Route Planner Mobile Web Application for UiTM Malaysia

Suzana Ahmad[1], Norizan Mat Diah, Nor Ashikin Mohamad Kamal and Siti Zalekha Mohd Jailani

Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA (UiTM), Malaysia
[1]Email: suzana@fskm.uitm.edu.my

ABSTRACT

Services that can be accessed by users with mobile devices for the purposes of route planning are increasingly in demand. UiTM Malaysia regularly host and conduct prestigious events, which involve outside participants, and therefore the development of an accurate and concise means to guide participants around the campus is very important. Convocations, Independence Day celebrations, New Year celebrations, Seminars, local and International Conferences are examples of events regularly conducted at UiTM. A Route Planner Mobile Web Application for UiTM Malaysia has been developed to produce a list of routes from two selected points in order of the shortest first. The user can select any listed location as a starting point and any listed location as a destination. Djikstra's algorithm has been used to find the appropriate shortest route from those two selected point. Djikstra's algorithm together with Google Maps API and Google Earth, which are popular web mapping services providing geospatial data, have been used to develop, a prototype route planner.

Keywords: *Route Planner, Web Application, Mobile web application*

Introduction

In the present mobile information society, the demand for services that can be accessed by users using mobile devices for the purposes of route planning is increasing. Route planning is a process by which an optimal

ISSN 1675-7009

route, based on the road network, topological structure and an optimal path criterion, is calculated and provides a path of minimum distance between the origin and the destination [1]. A route planner is a tool, which provides an optimal route from a user-defined origin to a user-defined destination [2].

UiTM Malaysia regularly conducts prestige events, which involve outside participants who tend to require guidance around the campus. This project focused on identifying available routes around UiTM Malaysia, for which 48 junctions, 96 nodes and 99 road segments have been identified.

Based on the number of junctions, nodes and road segments identified, a prototype Route Planner Mobile Web Application has been developed. The resultant web application can be used to find the shortest route from any origin to any destination in the UiTM Malaysia Campus.

Methodology

Having identified the requirements for the development of a mobile route planner; a method of implementation of this project was divided into several phases each tailored to address issues specific to the UiTM Malaysia Campus. The first phase of the methodology involved data collection, the second phase concerned the system design, the third phase concerned system implementation followed by the final phase, which was functionality testing.

Data Collection

Data collection required the definition of three types of raw data; edges, vertexes and distance. The edges, vertexes and distance provide the constraints for this system. The edges and vertexes were defined using the KML element called path as in Google Earth. The length for each edge was calculated by the Movable Type Script that calculates the shortest geodesic distance between two points with respect to their latitudinal and longitudinal values [3].

Defining Edges

The map used in this project is the complete UiTM Malaysia map acquired from Google Maps and consists of 48 junctions, 96 nodes and 187 road segments or edges. Each route comprises of a sequence of road segments from a start location (a node) to an end location, which is another node, whereby an edge is a link between two nodes. A link is the infrastructure that supports movement between nodes. Links may be either directed or undirected (bi-directional) [4]. According to Google Maps there are 187 edges or road segments in the UiTM Malaysia Campus.

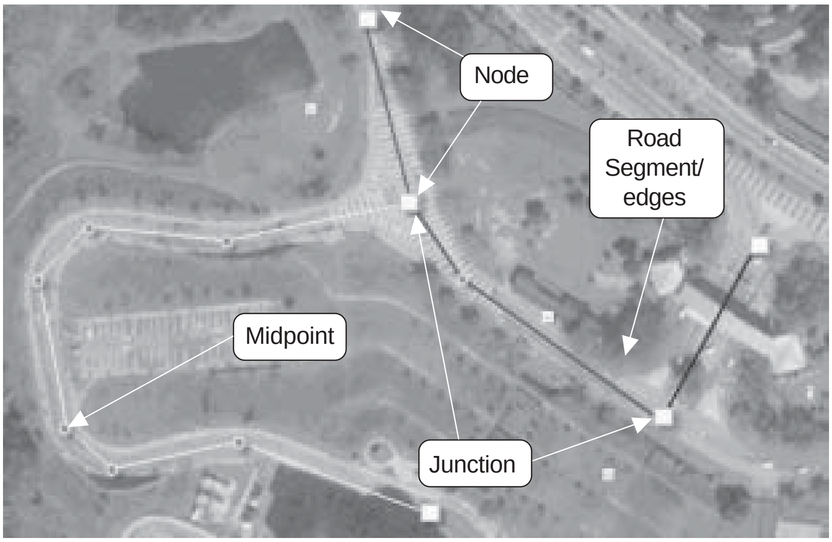


Figure 1: Identified Edges or Road Segments, Junctions, Nodes or Vertexes and Mid Point

The edges have been identified using junctions, for example, Figure 1 shows a path from the node or vertex from the main entrance to a T-junction. This junction is declared to be another node or vertex and the directed link between the two nodes is defined as an edge or road segment.

All the edges seen in Google Earth have been drawn using KML elements [5]. The KML element has been used to determine the drawing style: colour, colour mode and line width, for all linear geometries [5]. The KML file is stored in a server to enable production of appropriate routes in Google Maps. Figure 2 shows all the edges (in red line) and the junctions that are used to differentiate each edge or road segment.

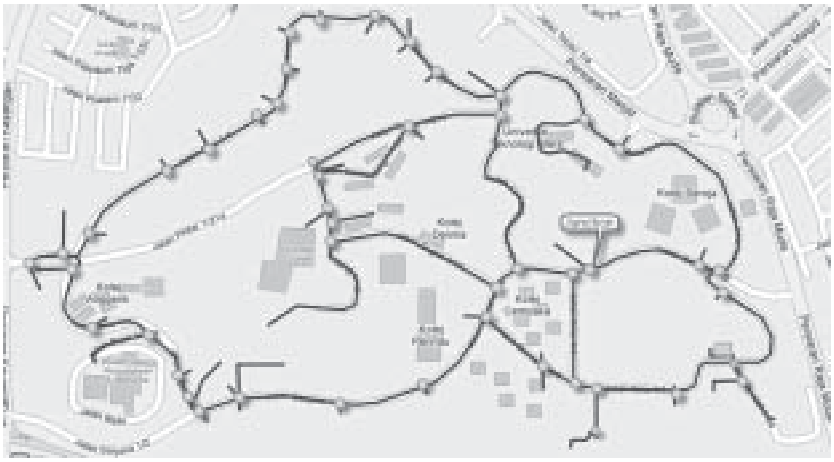


Figure 2: Edges or Road Segments Defined Based on Junctions

Defining Nodes / Vertexes

There have been 48 nodes or vertexes identified in the UiTM Malaysia Campus. A node is distinguished from a road segment in that it is either a start or end point of an edge [6]. Multiple node pairs may be linked together to form a route identified in the Campus. Figure 3 presents a simple example of a node.

Figure 4 presents all 48 nodes or vertexes successfully identified in the Campus. Figure 5 presents all identified edges and vertexes in the Campus. The green circles represent junctions and the blue rectangles represent the vertexes.

All identified edges are saved in KML or KMZ file format, which use xml. Data in the KML file are used to draw appropriate routes in Google Maps within the Campus. Coordinates (longitude and latitude) for each midpoint and node are also included in the KML file. Data defined as LineString in the KML file are used to draw the line path. An example of a KML file is presented in Figure 6.

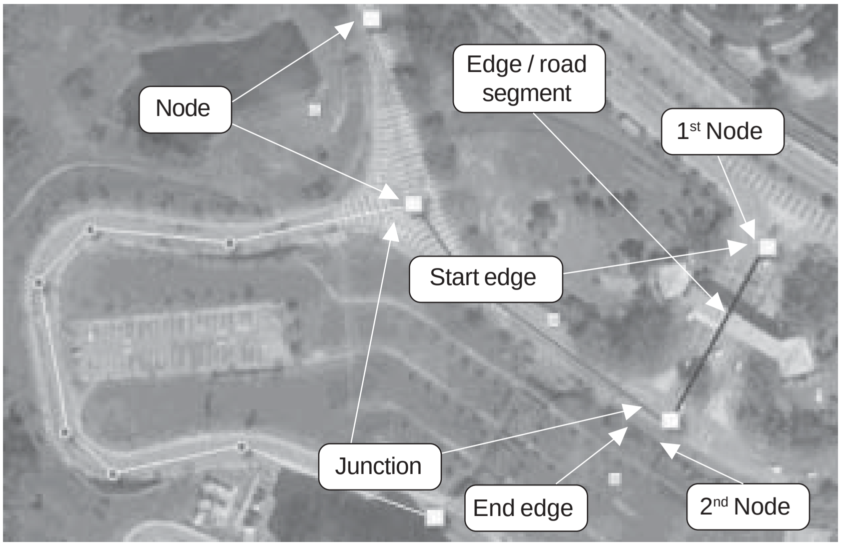


Figure 3: Defined Node or Vertex

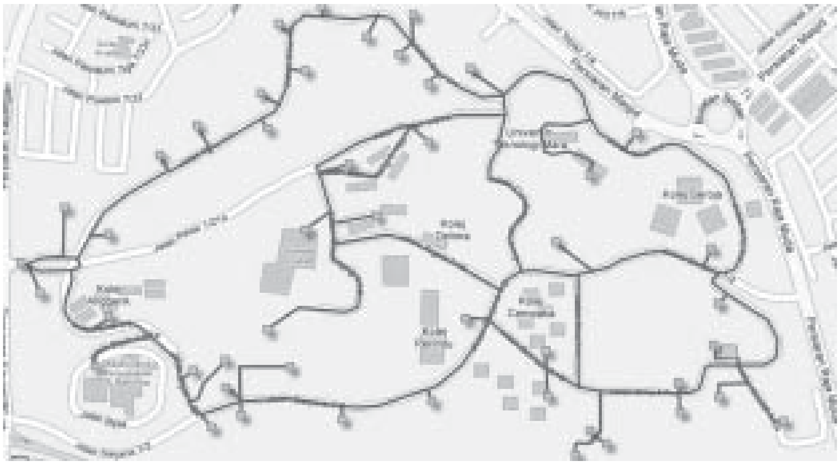


Figure 4: Nodes or Vertexes Defined in UiTM Maps

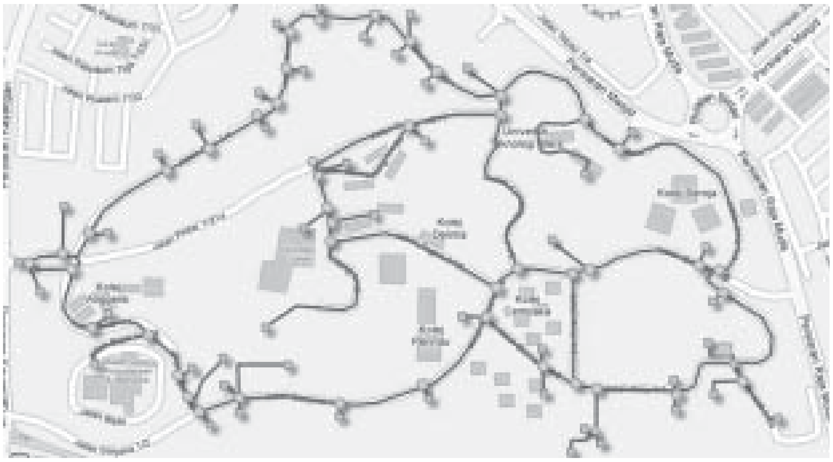
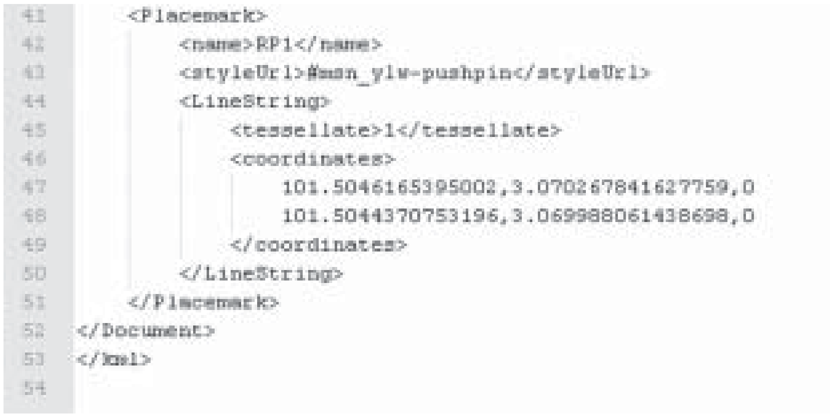


Figure 5: Vertex and Edges Defined in UiTM Maps

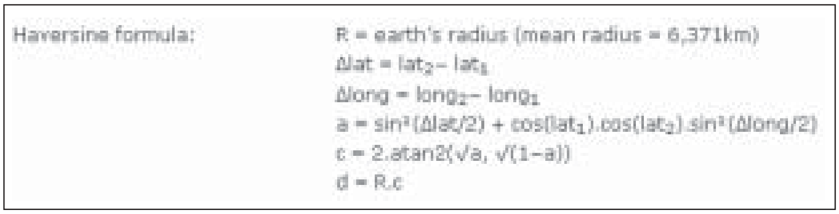


```
    <Placemark>
        <name>RP1</name>
        <styleUrl>#msn_ylw-pushpin</styleUrl>
        <LineString>
            <tessellate>1</tessellate>
            <coordinates>
                101.5046165395002,3.070267841627759,0
                101.5044370753196,3.069988061438698,0
            </coordinates>
        </LineString>
    </Placemark>
</Document>
</kml>
```

Figure 6: Sample Data in KML file

Defining Distance

The ‘Haversine Formula’ has been adapted to define the distance. This script is used to calculate great-cycle distances between two points, and identify the shortest distance over the earth’s surface [3]. Figure 7 shows the Haversine formula used to calculate a distance between two latitudinal and longitudinal points.

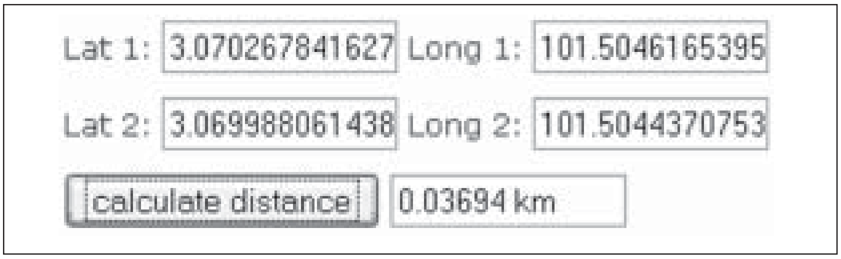


Haversine formula:

R = earth's radius (mean radius = 6,371km)
$\Delta lat = lat_2 - lat_1$
$\Delta long = long_2 - long_1$
$a = \sin^2(\Delta lat/2) + \cos(lat_1).\cos(lat_2).\sin^2(\Delta long/2)$
$c = 2.\mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)})$
$d = R.c$

Figure 7: Haversine Formula

Figure 8 presents a distance calculation example.



Lat 1: 3.070267841627 Long 1: 101.5046165395
Lat 2: 3.069988061438 Long 2: 101.5044370753
calculate distance 0.03694 km

Figure 8: Example of Calculated Distance

After all available edges, vertexes and distances in the Campus were identified, and saved in the KML file; calculating the distance between two specific locations within the Campus can be performed readily and the corresponding route is drawn on the Campus map. Figure 9 presents an example of a route between two locations along with the corresponding distances.

System Design

This system has been design in such a way that the user can access the system either using a mobile phone, handheld computer or portable computer, Figure 10. Users can access the system and enter any Campus locations and obtain the shortest path between the defined locations.

The system has been split into three modules, namely the data entrance, data processing and results modules. The first module requires users to select two locations: the current location and the destination.

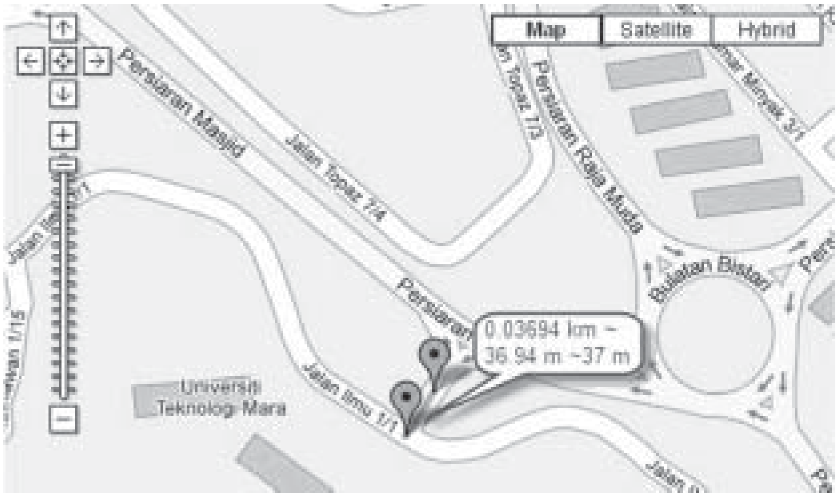


Figure 9: The Distance of the Edge (37 meters)

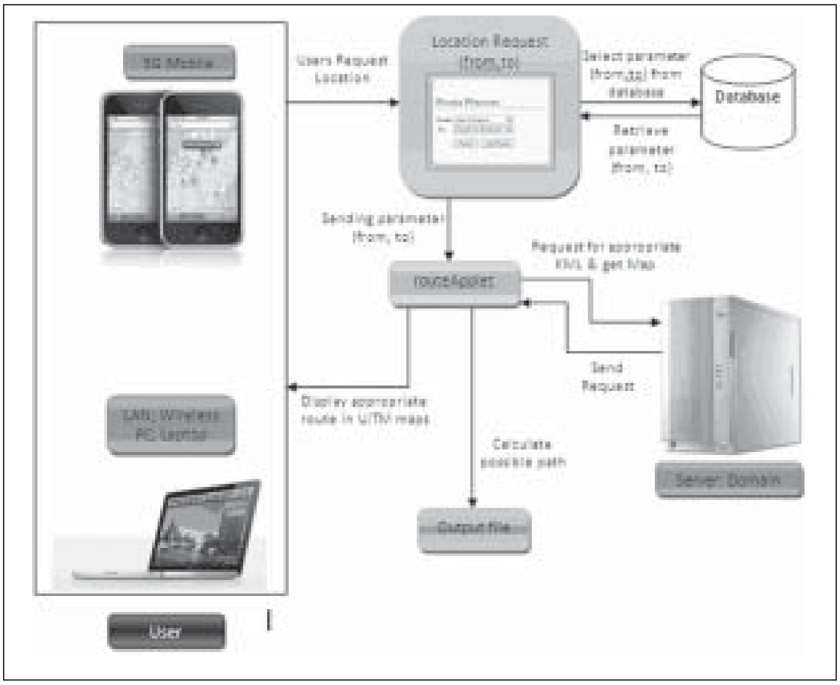

3G Mobile
Users Request Location
Location Request (from, to)
Select parameter (from, to) from database
Database
Retrieve parameter (from, to)
Sending parameter (from, to)
Request for appropriate KML & get Map
routeApplet
Send Request
Display appropriate route in UiTM maps
LAN, Wireless PC Laptop
Calculate possible path
Server Domain
Output file
User

Figure 10: Design for Route Planner Mobile Web Application

These corresponding parameters are sent to a routeApplet and the shortest route is calculated accordingly.

The second phase involves the calculation of the shortest route for the given current location and destination. The routeApplet will perform the appropriate calculations and return the shortest available route. Once the shortest route has been calculated, it will be stored in the output file together with the KML file.

The last phase involves retrieving the output: the shortest route together with other appropriate KML data, such as time, regions, super-overlays, models, photo overlays, cameras and sky data, and displaying it on the available medium. Medium such as mobile phones, PDA's, laptops or personal computers must of course be connected to the internet and be running compatible browsers in order to display the output.

System Implementation

This study used JAVA, Apache, PHP and MySQL in the implementation phase. Dijkstra's algorithm was applied to find the possible routes. Implementation was performed by constructing three different classes, namely the graph class, dijstra's class and routeApplet class. All three classes are related to each other, Figure 11.

The graph class reads the vertexes and edges via input files db.txt and path.txt. The graph class then prints the adjacency matrix based on the db.txt and path.txt files. Adjacency matrix values are then passed to the Djikstra class.

The Djikstra class is where the calculation for all possible paths for the two specified locations is performed. The adjacency matrix is used to calculate all possible routes between the two specified locations. The shortest path from the list of available paths is then determined. The parameters relating to the shortest available path are then passed to the routeApplet class for further processing.

The routeApplet processes the parameters provided with the appropriate data from the KML file. The shortest possible path is produced by the routeApplet and saved in the out.txt file. An appropriate map is then produced based on the routeApplet output. A sample of the data input window is presented in Figure 12 and an example of a successfully determined shortest route is presented in Figure 13.

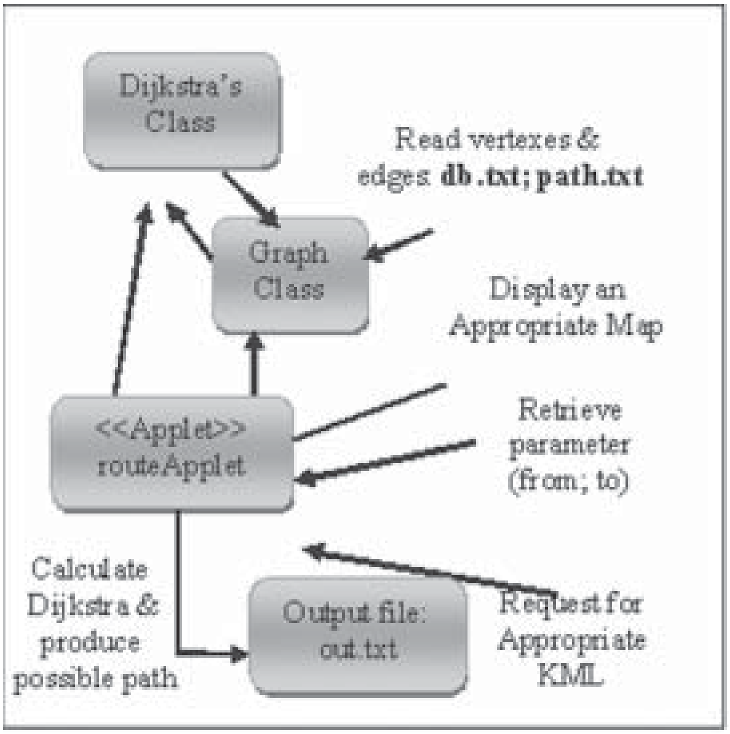

Dijkstra's Class
Read vertexes & edges: **db.txt; path.txt**
Graph Class
Display an Appropriate Map
<<Applet>> routeApplet
Retrieve parameter (from; to)
Calculate Dijkstra & produce possible path
Output file: out.txt
Request for Appropriate KML

Figure 11: Linkage Between Classes

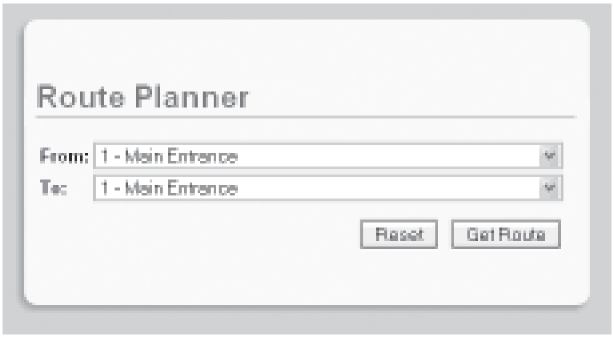


Figure 12: Sample Snapshoot of Data Entrance

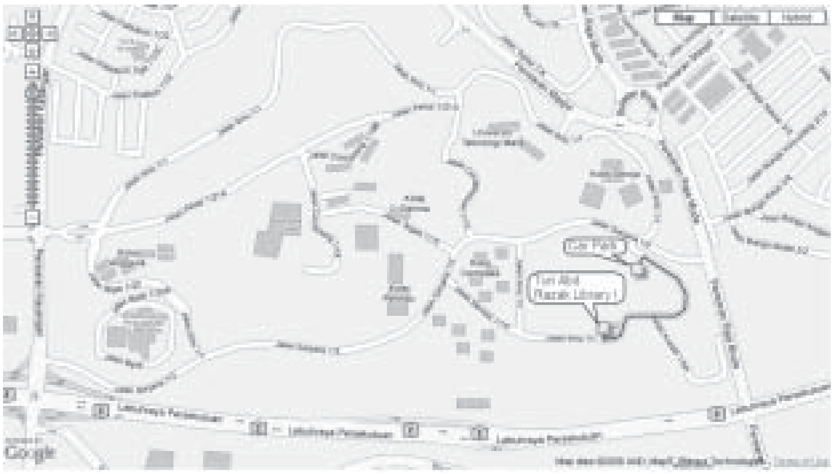


Figure 13: Sample Snapshoot of a Successful Output

System Functionality Testing

The functionality of the prototype has been evaluated with respect to a few test cases. The testing was performed using sets of 30 parameters. Figure 14 presents the results for some of the tests performed.

All results derived from the test have been validated manually. For each set of test, all possible routes were listed with their corresponding lengths. All 30 test cases successfully listed the shortest path among the possible paths identified.

Conclusions

This study has developed a prototype Route Planner capable of finding the shortest route between two predefined locations within the UiTM Malaysia Campus. This route planner could provide a failsafe method for people, familiar or otherwise, to get from A to be B anywhere in the University Campus. The concepts contained within this study could be applied to other popular places and locations thus simplifying travel through journey distance optimization.

No	Data set	Appropriate Shortest Route
1	Car Park (Tunas Mekar) → Tun Abdul Razak Library I	[Car Park (Tunas Mekar) → main 3→ main 4→ main 6→ Tun Abdul Razak Library I]
2	Second Entrance → Faculty of Information Technology & Quantitative Sciences	[Second Entrance → main 54 → main 34 → main 33 → main 32 → main 31→ main 30 →main 29→ main 28→ main 27 → main 26 →main 25 →main 24→ main 22→ Faculty of Information Technology & Quantitative Sciences]
3	Second Entrance → Faculty of Architecture, Planning & Surveying	[Second Entrance→ main 54 →main 34→ man 35→ main→ 36 main 37 → Faculty of Architecture, Planning & Surveying]
4	Main entrance → Seroja College	[Main entrance →main 1→ main 19 →main 20→ main 21→ main 14 → main 10→ main 16 →main 17→ Seroja College]
5	Main entrance → Science & Technology Twin Tower	[Main entrance →main→ 1→ main 19 →main 20 →main 22→ →main 24→ main 25→ main 26→ main 27 →Science & Technology Twin Tower]

Figure 14: Sample of Test

References

[1] J. L. Meng Yin Fu, 2004. A Practical Route Planning Algorithm for Vehicle Navigation System, *Proceeding of the 5th World Congress on Intelligent Control and Automation*, Hang Zhou, pp. 5326-5329.

[2] H. H. Hochmair, 2007. Optimal Route selection with Route Planners: Results of a Desktop Usability Study, *Proceeding of the 15th International Symposium on Advances in Geographic Information Systems ACM GIS 2007*, pp. 1-4.

[3] C. Veness, 2008. Calculate distance, bearing and more between two Latitude/Longitude points, retrieved July 15, 2008, from Movable Type Scripts: http://movable-type.co.uk/scripts/latlong.html.

[4] D. J. P. Rodrigue, 2008. Graph Theory: Definition and Properties, retrieved October 6, 2008, from the Goegraphy of Transport System: http://people.hostra.edu/geotrans’eng’ch1en/meth1en/ch1m2en_2ed.html.

[5] Google, 2008. KML Documentation Introduction, Retrieved July 15, 2008, from Google Code: http://code.google.com/apis/kml/documentation.

[6] L. M. Smyth, 2000. *Personalised Route Planning: A Case-Based Approach*, Springer-Verlag Berlin Heidelberg, pp. 431-433.