ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

Version 1

CENG 407

Innovative System Design and Development I

P2018 - 3
Route Finder

İrfan Doğan BACAKSIZ
201411010
Ebru GÜNDÜZ
201411027
Ali Cem KOÇ
201411034
Nilay Gizem TEZER
201411059

Advisor: *Dr. Instructor Roya CHOUPANI*
Co-Advisor: *Prof. Dr. Erdoğan DOĞDU*

Table of Contents

Abstract

In recent years, with the development of smartphones, various applications have been developed and continued to be developed to help passengers plan and complete their journeys. Many of these applications are designed for trips with personal vehicles. However, people's tendencies towards public transportation have increased due to both economic reasons and ease of use. In this study, we aimed to develop a mobile trip planner for people using public transport. The main objective of this application is to serve the users in the most efficient way by making distance and time optimization, taking real-time GPS location and destination preference from the user. The applications also use road conditions, road traffic, speed limits and environmental factors to calculate this optimization.

Keywords: Smart Navigation, Artificial Intelligence, Mobile Application, Machine Learning, Travel Planning, Public Transportation, Cost, Time Schedule, Travel Time, Travel Distance, Public Road Network, Localization, Services Recommendation, Mobility As a Service (MaaS).

Özet

Son yıllarda, akıllı telefonların gelişmesiyle birlikte, yolcuların yolculuklarını planlamalarına ve tamamlamalarına yardımcı olmak için çeşitli uygulamalar geliştirilmiş ve geliştirilmeye devam edilmiştir. Bu uygulamaların çoğu, kişisel araçlarla yapılan geziler için tasarlanmıştır. Ancak, insanların hem ekonomik nedenlerden hem de kullanım kolaylıklarından dolayı toplu taşıma eğilimleri artmıştır. Bu çalışmada, toplu taşıma araçlarını kullanan kişiler için mobil bir gezi planlayıcısı geliştirmeyi amaçladık. Bu uygulamanın temel amacı, kullanıcıdan başlangıç konumu ve hedef tercihi alarak mesafe ve zaman optimizasyonu yaparak kullanıcılara en verimli şekilde hizmet etmektir. Uygulamalar ayrıca bu optimizasyonu hesaplamak için yol koşullarını, yol trafiğini, hız sınırlarını ve çevresel faktörleri kullanır.

Anahtar Kelimeler: Akıllı Navigasyon, Yapay Zeka, Mobil Uygulama, Makine Öğrenimi, Seyahat Planlaması, Toplu Taşıma, Maliyet, Zaman Çizelgesi, Seyahat Süresi, Seyahat Mesafesi, Herkese Açık Yol Ağı, Yerelleştirme, Hizmet Önerisi, Hizmet Olarak Hareketlilik.

1. Introduction

1.1. Problem Statement

One of the biggest problems faced by foreign tourists when they visit our country is the problems they experience in finding the places which they will visit. The most important of these problems is that tourists often don't get the answer they want when they want to ask the address to any other person walking down the street. Due to the lack of knowledge of foreign languages in our country, people may not be able to clearly describe the address. Because of these problems, tourists are often forced to travel by taxi. However, due to the problems experienced and they pay much more than they would pay for public transport in taxis. This situation causes tourists to have difficulty in traveling in the city and travel less.

There are similar problems not only for foreign tourists but also for local tourists in our country. One of the biggest problems for local tourists is that traveling with an individual vehicle in an unknown city leads to increased accidents and other traffic problems. Because of this situation, the experience of visiting the city does not give the desired satisfaction.

Due to these reasons, route finder will minimize the problems faced by domestic and foreign tourists and will enable them to increase the use of public transportation in our country and to travel more by public transportation.

1.2. Solution Statement

In this project, a travel planning application will be developed with the aim of finding the best and shortest route according to the personal preferences of the people using public transport. This application aims to make the user experience more conscious and effective in the use of public transport. Mobile application allows users to easily search for their goals. As a result of this search, the application will inform the user about the minimum, maximum and average cost that can be given along the route by selecting the best route for both time and way. Many algorithms are used in the route calculation. The most appropriate ones will be developed and adapted in accordance with the usage scenario.

In addition, our artificial intelligence-driven application aims to improve the user's travel experience by showing users the important places they should travel while traveling. This helps the user not to miss attractions such as parks, museums, banks or historical sites.

In summary, the mobile trip planner will help travelers to design their travel routes in the best way before the journey begins.

1.3. Related Work

There are many applications has developed similar to Route Finder. Like Google Maps, Yandex Navi, Moovit, 9292.nl, İBB Yol Gösteren, Ego Cepte. However, every application written so far has certain deficiencies. For example, the Yandex Navi app was written for personal vehicles only. Although the İBB Yol Gösteren and Ego Cepte applications are written for public transportation, they do not fully meet the needs of tourists.

2. Literature Search

2.1 What is Public Transportation?

Leading agencies and experts in the field define public transportation systems today by their basic forms, operation, and design [96]. However, some of these definitions hint at how we should think of public transport. The American Public Transportation Association (APTA) defines public transportation as “transportation by a conveyance that provides regular and continuing general or special transportation to the public” [97]. Vukan Vuchic defines urban public transportation as “Transport system(s) for intraurban or intraregional travel available for use by any person who pays the established fare” [98]. He categorizes public transportation as consisting of both transit and paratransit. While transit is noted as having “fixed routes and schedules, such as bus, trolleybus, and rail services,” paratransit includes “modes of passenger transportation consisting of small or medium capacity highway vehicles offering service adjustable in various degrees to individual users’ desires. [96] ”Wikibooks’ Fundamentals of Transportation/Transit defines public transit as “a mode of transportation that involves moving persons from one place to another using a common form of conveyance, allowing multiple persons to share a common vehicle while traveling” [99]. If we to summarize public transport in line with these definitions; public transport is a form of transportation shared by multiple users.

2.2. Why Should We Use a Public Transportation?

It is possible to show the answer of this title with a single image. A figure 1 that has circulated for years that originated with the City of Muenster in Germany shows space requirements for travel by car, bus, or bicycle. As seen in the first picture, if people try to travel with their personal vehicles, they will face various problems such as traffic, parking problem and time. When we look at the third picture, although traveling by bike may seem a bit more advantageous than traveling by car, it has its disadvantages. For example, it is very difficult to travel by bicycle in difficult weather conditions (etc. snow or rain). When we compare bus and other 2 transportation types (bike and car), we can see that all kinds of buses are advantageous.

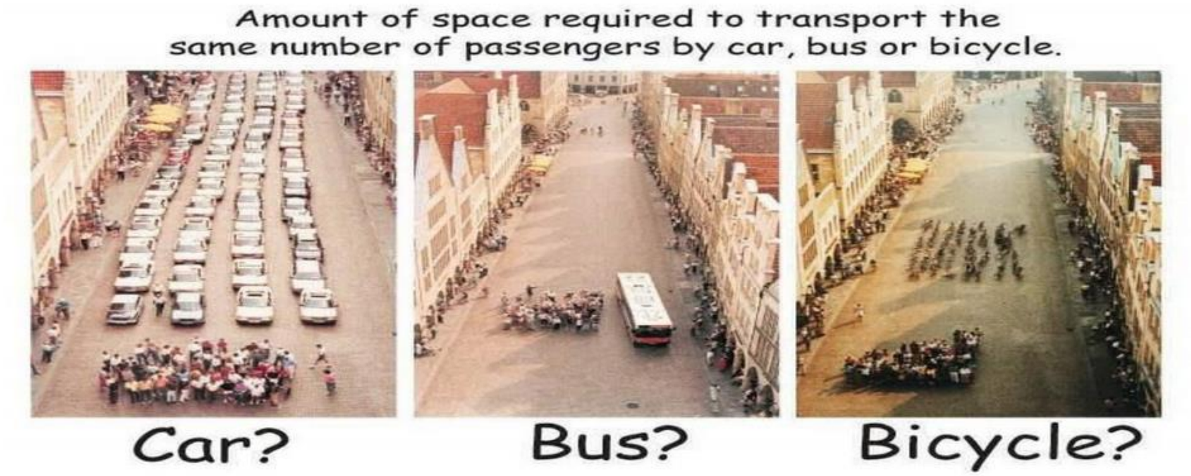

Figure 1: City of Muenster Image [100]

2.3. Mobility As A Service (MaaS)

Mobility as a Service (MaaS) is a new transportation solution as a mobility concept. In other words, it is the services offered to go from one place to another. The definition of MaaS was made by Hietanen in 2004: In recent years, the increasing number of transport services offered in cities and the advancements in technology and ICT have introduced an innovative Mobility as a Service (MaaS) concept. It combines different transport modes to offer a tailored mobility package, similar to a monthly mobile phone contract and includes other complementary services, such as trip planning, reservation, and payments, through a single interface (Hietanen, 2014) [101]. Mobility as a Service (MaaS) is a new concept that is expected to create significant changes in transport applications. This conceptual model consists of two components: customer (user) and Mobility as a Service.

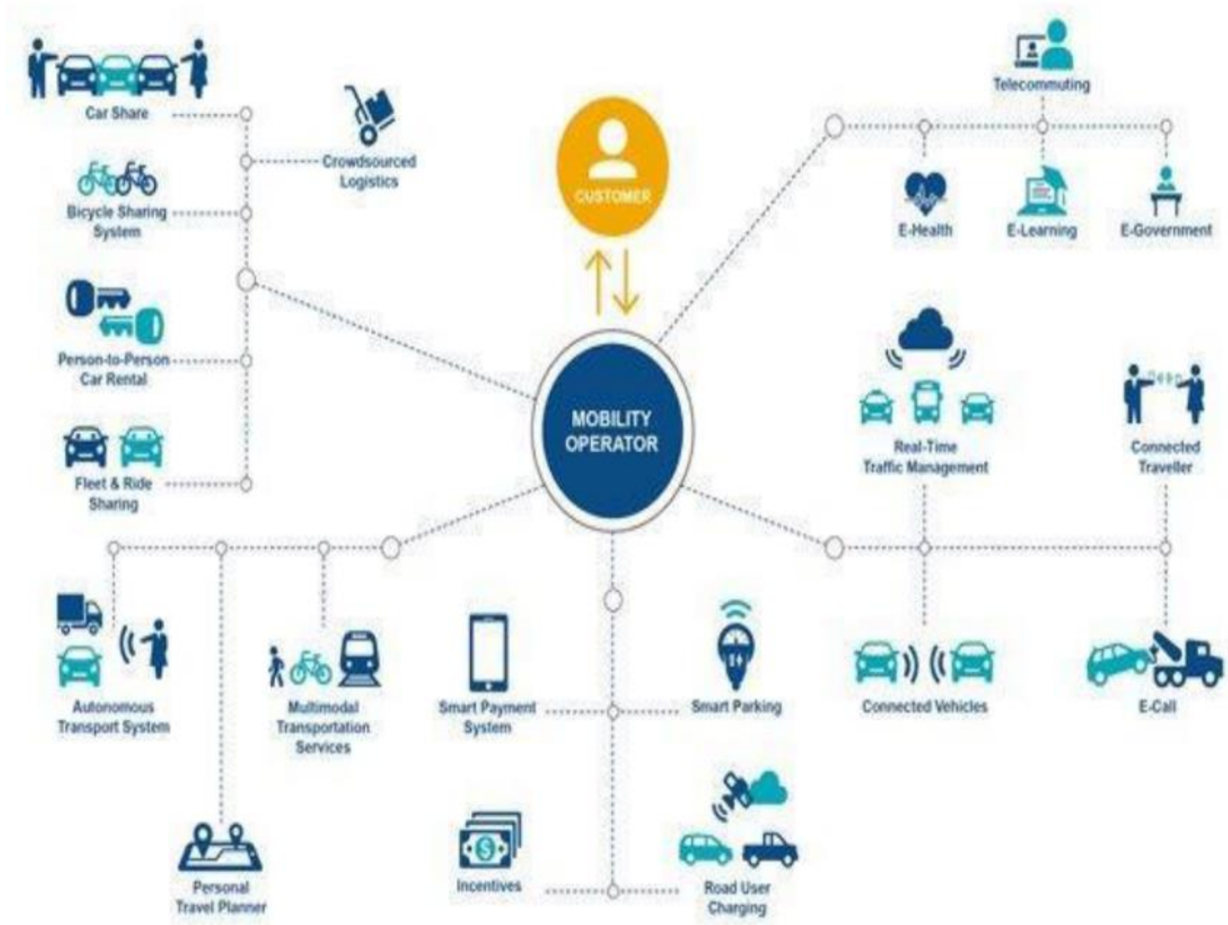

Figure 2 Mobility as a Service

The MaaS system offers alternatives to meet different demands for transportation:

- **Owned vehicles:** The system will integrate all available vehicles (the models and limitations of current ride sharing platform can be seen as a first step implementing this).
- **Public Transport:** This is the main component of the transport system which will provide transport and mobility services in the MaaS system. The role of public transport will be increased and the priority in traffic as well as in investment will be main aspects in planning new systems.
- **Taxi, Car Sharing:** The main characteristics of the user's profile as well as the availability of vehicles will include taxi system in MaaS system as provider of specific transport services (and other similar systems).
- **Bikes:** This component includes parking areas for bikes and rent-a-bike systems for both, private and public bikes.
- **Ped. Ways:** Another direction is to include walking as an important mode for a multimodal transport system support for MaaS.
- **Shops:** The role of shops is, on the one hand, as destinations for urban trips (transport of people) and, on the other hand, as provider of goods (transport of goods).

- **Other services:** Any other services could be integrated in MaaS to increase the value of the service (for instance, fast food service integration – drive in service as part of the mobility as a service) [102].

The main components of MaaS:

- Registeration
- Travel planning
- Reservation
- Ticket system
- Payment

2.3.1. Examples of Mobility As A Service

In this section, we will illustrate similar applications to the application we will develop.

2.3.1.1. Google Maps

Google Maps application offers users the public transportation, walking route, vehicle route and instant traffic information livingly. After receiving the user's location automatically, the searched place, together with the current traffic information, provides the routes to be used to reach the destination to user. When the Google Maps application offers these routes to the user, offers the possibility to choose different transportation methods such as walking, personal vehicle, public transport, bicycle and airplane.

There are also some disadvantages, as there are many advantages of this application. If we are going to use public transport to reach the target, this application may not always give the right results. The arrival time is not correct. It doesn't mention the amount of money that can be spent on average. You cannot use the application in real time according to the public transport mode. As a difference of our application from Google Maps application, we aim to correct these disadvantages this application.

2.3.1.2. Moovit

The Moovit app delivers its users to the destination through only public transport. Compared to Google Maps, the time spent on reaching the goal is more accurately calculated. Moovit is developing its database through volunteer users, not by its own means. This makes the Google Maps application more comprehensive and more reliable than Moovit. OpenStreetMap provides the infrastructure of the map to its users. This application is similar to the application we will develop. However, in addition to this application, we will make recommendations to popular places near targeted route.

2.3.1.3. İBB Yol Gösteren

İBB Yol Gösteren is an application developed for İstanbul only. The application provides users with access to mobese cameras and time of travel of public transport, routes and location of its and location of stations in the city. The biggest disadvantage of İBB Yol Gösteren is that it does not work when the user cannot determine user's position.

2.3.1.4. Yandex Navi

The Yandex Navi app allows users to reach their goals only when using their personal tools. In addition to using up-to-date traffic information such as Google Maps; It provides vital information for drivers, such as road works, road accidents, speed limits, the user's instantaneous speed. This information is obtained adding by users to the map. Yandex Navi also shows the empty parking spaces in your area with the latest update. However, this feature is still in the beta version does not work exactly right.

2.3.1.5. Ego Cepte

Ego Cepte app is an application developed only for Ankara. It offers its users the facilities such as the time of public transport, the routes of public transport, the instant location of public transport, and the location of public transport in the city. However, we can use this application to load money to our Ankara Card.

2.3.1.6. 9292.nl

9292.nl has provided travel information about public transport in the Netherlands for more than 20 years. The main goal of 9292.nl is to make life easier for all people involved in public transport. Only for the Netherlands, this application allows you to create a travel route using public transport options such as buses, trains, ferries, trams and subways. Also in this application disabled users are not forgotten. There is a different category for disabled users. The use of public transport for animals is also not forgotten. In addition, the application creates a route plan for the user by obtaining the information on which time and date you will use. It shows the user the average fare to be spent on the journey and the details of how much he would pay to the public transport vehicle. The application we will sample ourselves while developing our application is 9292.nl.

2.4. Finding Best Path

In this section we mention about algorithms commonly used in similar jobs. The shortest path problem was first discussed in 1958 with the Bellman algorithm [81]. Then in 1959 the Dijkstra algorithm and in 2013 Bousquet, Constans and El Faouzi (2009) [82] presented a dynamic labeling algorithm to calculate the shortest path in a one-way and multimode and proposed a strategy to solve the two-way problem [3]. Khani (2012) and others [91][94] propose a simple but effective algorithm to find the most appropriate way in the intermodal urban transport network. As part of this algorithm, they use the Trip-Based Transit Shortest Path algorithm [3], which takes advantage of the hierarchical features of transfer stations and transfer stations to find the shortest path from common tagging algorithms. Nassir (2012) and others [91][94] offers an effective Repeater Tagging algorithm for finding an intermodal optimal tour (from origin to origin) in a time-dependent transport network for a traveler with a number of targets to visit [3].

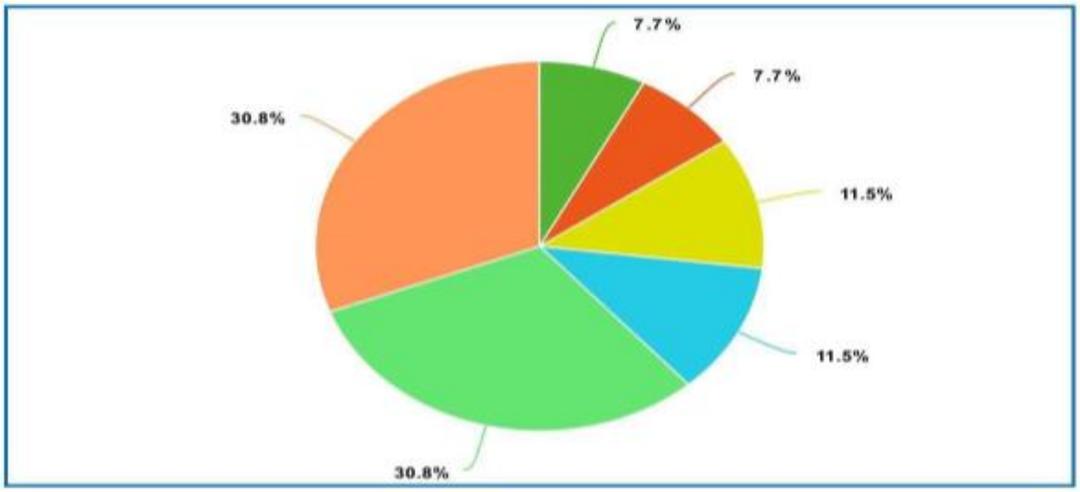

Figure 3: Algorithms used in the articles reviewed

2.4.1. Bellman-Ford Algorithm

Bellman-Ford Algorithm stands out for its simplicity and comprehensiveness. The Bellman-Ford algorithm solves the single-source problem in the general case, where edges can have negative weights and the graph is directed. If the graph is undirected, it will have to modified by including two edges in each direction to make it directed [29].

Bellman-Ford has the property that it can detect negative weight cycles reachable from the source, which would mean that no shortest path exists. If a negative weight cycle existed, a path could run infinitely on that cycle, decreasing the path cost to. If there is no negative weight cycle, then Bellman-Ford returns the weight of the shortest path along with the path itself.

```
function bellmanFord(G, S)
    for each vertex V in G
        distance[V] <- infinite
        previous[V] <- NULL
    distance[S] <- 0
    for each vertex V in G
        for each edge (U,V) in G
            tempDistance <- distance[U] + edge_weight(U, V)
            if tempDistance < distance[V]
                distance[V] <- tempDistance
                previous[V] <- U
    for each edge (U,V) in G
        If distance[U] + edge_weight(U, V) < distance[V}
            Error: Negative Cycle Exists
    return distance[], previous[]
```

Figure 4: Bellman-Ford Algorithm Pseudocode [5]

2.4.2. Djikstra's Algorithm

This algorithm delivers the shortest path from a given node i to a single destination node or all other nodes within a graph with nonnegative edge path costs [21].   In this algorithm, we first determine a node. Then algorithm works on this node. The algorithm initially assigns endless (∞) value to all nodes, assuming that there is no access yet. So, in the initial state yet we cannot go to any node. It then navigates all nodes that are adjacent to the start node and updates the reach to these nodes. After this update, the updated nodes update their neighbors and continue until all nodes are updated and there is no new update on the shape. The weakness of the Dijkstra algorithm: The algorithm does not work successfully if there is an edge minus (-). This is because the edge minus (-) produces a better result than the current state and the algorithm can never stabilize.

Dijkstra Algorithm gives more accurate results than Bellman-Ford. The values found are precise and unchanging. According to the researches [1], the Dijkstra algorithm is better than BellmanFord Algorithm when performing the shortest path calculations.

```
function
dikstra(G,S)


    for each vertex V in
    G
    distance[V] <- infinite previous[V] <-
    NULL If V ! = S, add V to Priority Queue
    Q distance[S] <- 0 while Q IS NOT
    EMPTY
        U <- Extracr MIN from Q for
        each unvisited neighbour V of U
            tempDistance <- distance[U] +
            edge_weight(U, V) if tempDistance <
            distance[V]
                distance[V] <- tempDistance
           previous[V] <- U return distance[],
                              previous[]
```

Figure 5: Djikstra's Algorithm Pseudocode[28]

2.4.3. Floyd-Warshall Algorithm

The Floyd-Warshall algorithm solves the all-pairs shortest path problem. It uses a dynamic programming approach to do so. Negative edge weight may be present for Floyd-Warshall. Floyd-Warshall takes advantage of the following observation: the shortest path from A to C is either the shortest path from A to B plus the shortest path from B to C or it's the shortest path from A to C that's already been found. This may seem trivial, but it's what allows Floyd-Warshall to build shortest paths from smaller shortest paths, in the classic dynamic programming way. [29]

```
for all nodes i of N do
    for all nodes j of N do
        if there is an edge from i to j then Dist[i, j] := d(i, j) else Dist[i, j] := ∞
for all nodes i of N do
    for all nodes j of N do
        for all nodes k of N do
            if Dist[j,i] + Dist[i,k] < Dist[j,k] then Dist[j,k] := Dist[j,i] + Dist[i,k];   Next[j,k] := i;
```

Figure 6: Floyd-Warshall Algorithm Pseudocode [21]

If we compare Floyd-Warshall and Dijkstra algorithms time complexity, Floyd-Warshall algorithm has a time complexity that can perform the Dijkstra algorithm n times. However, Floyd is usually faster than executing Dijkstra's algorithm for each node [21].

2.4.4. Johnson's Algorithm

While Floyd-Warshall works well for dense graphs (meaning many edges), Johnson's algorithm works best for sparse graphs (meaning few edges) [29]. If we compare Johnson's and Floyd-Warshall algorithms in sparse graphs, Johnson's algorithm has a lower running time.

Johnson's algorithm takes advantage of the concept of reweighting, and it uses Dijkstra's algorithm on many vertices to find the shortest path once it has finished reweighting the edges [29]. It works by using the Bellmann-Ford algorithm to compute a transformation of the input graph that removes all negative weights [21].

```
Johnson(G)

    create G` where G`.V = G.V + {s},
        G`.E = G.E + ((s, u) for u in G.V), and
        weight(s, u) = 0 for u in G.V
    if Bellman-Ford(s) == False
        return "The input graph has a negative weight cycle"

    else:
        for vertex v in G`.V:
                h(v) = distance(s, v) computed by Bellman-Ford
        for edge (u, v) in G`.E:
                weight`(u, v) = weight(u, v) + h(u) - h(v)
    D = new matrix of distances initialized to infinity
    for vertex u in G.V:
            run Dijkstra(G, weight`, u) to compute distance`(u, v) for all v in G.V
            for each vertex v in G.V:
            D_(u, v) = distance`(u, v) + h(v) - h(u)
    return D
```

Figure 7: Johnson's Algorithm Pseudocode [12]

2.4.5. Topological Sort [29]:

For graphs that are directed acyclic graphs (DAGs), a very useful tool emerges for finding shortest paths. By performing a topological sort on the vertices in the graph, the shortest path problem becomes solvable in linear time.

A topological sort is an ordering all of the vertices such that for each edge (u,v) in E ,u comes before v in the ordering. In a DAG, shortest paths are always well defined because even if there are negative weight edges, there can be no negative weight cycles.

```
for i from 1 to
N
        select some vertex v with indegree(v) equal
        to 0


        add v to our topological sort
        remove v and all edges leading
        from it
```

Figure 8: Topological Sort Pseudocode [13]

2.4.6. Breadth-First Search Algorithm (BFS) [30]:

Breadth-First Traversal (or Search) for a graph is a similar to Breadth First Algorithm of a tree. The only catch here is, unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a Boolean visited array. For simplicity, it is assumed that all vertices are reachable from the starting vertex.

For example, in the following graph, we start traversal from vertex 2. When we come to vertex 0, we look for all adjacent vertices of it. 2 is also an adjacent vertex of 0. If we don't mark visited vertices, then 2 will be processed again and it will become a non-terminating process. A Breadth First Traversal of the following graph is 2, 0, 3, 1.

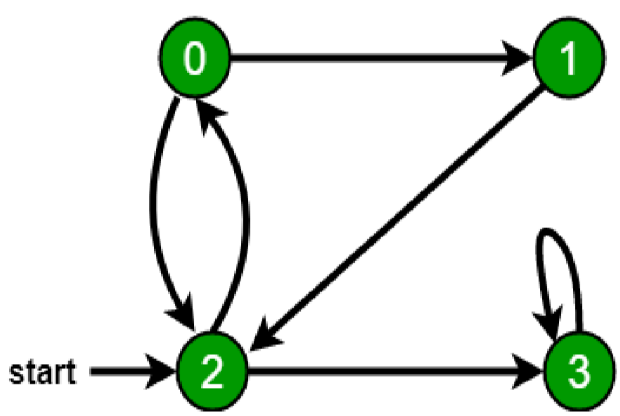

Figure 9: Breadth-First Search Traversal [30]

```
Starting from vertex v:
create a queue Q
mark v as visited and put v into Q
while Q is non-empty
        remove the head u of Q
        mark and enqueue all (unvisited)
        neighbours of u
```

Figure 10: BFS Algorithm Pseudocode [14]

2.4.7. A* Algorithm

A* algorithm is developed by Hart, Nilsson and Raphael in 1968 [18]. The A* algorithm is a more general approach than Dijkstra's algorithm for finding the shortest path between two nodes in a graph [22][23]. It guarantees to find the optimum route from start point to end point if there is route. It is usually used for network routing, finding shortest path for games etc. [19][20]. It chooses the next point by current cost from starting point and heuristic distance to ending point

```
1. Initialize the open list
2. Initialize the closed list
   put the starting node on the open
   list (you can leave its f at zero)
3. while the open list is not empty
  a) find the node with the least f on
     the open list, call it "q"
  b) pop q off the open list
  c) generate q's 8 successors and set their
     parents to q
  d) for each successor
     i) if successor is the goal, stop search
       successor.g = q.g + distance between
                        successor and q
       successor.h = distance from goal to
       successor (This can be done using many
       ways, we will discuss three heuristics-
       Manhattan, Diagonal and Euclidean
       Heuristics)
       successor.f = successor.g + successor.
     ii) if a node with the same position as
        successor is in the OPEN list which has a
       lower f than successor, skip this successor
     iii) if a node with the same position as
        successor  is in the CLOSED list which has
        a lower f than successor, skip this successor
        otherwise, add  the node to the open list
   end (for loop)
  e) push q on the closed list
  end (while loop)
```

Figure 11: A Algorithm Pseudocode [4]*

Algorithm	Runtime
Bellman–Ford	O(|V|*|E|)
Dijkstra's	O(|V|*log|V|+|E|)
Topological sort	O(|V|*|E|)
Floyd-Warshall	O(|V|³)
Johnson's	O(|V|*|E|+|V| ²*log|V|)
A*	O(|V|)
Breadth-First Search (BFS)	O(|V|*|E|)

Figure 12: Runtime Complexities of Algorithms

3. Summary

3.1. Summary of Conceptual Solution

In this CENG407 project, we talked about the mobile excursion utility that aims to find the best and shortest route according to the personal preferences of people using public transport. This is aimed at making the use of public transport more conscious and effective for the best user experience; Using the easy-to-use Google Maps interface and SDK, it allows mobile app users to easily search for their destination. As a result of this search, the application will inform the user of the minimum, maximum and average cost that can be given along the route by selecting the best route both time and way. Although many algorithms are used in the calculation of the route, we decided to use the A* algorithm which is the most current and most effective among these algorithms. In addition, our artificial intelligence-assisted application aims to improve the user's travel experience by showing the user the important places to travel around while traveling, which helps the user not to miss out on attractions such as parks, museums, banks or historic sites. In general, the mobile trip planner can help travelers to design their travel routes in the best way before the trip begins.

3.2. Technology Used

Access to the app will be done via mobile for the first version. The mobile application is primarily intended for launch on iOS platforms. The application will be developed using the Swift [108] programming language. In addition, our application will be integrated with Google Maps. The map section to be used in the application and the majority of the information required for the application will be provided via Google Maps.

In addition, we will develop machine learning in CreateML [106] to show popular locations. The reason we use machine learning in this application is to teach to the machine the travel plans that will be created by the user and to suggest places where the user can travel more efficiently in line with these plans.

For the first version, the target platform will be MacOS [107] and the development environment will be Xcode [108].

4. Software Requirement Specification

4.1. Introduction

4.1.1. Purpose

The main objective of this project is to help local and foreign tourists to plan their travels in the city and to show the public transport vehicles they will use and the average cost for their travels. The main motivation of this project is to minimize the time a person spends on travel, to provide English language support for foreign tourists, to guide the public to public transport, and to calculate the average fare that the traveler will cost.

4.1.2. Scope of the Project

This software system will be a route finding and tour guide application for users using public transport and especially for locals and foreign tourists. This application will be designed to create the best route for reaching a desired location from a specified location and to determine the popular and touristic places on this route, to minimize the time loss and to ensure the efficient use of time. By helping users to use time efficiently, they will meet the needs of the users. More specifically, the application is designed to list the most appropriate route to reach the desired location from the specified location, the public transports to be used to reach the destination, and their average cost, the popular places to visit on the designated route.

4.1.3. Glossary

Term	Definition
SRS	Software requirements specification.
Stakeholder	People with any interest in project's outcome.
A*	A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.
Mobile Application	A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. [1]

Table 1: Glossary

4.1.4. References

[136] IEEE. “IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications”. IEEE Computer Society. October 20, 1998.

4.1.5. Overview of the Document

The rest of this paper is organized as follows: Section 2 describes the definition, specification of the project and properties of the method and its application for users who use the system and read the document. The constraints and risks of this method are mentioned. Section 3 is mainly written for developers of this system and describes in technical terms the details of the requirements of this system. Section 4 describes the specifications of actors and stakeholders and their interactions with the developed method are presented. The functions used by the user in order to use the project software and the tasks of these functions are described.

4.2. Overall Description

4.2.1. Product Perspective

The software described in this SRS allows users to find the best route to reach the desired destination from the specified location. After determining the best route, the software lists the possible transportation vehicle and the average fare for the user to reach the destination. The software informs the user of the popular and touristic places on the route, increasing the efficiency of the time spent reaching the destination.

4.2.2. User Characteristics

Expectation from users who read the document is that they have information about A* algorithm, best path problem and swift programming language. Expectation of users using the system is that having a phone using iOS operating system, using an active icloud account to download the application through the app store and the basic smart phone usage capabilities.

4.2.3. Contraints

Some of the operating system and programming languages to be used in the system are a limitation for this software. The Mac environment and swift language are required to develop programs for the IOS operating system.

This software is also limited to the cellular data, as it will use the internet, and the storage capacity of the device as it takes up space in the device to be used.

4.2.4. Risks

Failure to draw the desired route due to the bugs in the system or misuse of the data from other sources may be considered as a risk because it may mislead the use.

4.2.5. Assumptions

During the use of the application, the phone with the IOS operating system correctly calculates the instant location and sends it to the application. The application algorithm works correctly and shows the user the necessary information. The public transport vehicle to be used arrives at the stop of the user without any delay.

4.3. Requirements

4.3.1. Specific Requirements

4.3.1.1. User Interfaces

The UI informations shown in this project is the main form divided into sub parts which are:

- ➔ Search bar: to search for and search for the desired location.
- ➔ The map will appear in the background.
- ➔ The destination for public transportation charges for the destination.
- ➔ The part of the public transport that the user can use.

4.3.1.2. Hardware Interfaces

No hardware interfaces needed to run this software.

4.3.1.3. Software Interfaces

Software presented in this SRS only needs an IOS Operating System.

4.3.1.4. Communications Interfaces

There is an internet connection is required to run this software.

4.3.2. Functional Requirements

➔ the user should be able to search for the location they want to go to.

➔ user should be able to see the cost of going to the location.

➔ user should be able to see the cost of going to the location.

➔ the user should be able to view the entire trip plan.

4.3.3. Software System Attributes

4.3.3.1. Performance

Depending on the user's cell data provider, the application will not take more than 10 seconds to determine the user's current location. It will not take more than 10 seconds for the application to find and show the route and the public transport vehicles to be used.

4.3.3.2. Availability

Users can use the application in an IOS environment. To be able to use the application in an iOS environment, the application is initiated with determination the user's location information and the location of the desired target to be reached.

4.3.3.3. Security

There is no security limitation as there is no important information to keep. In no event will the user location be recorded by us.

4.3.3.4. Portability

Because the system will be developed on iOS environment, it can work only on iOS.

4.3.3.5. Usability

➔ The application is battery friendly.

➔ It will be designed so that the users can use it very easily.

➔ It determines by public transport the best route between your current location and the location you want to go.

➔ The user can navigate to the location he / she wants to go.

4.3.3.6. Scalability

There is no user limitation at the same time. But in the free version of the application, total route search limit is 25.000 search per day.

4.3.3.7. Ease of Use

Since the developed application is a user-oriented project, the it should have a user interface that will provide simple usage to user and this interface should be understandable and user- friendly.

4.4. UML Analysis Model

4.4.1. Use Cases

4.4.1.1. Actor(s)

User: The user is a person who download the application and uses it. For using the application, user must accept the Loc. Sharing Permission only one time, because route will created according to user's current location. The user should enter destination location for creating route. The user accept the route which suggested by application. The user can see Route, Average cost, Time and Public transport information.

4.4.1.2. Stakeholder(s)

Project Advisor: The project advisor is responsible for delivering the project on time and on a given budget. They usually work together and guide the team that develops the project so that the goals in the project can be fulfilled correctly.

The Project Manager's goals are; To follow the project and check if it is done as requested. To ensure that the risks and problems are properly handled. Working with the group to motivate them and give the necessary support.

Project Development Team: It is a team of people who design the project given by the project advisor according to the desired characteristics and who play an active role in the project by working together within the project.

The Project Development Team's goals are; To make the project timely and correctly. To be able to complete the project in accordance with the quality and rules as much as possible. [110]

User: The user is the person who will use the project after the project is finished. User is the person who gives an idea to the project developer about the needs of the project and be reference in the project development process.

The User’s Goals are;

- Accessing the best route information to achieve the desired destination.

- Receiving suggestions to use time efficiently.

- Accessing to the type of transportation vehicle, travel time and average cost information.

4.4.1.3. Use Case Diagram

Figure 13 presents a use case diagram for the subject route finder application. The system shows the operations that end users can perform.

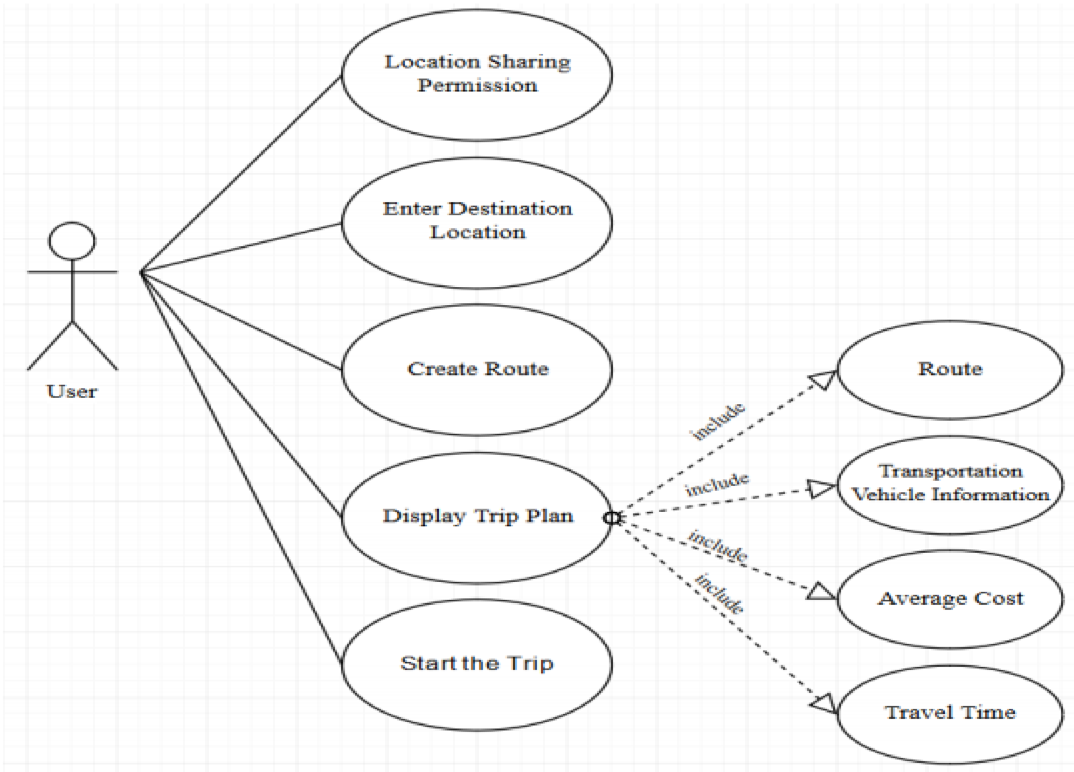

Figure 13: Route Finder Use Case Diagram

4.4.1.3.1. Brief Description of Use Case Diagram

After the program has been run by the user, if the location share is not allowed, the user will be asked to allow location information sharing first. The program then asks the user to enter the address of the destination. After entering the address, user must click “Create Route" button to continue. After user clicking "Create Route" button, program makes distance and time optimization. The app then shows a “Trip Plan” containing the route, vehicle information, average fare and time to the user on a new page. When the user clicks the “GO!” button, the Start the Trip step works and the user starts his / her journey with the Route Finder program.

4.4.1.3.2. Use Case Description

Use Case	Location Sharing Permission
Primary Actor	User
Goal in Context	Sharing the current location into the system for route calculation.
Preconditions	None
Trigger	None
Scenario	If the user wants to use the application, he must accept that the location information will be used by the application.
Exceptions	Location error due to cellular data or another external factor.

Table 2: Location Sharing Permission Use Case Description

Use Case	Enter Destination Location
Primary Actor	User
Goal in Context	Locating / identifying places to visit
Preconditions	Acceptance of location information sharing.
Trigger	None
Scenario	The user enters the destination in the search bar.
Exceptions	Incorrect orientation if map information is out of date.

Table 3: Enter Destination Location Use Case Description

Use Case	Create Route
Primary Actor	User
Goal in Context	User acceptance of the route found by the application.
Preconditions	The destination should be selected.
Trigger	Click “Create Route” Button
Scenario	If the user likes the route after selecting the destination, he can accept the route drawn with the create route and calculate the trip plan.
Exceptions	None

Table 4: Create Route Use Case Description

Use Case	Start the Trip
Primary Actor	User
Goal in Context	After the trip plan is set, the trip start if the user click the “GO!” Button.
Preconditions	Click “GO!” Button
Trigger	Click “GO!” Button
Scenario	The user starts his / her journey with the route he / she desires and gets in the way with all the information he needs.
Exceptions	Confirmation of an incorrect or out-of-date route.

Table 5: Start the Trip Use Case Description

Use Case	Display Trip Plan
Primary Actor	User
Goal in Context	Display of accepted route information. This information: route, average cost, estimated time, and public transport information.
Preconditions	Click “Create Route” Button
Trigger	Click “Create Route” Button
Scenario	The user can easily see the information such as the time and cost to pass to the destination location.
Exceptions	Missing or incomplete data from 3rd party applications.

Table 6: Display Trip Plan Use Case Description

4.4.2. State Chart Diagram

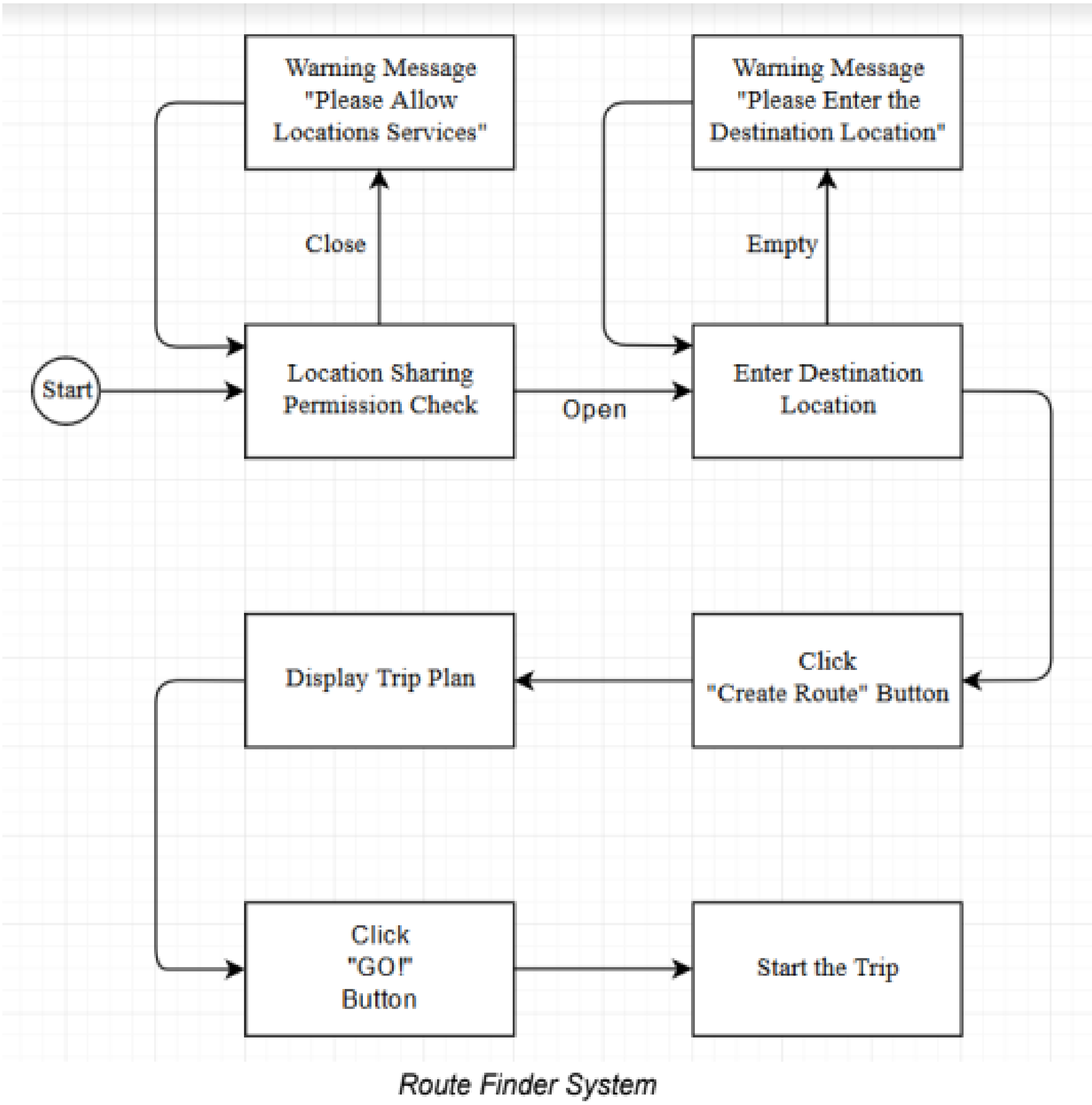

Figure 14: State Chart Diagram

5. Software Design Description

5.1. Introduction

This document provides detailed information about the requirements of the route finding system software. It will explain the working principles of the proposed method, and design of the user interface of this method. This document prepared to guide to users.

5.1.1. Purpose

The purpose of this document is providing the details of the proposed project "Optimal route, public transportation and average cost detection to reach the desired location". This determination will help local and foreign tourists to plan their travels in the city and see the average cost of travel for public transport. Our goal is to minimize the time spent by the user while traveling, to provide English language support to foreign tourists, to guide the public to public transportation and to calculate the cost of the journey.

5.1.2. Scope

For a tourist visiting our country, considering the transportation to the desired destination, s/he will want to do this in the easiest way, in the shortest time and with the mean cost. The main purpose of this project is to help travelers to design travel routes in the best way before they start traveling. The main motivation of the traveler to use this application is to see the public transportation information and average fare information that will be used to plan its journey in the best way and to get the knowledge of the popular places on the route. The software itself will be developed using Swift Programming Language and popular place suggestions will be made by using CreateML.

5.1.3. Glossary

Term	**Definition**
SDD	Software Design Description
Stakeholder	People with any interest in project's outcome
A*	A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.
Mobile application	A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer.

Table 7: Glossary for SDD

5.1.4. References

[137] IEEE. "IEEE Std 1016-2009 IEEE Recommended Practice for Software Design Description". IEEE Computer Society. 20 July 2009.

5.1.5. Overview of the document

This section provides information about the contents of the rest of the document as follows: Section 2 describes the problem and details the design of this project along with the class architecture. Section 3 displays and explains the block diagram of the system, which is designed according to use cases in SRS document.

5.1.6. Motivation

We are a group of senior students in computer engineering department who are interested in machine learning. As a group, in this project we aimed to help tourists to plan their travels in the city, to see the average cost of transportation for public transport and travel. We are working on the IOS application that will determine the most optimal travel plan, and we have searched how to use the Swift Programming Language to improve the processes required for this subject. On the other hand, we have searched the CreateML for machine learning to improve this process.

5.2. Design Overview

5.2.1. Description of Problem

Especially when tourists visit our country, they encounter various problems in address descriptions about places to visit. The most important of these problems is the lack of foreign language knowledge in our country, so people cannot clearly describe the address. That's why tourists usually want to travel by taxi. However, as they will pay money a lot more than normal public transport when using a taxi, this situation of causes tourists to travel less. Also, there are some problems for local tourists in our country. For example, according to statistical research conducted in October, the highest number of accidents occurred in Ankara [138]. If people prefer public transport, the accident rate will also decrease. The software provided in this SDD enables local and foreign tourists to easily use public transport.

5.2.2. Technologies Used

This software will be developed with Swift programming language for use on iOS platforms. The target platform will be Mac OS X and development environment will be Xcode.

5.2.3. Architecture Design

5.2.3.1. Simulation Design Approach

5.2.3.1.1. Sequence Diagram

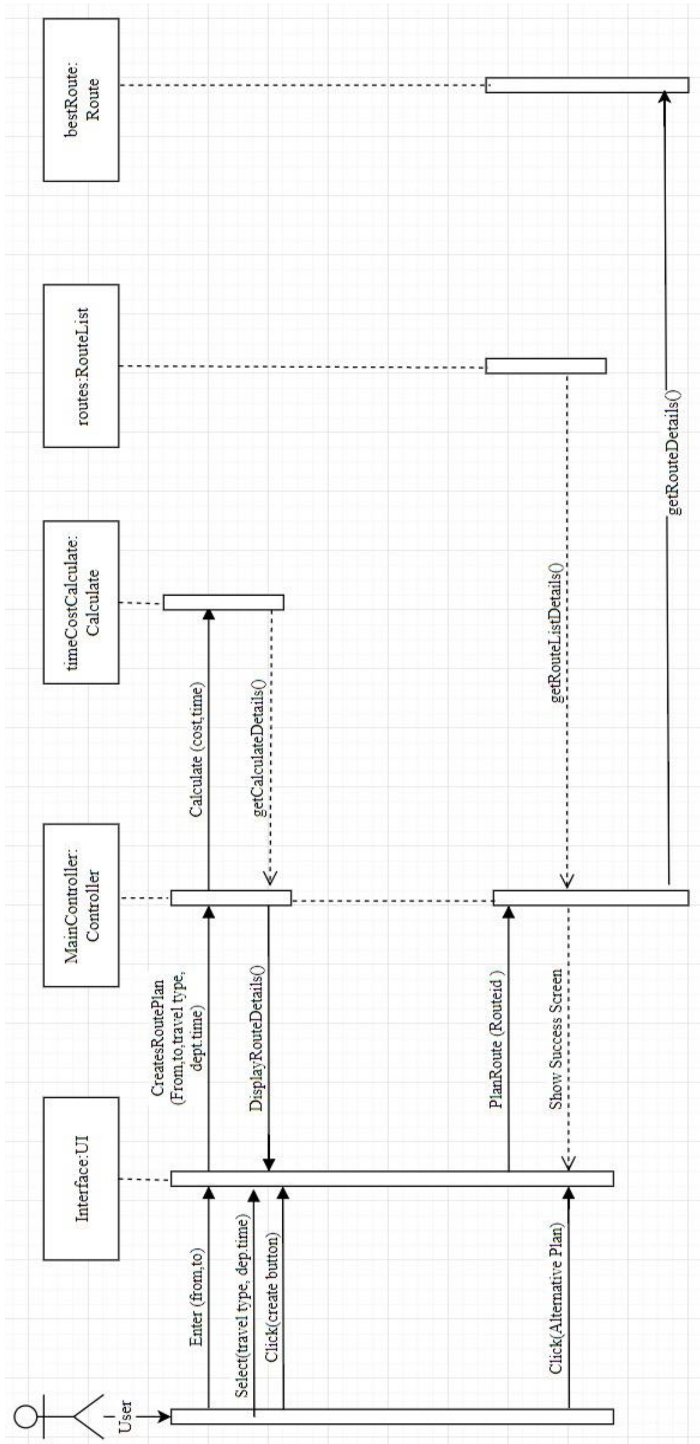

Figure 15: Sequence Diagram

Figure 15 displays sequence diagram. The user enters the location information he wants to use. It will create the route list via Routeid by selecting the type of travel type and departure time and clicking on the create button. After the created plan, all routes will be listed to the user in the interface section. Once the user selects the route he wants to use, the time and cost between routes are calculated again. The information returned is displayed to the user.

5.2.3.1.2. Database Diagram

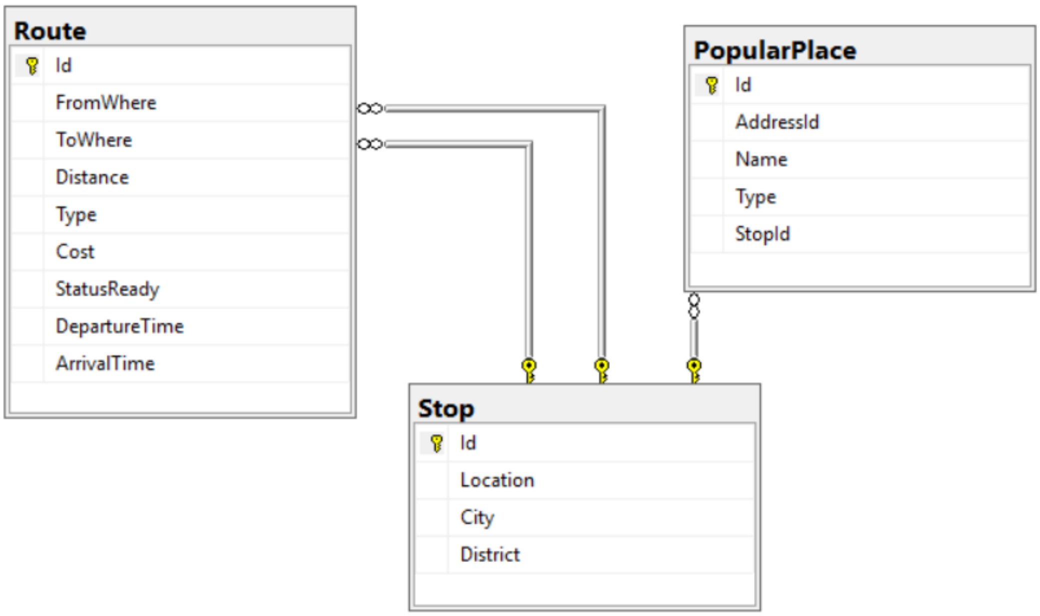

Figure 16: Route Finder Database Diagram

As stated in the Figure 16, there are related tables. The relationships between them are provided according to the choices made. In our system, the Route, PopularPlace and Stop cluster and document tables are linked together. Once the user has entered the location information of the start and destination, this information is kept in the Route table. The Route table is linked to the Stop table. Bus stop information is kept in this table. The Stop table is related with the PopularPlace table. The PopularPlace table holds the data of popular places. Based on this data from the database, system creates the most appropriate route between the user's starting position and the destination he wants to go to and provides the popular places on the route to the user.

5.2.3.1.3. Entity Relationship (ER) Model

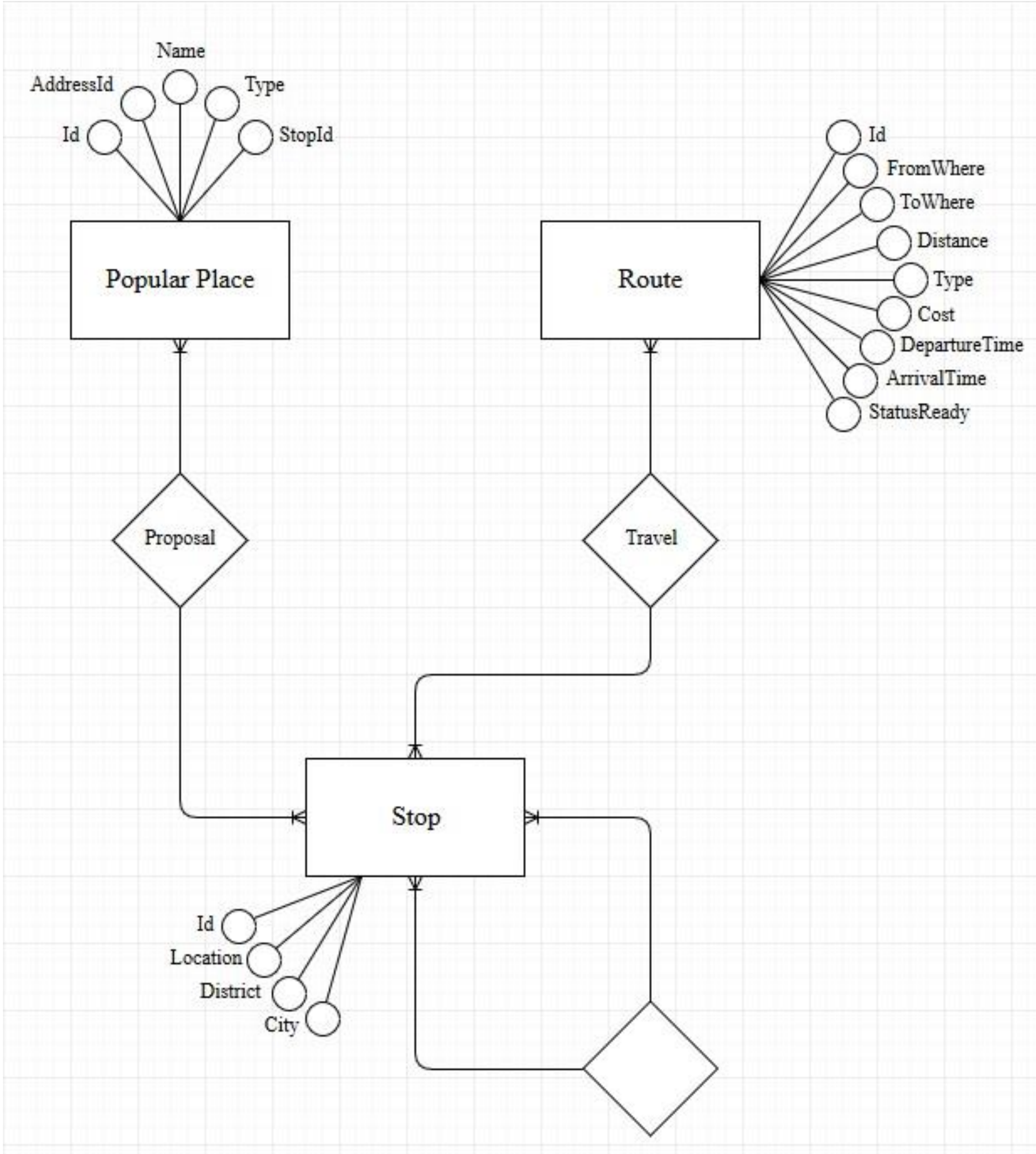

Figure 17: Route Finder ER Model

In the Figure 17, we showed the Route Finder Entity Relationship Diagram. The ER Diagram shows Tables, Attributes, Relationships between tables. The Stop table has a recursive relation. The Stop and Route has Travel relationship. This relationship takes datas from Stop and Route, after that this relationship creates Travel plan. The Stop and Popular Places table has Proposal relationship. This relationship takes datas from Stop and Popular Places tables and shows the popular places on the route.

5.2.3.1.4. Activity Diagram

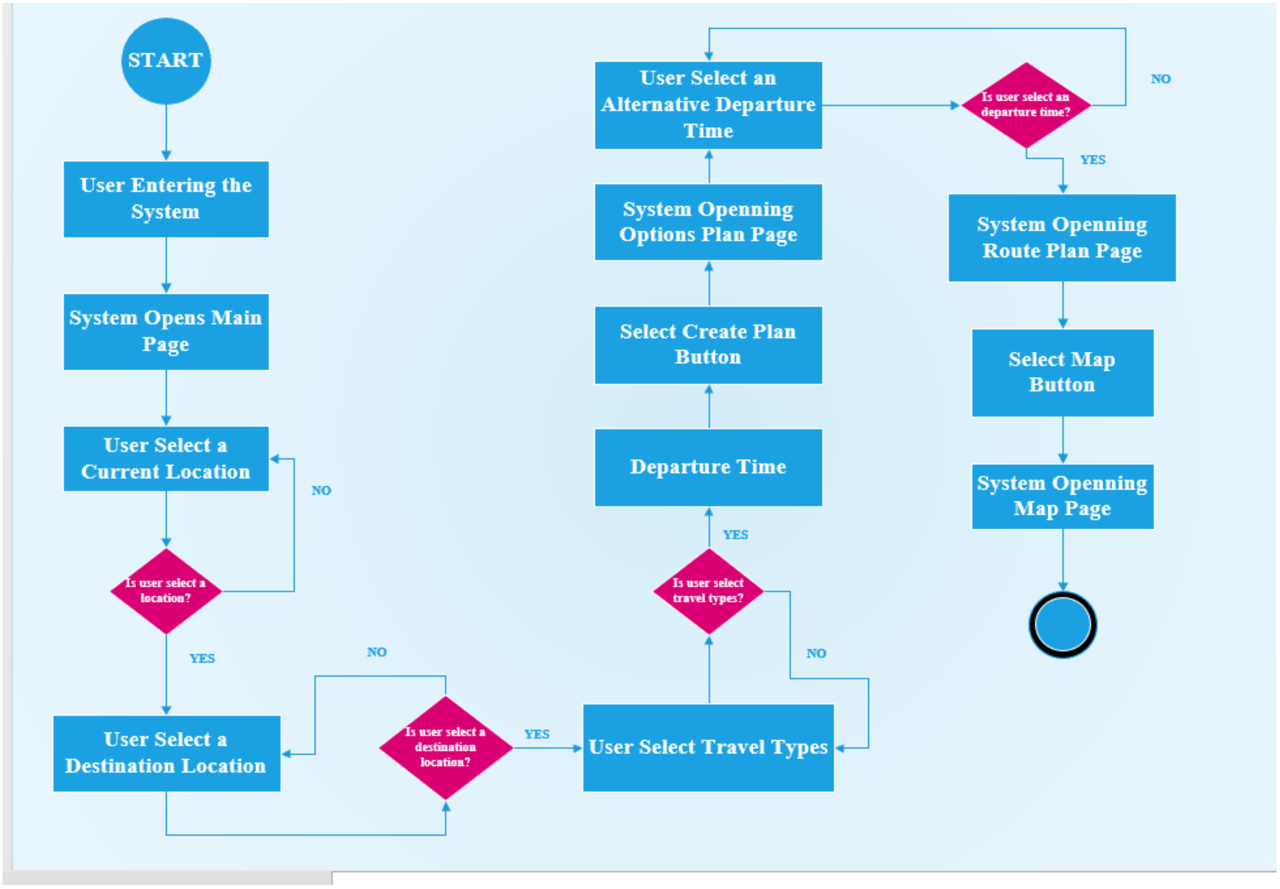

Figure 18: Route Finder Activity Diagram

The Figure18, shows that how the scenario generation works as an activity diagram. When the user open to the application, She/he sees the main page of the application. If the user inputs current/start location, destination location, travel type and departure time user can use "Create Plan" button. After that the system opens the plan page and user will select alternative departure time. After this step system will open the Route Plan Page if the user select "Map" button he/she can see travel map.

5.2.3.2. Architecture Design of Route Finder System

5.2.3.2.1. Create Plan

Summary: This system is used by participant. The participant can select his/her location, determine the target position, select the time to start the journey, select the travel types.

Actor: User

Precondition: The user must be connected to the Internet and keep the location settings open.

Basic Sequence:

I. The user selects his/her own location to begin the journey or sharing the current location into the system.
II. The user selects his/her own location to begin the journey or sharing the current location into the system.
III. The user selects his/her own location to begin the journey or sharing the current location into the system.
IV. The user selects his/her own location to begin the journey or sharing the current location into the system.
V. The user can start to create his/her plan by selecting the create plan button.

Exception:

● Location error due to cellular data or another external factor.

● Incorrect orientation if map information is out of date.

Post Conditions: User must fill start and destination location information, travel type information and start time information.

Priority: High

5.2.3.2.2. Plan Menu

Summary: This system is used by participant. The participant can see the details of the plan that he/she selected on the previous page on this page.

Actor: User

Precondition: The user must enter the information requested on the previous page correctly. In other words, click “Create Route” button.

Basic Sequence:

I. The user can select one of the alternative departure time.
II. The user can select one of the alternative departure time.
III. The user can select one of the alternative departure time.

Post Condition: None

Exception: Confirmation of an incorrect or out-of-date route

Priority: Medium

5.2.3.3. Project Plan

Figure 19: Project Plan

Descriptions for the work packages and their assignees are presented in Table 2. We decided that we can obtain best results if we distribute the workload evenly.

Work Package	Brief Description	Assignee(s)
Project Proposal Form	Making a project proposal to the consultant to get the project.	Team
Project Selection Form	The document of formalize the project, after the project advisor has accepted the project proposal.	Team
Research About Project	Investigation of similar projects and applications in our project.	Team
GitHub Repository	Opening and organizing of the project's GitHub page.	Team
Project Work Plan	Represents our whole development plan of the project.	Team
Literature Review	Researching, reading, documenting and making	Team

	comparison the articles about the project and the algorithms used in the project.	
Preparing Software Requirement Specification	Determination, research and documentation of what the project will do	Team
Preparing Project Webpage	Designing, writing and publishing of the project web page.	Team
Preparing Software Design Document	Specifying how to perform the actions specified in Software Requirement Specification and the design of the project.	Team
Project Report	Collecting every data, research, design and work in a single document.	Team
Meeting and Tracking Form	Document each meeting about the project and sign the project advisor.	Team
Preparing Presentation	Presentation slide designing and scripting the presentation flow.	Team
Dataset	Research about popular places. The names, google ratings, addresses and phone numbers of these places are written to the excel file according to the type of places.	Team
Learning Swift Programming Language	Researching and learning the Swift programming language.	Team

5.3. USE CASE REALIZATION

5.3.1. Project Components

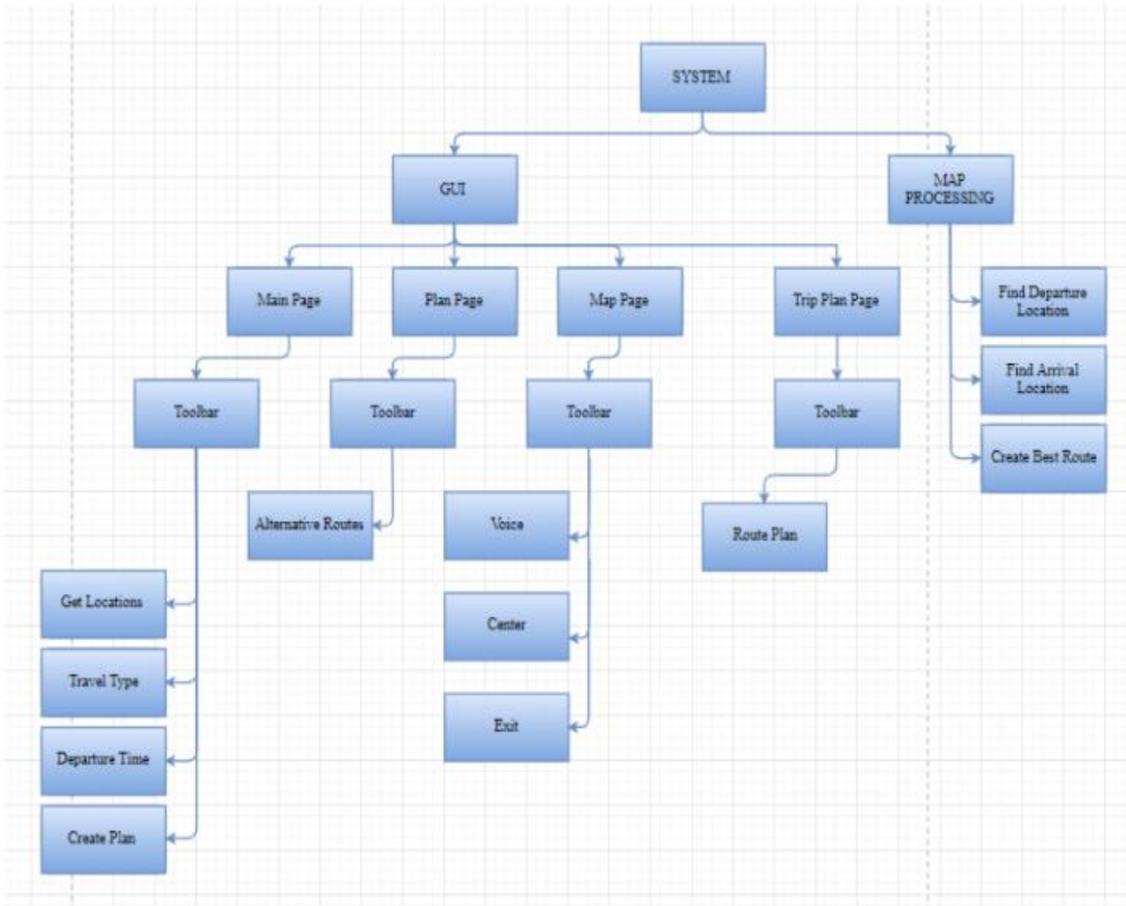

Figure 20: Route Finder Components Block Diagram

5.3.1.1. Brief Description of Block Diagram

The components of the Route Finder project are shown in the figure 20. All designed systems of the system are displayed in the block diagram in the figure. There are two main components of the system which have their own sub-systems.

5.3.1.1.1. GUI Design

The GUI design is responsible for the interaction between actors and the system. The components of the Route Finder project are shown in the figure 20. All designed systems of the system are displayed in the block diagram of the figure. The system has four main components with its own subsystems. The Main Page has a sub-system named toolbar, and the toolbar has four subsystems called Get Location, Travel Type, Departure Time, and Create Plan. The subsystem of the Plan Page called the toolbar has a subsystem called Alternative Routes. The subsystem of the Map Page called the toolbar has three subsystems, Voice, Center, and Exit. Finally, the subsystem of the Trip Plan Page's toolbar has a subsystem called Route Plan. The Main Page is a start page. Here the user is expected to enter the desired information to create a travel plan. In the Plan page offers an alternative route within the information received from the user in the Main Page. The desired type of route is created on the Trip Plan Page.

5.3.1.1.2. Map Processing

Map Processing is responsible for all locations processing. Find Departure/Arrival Location checks the accuracy of the user's location on Google Maps. If the location entered is not found by google map, the user is asked to re-enter the information. In the Create Route section, various routes are created with the location information received.

5.3.2. User Interface

In this section of the document, layout of the pages are presented in below figures.

5.3.2.1. Overview of User Interface

5.3.2.1.1. Main Page Layout

The main page is the interface that the user of the software will see when opening the application (Figure 21). On the main page, the user is asked to enter the location information and the location information of the destination that user wants to go to. Then the user is expected to choose the travel type. Finally, on this page, the user is expected to select the time to start travel. The user can create the plan by pressing the “Create Plan” button after entering the required information. After this button is pressed, the plan page is displayed to the user (Figure 22).

Figure 21: Main Page Layout

5.3.2.1.2. Plan Page Layout

In this page, the user sees alternative times to selected departure time on the main page. In addition to this, the user can select any of the alternative routes on this page.

Figure 22: Plan Page Layout

5.3.2.1.3. Trip Plan Page Layout

The Trip Plan page is the page shown after the user created the plan on the main page and plan page. In the other words, it is the page showing the details of the plan the user created. These details indicate;

- The departure time at which the user will take the bus.
- The station at which the user should take the bus/subway/train
- The average time to arrive at the destination
- The name of the station that should land
- The plan of the route along the journey
- The average fare that you will spend on your journey.

After checking all the details, the user clicks on the "MAP" button and the user sees route on the map (Figure 23).

Figure 23: Trip Plan Page Layout

5.3.2.1.4. Map Page Layout

On this page, the user can see the plan she created on the map.

Figure 24: Map Page Layout

6. Conclusion

In this CENG407 project, we talked about the mobile excursion utility that aims to find the best and shortest route according to the personal preferences of people using public transport. This is aimed at making the use of public transport more conscious and effective for the best user experience; Using the easy-to-use Google Maps interface and SDK, it allows mobile app users to easily search for their destination. As a result of this search, the application will inform the user of the minimum, maximum and average cost that can be given along the route by selecting the best route both time and way. Although many algorithms are used in the calculation of the route, we decided to use the A* algorithm which is the most current and most effective among these algorithms. In addition, our artificial intelligence-assisted application aims to improve the user's travel experience by showing the user the important places to travel around while traveling, which helps the user not to miss out on attractions such as parks, museums, banks or historic sites. In general, the mobile trip planner can help travelers to design their travel routes in the best way before the trip begins.

7. References

[1] Demirkol, Ö. E., & DEMİRKOL, A. (2003). DIJKSTRA VE BELLMAN-FORD EN KISA YOL ALGORİTMALARININ KARŞILAŞTIRILMASI. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 7(3), 5562.

[2] Flushing, E. F., Kudelski, M., Gambardella, L. M., & Di Caro, G. A. (2014, June). Spatial prediction of wireless links and its application to the path control of mobile robots. In Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on (pp. 218-227). IEEE.

[3] Casey, B., Bhaskar, A., Guo, H., & Chung, E. (2015). Design and implementation of an intermodal trip planner. Road & Transport Research: A Journal of Australian and New Zealand Research and Practice, 24(4), 45.

[4] VibhakarMohta. A Search Algorithm. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org

[5] Anonymous. Bellman Ford Algorithm. Programiz. Retrieved from https://www.programiz.com/dsa/

[6] Chumbley A., Moore K., Yang J. Floyd-Warshall Algorithm. Brilliant. Retrieved from https://brilliant.org/wiki/

[7] Crainic, T. G., Ricciardi, N., & Storchi, G., "Advanced freight transportation systems for congested urban areas", Transportation Research, Part C: Emerging Technologies, 12(2), 119 137, 2004.

[8] Taniguchi, E., Thompson, R. G., Yamada, T., & Van Duin, R., “City Logistics. Network modelling and intelligent transport systems”, Amsterdam : Pergamon, 2001.

[9] Crainic, T. G., Gendreau, M., & Potvin, J. Y., “Intelligent freight- transportation systems: Assessment and the contribution of operations research”, Transportation Research Part C: Emerging Technologies, 17(6), 541-557, 2009.

[10] Azevedo, J.A., & Martins, E.Q.V., “An algorithm for the multiobjective shortest path problem on acyclic networks”, Investigação Operacional, 11 (1), 52–69, 1991.

[11] Clímaco, J.C.N., & Martins, E.Q.V., “On the determination of the nondominated paths in a multiobjective network problem”, Methods in Operations Research 40, 255–258, 1981.

[12] Chumbley A., Moore K., Khim J. Johnson's Algorithm. Brilliant. Retrieved from https://brilliant.org/wiki/

[13] Anonymous (2015). Topological Sort. Algorithmist. Retrieved from http://www.algorithmist.com/index.php/

[14] Anonymous. University of Nottingham. Retrieved from http://www.cs.nott.ac.uk/~psznza/G5BADS04/graphs2.pdf

[15] Corley, H.W., & Moon, I.D., “Shortest paths in networks with vector weights”, Journal of Optimization Theory and Applications 46, 79–86, 1985.

[16] Tung, C.T., Chew, K.L., “A multicriteria Pareto-optimal path algorithm”, European Journal of Operational Research 62, 203–209, 1992.

[17] Luè, A., Ciccarelli, D., Colorni, A., “A GIS-Based Multi-Objective Travel Planner for Hazardous Material Transport in the Urban Area of Milan.. In: Bersani, C., Boulmakoul, A., Garbolino, E., & Sacile, R. (Eds)”, Advanced Technologies and Methodologies for Risk Management in the Global Transport of Dangerous Goods, 45, IOS Press, 2008.

[18] Hart, P. E., Nilsson, N. J., Raphael B., "A formal basis for the heuristic determination of minimum cost paths", Systems Science and Cybernetics, IEEE Transactions on 4.2, pp. 100-107, 1968.

[19] Cui, X., & Shi, H., “A*-based pathfinding in modern computer games”.International Journal of Computer Science and Network Security, 11(1), 125-130,2011.

[20] Wang, J. Y., Lin, Y. B., “Game AI: Simulating Car Racing Game by Applying Pathfinding Algorithms. International Journal of Machine Learning and Computing”, 2(1).

[21] K. Gutenschwager, S. Völker, A. Radtke and G. Zeller, "The shortest path: Comparison of different approaches and implementations for the automatic routing of vehicles," Proceedings of the 2012 Winter Simulation Conference (WSC), Berlin, 2012, pp. 1-12.

[22] Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. IEEE Transactions on Systems Science and Cybernetics 4:100–107.

[23] Hart, P. E., N. J. Nilsson, and B. Raphael. 1972. “Correction to ’A Formal Basis for the Heuristic Determination of Minimum Cost Paths”’. SIGART Newsletter 37:28–29.

[24] Luè, A., Colorni, A., “A multicriteria spatial decision support system for hazardous material transport In: R. Bisdorff, R, Dias, L., Meyer, P., Mousseau, V., & Pirlot, M. (Eds)”, Evaluation and Decision Models with Multiple Criteria: Case studies, Springer-Verlag, Berlin, 2015.

[25] Sastry, V.N., Janakiraman, T.N., Mohideen, S.I., “New algorithms for, multi objective shortest path problem”, Opsearch 40, 278–298, 2003.

[26] Sastry, V.N., Janakiraman, T.N., Mohideen, S.I., “New polynomial time algorithms to compute a set of Pareto optimal paths for multiobjective shortest path problems”, International Journal of Computer Mathematics 82, 289– 300, 2005.

[27] Russell, S., Norvig, P., “A modern approach. Artificial Intelligence”, Prentice-Hall, pp. 69, 1995.

[28] Anonymous. Dijkstra's Algorithm. Programiz. Retrieved from https://www.programiz.com/dsa/

[29] Chumbley A. Shortest Path Algorithms. Brilliant. Retrieved from https://brilliant.org/wiki/

[30] Anonymous. Breadth First Search or BFS for a Graph. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org

[31] Karp, R. M., “Reducibility Among Combinatorial Problems”, Complexity of Computer Computations, New York, Plenum, pp. 85-103.

[32] Hart, P. E., Nilsson, N. J., Raphael B., "A formal basis for the heuristic determination of minimum cost paths", Systems Science and Cybernetics, IEEE Transactions on 4.2, pp. 100-107, 1968.

[33] Cui, X., & Shi, H., “A*-based pathfinding in modern computer games”.International Journal of Computer Science and Network Security, 11(1), 125-130,2011.

[34] Wang, J. Y., Lin, Y. B., “Game AI: Simulating Car Racing Game by Applying Pathfinding Algorithms. International Journal of Machine Learning and Computing”, 2(1).

[35] Dorigo M., “Optimization, Learning and Natural Algorithms”, PhD thesis, Politecnico di Milano, Italy, 1992.

[36] Gambardella, L. M., Taillard, É., Agazzi, G., "Macs-vrptw: A multiple colony system for vehicle routing problems with time windows. In New ideas in optimization",1999.

[37] Parpinelli, R. S., Lopes, H. S., Freitas, A. A. "Data mining with an ant colony optimization algorithm". Evolutionary Computation, IEEE Transactions on, 6(4), pp.321-332,2002.

[38] J. Pearl, Heuristics: Intelligent Search Strategies for computer Problem Solving. Addison Wesley, Reading, Massachusetts, 1984.

[39] S. Russell and Peter Norving, Artificial Intelligence a Modern Approach. Prentice Hall, Upper Saddle River, New Jersey, 2003.

[40] H. Halaoui, Smart Traffic Online System (STOS): Presenting Road Networks with time-Weighted Graphs". IEEE International Conference on Information Society (i-Society 2010) London, UK. June 2010, pp. 349-356.

[41] Google Earth Blog Google Earth Data Size, Live Local, New languages coming Available: http://whatis.techtarget.com/definition/Google-Maps. Retrieved: September, 2015.

[42] H. Halaoui, "Smart Traffic Systems: Dynamic A*Traffic in GIS Driving Paths Applications". Proceeding of IEEE CSIE09, IEEE, Los Angeles, USA. March, 2009, pp. 626-630.

[43] H. Halaoui," Intelligent Traffic System: Road Networks with Time-Weighted Graphs". International Journal for Infonomics (IJI), Volume 3, Issue 4, December 2010, pp. 350-359.

[44] Google Maps. Available: https:// Maps.google.com. Retrieved: September, 2015.

[45] H. Halaoui. "Spatial and Spatio-Temporal Databases Modeling: Approaches for Modeling and Indexing Spatial and Spatio-Temporal Databases". VDM Verlag, 2009.

[46]Arsanjani, J. J., A. Zipf, P. Mooney, and M. Helbich, ed. 2015. OpenStreetMap in GI-Science: Experiences, Research, and Applications. Springer International Publishing.

[47] Luxen, D., and C. Vetter. 2011. Real-time routing with OpenStreetMap data. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 513–516. GIS '11. New York, USA: ACM.

[48] Ozimek, A., and D. Miles. 2011. Stata utilities for geocoding and generating travel time and travel distance information. Stata Journal 11(1): 106–119.

[49] Voorheis, J. 2015. mqtime: A Stata tool for calculating travel time and distance using MapQuest web services. Stata Journal 15(3): 845–853.

[50] Huber, S., & Rust, C. (2016). Calculate travel time and distance with OpenStreetMap data using the Open Source Routing Machine (OSRM). Stata J, 16, 416-423.

[51] Kumbhar, M., Survase, M., Mastud, P., Salunke, A., & Sirdeshpande, S. (2016). Real Time Web Based Bus Tracking System. International Research Journal of Engineering and Technology, 3(02), 632-635.

[52] Fan, L., & Mumford, C. L. (2010). A metaheuristic approach to the urban transit routing problem. Journal of Heuristics, 16(3), 353-372.

[53] Dr. Saylee Gharge, Manal Chhaya, Gaurav Chheda, Jitesh Deshpande, "Real time bus monitoring system using GPS," An International Journal of Engineering Science and Technology, Vol. 2, Issue 3, June 2012.

[54] Abid Khan, Ravi Mishra, "GPS-GSM based tracking system," International Journal of Engineering Trends and Technology, Vol. 3, Issue 2, pp: 161-164, 2012.

[55] Swati Chandurkar, Sneha Mugade, Sanjana Sinha, Pooja Borkar, "Implementation of real time bus monitoring and passenger information system," International Journal of Scientific and Research Publications, Vol. 3, Issue 5, May 2013.

[56] Pankaj Verma, J. S. Bhatia, "Design and development of GPS-GSM based tracking system with Google map based monitoring," International Journal of Computer Science, Engineering and Applications, Vol. 3, No.3, June 2013.

[57] Madhu Manikya Kumar, K. Rajesekhar, K. Pavani, "Design of punctually enhanced bus transportation system using GSM and Zigbee," International Journal of Research in Computer and Communication Technology, Vol. 2, Issue 12, December 2013.

[58] N. Vijayalashmy, V. Yamuna, G. Rupavani, A. Kannaki@VasanthaAzhagu," GNSS based bus monitoring and sending SMS to the passengers," International Journal of Innovative Research in Computer and Application Engineering, Vol. 2, Special Issue 1, March 2014.

[59] R. Manikandan, S. Niranjani, "Implementation on real time transportation information using GSM query response system," Contemporary Engineering Sciences, Vol. 7, No.11, pp: 509-514, 2014.

[60] G. Raja, D. NaveenKumar, G. Dhanateja, G. V. Karthik, Y. Vijay Kumar, "Bus Position monitoring system to facilitate the passengers," International Journal of Engineering Science and Advanced Technology(IJESAT), Volume-3, Issue-3, pp: 132-135, 2014.

[61] S. Eken, A. Sayar, "A Smart bus tracking system based on location aware services and QR codes," IEEE International Symposium on Innovations in Intelligent and Applications Proceedings, pp: 299-309, 2014.

[62] Vishal Bharte, Kaustubh Patil, Lalit Jadhav, Dhaval Joshi , " Bus Monitoring System Using Polyline Algorithm ," International Journal of Scientific and Research Publications, Volume 4, Issue 4, April 2014.

[63] Shekhar Shinde , Vijaykumar Nagalwar , Nikhil Shinde , B.V.Pawar, " Design of E-City Bus Tracking System ,"Int. Journal of Engineering Research and Applications, ISSN : 2248-9622,

[64] Biesinger, J., Burr, L., & Terry, T. (2017). Final Group Project-Optimal Route Finding

[65] Amirgholy, M., Golshani, N., Schneider, C., Gonzales, E. J., & Gao, H. O. (2017). An advanced traveler navigation system adapted to route choice preferences of the individual users. International Journal of Transportation Science and Technology, 6(4), 240-254.

[66] Szabó, C., & Sobota, B. (2012). Path-finding algorithm application for route-searching in different areas of computer graphics. In New Frontiers in Graph Theory. InTech.

[67] Verma, R., Shrivastava, A., Mitra, B., Saha, S., Ganguly, N., Nandi, S., & Chakraborty, S. (2016, April). UrbanEye: An outdoor localization system for public transport. In INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE (pp. 1-9). IEEE.

[68] H. Aly and M. Youssef, "Dejavu: An accurate energy-efficient outdoor localization system," in Proc. of SIGSPATIAL'13. New York, NY, USA: ACM, 2013, pp. 154–163.

[69] R. Mandal, N. Agarwal, S. Nandi, P. Das, A. Anvit, S. Sanyal, and S. Saha, "Stoppage pattern analysis of public bus GPS traces in developing regions," in Proc. of PerCom '15, 2015, pp. 276–279.

[70] P. Zhou, Y. Zheng, and M. Li, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," IEEE Transactions on Mobile Computing, vol. 13, no. 6, pp. 1228–1241, June 2014.

[71] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, "Easytracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones," in Proc. of SenSys '11. New York, NY, USA: ACM, 2011, pp. 68–81.

[72] T. Vincenty, "Transformation of co-ordinates between geodetic systems," Survey Review, vol. 18, no. 137, pp. 128–133, 1965.

[73] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakr- ishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in Proc. of MobiSys '08. ACM, 2008, pp. 29–39.

[74] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, "Wolverine: Traffic and road condition estimation using smartphone sensors," in Proc. of COMSNETS '12, Jan 2012, pp. 1–6.

[75] Halaoui, H. (2015). Smart navigation: using artificial intelligent heuristics in navigating multiple destinations. Proceedings of SOTICS.

[76] Casey, B., Bhaskar, A., Guo, H., & Chung, E. (2015). Design and implementation of an intermodal trip planner. Road & Transport Research: A Journal of Australian and New Zealand Research and Practice, 24(4), 45.

[77] Ahuja, RK, Mehlhorn, K, Orlin, J & Tarjan, RE 1990, ‘Faster algorithms for the shortest path problem’, Journal of the Association for Computing Machinery, 37, 213–23.

[78] Casey, B, Bhaskar, A, Guo, H & Chung, E 2014, ‘Critical review of time dependent shortest path algorithms, an intermodal trip planner perspective’, Transport Reviews: A Transnational Transdisciplinary Journal, DOI: 10.1080/01441647.2014.921797.

[79] Dantzig, GB 1960, ‘On the shortest route through a network’, Management Science, 6, 187–90.

[80] Dial, RB 1969, ‘Algorithm 360: shortest-path forest with topological ordering [H]’, Communications of the ACM, 12, 632–33.

[81] Bellman, RE 1958, ‘On a routing problem’, Quarterly of Applied Mathematics, 16, 87–90.

[82] Bousquet, A, Constans, S & El Faouzi, N-E 2009, ‘On the adaptation of a label-setting shortest path algorithm for one-way and two-way routing in multimodal urban transport networks’, Proc. International Network Optimization Conference 2009, available at http://www.di.unipi.it/ optimize/Events/proceedings/.

[83] Dijkstra, EW 1959, ‘A note on two problems in connexion with graphs’, Numerische Mathematik, 1, 269–71.

[84] Fu, L, Sun, D & Rilett, LR 2006, ‘Heuristic shortest path algorithms for transportation applications: State of the art’, Computers and Operations Research, 33, 3324–43.

[85] Goldberg, AV. & Harrelson, C 2005, ‘Computing the shortest path: A search meets graph theory’, Proc. 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 156–65.

[86] Hart, PE, Nilsson, NJ & Raphael, B 1968, ‘A formal basis for the heuristic determination of minimum cost paths’, IEEE Transactions on Systems Science and Cybernetics, 4, 100–07.

[87] Johnson, EL 1972, ‘On shortest paths and sorting’, Proc. ACM Annual Conference 1972 – Volume 1, Boston, Massachusetts, United States, pp. 510–17.

[88] Research Record: Journal of the Transportation Research Board, No. 2284, 40–46.

[89] Miska, MP, Nantes, A, Torday, A, Jin, H & Chung, E 2013, ‘Model-free networks as basis for transport data hub’, Transportation Research Board 92nd Annual Meeting, Compendium of Papers, Paper 13-3481.

[90] Pohl, I 1969, Bi-directional and Heuristic Search in Path Problems. Department of Computer Science, Stanford University.

[91] Khani, A, Lee, S, Hickman, M, Noh, H & Nassir, N 2012, 'Intermodal path algorithm for time-dependent auto network and scheduled transit service', Transportation

[92] Schulz, F 2005, Timetable Information and Shortest Paths. PhD Dissertation, Karlsruhe Univ.

[93] Translink (Online), Translink Public Transport Performance Data - General Transit Feed Specification Translink, Brisbane. Available at: http://translink.com.au/about-translink/reporting-and-publications/public-transport-performance- data.

[94] Nassir, N, Khani, A, Hickman, M & Noh, H 2012, 'Algorithm for intermodal optimal multidestination tour with dynamic travel times', Transportation Research Record: Journal of the Transportation Research Board, No. 2283, 57–66.

[95] Nalepa, G. J., & Bobek, S. (2014). Rule-based solution for context-aware reasoning on mobile devices. Computer Science and Information Systems, 11(1), 171-193.

[96]https://scholarcommons.usf.edu/cgi/viewcontent.cgi?referer=https://scholar.google.com.tr/&httpsredir=1&article=1740&context=jpt

[97]Neff, J. and M. Dickens. 2017. 2016 Public Transportation Fact Book, 67th Edition. Washington, DC: American Public Transportation Association, Retrieved from http://www.apta.com/ resources/statistics/Documents/FactBook/2016-APTA-Fact-Book.pdf

[98]Vuchic, V. 2007. Urban Transit Systems and Technology. Hoboken, NJ: John Wiley & Sons, Inc.

[99] Wikibooks. 2017. Fundamentals of Transportation/Transit. Retrieved from https://en.wikibooks. org/wiki/Fundamentals_of_Transportation/Transit

[100] Richard, M. 2009. "Amount of Space Required to Transport People by Car, Bus, or Bicycle." Treehugger. Retrieved from https://www.treehugger.com/cars/amount-of-space-requiredto-transport-people-by-car-bus-orbicycle.html

[101] Jittrapirom, P., Caiati, V., Feneri, A. M., Ebrahimigharehbaghi, S., Alonso González, M. J., & Narayan, J. (2017). Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges.

[102] NEMÞANU, F., SCHLINGENSIEPEN, J., BURETEA, D., & IORDACHE, V. (2016). Mobility as a service in smart cities. RESPONSIBLE ENTREPRENEURSHIP VISION, DEVELOPMENT AND ETHICS, 425.

[103] Deloitte. Deloitte: Retrieved from https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/consumerbusiness/deloitte-nl-cb-ths-rise-of-mobility-as-a-service.pdf

[104] MaaS Alliance. Retrived From: https://maas-alliance.eu
[105]ÇELİKBAŞ, M. K. (2018, July 25). Prezi: Retrived From: https://prezi.com/6visatymves0/maas/

[106] Apple. *Developer.* Apple: Retrieved From: https://developer.apple.com/documentation/createml adresinden alındı

[107] Apple. *MacOs.* Apple: Retrieved From: https://www.apple.com/tr/macos/what-is/

[108] Apple. *Swift.* Apple: Retrieved From: https://www.apple.com/tr/swift/

[109] techopedia, «Mobile Application,» 16 11 2018. [Online]. Available:https://www.techopedia.com/definition/2953/mobile-application-mobile-app.

[110] E. Ö. -. A. ER, «Software Requirements Specification,» 12 Mar 2018. [Online]. Available: https://github.com/CankayaUniversity/ceng-407-408-2017-2018-project-blood-vessel-segmentation/wiki/Software-Requirements-Specification. [Accessed: 2018 11 16].

[111] S. W. H. T. W. S. w. Examples, «Software Requirements Specifications,» The Business of IT Blog, 18 September 2017. [Online].

Available: https://www.bmc.com/blogs/software- requirements- specification- how-to-write-srs-with-examples/. [Accessed: 16 11 2018].

[112] GeeksforGeeks. (2019). How to write a good SRS for your Project - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/ [Accessed 7 Jan. 2019].

[113] Flushing, E. F., Kudelski, M., Gambardella, L. M., & Di Caro, G. A. (2014, June). Spatial prediction of wireless links and its application to the path control of mobile robots. In Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on (pp. 218-227). IEEE.

[114] Casey, B., Bhaskar, A., Guo, H., & Chung, E. (2015). Design and implementation of an intermodal trip planner. Road & Transport Research: A Journal of Australian and New Zealand Research and Practice, 24(4), 45.

[115] BMC Blogs. (2019). Software Requirements Specifications: How To Write SRS with Examples. [online] Available at: https://www.bmc.com/blogs/software-requirements-specification-how-to-write-srs-with-examples/ [Accessed 7 Jan. 2019].

[116] Uml-diagrams.org. (2019). Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors).. [online] Available at: https://www.uml-diagrams.org/use-case-diagrams.html [Accessed 7 Jan. 2019].

[117] Sourcemaking.com. (2019). Design Patterns and Refactoring. [online] Available at: https://sourcemaking.com/uml/modeling-business-systems/external-view/use-case-diagrams [Accessed 7 Jan. 2019].
[118] Crainic, T. G., Ricciardi, N., & Storchi, G., “Advanced freight transportation systems for congested urban areas”, Transportation Research, Part C: Emerging Technologies, 12(2), 119-137, 2004.
[119] Taniguchi, E., Thompson, R. G., Yamada, T., & Van Duin, R., “City Logistics. Network modelling and intelligent transport systems”, Amsterdam : Pergamon, 2001.
[120] Crainic, T. G., Gendreau, M., & Potvin, J. Y., “Intelligent freight- transportation systems: Assessment and the contribution of operations research”, Transportation Research Part C: Emerging Technologies, 17(6), 541-557, 2009.
[121] itysdk.eu. (2019). [online] Available at: https://www.citysdk.eu/wp-content/uploads/2013/09/DELIVERABLE_WP4_TA_SRS_0.21.pdf [Accessed 7 Jan. 2019].
[122] Visual-paradigm.com. (2019). What is Use Case Specification?. [online] Available at: https://www.visual-paradigm.com/guide/use-case/what-is-use-case-specification/ [Accessed 7 Jan. 2019].
[123] SearchSoftwareQuality. (2019). What is use case? - Definition from WhatIs.com. [online] Available at: https://searchsoftwarequality.techtarget.com/definition/use-case [Accessed 7 Jan. 2019].
[124] Anonymous. University of Nottingham. Retrieved from http://www.cs.nott.ac.uk/~psznza/G5BADS04/graphs2.pdf
[125] Corley, H.W., & Moon, I.D., “Shortest paths in networks with vector weights”, Journal of Optimization Theory and Applications 46, 79–86, 1985.
[126] Shaw, M., & Garlan, D. (1996). Software architecture (Vol. 101). Englewood Cliffs: Prentice Hall.
[127] Jacobson, I., & Ng, P. W. (2004). Aspect-oriented software development with use cases (addison- wesley object technology series). Addison-Wesley Professional.
[128] Martin, R. C. (2002). Agile software development: principles, patterns, and practices. Prentice Hall.
[129] Gomaa, H. (2001, July). Designing concurrent, distributed, and real-time applications with UML. In Proceedings of the 23rd international conference on software engineering (pp. 737-738). IEEE Computer Society.
[130] Shaw, M., & Garlan, D. (1996). Software architecture (Vol. 101). Englewood Cliffs: Prentice Hall.
[131] Leffingwell, D., & Widrig, D. (2000). Managing software requirements: a unified approach. Addison-Wesley Professional.
[132] Constantine, L. L., & Lockwood, L. A. (1999). Software for use: a practical guide to the models and methods of usage-centered design. Pearson Education.
[133] Fielding, R. T., & Taylor, R. N. (2000). Architectural styles and the design of network-based software architectures (Vol. 7). Irvine, USA: University of California, Irvine.

[134] Gomaa, H. (2011). Software modeling and design: UML, use cases, patterns, and software architectures. Cambridge University Press.
[135] Richard N. Taylor, Nenad Medvidovic, and Eric M. Dashofy.Software Architecture: Foundations, Theory, and Practice.John Wiley & Sons, © 2010, 736 pages. ISBN-10: 0470167742. ISBN-13: 978- 0470167748
[138] Anonymous.Trafik. Retrieved from [http://www.trafik.gov.tr/SiteAssets/istatistik/Ekim18.pdf]
[139] Richard N. Taylor, Nenad Medvidovic, and Eric M. Dashofy.Software Architecture: Foundations, Theory, and Practice.John Wiley & Sons, © 2010, 736 pages. ISBN-10: 0470167742. ISBN-13: 978- 0470167748
[140] Gomaa, H. (2011). Software modeling and design: UML, use cases, patterns, and software architectures. Cambridge University Press.
[141] Fielding, R. T., & Taylor, R. N. (2000). Architectural styles and the design of network-based software architectures (Vol. 7). Irvine, USA: University of California, Irvine.
[142] Constantine, L. L., & Lockwood, L. A. (1999). Software for use: a practical guide to the models and methods of usage-centered design. Pearson Education.
[143] Gomaa, H. (2001, July). Designing concurrent, distributed, and real-time applications with UML. In Proceedings of the 23rd international conference on software engineering (pp. 737-738). IEEE Computer Society.
[144] Jacobson, I., & Ng, P. W. (2004). Aspect-oriented software development with use cases (addison-wesley object technology series). Addison-Wesley Professional.