

AUTO AMBULANCE REDEPLOYMENT IN USER'S DESTINATION

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Science degree in Computer Science

by

K.AKSHAYA (39290005)
S.MALINI (39290061)



DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **K.Akshaya (39290005)**, **S.Malini (39290061)**, who carried out the project entitled “ **AUTO AMBULANCE REDEPLOYMENT IN USER’S DESTINATION** ” under my supervision from November 2021 to May 2022.

Internal Guide

Dr.M.SELVI, M.E.,Ph.D.,

Head of the department

Dr.L.LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, **S.Malini (39290061)** hereby declare that the Project Report entitled **AUTO AMBULANCE REDEPLOYMENT IN USER'S DESTINATION** was done by me

Under the guidance of **Dr.M.SELVI, M.E.,Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI, M.E., Ph.D.**, and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, Head of the Department, Department of Computer Science for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M. SELVI, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science who were helpful in many ways for the completion of the project.

ABSTRACT

Emergency Medical Services (EMS) are of great importance to saving people's lives from emergent accidents and diseases by efficiently picking up patients using ambulances. The transporting capability of an EMS system significantly depends on the real-time redeployment strategy of ambulances. That is, which station should an ambulance be redeployed to, after it becomes available. However, it is a challenging task concerning with the multiple data. To this end, in this paper, we propose a data-driven real-time ambulance redeployment approach that redeploys an ambulance to a proper station after it becomes available, so as to optimize the transporting capability of an EMS system, considering the aforementioned multiple data. Specifically, the proposed approach is comprised of two stages to well considered. First, we propose a method, a safety time-based urgency index to incorporate into each ambulance station's urgency degree. Second, we propose an optimal matching algorithm to combine into the redeployment of the current available ambulance. Experimental results using data collected in real world demonstrate the significant advantages of our approach over many baselines. Comparing with baselines, our approach can save 4 minutes (35%) of the average pickup time for each patient, improve the ratio of patients picked up within 10 minutes from 0.684 and 0.803 (17%), and largely enhance the survival rate of patients (12% for patients in category A1 and 17% for patients in A2).

CONTENTS

CHAPTER No	TITLE	PAGE No
	ABSTRACT	V
	LIST OF FIGURES	VIII
	LIST OF ABBREVIATIONS	IX
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	5
	3.1 AIM OF THE PROJECT	5
	3.2 PROJECT SCOPE	5
	3.3 SYSTEM REQUIREMENTS	5
	3.3.1 HARDWARE REQUIREMENTS	5
	3.3.2 SOFTWARE REQUIREMENTS	6
	3.3.3 TECHNOLOGIES USED	6
	3.3.3.1 JAVA	7
	3.3.3.2 WORKING OF JAVA	7
	3.3.3.3 THE JAVA PLATFORM	8
	3.3.3.4 DATA MINING	9
	3.3.3.5 DATA MINING TECHNIQUES	9

4	EXPERIMENTAL OR MATERIAL METHODS	10
	4.1 DESIGN METHODOLOGY	10
	4.1.1 EXISTING SYSTEM	10
	4.1.2 PROPOSED SYSTEM	10
	4.2 MODULES	11
	4.2.1 MODULE EXPLANATION	11
	4.3 ARCHITECTURE DIAGRAM	12
	4.4 APPLICATION SCREENSHOTS	13
5	RESULTS AND DISCUSSION	16
6	SUMMARY AND CONCLUSIONS	17
	REFERENCES	19
	APPENDIX	21
	A.SOURCE CODE	21

LIST OF FIGURES

FIG NO	FIG NAME	PAGE NO
3.1	Java API	8
4.1	Architecture Diagram	12
4.2	Entry Page	13
4.3	Login Page	13
4.4	User's Profile page	14
4.5	Location page	14
4.6	Request Page	15

LIST OF ABBREVIATIONS

EMS	Emergency Medical Service
GIS	Geographic Information System
SMA	Spatial Moving Average
GPS	Global Positioning System
JVM	Java Virtual Machine

CHAPTER 1

INTRODUCTION

Emergency Medical Service (EMS) is branch of emergency services devoted to providing transport to care patients with illnesses and injuries. The main goal of the EMS to treat patients and arranging for timely removal of the patient to the next point of best cares usually from hospital to another place or vice versa. EMS operates primarily based on time. The capability of the Ems service is known by the average lift up time of the patients. There are many possibility that EMS cannot be not accessible by the patients because it went for another destination and not making the closest EMS to dispatch. The crucial moments in dispatching an ambulance correctly with the least response time in cases of trauma or other life threatening situations will contribute and determine patient survival. Our proposed model will used in order to avoid these problems.

CHAPTER 2

LITERATURE SURVEY

S.No	Year	Author Name	Project Title	Merits	Demerits
1.	2016	Albert ,Y.Chen,Tsung Yu Lu	Demand Forecast Using Data Analytics for the Pre allocation of Ambulances	A flexible model is implemented in GIS, through which training data are prepared with user-desired sizes for the spatial grid and discretized temporal steps.	Pre hospital emergency medical services(EMS) is to have a short response time.
2.	2015	Mr.Harshad.P.A, Mr.S.Saravanakumar, Dr.Rajalakshmi.S	Real Time City Scale Taxi Ride Sharing	The monetary constraints provide to incentives for both passengers and taxi drivers.	Real time taxi sharing has not been well explored.
3.	2013	Erik Jenelius, Haris N.Koutsopoulos	Travel Time Estimation For Urban Road Networks Using Low Frequency Probe Vehicle Data	The observation model presents a way to estimate the parameters of the network model, including the correlation structure, through low frequency sampling of vehicle traces.	The network model separates trip travel times into link travel times and intersection delays and allows correlation between travel times on different network links based on a spatial moving average (SMA) structure.

4.	2000	Michel Gendreau, Gilbert Laporte, Frédéric Semet	A Dynamic Model and Parallel Tabu Search Heuristic for Real Time Ambulance Relocation	The potential of such systems to reduce mortality is related to paramedics training and to the time needed by a paramedic team to arrive on scene.	The redeployment problem for a fleet of ambulances.
5.	2010	Verena Schmid and Karl F. Doerner	Ambulance Location and Relocation Problems with Time-dependent Travel Times	The problem is solved meta heuristically using variable neighborhood search.	Emergency service providers are facing the following problem: how and where to locate vehicles in order to cover potential future demand effectively.
6.	2017	C.J. Jagtenberg, S. Bhulai and R.D. van der Mei	Optimal Ambulance Dispatching	Dispatching the closest idle ambulance seems to be so natural that justification is not needed, but one should not overlook the possibility to change the dispatch policy in order to improve the objective.	The ambulance dispatch problem, in which one must decide which ambulance to send to an incident in real time.
7.	2015	C.J. Jagtenberg, S. Bhulai, R.D. van der Mei	An Efficient Heuristic for Real-Time Ambulance Redeployment	A polynomial-time heuristic that distinguishes itself by requiring neither assumptions on the region nor extensive state information.	The problem of dynamic ambulance repositioning, in which the goal is to minimize the expected fraction of late arrivals.

8.	2010	Jing Yuan, Yu Zheng, Chengyang Zhang	T-Drive: Driving Directions Based on Taxi Trajectories	Smart driving directions from the historical GPS trajectories of a large number of taxis, and provide a user with the practically fastest route to a given destination at a given departure time. Design a two-stage routing algorithm to compute the practically fastest route.	Experienced drivers but are too subtle and difficult to incorporate into existing routing engines.
9.	2008	Corrado de Fabritiis, Roberto Ragona, Gaetano Valenti	Traffic Estimation and Prediction Based On Real Time Floating Car Data	Gather accurate travel times/speeds in a road network and to improve short-term predictions of travel conditions.	The redeployment problem for a fleet of ambulances.
10.	2016	Bradford S. Westgate, Dawn B. Woodard	Large-Network Travel Time Distribution Estimation for Ambulances	It is important to capture the uncertainty in the travel time, by predicting the entire travel time distribution rather than just the expected travel time.	Allocation problem Redeployment problem

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 Aim of the project

The main aim of this project is to provide real time ambulance redeployment initiative to make the ambulance accessible to the appropriate station later it become available.

3.2 Project Scope

The Proposed System automatically assigning the ambulance to the requested user. This work proposed two methodologies for handling the multiple data request. A safety time-based urgency index and optimal matching algorithm First safety time-based urgency index organically merge a station's number of available ambulances number of nearby EMS requests in the future and geographical location into the urgency degree of this station Secondly an optimal matching algorithm, which can incorporate each station's urgency degree the travel time of the current available ambulance to reach each station and the status of other occupied ambulances into the redeployment of the current available ambulance when the request is made to the system and making the ambulance to reach the location in faster pace and saving time and also handling the multiple request at same time if the requested ambulance is not available at the station our system will redirect the request to the next nearest ambulance making it to reach the patient.

3.3 System Requirements

3.3.1 Hardware Requirements

- RAM : 6GB & Above
- Processor : i3 & Above

3.3.2 Software Requirements

- Windows 7 operating system and above
- JDK 1.7
- Tomcat7
- My-SQL 5
- Android Studio 3.4

3.3.3 Technologies used

- JAVA
- DATA MINING

3.3.3.1 JAVA

Java is an object- oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak tree outside Gosling's office), was intended to replace C++, although the Feature set better resembles that of Objective C.

3.3.3.2 Working of JAVA

For those who are new to object-oriented programming, the concept of a class Will be new to you. Simplistically, a class is the definition for a segment of code that can Contain both data (called attributes) and functions (called methods). when the Interpreter executes a class, it looks for a particular method by the name of main, which Will sound familiar to C programmers. The main method is passed as a parameter an Array of strings (similar to the argv [] of C, and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out**, Which is java's output stream. UNIX users will appreciate the theory behind such a Stream, as it is actually standard output. For those who are instead used to the Wintel Platform, it will write the string passed to it to the user's program. Java consists of two things :

- Programming language
- Platform

3.3.3.3 The JAVA Platform

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's software- only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system :

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

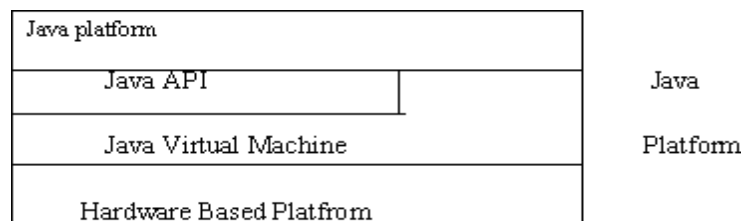


Fig 3.1 Java API

3.3.3.4 Data Mining

In general terms, “**Mining**” is the process of extraction of some valuable material from the earth e.g. coal mining, diamond mining etc. In the context of computer science, “Data Mining” refers to the extraction of useful information from a bulk of data or data warehouses. One can see that the term itself is a little bit confusing. In case of coal or diamond mining, the result of extraction process is coal or diamond. But in case of Data Mining, the result of extraction process is not data!! Instead, the result of data mining is the patterns and knowledge that we gain at the end of the extraction process. In that Sense, Data Mining is also known as Knowledge Discover.

Gregory Piatetsky-Shapiro coined the term “Knowledge Discovery in Databases” in 1989. However, the term ‘data mining’ became more popular in the business and press communities. Currently, Data Mining and Knowledge Discovery are used interchangeably.

Now a days, data mining is used in almost all the places where a large amount of data is stored and processed. For example, banks typically use ‘data mining’ to find out their Prospective customers who could be interested in credit cards, personal loans or insurances as well. Since banks have the transaction details and detailed profiles of their customers, they analyze all this data and try to find out patterns which help them predict that certain customers could be interested in personal loans etc.

3.3.3.5 Data Mining Techniques

1.Classification :

This analysis is used to retrieve important and relevant information about data, and metadata. This data mining method helps to classify data in different classes.

2.Clustering :

Clustering analysis is a data mining technique to identify data that are like each other. This process helps to understand the differences and similarities between the data.

3.Regression:

Regression analysis is the data mining method of identifying and analyzing the relationship between variables. It is used to identify the likelihood of a specific variable, given the presence of other variables.

4.Association Rules:

This data mining technique helps to find the association between two or more items. It discovers a hidden pattern in the data set.

5.Outer detection:

This type of data mining technique refers to observation of data items in the dataset which do not match an expected pattern or expected behavior. This technique can be used in a variety of domains, such as intrusion, detection, fraud or fault detection, etc. Outer detection is also called Outlier Analysis or Outlier mining.

6.Sequential Patterns:

This data mining technique helps to discover or identify similar patterns or trends in transaction data for certain period.

7.Prediction:

Prediction has used a combination of the other data mining techniques like trends, sequential patterns, clustering, classification, etc. It analyzes past events or instances in a right sequence for predicting a future event.

CHAPTER 4

EXPERIMENTAL OR MATERIAL METHODS

4.1 Design Methodology

4.1.1 Existing System

In Existing, EMS has become venues for initiating life-saving treatments prior to arrival at a health care. However many deaths happen around the cities because of EMS not reaching its location on time and deployed systems are not efficient in providing the nearest EMS to the patients and causing more and more time to reach the patient and it does not handle multiple request of the patients at time making patients to wait longer and risking their lives. The user communicates via phone and cannot provide the exact location of the patients and EMS cannot able to understand user location details. In case of patient emergency EMS moved to another location there is a chance that another patient can request the EMS in this scenario EMS is not available to the requested patient Every moment of delay in an emergency can significantly reduce a patient's chance of survival.

4.1.2 Proposed System

The Proposed System automatically assigning the ambulance to the requested user. This work proposed two methodologies for handling the multiple data request. A safety time-based urgency index and optimal matching algorithm First safety time - based urgency index organically merge a station's number of available ambulances number of nearby EMS requests in the future and geographical location into the urgency degree of this station Secondly an optimal matching algorithm, which can incorporate each station's urgency degree the travel time of the current

available ambulance to reach each station and the status of other occupied ambulances into the redeployment of the current available ambulance when the request is made to the system and making the ambulance to reach the location in faster pace and saving time and also handling the multiple request at same time if the requested ambulance is not available at the station our system will redirect the request to the next nearest ambulance making it to reach the patient.

4.2 Modules

- User registration
- Station Creation
- Update request to server
- Ambulance Deployment

4.2.1 Module Explanation

User Registration :

In this module, the user registration process will be done.

Station Creation :

The admin logs in and create the stations for the ambulances along with the Information Such as the name of the city and the number of ambulances for the respective stations.

Update Request to Server :

In this module, the user application will be provided from where the user will register and login and can request the ambulance. The requests for the need of Ambulances will get updated to the server.

Ambulance Deployment :

In this module, based on the number of requests, the ambulance will be deployed to those particular stations where the need of ambulance is requested.

4.3 Architecture Diagram

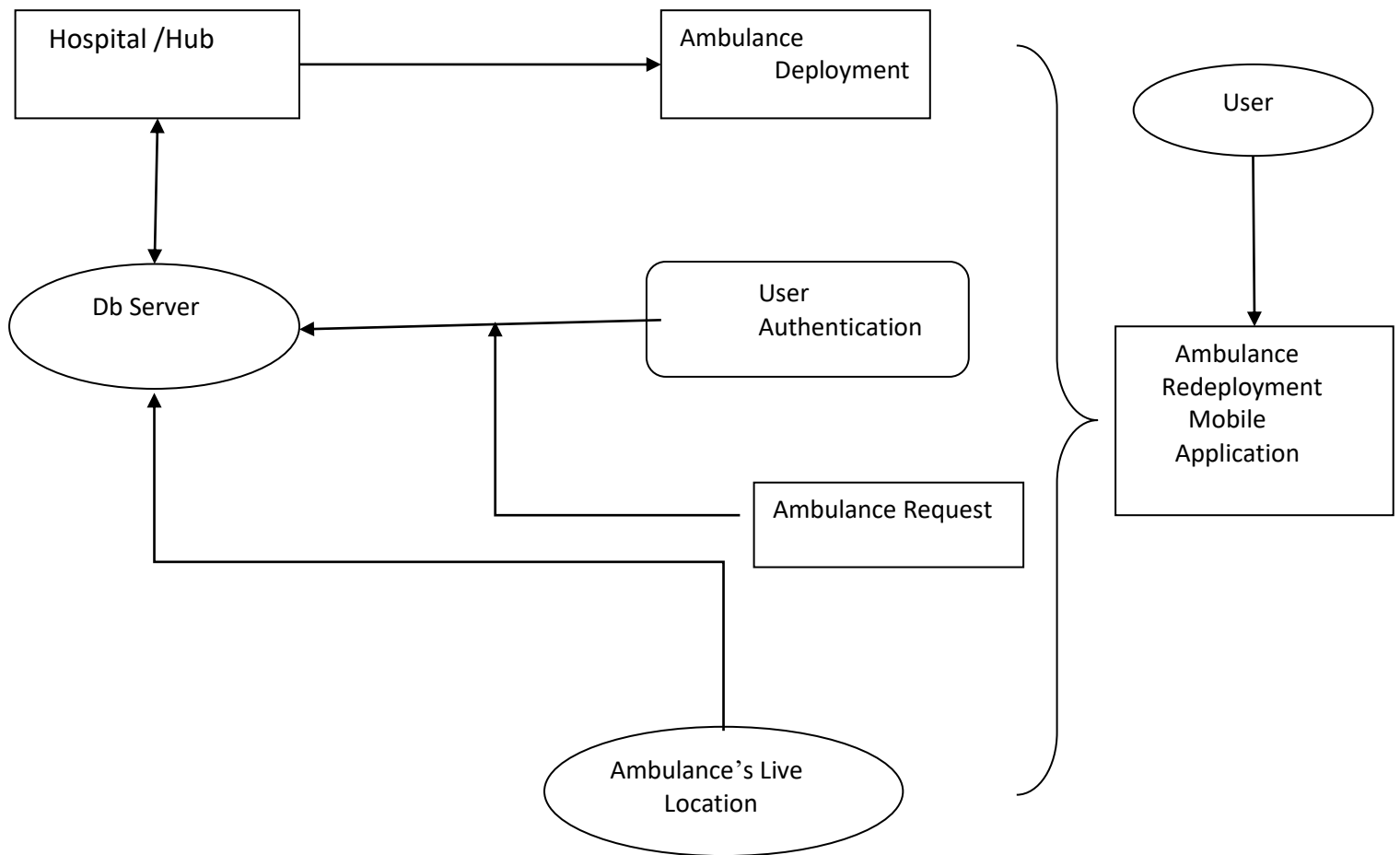


Fig 4.1 Architecture diagram

4.4 Application Screenshots

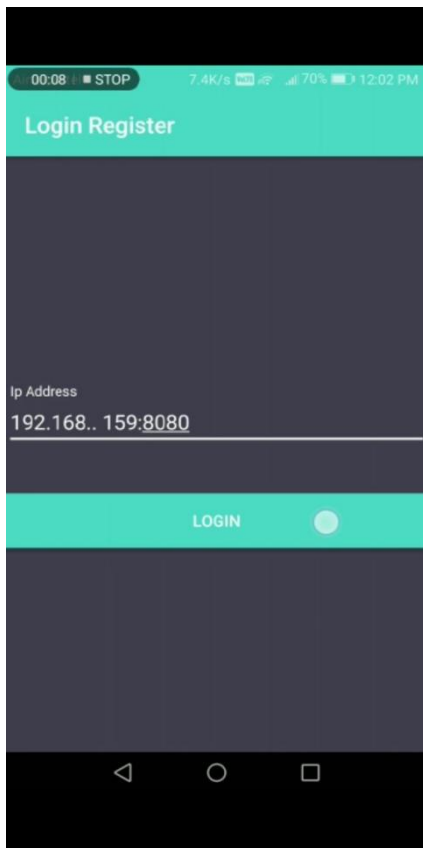


Fig 4.2 Entry Page

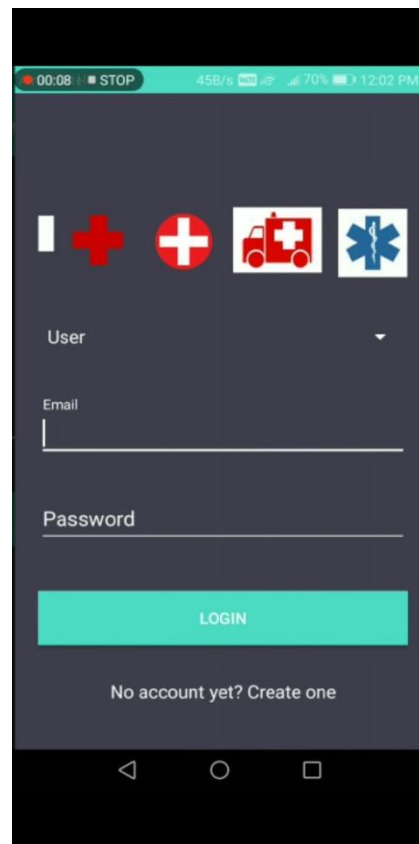


Fig 4.3 Login Page

This is the home page or the entry page of our application. In this page , the user fill the IP Address of our current emergent location. Next, it will go to the login page and the user just fill it ,then click the Login .

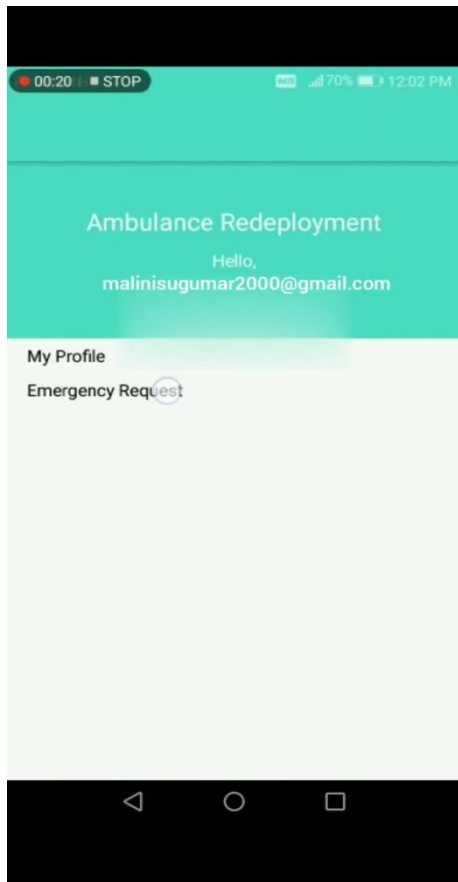


Fig 4.4 User's Profile Page

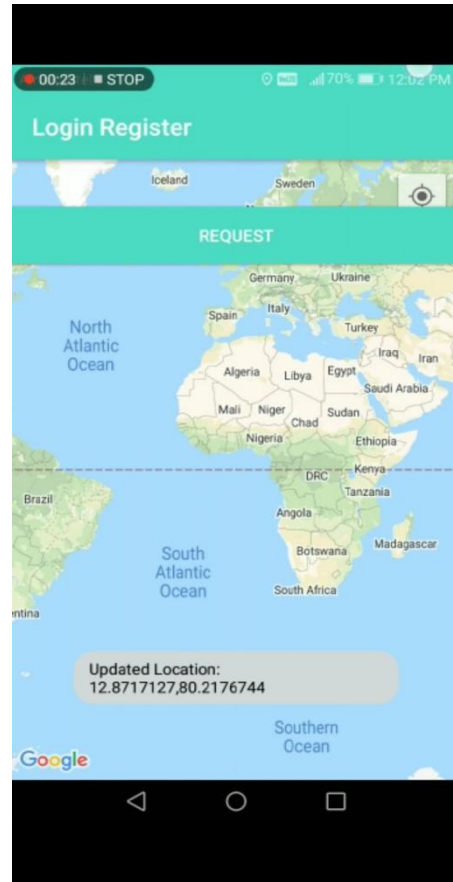


Fig 4.5 Location Page

The user after login, the user have a profile page. Now the User click into the emergency request if they needed ambulance. Next the given location updated in few seconds based on the internet speed.

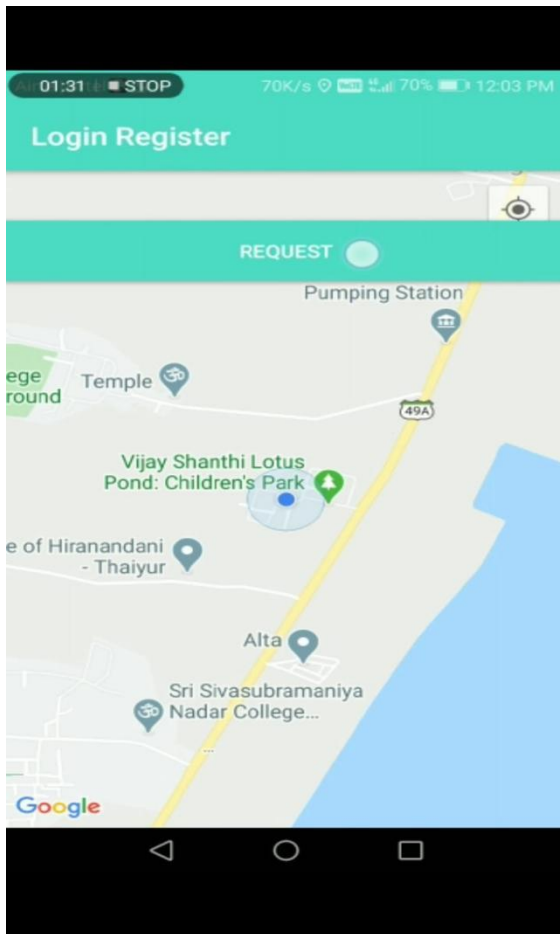


Fig 4.6 Request page

This is the final step of the requesting ambulance . In this page the given location searched by the server . Now the user give the request .

CHAPTER 5

RESULTS AND DISCUSSION

Thus, this project provides real time ambulance redeployment initiative to make the ambulance accessible to the appropriate station later it become available. The improvement of the transporting capability using complex dispatching strategies is highly limited. Besides, there exist quite a few hard constraints on the dispatching strategies in real world, e.g. picking up a patient in 10 minutes or picking up a patient as soon as possible if the patient is in extreme danger. Therefore, EMS systems in real life usually dispatch the nearest ambulance to pick up an EMS request. As a result, for an established EMS system, the real-time redeployment of each available ambulance, i.e. which station should an available ambulance be redeployed to, becomes more important to improving its transporting capability. For instance, when an EMS request comes, whether there are available ambulances in stations nearby the request significantly depends on the redeployment results of previous ambulances. However, real-time ambulance redeployment problem is challenging since multiple complex and dynamic data should be considered. For example, ambulance a1 has transported a patient to a hospital and becomes ready to be redeployed to a station. When redeploying ambulance a1, we propose to consider at least the following data. D1: the number of available ambulances at each station.

CHAPTER 6

SUMMARY AND CONCLUSIONS

We presented a parametric model for evaluating the distribution of emergency vehicle travel times between any areas in a street arrange. The strategy utilizes information from Chronicles ambulance trips that can be meager in time and System inclusion and is computationally tractable for enormous street systems and huge datasets of vehicle trips. The model parameters are interpretable and incorporate choices for the streets went by the vehicle and outing level choices, for example, time of day. We utilized a bayesian plan and markov affix monte carlo strategy to appraise the model parameters. We tried the strategy on a huge dataset of rescue vehicle trips from toronto. Exploratory investigation of the information showed that the dispersion of ambulance travel times between two axed areas is well modelled by a lognormal dissemination, with inconstancy parameter relying upon movement separation. These perceptions nuanced our demonstrating decisions. We thought about movement time expectations from our strategy with forecasts from an as of late distributed technique by budge et al. What's more, financially accessible travel time elective strategies in both point estimation and conveyance estimation. We likewise contrasted our strategy and the technique for westgate et al. On a subregion of toronto and found that our strategy performed nearly also in point estimation and better in interim estimation, while being unquestionably more computationally old. We adaptations of our strategy, to figure out which highlights were the most signification. Between time receptacles. We found that if the movement times during busy time were expanded by at any rate 20%, at that point time container factors gave a significant benefit to estimation. At last, we examined operational divergences for emergency vehicle eat the executives from utilized our technique versus the technique for budge et al. Subsequent to xing a lot of delegate emergency vehicle posts in toronto, we determined the likelihood that a rescue vehicle shows up at every crossing point in the city inside 2 min, reacting from the nearest post. We found that for about 10% of the convergences in the city, the two strategies gave appearance probabilities that redirecting by in excess of 15 rate focuses. We have dispatching and

redeployment framework to help realtime dynamic. The principle highlight of the framework lies in the precomputation of redeployment situations that permits prompt dynamic when calls are received. Top notch arrangements as estimated by inclusion indicators are controlled by methods for an incredible untouchable web index initially created for a static case. Computational efficiency is accomplished using a parallelization strategy which is fundamental to the achievement of this application. Computational results dependent on genuine information show that the proposed framework can adequately comprehend genuine occurrences ,For example, those emerging in montreal.

REFERENCES

- [1] C. J. Battenberg, S. Bhola, and R. D. van der Mei, "Optimal rescue vehicle dispatching," *Operations Research for Health Care*, vol. 248, pp. 269–291, March 2017.
- [2] V. Schmid, "Unraveling the dynamic emergency vehicle migration and dispatching issue utilizing inexact unique star gramming," *European Journal of Operational Research*, vol. 219, no. 3, pp. 611–621, April 2012.
- [3] M. S. Deskin, "A greatest expected covering area model: Formulation, properties and heuristic arrangement," *Transportation Science*, vol. 17, no. 1, pp. 48–70, 1983.
- [4] C. Battenberg, S. Bhola, and R. van der Mei, "An effective heuristic for ongoing emergency vehicle redeployment," *Operations Research for Health Care*, vol. 4, pp. 27–35, 2015.
- [5] D. Degele, L. Wyche, S. Rachana, and B. Werner's, "Time subordinate emergency vehicle allotment considering information driven empirically required inclusion," *Health Care Manga Sci*, vol. 18, pp. 444–458, 2015.
- [6] V. Schmid and K. F. Goerner, "Emergency vehicle area and movement issues with time-subordinate travel times," *Eu-ropean Journal of Operational Research*, vol. 207, no. 3, pp. 1293–1303, October 2010.
- [7] L. V. Snyder and M. S. Deskin, "Unwavering quality models for office area: The normal disappointment cost case," *Transporta-tion Science*, vol. 39, no. 3, pp. 400– 416, August 2004.

[8] M. Gaudreau, G. Laporte, and F. Semen, "A unique model and equal untouchable quest heuristic for continuous ambu-spear movement," *Parallel Computing*, vol. 27, no. 12, pp. 1641–1653, September 2001.

[9] Our sent web demo, http://101.124.0.58/rescue_vehicle/

[10] Y. Wang, Y. Zheng, and Y. Xu, "Travel time estimation of a way utilizing scanty directions," in *Proceedings of the twentieth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, September 2014, pp. 25–34.

[11] B. S. Westgate, D. B. Woodard, D. S. Matteson, and S. G. Henderson, "Huge system travel time dissemination estimation for ambulances," *European Journal of Operational Research*, vol. 252, pp. 322–333, 2016.

[12] J. Yuan, Y. Zheng, C. Zhang, W. Xin, X. Xin, G. Sun, and Y. Huang, "T-drive: driving bearings dependent on taxi trajectories," in *Proceedings of the eighteenth ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*, November 2010, pp. 99–108.

[13] E. Menelaus and H. N. Kotsiopoulos, "Travel time estimation for urban street systems utilizing low recurrence test vehicle information." *Transportation Research Part B: Methodological*, vol. 53, pp. 64–81, 2013.

[14] C. D. Fabritius, R. Ramona, and G. Valente, "Traffic estimation and forecast dependent on continuous coasting vehicle data," in *Proceedings of the eleventh International IEEE Conference on Intelligent Transportation Systems*, 2008.

[15] Y. Zheng, "Direction information mining: A diagram," *ACM Transaction on Intelligent Systems and Technology*, vol. 6, no. 3, pp. 29:1–29:41, 201

APPENDIX

A. SOURCE CODE

```
package com.google.maps;
import static com.google.maps.internal.StringJoin.join; import
com.google.gson.FieldNamingPolicy;
import com.google.maps.errors.ApiException;
import com.google.maps.internal.ApiConfig;
import com.google.maps.internal.ApiResponse;
import com.google.maps.model.LatLng;
import com.google.maps.model.PlaceType;
import com.google.maps.model.PlacesSearchResponse;
import com.google.maps.model.PlacesSearchResult;
import com.google.maps.model.PriceLevel;
import com.google.maps.model.RankBy;
/**
A<ahref="https://developers.google.com/places/web-
service/search#PlaceSearchRequests">Nearby
Search</a> request/

public class NearbySearchRequest

extends PendingResultBase<PlacesSearchResponse, NearbySearchRequest,
NearbySearchRequest.Response>
{

static final ApiConfig API_CONFIG =

new ApiConfig("/maps/api/place/nearbysearch/json")
```

.fieldNamingPolicy(FieldNamingPolicy.LOWER_CASE_WITH_UNDERSCORES);
Constructs a new {@code NearbySearchRequest}.

@param context The {@code GeoApiContext} to make requests through. public
NearbySearchRequest(GeoApiContext context)

{

super(context, API_CONFIG, Response.class);

}

Specifies the latitude/longitude around which to retrieve place information.

@param location The location to use as the center of the Nearby Search.

@return Returns this {@code NearbyApiRequest} for call chaining.

public NearbySearchRequest location(LatLng location)

{

return param("location", location);

}

Specifies the distance (in meters) within which to return place results. The maximum
allowed

radius is 50,000 meters. Note that radius must not be included if

{

@code rankby=DISTANCE} is specified.

@param distance The distance in meters around the

{@link #location(LatLng)} to search.

@return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest radius(int distance) { if (distance > 50000)
{
throw new IllegalArgumentException("The maximum allowed radius is 50,000 meters.");

}

return param("radius", String.valueOf(distance));

}
```

Specifies the order in which results are listed. @param ranking The rank by method.

@return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest rankby(RankBy ranking)

{

return param("rankby", ranking);

}
```

Specifies a term to be matched against all content that Google has indexed for this place. This includes but is not limited to name, type, and address, as well as customer reviews and other third-party content.

@param keyword The keyword to search for.

@return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest keyword(String keyword)
```

```
{
```

```
    return param("keyword", keyword);
```

```
}
```

Restricts to places that are at least this price level.

@param priceLevel The price level to set as minimum.

@return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest minPrice(PriceLevel priceLevel)
```

```
{
```

```
    return param("minprice", priceLevel);
```

```
}
```

Restricts to places that are at most this price level.

@param priceLevel The price level to set as maximum.

@return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest maxPrice(PriceLevel priceLevel)
```

```
{
```

```
    return param("maxprice", priceLevel);
```

```
}
```

Specifies one or more terms to be matched against the names of places, separated by spaces.

@param name Search for Places with this name.

@return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest name(String name)
```

```
{
```

```
    return param("name", name);
```

```
}
```

Restricts to only those places that are open for business at the time the query is sent.

@param openNow Whether to restrict to places that are open. @return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest openNow(boolean openNow)
```

```
{
```

```
return param("opennow", String.valueOf(openNow));
```

```
}
```

Returns the next 20 results from a previously run search. Setting

{@code pageToken} will

execute a search with the same parameters used previously — all parameters other than {@code pageToken} will be ignored.

@param nextPageToken The page token from a previous result. @return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest pageToken(String nextPageToken)
```

```
{
```

```
return param("pagetoken", nextPageToken);
```

```
}
```

Restricts the results to places matching the specified type.

@param type The {@link PlaceType} to restrict results to.

@return Returns this {@code NearbyApiRequest} for call chaining.

```
public NearbySearchRequest type(PlaceType type)
```

```
{

return param("type", type);

}
```

`@deprecated` Multiple search types are ignored by the Places API. `@param` types The `{@link PlaceType}`s to restrict results to. `@return` Returns this `{@code NearbyApiRequest}` for call chaining.

`@Deprecated`

```
public NearbySearchRequest type(PlaceType... types)
```

```
{
return param("type", join('|', types));
}
```

`@Override`

```
protected void validateRequest()
```

```
{
// If pagetoken is included, all other parameters are ignored. if
(params().containsKey("pagetoken"))
return;
}
```

```

// radius must not be included if rankby=distance if (params().containsKey("rankby")
&& params().get("rankby").get(0).equals(RankBy.DISTANCE.toString())
&& params().containsKey("radius"))

{

throw new IllegalArgumentException("Request must not contain radius with
rankby=distance");
}

// If rankby=distance is specified, then one or more of keyword, name, or type is required.
if (params().containsKey("rankby")

&& params().get("rankby").get(0).equals(RankBy.DISTANCE.toString())
&& !params().containsKey("keyword") && !params().containsKey("name") &&
!params().containsKey("type")) {
throw new IllegalArgumentException(

"With rankby=distance is specified, then one or more of keyword, name, or type is
required");
}

}

public static class Response implements ApiResponse<PlacesSearchResponse>
{
public String status;

public String htmlAttributions[];
public PlacesSearchResult results[];
public String nextPageToken;
public String errorMessage;

```

```

public boolean successful()
{

return "OK".equals(status) || "ZERO_RESULTS".equals(status);

}

public PlacesSearchResponse getResult()
{
PlacesSearchResponse result = new PlacesSearchResponse();
result.htmlAttributions = htmlAttributions;
result.results = results;
result.nextPageToken = nextPageToken;
return result;

}

public ApiException getError()
{
if (successful())
{
return null;

}

return ApiException.from(status, errorMessage);

}

}

}

```