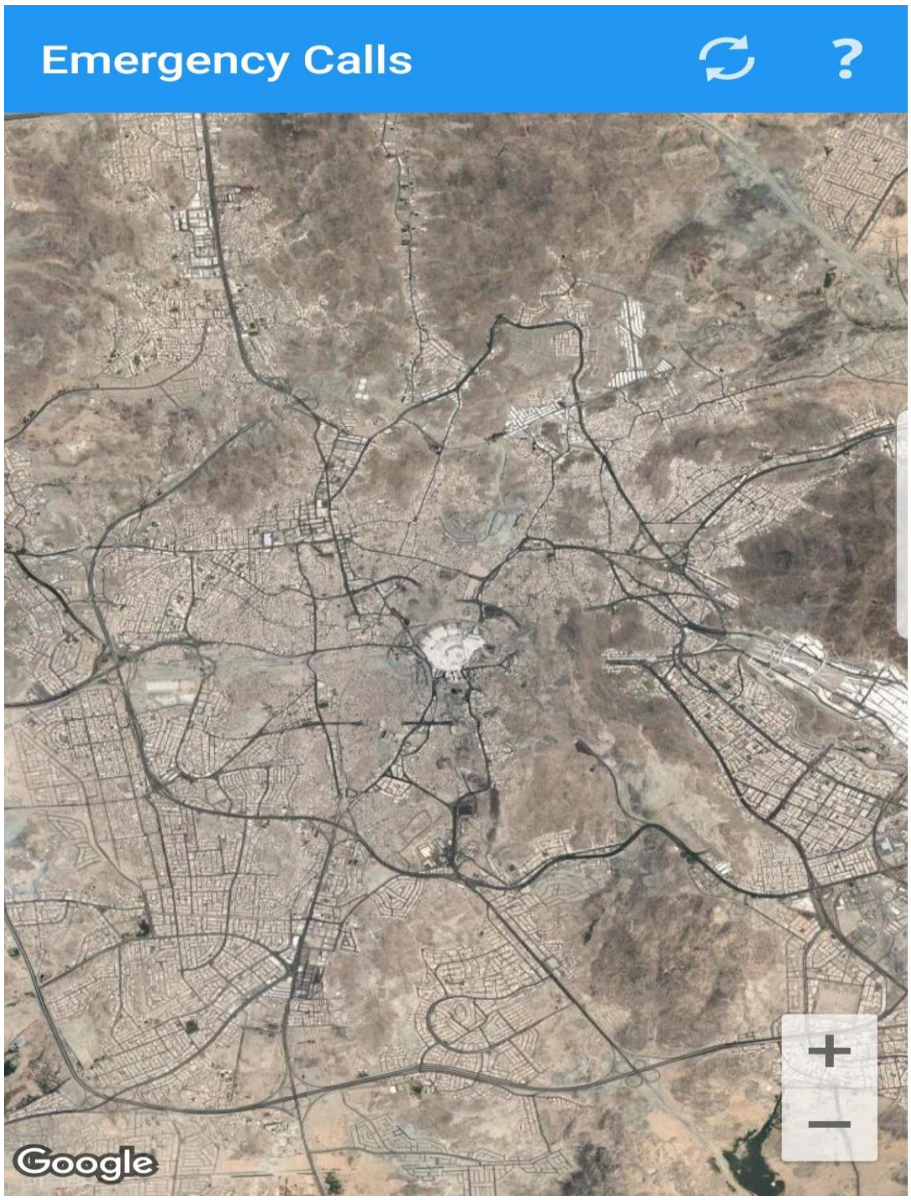

Figure 15: HajiRelief Main Screen

Emergency calls map is the main screen for HajiRelief completely full mobile screen with application map that displays new emergency calls still not assigned and served by others, when medical team member decided to serve and assigned to himself by selecting call's marker identified by call id and located on the map, Real-time update to new call to be in progress and removed from all other HajiRelief mobile application

 This icon opens new calls list from the right side of the screen order by call time and displaying call's id

 Reset map displaying area to start extent and refresh features displaying on the map

Swapping map screen from the left side will open received calls list that employee chooses to serve and assigning to him.

3.4.3 New emergency calls list

New emergency calls listed at right side ordered list that hidden and show by touching the screen by user swapping from right or clicking question mark icon at upper right corner, clicking list items will be panning and zooming to clicked call's location that displayed on the map by a marker, Navigate to any call location by clicking it from the list that will panning map to focus on this call location, clicking this calls to display its information at Map Info popup

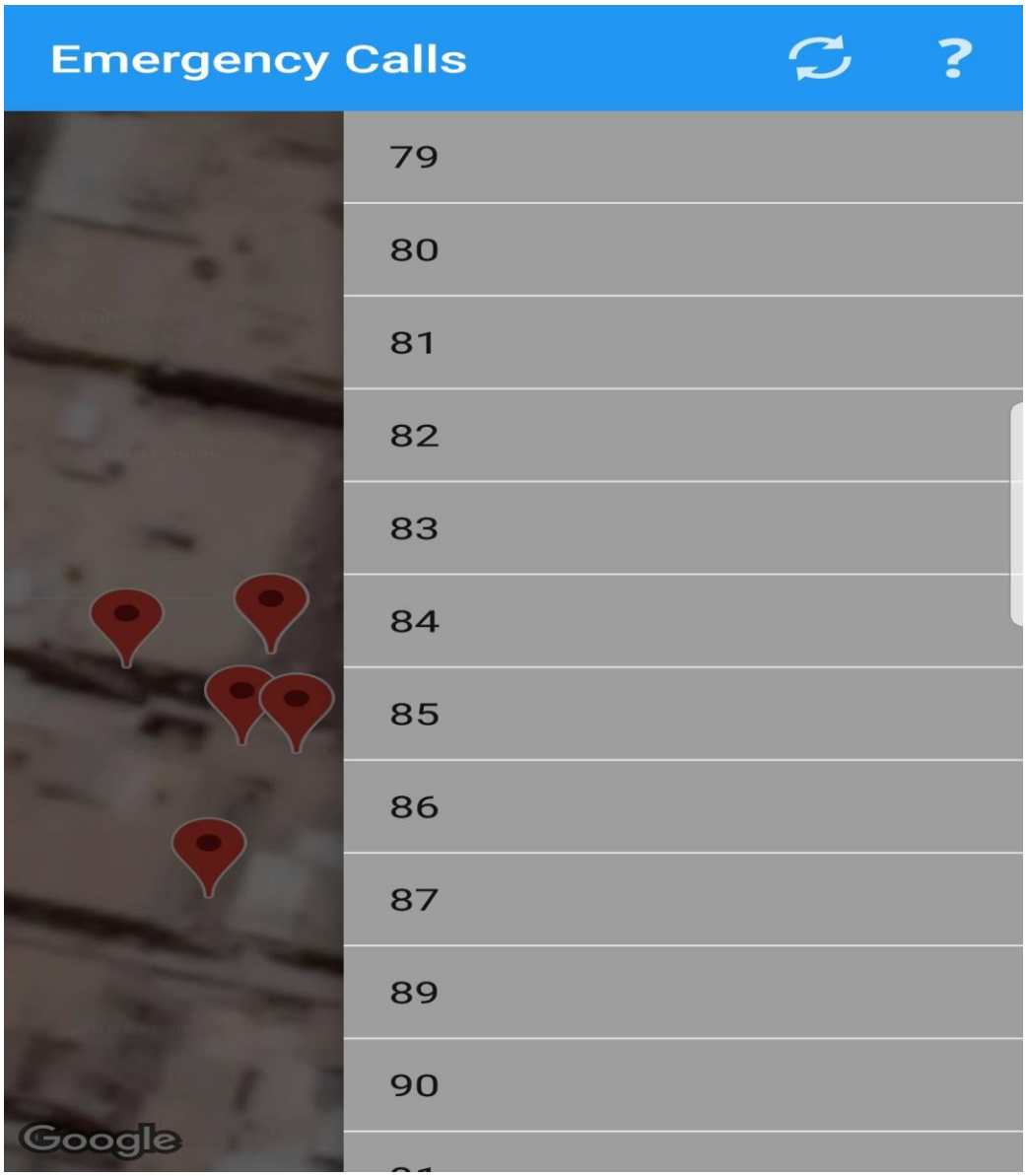

Figure 16: HajiRelief new calls list

Swap from the left side screen to open assigned tasks list and double click to close and finish this task

3.5 Emergency Calls Mapping Explorer

Reaching to last station of our story emergency calls map explorer and also acting as a middle tier for designed mobile applications to contacting the service by data in real-time

modifications, web service described before so interesting here map explorer that built on open source OpenLayers JavaScript APIs and Open Map Street data that is available as free and open source mapping data resource.

Home page designed to give a brief description about the system aims and components, Hajj duties, and resources prepared yearly to cover all Hajj's sites, First section talking about Masjid Al-Ahram and picture to Ka'ba during people surrounding it to do tawaaf followed by a section discussing emergency calls identification.

Figure 17: Hajj Emergency Management System Homepage

HajiPilot section describing first mobile application that available for Pilgrims to using on their phones and give a simple explanation for usability, the other mobile application HajiRelief also described at the next section and giving small brief about its functionalities also add another section for emergency call map explorer talking about it features and finally relating to important of Arafat day I preferred to write short words about it.

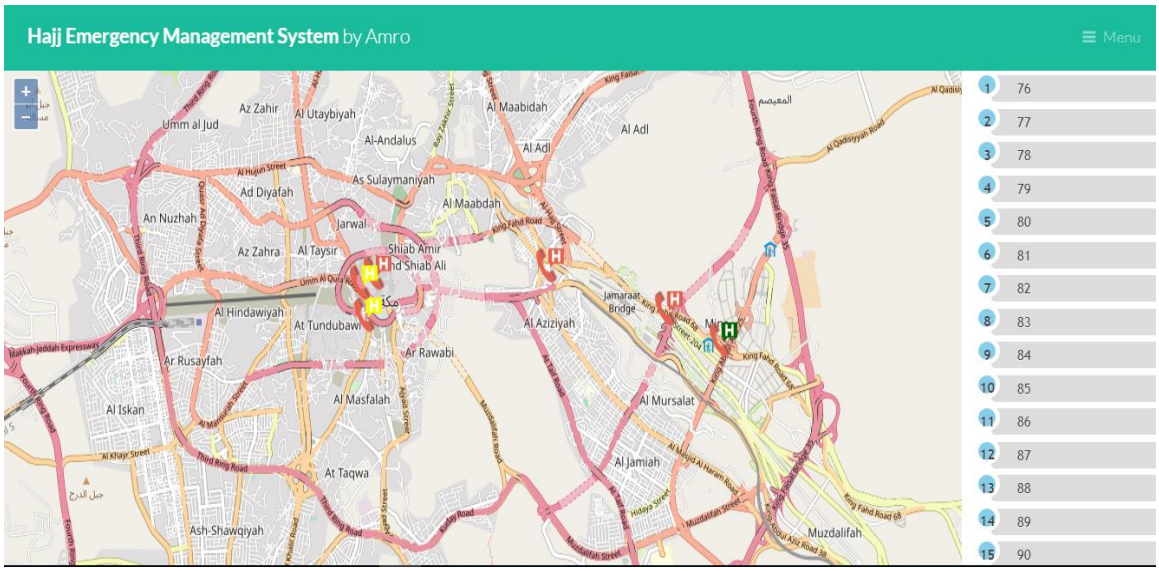

Figure 18: Hajj Emergency Wepmap Explorer

The second page designed to display full map screen displaying emergency calls located by their locations and three types of symbology depends on on-call status

Table 17: Call Status

Symbol	Meaning
	A new emergency call received from Pilgrim
	In-Process call while teams moving and dealing with the situation
	Served and covered case that no need for more actions

Emergency call explorer and SQL Server database hosted by Godaddy on Microsoft Windows Server running Microsoft Internet Information Services (IIS) including Web APIs Service that will transfer data between mobile applications and the system.

3.6 How to test the system

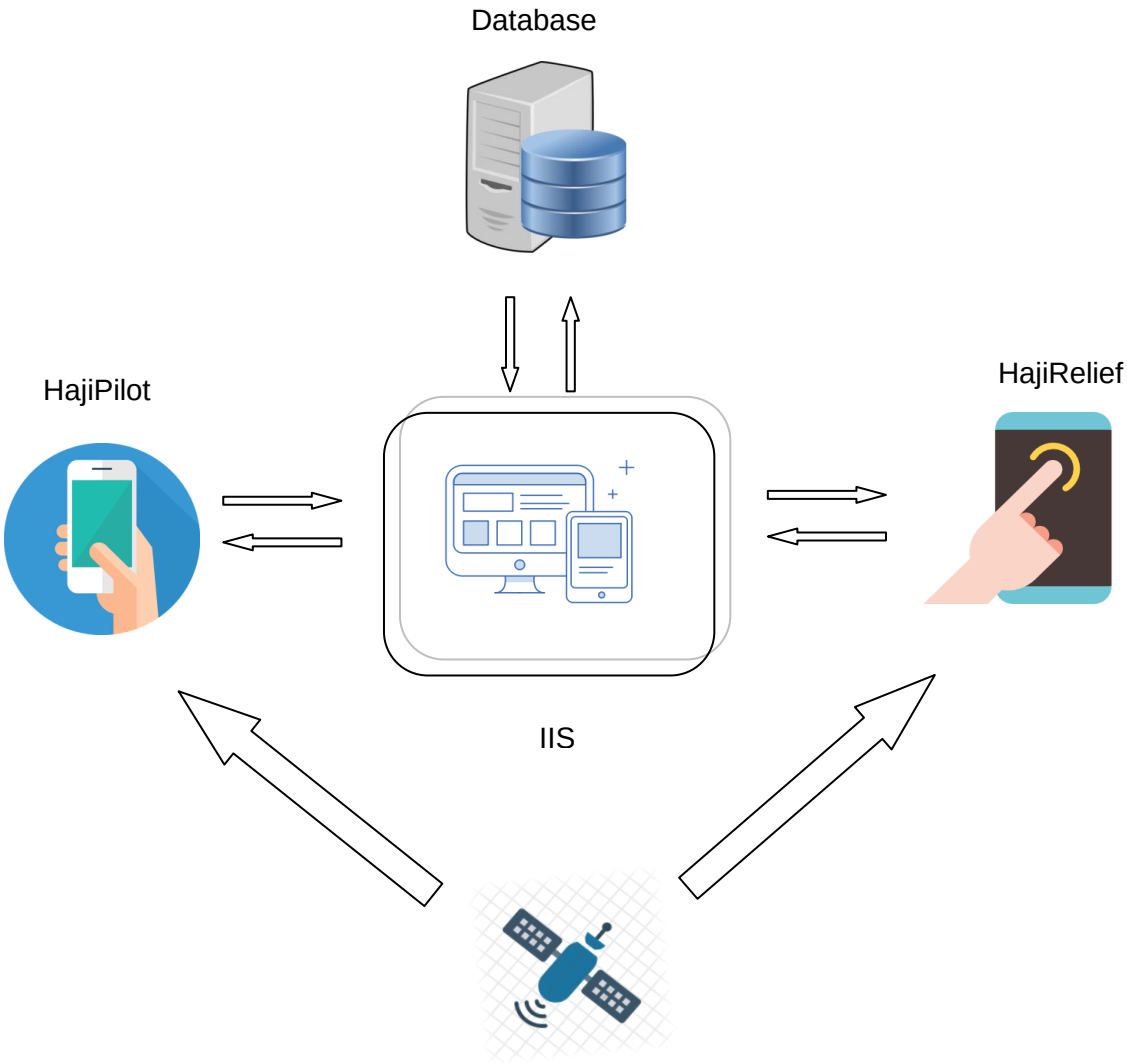

Figure 19: Hajj management System Architecture

We have three application developed to achieve our goal that should be tested as it is the last step of software developing to accredit stability and functionality covered; we will have a brief description about implementation structure that will cover required infrastructure meet requirements of stability and high performance

1 Install HajiPilot at tow mobiles

2 Open and register mobile number for each one

3 Use one as Pilgrim and enabling location service from mobile settings and have enough credit at least for one SMS

4 Use the other one as a family member to send location request but first add the other contact number from phone addresses to HajiPilot watching contact list

5 Click this contact at watching contact list will send location request message received by Pilgrim mobile

6 HajiPilot application at Pilgrim mobile has the permission to Access messages licensing and filter these message if matching filter properties that mark HajiPilot request auto-response message sent containing on current Pilgrim's location link even if the application at Android background

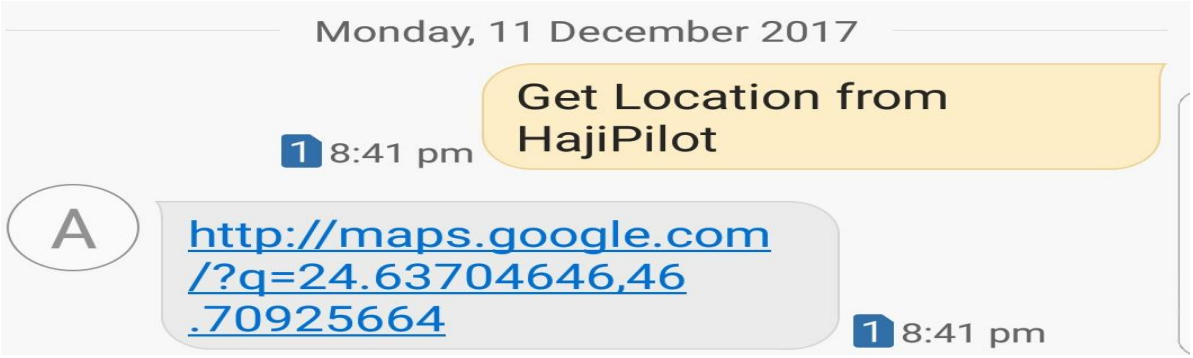

Figure 20: Response location Message

7 To ask for something at Hajj duties there are accredit center for response to any question clicking Fatwa button will open phone dial by central fatwa phone number ready for a call

3.6.1 Emergency Call

The three application will be required here to observing call during its process cycle so tow mobiles will be used one for HajiPilot and the other will be for HajiRelief applications, emergency calls map explorer published and working on public domain name HajiPilot.com no need to implementation to any device.

HajiPilot installed and register as a pilgrim by Visa Number on the other device installed HajiRelief use employee user name and password to log in there are initial users provided as the following employee users can use anyone to log in.

Table 18: HajiRelief Users

User Name	Password
User1	123456@Salz
User2	123456@UNIGIS
TestUser1	Passw0rd2244

Three applications should be opened at the same time to observe the emergency call process cycle and calls' status changes start by new, change to in-progress finally closed

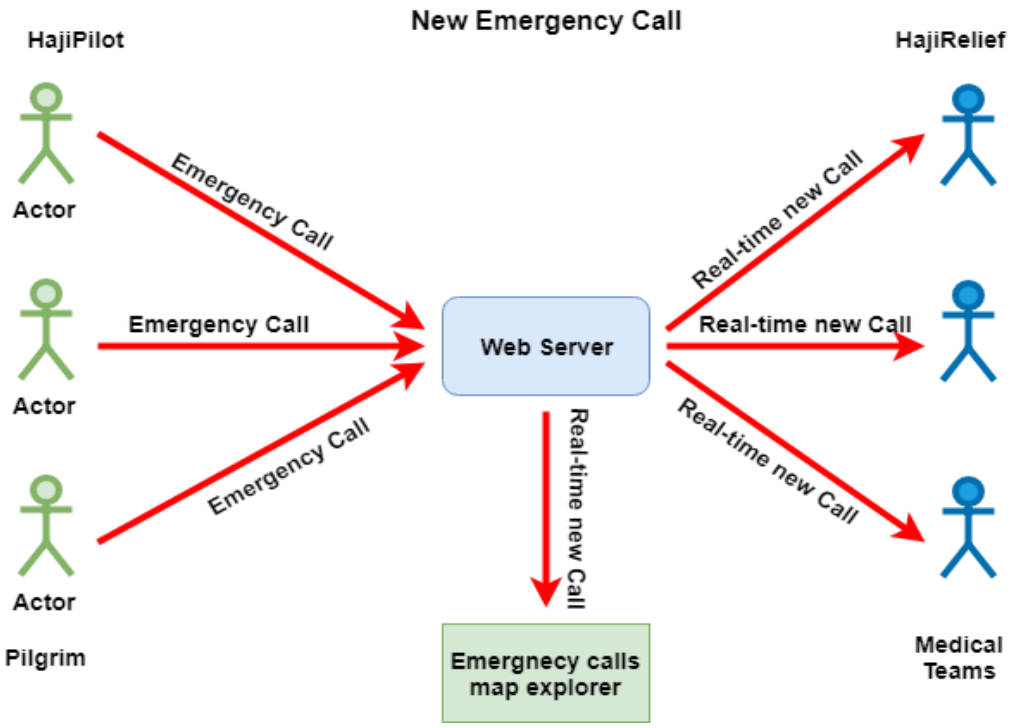

Figure 21: New Emergency Call Cycle

Staring by new emergency call will received by the server and send in the real-time channel to all HajiRelief applications with medical teams appended to application map in red symbology located by call's location and also added at the end of the list of new tasks.

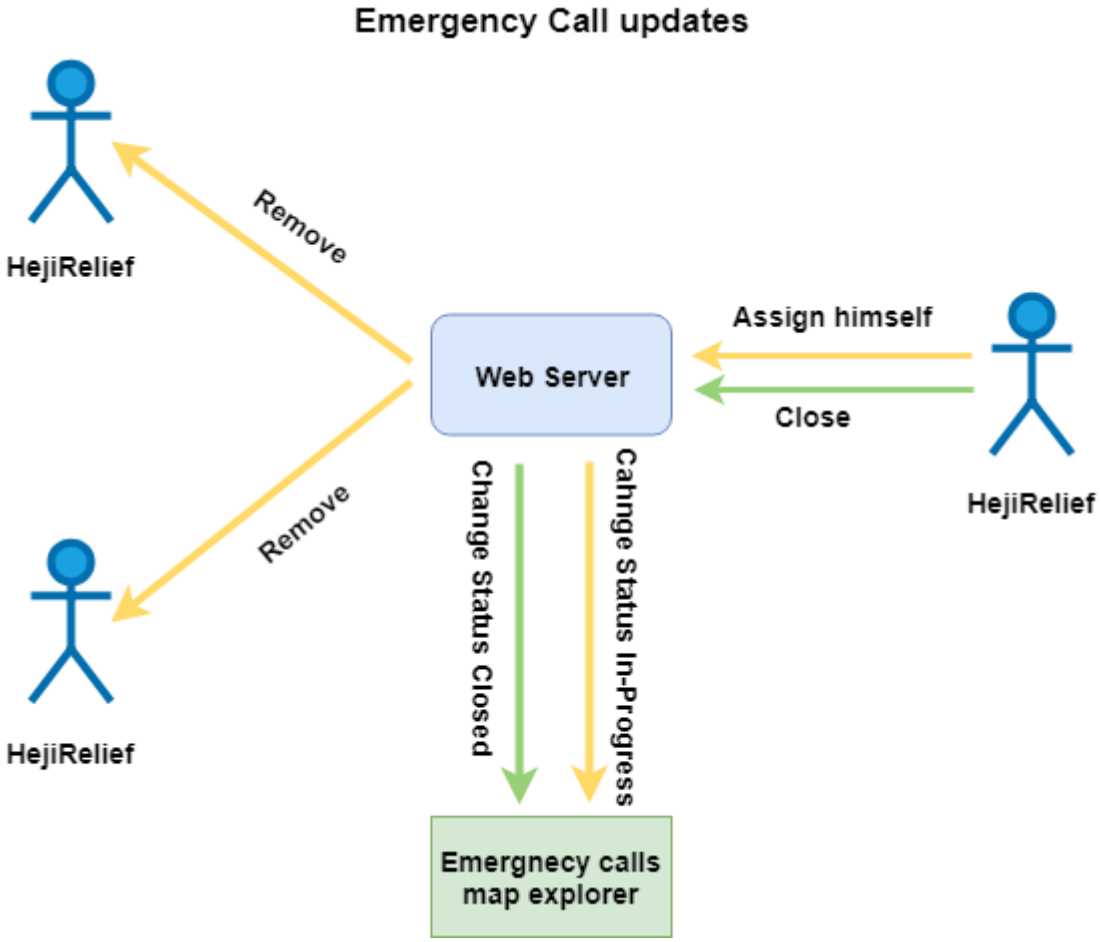

Figure 22: Emergency Call Update Process

Emergency calls map explorer displaying new task in red symbology color received from Pilgrim by HajiPilot application, Medical teams using HajiRelief invoked by this new real-time push by new calls at HajiRelief map and tasks list, nearest team supposed to take this call and assigning himself by selecting call on map info popup will appeared select assign and confirm that is will move task from right new tasks list to in progress tasks at left tasks list.

The server will receive notification about this task assigned for a team and invoke to change its status to be In-progress identifying by yellow symbology on calls map explorer at the same moment should be removed from other HajiRelief applications to giving teams chance to serving other emergency calls.

Task received now and Google maps direction service may be used to arrive in the shortest time closing this task now will be the last step at treatment processes closing task will be from received tasks at the left list double click task and confirm that will remove this task from received tasks and map invoking the server to real-time update this call status on emergency calls map to closed in green symbology.

Figure 23: Map Navigation Menu

Emergency Map explorer published and has domain name HajiPilot so can be browsed at www.HajiPilot.com enabled SSL certification contains on two pages default talking about Hajj and developed the application and the other is the emergency calls map explorer you can open from the menu on the right and select map item.

Chapter-4: Results

4.1 Emergency call identified by location

Emergency calls' coordinates located using American Global Positioning System GPS that has become used by most location services at most mobile operating systems that received by the system and using maps to locate each call to help head management supervising the process and medical teams improving time response for any call.

4.2 Pilgrim's identification card

Avoiding pilgrims lost by displaying identity screen at HajiPilot application for his personal and staying information at different Hajj' sites that can be used anytime pilgrim forget his staying information while most of pilgrims elder that needs to private care any may suffer from memorization problems to remember or save personal information HajiPilot also providing the ability for calling Fatwa center no need know its number.

4.3 Pilgrim tracking by friend and family

Families and friend now have the ability to know pilgrim's location at any time no worry or fear use HajiPilot application receiving auto response to current coordinates for Pilgrim to be assured at safe and planned hajj site no need to internet connection for send and response location request while application send location request SMS and auto responses from pilgrim's mobile received containing location link opened by Google maps.

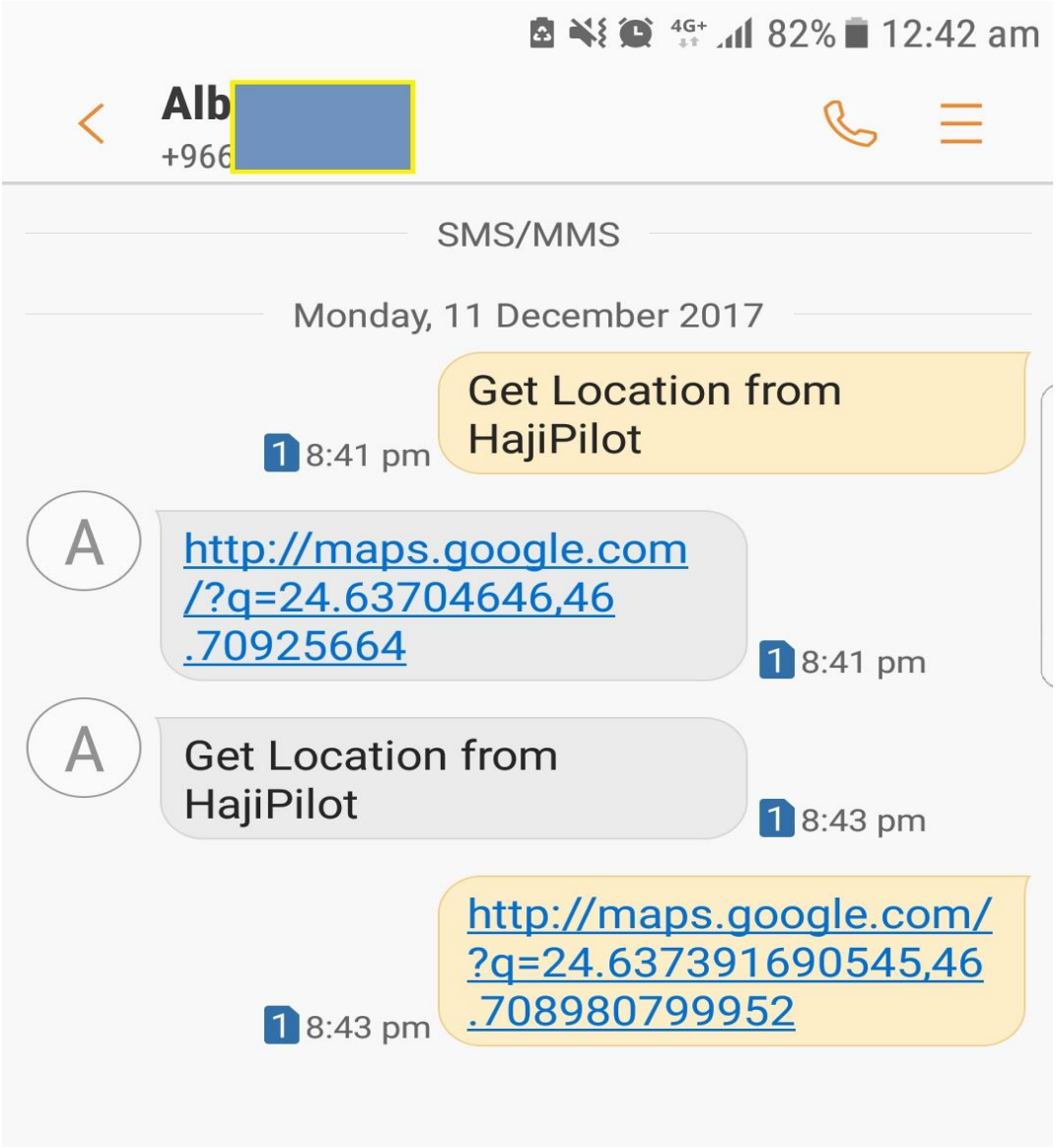

Figure 24: Auto Location Message

4.4 Location request auto SMS location response

By clicking contact from watching list message send automatically, HajiPilot detect this message and auto response message containing on clickable location message opened using any mapping application specially Google maps that is already installed by default to Android systems displaying target location by Mark, Google maps or any other application can be routing to guide you for pilgrim's location.

4.5 Mobile mapping for emergency calls relief

No doubt about big effective role for technology using at our life today and success story for most organizations depends level of technologies have and relating to critical needs at emergency process covering at Hajj to nearly two millions of pilgrims from all over the world traditional way of emergency process resulting fail to save and meet all emergency cases at short time.

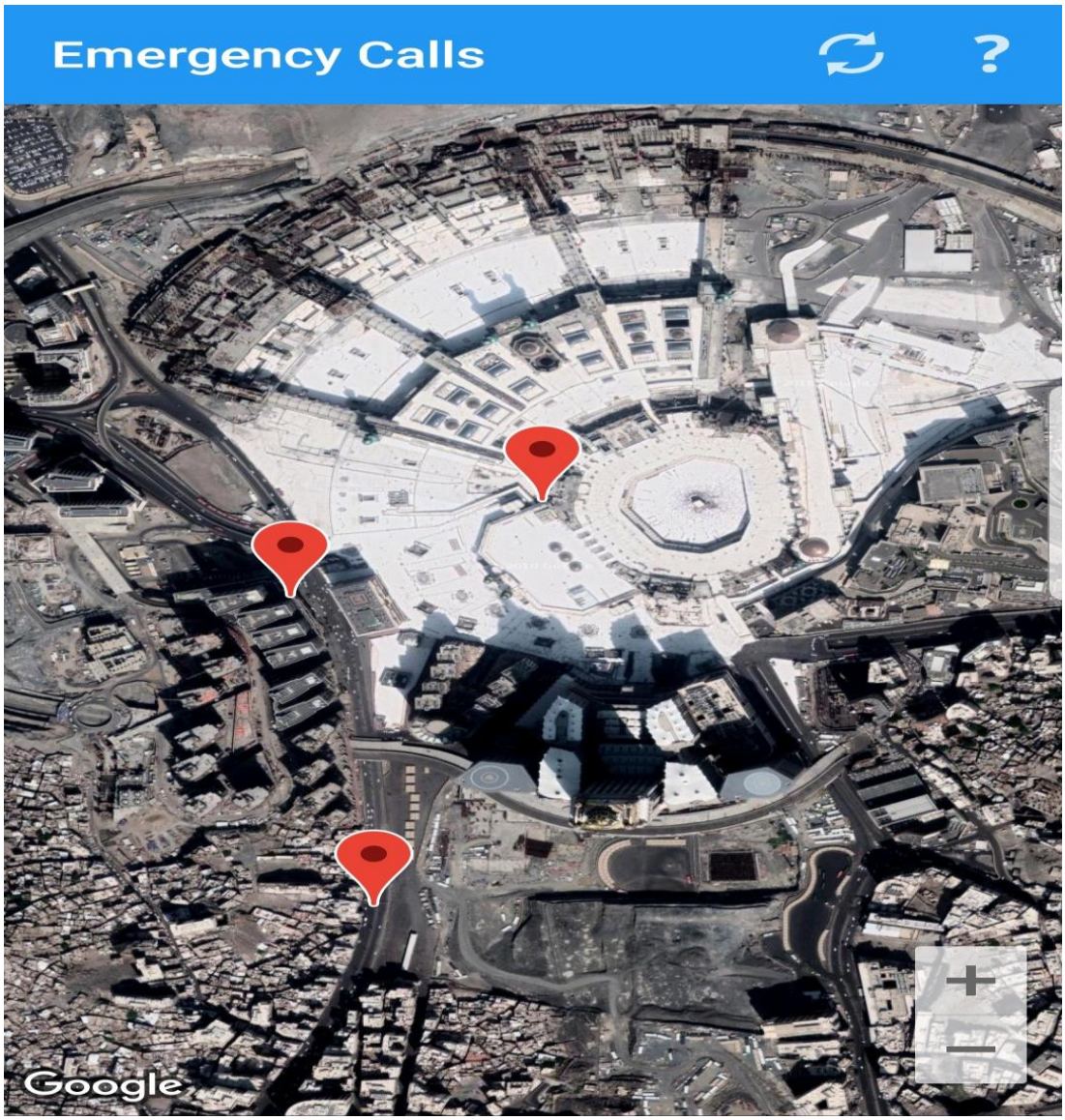

Figure 25: New Emergency Calls at the map of HajiRelief

HajiRelief automat process or emergency call treatment starting from receiving call, treatment and finally closing it enabling teams for routing and navigate to call's location depending Google maps direction service, emergency call status has real-time updates once this call assigned for team and also closing it will hide from all client mobiles to focus only on non-served calls.

4.6 Real-time emergency call receiving

As we dealing here with people lives time have a great value any response delay may causes high cost that is mean lost some pilgrim's lives so it's necessary to have real-time calls receiving and manipulations that are achieving by integrating SignalR component to the system receiving calls at the server pushing at the same time to all clients that appeared on emergency calls to map and also HajiRelief application map.

4.7 Monitoring emergency calls on Web mapping

Using GIS mapping to displaying emergency call on the map located by call's location identified by colored marker present each call status to help observers at the managing centers of Hajj graphical way following recovery processes on the land that will motivate teams to do the best to recovery cases at shortest time, also management can redistribute teams and assets relating to hot points on the map.

Storing emergency call time and tracking status changes timeline ended by closing the task will use for analyzing process performance and teams' attitude leads to planning for improving serving time, work strategies and teams' skills that will contribute our main saving pilgrim lives, avoiding crises happening or decreasing its bad effects.

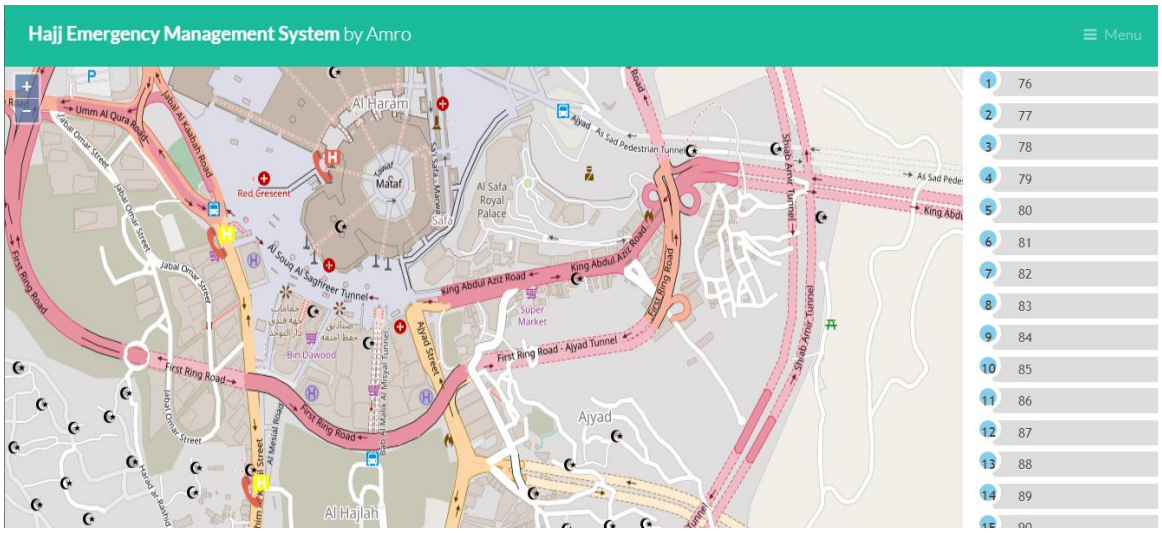

Figure 26: Hajj Emergency Calls Explorer

Colored marker on the map will distinguish among new calls (colored by red), at serving calls (colored by yellow), and completed tasks (colored by green), visualizing calls on the map and locating by calls' location will present a whole look at all operation by in attractive and understood way for anyone browsing the system from anywhere at the world that's mean all government organizations concerned about Hajj can follow the operations at real-time emergency calls mapping from any computer, tabs and also mobile while portal is responsive designed as we can see at the following screen.

Depending on Geographic information system to receiving and locating emergency calls that will give the ability to use GIS mapping functionality such as zooming in, zooming out, panning to browsing landmarks or emergency calls on map avoiding time wasting to catch the right direction and location.

Now we have full automation Hajj emergency management system depending on Geographic information system starting by sending emergency call, location request from friend and families, real-time emergency call receiving, real-time pushing for teams' mobile, real-time call status updates، pushing new calls for all client web mapping explorer

at the same time no refresh or updates needed and ordered list of calls appeared on the left side.

High-performance attitude resulting to using latest technologies to improving government process facing any emergencies that are achieved by developing mobile applications that will be available for android mobiles that are connected to the server to transfer emergency calls real-time updates to step on recovery situation and also following to assure no emergency cases drop.

4.8 Map analysis

As we know photo and visualization can brief and save signs about thousands of lines and thinking especially for critical situations that are related to people lives so displaying calls located at map will give the management a wide and clear view for all process enabling to ask for more resources for some regions to cover hot points, reports and analysis base on changes on the map will give the ability to preplan the strategies in the future and also gives scientific statistics and screens for efforts and covering tasks.

Chapter-5: Conclusions

An average of three millions of people safety during Hajj is a huge and the primary responsibility for all Saudi governments' organization in the kingdom, there are some critical situations that may happen result of any crises anywhere at Hajj sites poor planning and recovery strategy will increase the size of human losses, Haji Management System aims to put advanced and new methodology as web and mobile applications fast and accrued procedures taken for recovery steps.

Analysis Haji Management system requirements and moving to smartphones and GIS Web mapping to encouraging General Management of Hajj and Umrah Affairs to using developed applications control to receiving emergency needs and recovery process at all Hajj sites depending on maps and location service to best usage of available resources and assets by relocating and redistributing as needed relating to rushes and emergency situations.

Reviewing some similar available systems and applications related to emergency control and tracking family mobile applications to step on provided functionalities I did not find any system all planned functionalities as one system or application for example Smart Hajj mobile application that focus on just religion sides for Hajj, Family Locator mobile applications focused on tracking members and sending receiving messages so it is pretty to merge different functionalities at one system designed for Pilgrim needs covering integrating with GIS web mapping for locating and displaying emergency calls.

Users and functionalities required to developing applications covering emergency life cycle identified as

1. HajiPilot Mobile Android application for Pilgrims, family, and friends

2. HajiRelief Mobile Android applications for medical teams
3. Web mapping application for the head department and Monitoring centers

Tools required for implementation planned applications installed depending on one Database, OpenLayers, and Open Street Maps and also Google Maps APIs used
The database, emergency calls map explorer and WebAPI service hosted by Godaddy also domain HajiPilot.com bought enabling SSL Certificate for more secure data exchange.

Tow Android mobile applications HajiPilot and HajiRelief installation APKs released and signed by Microsoft visual studio ready to install but there are some concerns relating to interface design while I poor design skills for designing icons and also native applications mostly have simple interface, no software free bugs it is known phrase at software developing field so maybe there are some scenarios not tested at the system and also its supposed to use SMS bulk for registration and verify mobile numbers registered but keeping cost while domain and host cost much money for me.

Google Maps used at HajiRelief mobile application to allocating emergency calls and guiding medical teams by routing، HajiPilot depends on background Google play location service that updates pilgrim's location information and auto-response by location link for families or friends and finally emergency calls explorer using free open source OpenStreetMap and OpenLayers JavaScript APIs exploring emergency calls by different marker represents an emergency call's status and its status changes at Real-Time by using Microsoft ASP.Net MVC SignalR tool.

Additional improving in the future for functionalities and interfaces will give high rank for the system and increasing number of users besides public DNS (Domain Name Service)

availability, Social integration services will support the system functionalities to get all Pilgrims needs at one place, there is a plan to develop another application for HajiPilot compatible with smart swatches that have become popular for user and small size still I did not investigate at this direction yet but this will be great value for Pilgrims that are busy among rush Hajj Sites in continues move.

References

Ahmed, Q.A., Arabi, Y.M. and Memish, Z.A., 2006. *Health risks at the Hajj*. *The Lancet*, *367*(9515), pp.1008-1015.

Amro, A. & Nijem, Q. 2012. *Pilgrims"Hajj" Tracking System (e-Mutawwif)*. *Contemporary Engineering Sciences,* 5**,** 437-446.

Android Developers 2011, *what is an android*? Available from: https://developer.android.com/about/index.html. [16 February 2019].

Arab News 2018. *Smooth movement of pilgrims from Makkah to Mina*. Available from: http://www.arabnews.com/node/1359046/saudi-arabia. [13 February 2019].

BOIN, A. & MCCONNELL, A. 2007. *Preparing for critical infrastructure breakdowns*: the limits of crisis management and the need for resilience. *Journal of Contingencies and Crisis Management,* 15**,** 50-59.

BRANCH, C. N., NICHOLS, M. E. & JANKY, J. M. 1998. *Integrated mobile GIS/GPS/AVL with wireless messaging capability*. Google Patents.

CHARLAND, A. & LEROUX, B. 2011. *Mobile application development*: web vs. native. *Communications of the ACM,* 54**,** 49-53.

DJUKNIC, G. M. & RICHTON, R. E. 2001. Geolocation and assisted GPS. *Computer,* 34**,** 123-125.

General Authority for Statistics 2016, *Hajj Statistics*. Available from: https://www.stats.gov.sa/sites/default/files/hajj_1437_en_0.pdf. [10 February 2019].

General Authority for Statistics 2018, *Hajj Statistics* . Available from: https://www.stats.gov.sa/sites/default/files/hajj_1439_en.pdf. [10 February 2019].

GUNES, A. E. & KOVEL, J. P. 2000. *Using GIS in emergency management operations*. *Journal of Urban Planning and Development,* 126**,** 136-149.

HADDOW, G., BULLOCK, J. & COPPOLA, D. P. 2017. *Introduction to Emergency Management*, Elsevier Science.

HEALTH, M. O. 2017, *Health services provided for Pilgrims*. Available from: https://www.facebook.com/SaudiMOH/photos/a.195136303888893.45096.142409672494890/1537135736355603/?type=3&theater. [12 February 2019].

JOHNSON, R. 2000. *GIS technology for disasters and emergency management*. *An ESRI white paper*.

HUDA 2018. *Hajj pilgrimage statistics*. Available from: https://www.thoughtco.com/hajj-by-the-numbers-2004319. [9 February 2019].

MISRA, P. & ENGE, P. 2006. *Global Positioning System*: signals, measurements and performance second edition. *Massachusetts: Ganga-Jamuna Press*.

MODSCHING, M., KRAMER, R. & TEN HAGEN, K. Field trial on GPS Accuracy in a medium size city: The influence of built-up. 3rd workshop on positioning, navigation and communication, 2006. 209-218.

Nicole L., Teri M., Amy P., Francis C. & Thomas R Frieden. 2013. Research as a Part of Public Health Emergency Response. *The New England journal of medicine,* 368 13**,** 1251-5.

RODGER, J. 2017. *When is Hajj in 2017 - and what is the Muslim pilgrimage*? Coventry Telegraph, 09:10, 29 JUN 2017.

RODRIGUEZ, A. 2008. *Restful web services*: The basics. *IBM developerWorks,* 33

ROGERS, R., LOMBARDO, J., MEDNIEKS, Z. & MEIKE, B. 2009. *Android Application Development: Programming with the Google SDK*, O'Reilly Media, Inc.

Saudi Arabian Scout Assoiciation 2018. *Guidance maps.* Available from: http://www.scouts.org.sa/haj39/arafat.htm. [15 February 2019].

SOLUTIONS, H. 2017. *Hajj and Umrah Tips*. Available from: https://www.hajjsolutions.com/hajj-and-umrah-tips/#geoandtrans. [13 February 2019].

Zandbergen, P., & Barbeau, S. 2011. Positional Accuracy of Assisted GPS Data from High-Sensitivity GPS-enabled Mobile Phones. *Journal of Navigation, 64*(3), 381-399. doi:10.1017/S0373463311000051