**LOVELY PROFESSIONAL UNIVERSITY**

**END TERM PROJECT**

# SIMPLE NAVIGATION SYSTEM

**Submitted by:**


| | | |
|---|---|---|
| S.praveen Kumar | 11809946 | K18RH |
| N.Kavi Prakash | 11804098 | K18RH |
| Gone Varun Raj | 11810097 | K18RH |
| Suman Shreevastav | 11804106 | K18RH |

**DEPARTMENT OF INTELLIGENT SYSTEM**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**LOVELY PROFESSIONAL UNIVERSITY**

**( JALANDHAR)**

# Table of Contents

# Student Declaration

This is to declare that this report has been written by us.no part of the report is copied from other source. All information included from the other source have been duly acknowledged. We aware that if any part of the report is found to be copied, We shall take full responsibility for it.

BY,
N.Kavi Prakash
11804098

S.Praveen Kumar
11809946

Gone Varun Raj
11810097

Suman Shreevastav
11804106

## Abstract

Navigation system is very Essential for every person, let it be travelling from one place to another or to find the distance between multiple locations or to get information about a place. This paper gives the review of different web-based applications which are used in Mumbai for travelling by many users. This paper gives us a study of different applications used for travelling to multiple destinations via different modes of transportation i.e. Bus, Train, Taxi, Metro, etc. Each and every application have some pros and cons because the working of all the applications are different and Each and every application tries to solve a single problem

## REASEARCH METHODOLOGY

### 1) Introduction:-

In today's life, Time and Money are very important; no one wants to waste their time and money, be it travelling on a long route, instead of a short route. or it can be giving extra money for public transportation as you were unaware of the cost of the transportation.Google maps has been a great application the travellers asit provides the user with the route, they want to travel. The user can find the route from source to destination and even Google maps can guide those using GPS to reach their destination. Google maps help users to choose different types of fields. Navigation systems have become common in the last decade. They are mainly classified as driver comfort systems (Brookhuis et al.), but their economic and ecological benefits (due to shorter routes) are unequivocal. Although we know a lot about how navigation systems affect driving.

The current study focuses on the use of nomadic navigation systems: how and when they are used in naturalistic driving and whether this can affect driving speed. First, we address the literature regarding 2 distinct tasks involved with navigation systems, namely, following route guidance instructions and operating the system.

### 2) Objectives of Project Report:-

The objective of the project is to provide navigate the person to find shortest path to destination and to show directions in which it leads to the destination using shortest path to travel.

- containing maps, which may be displayed in human readable format via text or in a graphical format
- via sensors, maps, or information from external sources
- providing suggested directions to a human in charge of a vehicle or vessel via text or speech
- providing directions directly such as a robotic probe or guided missile
- providing information on nearby vehicles or vessels, or other hazards or obstacles
- providing information and suggesting alternative directions
- simultaneous localization and mapping
- acoustic positioning for underwater navigation

### Sources of Data Collection: -

Data Collection is key part of project work. There are two types of data collection, first is primary source and second is secondary of data collection.

**Primary Sources: -**

The primary data includes company profile, financial statements, and documents relevant have been obtained from Delhi metro.

**Secondary Sources:-**

The secondary data relating to the procedures of assessment of project financing in MRTS, RBI guidelines etc. have been sourced from reference books and websites.

**Scope of the project:-**

GPS is the future technology which helps in navigation and other source of help in which the world depends on it. Every electronic devices which made today has GPS for navigation. So this technology helps the people in the world to travel even without the help of guides. It make the travel easier and easy

**Limitation of the study:-**

- The time limitation is the most important problem to collect the various information.
- It requires lot of time & is more expensive

# LITERATURE REVIEW

46 Satellite Navigation Systems: Policy, Commercial and Technical Interaction areas utilizing GNSS are now transport, agriculture, science, and spacecraft navigation, while hand-held GPS receivers are used for leisure activities such as hiking and boating. The aviation industry is accepting the use of GNSS and augmented systems for long-haul flights, landing at airports and airspace surveillance, to improve the effectiveness and security of the increasingly congested airspace. Unmanned Aerial Vehicles (UAY) take navigation systems a step further and closer to fully autonomous aircraft. In emergency situations GNSS are useful. For example, a hiker lost in fog can find his way home safely, or a ship in distress can be located. In the near future, when devices combining cell phones and GNSS receivers become common, anyone involved in, or witness to, an accident will be able to inform the emergency services of its exact location when calling for assistance. The environment will also benefit from the use of these systems.

A farmer will no longer have to fertilize the entire field with a fixed amount per square meter, sea and coastal regions will suffer from fewer oil spills when ships are navigating safely around dangerous cliffs, and the air pollution in cities will be reduced by traffic management systems, which remove cars from dense traffic and areas with many traffic lights. Also, the scientific community and the spacecraft industry itself are increasingly using GNSS. On board satellites or aircraft, GNSS receivers are aiding studies in geology, hydrology, meteorology and oceanography, leading to an increasing knowledge of both the Earth's climate and solid Earth dynamics, as well as providing the possibility for early warning systems for storms and earthquakes. GNSS for orbit determination of spacecraft used for more than a decade are now finding other application areas such as altitude determination and maintaining a formation-flying constellation. For example, the navigation system of the Automated Transport Vehicle (ATV), which will perform rendezvous-and-docking operations with the International Space Station (ISS), includes GPS receivers.

Aspects of Applications Navigation systems today rely on either a pure or an augmented GNSS, while only interplanetary navigation relies on a totally ground-based navigation system. The range of applications covers a wide spectrum besides the military arena that Originally spurred the creation of the system.

46 Satellite Navigation Systems: Policy, Commercial and Technical Interaction areas utilizing GNSS are now transport, agriculture, science, and spacecraft navigation, while hand-held GPS receivers are used for leisure activities such as hiking and boating. The aviation industry is accepting the use of GNSS and augmented systems for long-haul flights, landing at airports and airspace surveillance, to improve the effectiveness and security of the increasingly congested airspace. Unmanned Aerial Vehicles (UAY) take navigation systems a step further and closer to fully autonomous aircraft. In emergency situations GNSS are useful. For example, a hiker lost in fog can find his way home safely, or a ship in distress can be located. In the near future, when devices combining cell phones and GNSS receivers become common, anyone involved in, or witness to, an accident will be able to inform the emergency services of its exact location when calling for assistance.

The environment will also benefit from the use of these systems. A farmer will no longer have to fertilize the entire field with a fixed amount per square meter, sea and coastal regions will suffer from fewer oil spills when ships are navigating safely around dangerous cliffs, and the air pollution in cities will be reduced by traffic management systems, which remove cars from dense traffic and areas with many traffic lights. Also, the scientific community and the spacecraft industry itself are increasingly using GNSS. On board satellites or aircraft, GNSS receivers are aiding studies in geology, hydrology, meteorology and oceanography, leading to an increasing knowledge of both the Earth's climate and solid Earth dynamics, as well as providing the possibility for early warning systems for storms and earthquakes. GNSS for orbit determination of spacecraft used for more than a decade are now finding other application areas such as altitude determination and maintaining a formation-flying constellation. For example, the navigation system of the Automated Transport Vehicle (ATV), which will perform rendezvous-and-docking operations with the International Space Station (ISS), includes GPS receivers

## Background

The main idea of the project is to show the shortest path from the location of start to destination in efficient manner. Navigation system uses shortest path to find the route for travel. So we used A* algorithm and Dijkstra's algorithm which is used to find the shortest path or route Between the start and end. Dijkstra's algorithm finds a shortest path tree from a single source node, by building a set of nodes that have minimum distance from the source

We need to maintain the path distance of every vertex. We can store that in an array of size v, where v is the number of vertices. We also want to able to get the shortest path, not only know the length of the shortest path. For this, we map each vertex to the vertex that last updated its path length.Once the algorithm is over, we can backtrack from the destination vertex to the source vertex to find the path.

A minimum priority queue can be used to efficiently receive the vertex with least path distance. We have implemented the project by every location of the LPU and then we added with the distance using graph so which helps in finding the shortest path and connect other nodes. An inertial navigation system comprises two-distinct parts; the first is the IMU (inertial measurement unit)—sometimes called the IRU (inertial reference unit). This is the collective name for the accelerometers and gyros that provide acceleration and angular velocity measurements. The second part is the navigation computer. The navigation computer takes measurements from the IMU and uses them to calculate the relative position, orientation and velocity of the INS.

There are essentially two kinds of navigation computers in use; stabilised platforms and strap-down navigators. Stabilised platforms use real, spinning mechanical gyroscopes to stabilise a platform that rotates independently to the INS. So, as the inertial navigation system rotates, the stabilised platform inside it does not. In this way, the system learns about its orientation and can make use of the measurements from the accelerometers. The downsides of this type of system are gimbal lock (see the section on gyros for a full explanation), the high cost and complexity.

In contrast, the sensors inside a strap-down navigator do not move independently of the INS. They are, if you like, strapped down. This overcomes many of the problems associated with stabilised platforms and is the main reason why inertial navigation systems are now affordable to a lot more people. Unlike the spinning, mechanical gyros inside a stabilised platform, the gyros used inside a strap-down navigator are typically MEMS (microelectromechanical systems), which don't appear to have any moving parts. In fact, it's better to think of them as angular rate sensors, rather than gyros, although that's what they're typically called.

Dijkstra's algorithm is an iterative algorithm that provides us with the shortest path from one particular starting node (**a** in our case) to all other nodes in the graph.

To keep track of the total cost from the start node to each destination we will make use of the **distance** instance variable in the **Vertex** class. The **distance** instance variable will contain the current total weight of the smallest weight path from the start to the vertex in question. The algorithm iterates once for every vertex in the graph; however, the order that we iterate over the vertices is controlled by a **priority queue** (actually, in the code, I used **heap**).

The value that is used to determine the order of the objects in the priority queue is **distance**. When a vertex is first created **distance** is set to a very large number.

When the algorithm finishes the distances are set correctly as are the predecessor (**previous** in the code) links for each vertex in the graph.

A node is just some object, and an edge is a connection between two nodes. Depicted above an *undirected graph*, which means that the edges are bidirectional. There also exist *directed graphs*, in which each edge also holds a direction.

Both nodes and edges can hold information. For example, if this graph represented a set of buildings connected by tunnels, the nodes would hold the information of the name of the building (e.g. the string “Library”), and the edges could hold information such as the length of the tunnel.

Graphs have many relevant applications: web pages (nodes) with links to other pages (edges), packet routing in networks, social media networks, street mapping applications, modeling molecular bonds, and other areas in mathematics, linguistics, sociology, and really any use case where your system has interconnected objects.

We want to find the shortest path in between a source node and all other nodes (or a destination node), but we don’t want to have to check EVERY single possible source-to-destination combination to do this, because that would take a really long time for a large graph, and we would be checking a lot of paths which we should know aren’t correct! So we decide to take a *greedy* approach! You have to take advantage of the times in life when you can be greedy and it doesn’t come with bad consequences!

In the code, we create two classes: **Graph**, which holds the master list of vertices, and **Vertex**, which represents each vertex in the graph.

The function **dijkstra()** calculates the shortest path. The **shortest()** function constructs the shortest path starting from the target ('e') using predecessors.

## Algorithm Overview

Now let's be a **little** more formal and thorough in our description.

INITIALIZATION

1. Set the provisional_distance of all nodes from the source node to infinity.
2. Define an empty set of seen_nodes. This set will ensure we don't re-evaluate a node which already has the shortest path set, and that we don't evaluate paths through a node which has a shorter path to the source than the current path. Remember that nodes **only** go into seen_nodes once we are **sure** that we have their absolute shortest distance (not just their provisional distance). We use a set so that we get that sweet O(1) lookup time rather than repeatedly searching through an array at O(n) time.
3. Set the provisional_distance of the source node to equal 0, and the array representing the hops taken to just include the source node itself. (This will be useful later as we track which path through the graph we take to get the calculated minimum distance).

ITERATIVE PROCEDURE

4. **While** we have not seen all nodes (or, in the case of source to single destination node evaluation, while we have not seen the destination node):

5. Set current_node to the node with the smallest provisional_distance in the entire graph. Note that for the first iteration, this will be the source_node because we set its provisional_distance to 0.

6. Add current_node to the seen_nodes set.

7. Update the provisional_distance of each of current_node's neighbors to be the (absolute) distance from current_node to source_node plus the edge length from current_node to that neighbor IF that value is less than the neighbor's current provisional_distance. If this neighbor has never had a provisional distance set, remember that it is initialized to infinity and thus must be larger than this sum. If we update provisional_distance, also update the "hops" we took to get this distance by concatenating current_node's hops to the source node with current_node itself.

8. **End While**.

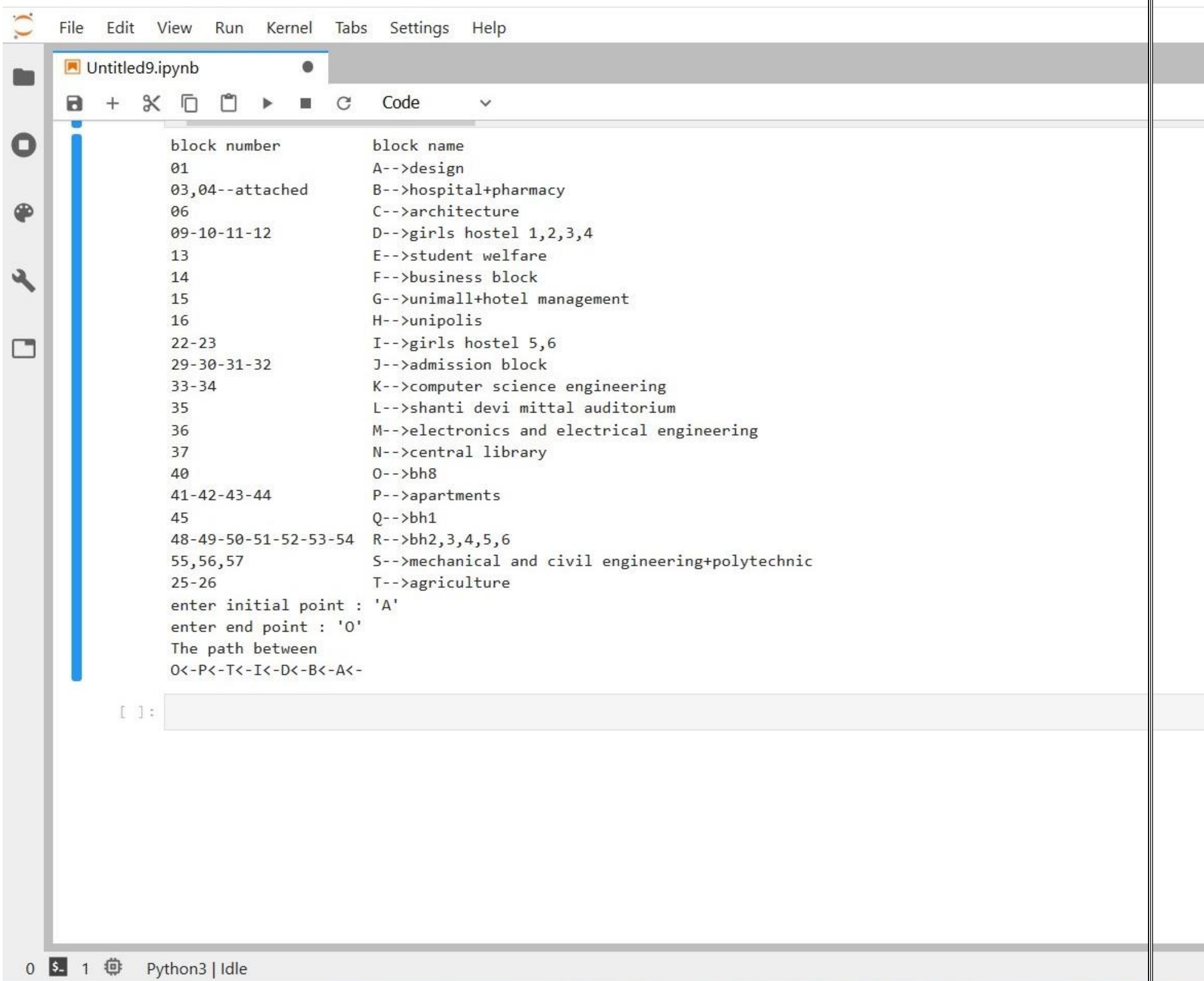
**MAP USED FOR THE PROJECT :**

**CODE:**

```
def initial_graph() :

    return {

        'A': {'B':50},
        'B': {'A':50, 'C':50, 'D':80},
        'C': {'A':50, 'D':30, 'E':80, 'G':90},
        'D': {'B':80, 'C':30, 'E':110, 'G':120, 'I':100},
        'E': {'C':80, 'D':110, 'F':20, 'G':10, 'H':20},
        'F': {'E':20, 'G':30, 'H':20, 'J':80, 'T':100},
        'G': {'C':90, 'D':120, 'E':10, 'F':30, 'H':10, 'I':120},
        'H': {'E':20, 'F':20, 'G':10, 'J':70, 'T':90},
        'I':{'D':100, 'G':120, 'T':50},
        'J':{'F':80, 'H':70, 'T':50, 'L':10, 'K':20},
        'K':{'J':20, 'L':10, 'M':10},
        'L':{'J':10, 'K':10, 'M':10, 'T':20, 'N':10},
        'M':{'N':5, 'L':'10', 'K':5},
        'N':{'L':10, 'M':5, 'O':60, 'P':50, 'T':20},
        'O':{'N':60, 'P':10, 'Q':70, 'R':50, 'T':40},
        'P':{'N':50, 'O':10, 'Q':60, 'R':50, 'T':10},
        'Q':{'O':70, 'P':60, 'R':30},
        'R':{'O':50, 'P':50, 'Q':60, 'S':20},
        'S':{'R':20},
        'T':{'F':100, 'H':90, 'I':50, 'J':50, 'L':20, 'N':20, 'O':40, 'P':10}



           }


initial = eval(input("enter initial point :"))
path = {}
adj_node = {}
queue = []
graph = initial_graph()

for node in graph:
    path[node] = float("inf")
    adj_node[node] = None
    queue.append(node)

path[initial] = 0
```

```
while queue:
    key_min = queue[0]
    min_val = path[key_min]
    for n in range(1, len(queue)):
        if path[queue[n]] < min_val:
            key_min = queue[n]
            min_val = path[key_min]
    cur = key_min
    queue.remove(cur)


    for i in graph[cur]:
        alternate = graph[cur][i] + path[cur]
        if path[i] > alternate:
            path[i] = alternate
            adj_node[i] = cur

x =eval(input("enter end point :"))
print('The path between ')
print(x, end = '<-')
while True:
    x = adj_node[x]
    if x is None:
        print("")
        break
    print(x, end='<-')
```

**OUTPUT:**


File Edit View Run Kernel Tabs Settings Help
Untitled9.ipynb
Code
block number block name
01 A-->design
03,04--attached B-->hospital+pharmacy
06 C-->architecture
09-10-11-12 D-->girls hostel 1,2,3,4
13 E-->student welfare
14 F-->business block
15 G-->unimall+hotel management
16 H-->unipolis
22-23 I-->girls hostel 5,6
29-30-31-32 J-->admission block
33-34 K-->computer science engineering
35 L-->shanti devi mittal auditorium
36 M-->electronics and electrical engineering
37 N-->central library
40 O-->bh8
41-42-43-44 P-->apartments
45 Q-->bh1
48-49-50-51-52-53-54 R-->bh2,3,4,5,6
55,56,57 S-->mechanical and civil engineering+polytechnic
25-26 T-->agriculture
enter initial point : 'A'
enter end point : 'O'
The path between
O<-P<-T<-I<-D<-B<-A<-
[ ]:
0 1 Python3 | Idle

**CONCLUSION:**

GPS is a fantastic tool of the 21st century offering many functions: waypoint, MOB…. However, received information is not always reliable, and it would have been interesting to see in which cases it is not. Furthermore, the space segment is completely controlled by the American Army which enables them to completely remove the satellite cover in certain countries in the event of a war… This aspect mobilized Europe towards the creation of its own GPS system: the project GALILEO, which should be in place2008. The future of GPS appears to be virtually unlimited; technological fantasies abound. The system provides a novel, unique, and instantly available address for every square yard on the surface of the planet–a new international standard for locations and distances. To the computers of the world, at least, our locations may be defined not by a street address, a city, and a state, but by a longitude and a latitude. With the GPS location of services stored with phone numbers in computerized "yellow pages," the search for a local restaurant or the nearest gas station in any city, town, or suburb will be completed in an instant. With GPS, the world has been given a technology of unbounded promise, born in the laboratories of scientists who were motivated by their own curiosity to probe the nature of the universe and our world, and built on the fruits of publicly supported basic research.