E-Guide for University Students

Ammar M. Huneiti[1] , Rizik M. H. Al-Sayyed[2] , Rola I. Al-Khalid[1], Dima Khalil[1] , Heba Mousa[1]

[1.] Department of Computer Information Systems, King Abdullah II School for Information Technology, The University of Jordan, Amman 11942 Jordan
[2.] Department of Business Information Systems, King Abdullah II School for Information Technology, The University of Jordan, Amman 11942 Jordan
a.huneiti@ju.edu.jo

Abstract: Universities all over the world have many buildings and facilities distributed inside and outside of their campus location, and they are used by students for different reasons such as academic, entertainment, services, etc. These may include faculty buildings, cafés, libraries, restaurants and others. For newly registered students and even current students, locating the location and the route to a certain facility is a troublesome task. This task is even more difficult for foreign and non-local students. Therefore, taking into consideration the advantages of digital guiding and navigation systems over static paper maps, a system was developed in order to guide students to facilities both inside and outside the university campus. This paper introduces the stages of development of such systems and their main features. In addition, the search operation and locating the shortest path to any place of interest are also described.

[Ammar Huneiti, Rizik Al-Sayyed, Rola Al-Khalid, Dima Khalil, Heba Mousa. **E-Guide for University Students.** *J Am Sci* 2012;8(10):574-581]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 80

Keywords: Maps, shortest path, E-Guide system, Dijkstra's algorithm, university.

1. Introduction

Maps are visual representations that show all or part of the Earth's surface with geographic features, universities, roads, and other details. Also maps are ordinarily used to show details in specific areas. Traditional maps, while useful, are limited to one view and can include only a limited amount of information (Kuhn et al., 2004; Jiang et al., 2006). In this paper, we present an E-Guide system to guide students inside or outside the university. By entering the name of your desired location (for example, a specific region) in the search field, the GIS system provides a list of possible locations. You can select the location that meets your needs. University of Jordan; as other universities around the world; is supposed to serve its' students in the best way. That is why this system is proposed and developed.

Many applications focused on guiding students only inside the university and telling them about faculties and all other university's components (Zheng et al., 2009). Our system guides students inside the university campus as well as outside the campus in regions surrounding the university, telling them about the places that are expected to be known to most students. For example; some students need to know about the accommodations that are very close to the university or they need to know about libraries, cafeterias and student services in the vicinity.

The Innovative E-Guide can be used in two languages: Arabic and English; the goal is to benefit all students who come to the university from different nationalities.

In order to make this system useful and convenient, many services are provided:

1. Query which is used to help students to know all places names that are far from the gates of the university.
2. Finding the shortest path between two places by using the Dijikstra algorithm.
3. View map which gives the student an overview of the system scope.
4. Searching either General or Specific.

2. Related Work

Mytilene E-guide (Michael et al., 2011) is a multiplatform mobile application tourist guide exemplar. In this paper, the authors introduced an approach of enhancing the tourism experience by incorporating a tool that adopts a "web-to-mobile" model into tourism websites. That allows for the adaptation of personalized tourism web content to be transferred to a mobile application hosted at users' mobile device. The advantage of this work is the multiplatform system combination between the nature of the mobile and the web. The disadvantage of this work is that the system depends on the users to "tag" the content which interests them through a familiar PC web interface and select their mobile to download the application to a PC and this operation wastes time.

In reference (Cheverst et al., 2000), the authors presented an intelligent electronic tourist guide system; they developed a Context-aware Electronic Tourist Guide. This system has been built to overcome many of the limitations of the traditional information and navigation tools available to city visitors. The system combines mobile computing technologies with a wireless infrastructure to present city visitors with information tailored to both their personal and environmental contexts. The advantage of this approach is that one can install this system on the mobile with a wireless infrastructure. While the disadvantage is that the modified browser metaphor used by GUIDE caused some confusion because of the semantic overlap between the standard back and forward buttons and the buttons for requesting to view the next or previous navigation instruction. For a system like this system to be accepted by the public at large it needs to continuously show clear benefits over the traditional facilities available to tourists.

The authors of reference (Simcock et al., 2003) developed a Location Based Tourist Guide Application. This system is supported for the location and other elements of the users such as buildings in view, attractions and equipment nearby, such as public telephones and toilets. This system will describe the Tourist Guide system and discuss the processes involved in the development of the application. More specifically, this application will look at the design and usability issues. The advantage of this approach is that this application looks at the design and usability issues while the disadvantage is the lack of functionality such as sound and streaming video. These media could be looked at if a new hand held device with faster processing and storage capability is used.

In reference (Cambria et al., 2009), an International Student Admission Guide is created to provide comprehensive information to international students who are applying for the Master of Science in Project Management (MSPM) program. This guide includes a helpful admissions checklist, application deadlines and process, the visa process, and frequently asked questions. The advantage of this approach is that it helps in the admission process while its disadvantage is that it is only restricted to George Washington University.

In reference (Michael et al., 2009), the authors presented an innovative mobile electronic tourist guide application that represents a relatively new trend in the field of tourism and involves the use of mobile devices as electronic tourist guide. It enables the creation of portable tourist applications with rich content that matches user preferences. The advantage of this approach is that portable tourist guide, network coverage is not further required and the application content may be updated when the user returns online. On the other hand, the disadvantage is that it implies using a mobile or wireless network to access Internet resources; wireless web access is typically enabled by WAP or i-mode compatible devices. Here, the requirement for constant connection (airtime) of the mobile device with a mobile network to offer access to web content is the main disadvantage of using i-mode WAP-based services. Also, since this system is implanted in J2ME, one should be careful of J2ME technology weaknesses including "increased requirements on devices' resources: and the download of new J2ME applications (jar files) is costly, slow and consumes network resources".

3. Innovative E-Guide Model

In this section we will present our application the **Innovative E-Guide;** it is a desktop application that is designed to be used by novice and foreign students. Students can use Innovative E-Guide to guide them inside and outside Jordan University.

As mentioned above, the model provides four services: Query, finding the shortest path between two places, viewing map and search. Query and the shortest path are the main services the system provides. Query is a service used when the student wants to know all places names that are far away from the gates of the university a specific distance determined by the student. We focused our work on The University of Jordan Main and Northern gates; this is due to the large number of interesting places located opposite to these gates. The search results can be viewed in two ways: on map and as a list.

View map is a service that gives an overview of the scope of the system. It gives a map of The University of Jordan as well as Jubeeha district and Queen Ranya Street.

The shortest path is a service that allows users to ask about the shortest path between two places one is the source and the other is the destination. However, the result is computed by using Dijisktra algorithm. The algorithm computes the distance of all possible paths between the determined source and destination and chooses the minimum one. The shortest path is shown by two ways: either by drawing the path on the map or by listing all the places and the streets between the source and the destination.

All these services are bilingual in both Arabic and English; in order to make sure that the system is easily accessible by students who have different nationalities.

There are two types of search, general search and search by name. In general search**,** a student must determine several requirements, for example the student should determine the scope of search whether it is inside or outside the university. If the search is inside the university, there is no need to determine a region of search. Otherwise, the region of search must be determined. Only two regions are involved in our application: Jubeeha district and Queen Ranya Street. Another thing the student should determine is the type of object he/she is searching for: restaurants, cafeterias

… etc. The result of the general search is shown in two ways depending on student' choice i.e. on the map or as a list.

The other type of search is search by name. Students use this type of search when they know nothing about the target place except its name. Therefore the student should determine scope of search, type of target and the name of the place. In search by specific name the result can be shown only on the map displaying only the places that match the name entered by the student. When a student uses the search either general or specific, the result will be displayed, and then the student can click this result to find a list of information containing the place name, address, phone, and logo.

4. Model description

4.1 Functional requirements

In this section we introduce the functional and nonfunctional requirements (see Table1).

Table 1. Functional Requirements

No.	***Requirement Description***
R1.	This system provides an electronic map for guiding foreign/novice students.
R2.	This map gives the student the main features of the university such as faculties , main library, gates … etc.
R3.	In addition to the university ,the map gives the student main features of all the places around the university such as housing student , super markets … etc.
R4.	This system gives the student the ability to ask for places by using the search operation.
R5.	The student can search for places inside and outside the university.
R6.	If the student asks about the places inside the university, he/she should first determine what type of object he/she wants such as faculties, gates … etc?
R7.	The system gives a list of objects the student may ask for.
R8.	If the student asks about the places outside the university, he/she should determine the region and the object itself such as accommodations, supermarkets, pharmacies … etc.
R9.	The system gives a list of objects the student asks for within the specified region.
R10.	The system provides the shortest path between two places; the student only determines the source and the destination.
R11.	The student can query about a specific object within a specific distance.
R12.	The student has a help button in each form to clarify the task of this form.
R13.	The administrator of our system has an initial username and password to sign in and later on, the password can be changed.
R14.	The administrator can add a new place that has been built in the real life.
R15.	The administrator can delete a place that has been removed from the real life.
R16.	The administrator can update information of specific place such as the phone number.
R17.	The administrator can add a new place with specific information to calculate the shortest path.
R18.	The student can get more details about a specific place outside.
R19.	The student can get more details about a specific place inside.
R20.	The system provides services in two languages: Arabic and English.
R21	The student can get the result either by list or by drawing lines on the selected map.
R22	The student can get the result either by list or by map.

Block diagrams are composition of different shapes and lines showing how the components of an application, process, or system are related and depend upon one another. Figure 1 shows the block diagram of our model. More description of the block diagram and the requirements R1, 2, 3.

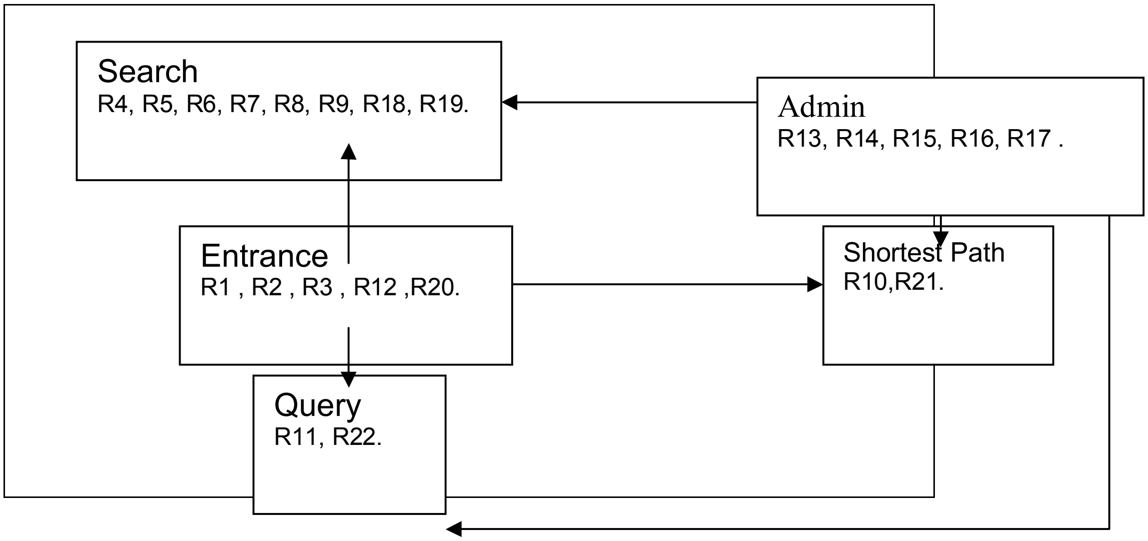


Figure 1. Innovative E-Guide Block Diagram

4.2 Use case

The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. However, the use case should contain all system activities that have significance to the users. Here we introduce Administrator and student use cases (see figure 2, figure 3).

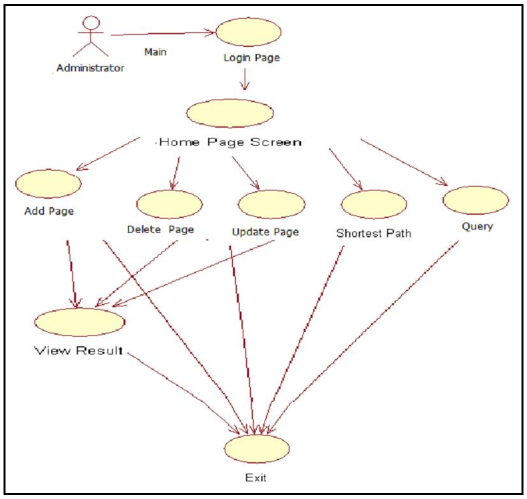


Figure 2. Administrator Use Case

Use case Data Dictionary defines each term in the use case diagram and provides explicit clarification for any system-specific uses of general terms (such as student, search … etc). It explains interactions and uses reader-friendly cross-references to other entries for simplifying the process of understanding. Table 2 shows the use case data dictionary of our proposed model.

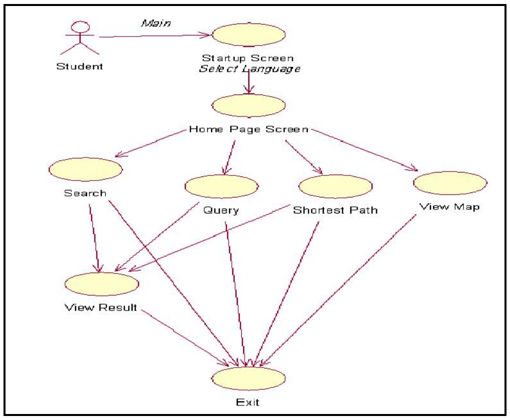


Figure 3. Student Use Case

5. Algorithm

Dijkstra's algorithm (Merin, 2011; Thomas et al., 2001) is called the single-source shortest path. It computes the length (Distance) of the shortest path from the source to each of the remaining vertices in the graph.

The single source shortest path problem can be described as follows:

Let $G = \{V, E\}$ be a directed weighted graph with V having the set of vertices.

The special vertex $s \in V$, where s is the source and let for any edge e in E, $EdgeCost(e)$ be the length of edge e. All the weights in the graph should be non-negative.

Dijkstra's algorithm is considered a *Greedy Algorithm.* Greedy algorithms use problem solving methods based on actions to see if there's a better long term strategy. Dijkstra's algorithm uses the greedy approach to solve the single source shortest problem. It repeatedly selects from the unselected vertices, vertex *v* nearest to source *s* and declares the distance to be the actual shortest distance from *s* to *v*. The edges of *v* are then checked to see if their destination can be reached by *v* followed by the relevant outgoing edges.

Although Dijikistra algorithm is very common in finding the shortest path but still it has some disadvantages .The major disadvantage of the algorithm is the fact that it does a blind search; i.e. it consumes a lot of time. Another disadvantage is that it cannot handle negative edges measures. This leads to acyclic graphs and most often cannot obtain the right shortest path.

5.1 Description of the algorithm

Before going into details of the pseudo-code of the algorithm it is important to know how the algorithm works. Dijkstra's algorithm works by solving the sub-problem *k*, which computes the shortest path from the source to vertices among the *k* closest vertices to the source.

For Dijkstra's algorithm (Dijkstra, 1959) to work; the graph should be directed and weighted and the edges should be non-negative. If the edges are negative then the actual shortest path cannot be obtained. At the *kth* round, there will be a set of *k* vertices that will consist of the vertices closest to the source. The shortest distance obtained is maintained in a database table; it holds the estimate of the distance from *s* to *w*. Dijkstra's algorithm finds the next closest vertex by keeping track of the new closest vertices.

5.2 Pseudo code of the algorithm

The following pseudo-code (Figure 4) gives a brief description of the Dijkstra's algorithm (Italic=comments; Underline=name of sub procedure).

5.3 Example

We will give an example that illustrates how the Dijkstra's algorithm works. As shown below the direct link between vertex "2" and "5" is 10m but the actual shortest path between both vertex is "2 ,4 ,6 ,5" because the distance between these vertexes is 9m and it is less than 10m (direct link) (see figure 5).

```
'if the computer program has not read in the input data yet
then tell it to go get it
If have_input_data <> 1 Then
   get_nodes 'get the nodes list
   get_links 'get the link list
   ReDim node_table(num_nodes, 4)
  ReDim all_origins_node_table(
     num_nodes*num_zones, 5)
'this indicates whether or not we have done the node table
for the first origin (0 means no)
'knowing this will also tell us whether or not to set output
counters to zero
   first_origin = 0
End If
'know that we have all of the network input data we will not
need to get the network input data any more. 'This will help
when using this code within the four step travel demand
modeling framework.
have_input_data = 1
'this will calculate costs depending on the link volumes
assigned to the link
calc_costs
'This loop goes through each node in the list of nodes. As it
finds zone nodes, it sets them up as the origin node and
completes Dijkstra's algorithm for them.
For node_count = 1 To num_nodes
'if this condition is true then the computer has found an
origin node and a node table needs to be generated for it
  If nodes(node_count, 2) = "z" Then
   origin_zone = node_count   'the next zone was found. Set it
as the origin node
```

```
   initialize_node_table
   debug_output  'output the current node table and
information for the list of tentative nodes
   origin_done = "no" 'need to start working on the table for
the origin node
   Do Until origin_done = "yes"
     find_neighboring_nodes 'Step one in Dijkstra's algorithm
     update_node_table 'Step one in Dijkstra's algorithm
     debug_output
     finalize_current_node 'Step two in Dijkstra's algorithm
     debug_output
     find_current_node 'Step three in Dijkstra's algorithm
   Loop
'output the completed node table to the spreadsheet and to
the array containing all of the node table for each of the
origins.
   output_origin_node_table
   'when the computer reaches this point, it will have done
the at least the first origin.
            'if it has only done the first origin then the
first_origin variable would still be at zero.
'if it is still zero then change the first_origin variable to 1 to
say that the first origin has been 'done.
   if first_origin = 0 then
      first_origin = 1
   end if
  End If
Next
```

Figure 4.Pseudo-code of the Dijkstra's algorithm

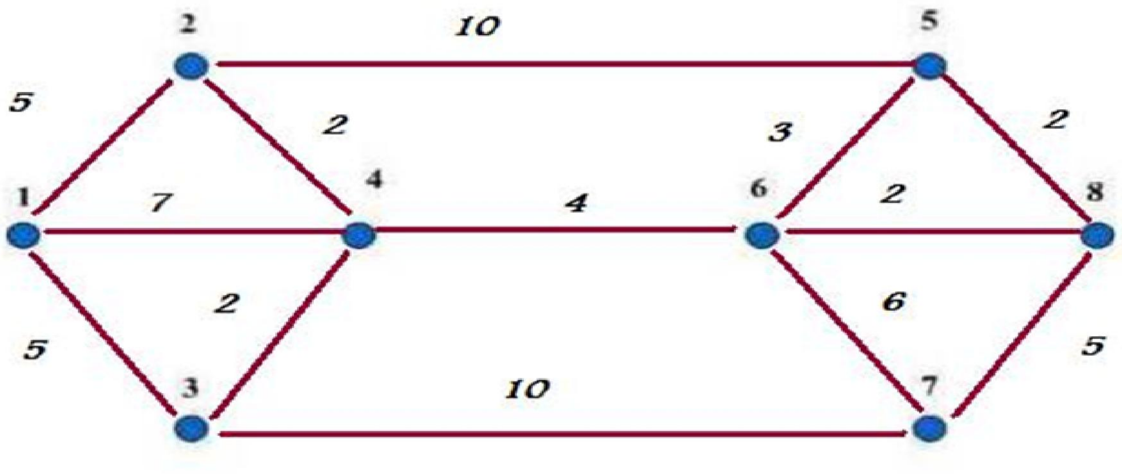


Figure 5. Shortest path example

Table 2. Use case data dictionary

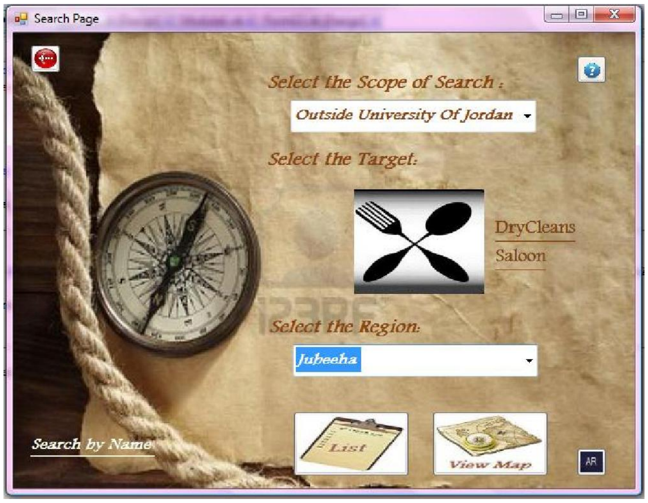
Use Case Name	Description
Student	Select Language View Map Search Query Find Shortest Path
Start Up Page	The student selects the language from this page
Home page screen	This Page contains the main process the student can use like search, query etc
Search	This form allows the student to define the search scope, region and type or the name of specific place to find
Query	The student can query about specific place within allowed distance using this form
Shortest Path	The student defines the source and the destination place inside the University of Jordan to find the shortest path
View map	The map of the University of Jordan and around it is displayed using this form
View result	Using this form the result of the previous process is displayed by a map or a list according to the student selection.
Exit	This command allows the user to quit the application

6. System Usage: Screen shots

In this section, we will present the main services that our system provides:

6.1 General Search operation

As shown below the student must select the scope of search "inside or outside" the University of Jordan, then select the target such as "Dry-clean, Restaurant, Cafeteria" and finally if the search is outside the university, she/he must select the region of search "Jubeeha or Queen Rania Street" then the search will be represented as a map or a list (see figures 6 and 7).


Search Page

Select the Scope of Search : Outside University Of Jordan

Select the Target:

Select the Region: Jubeeha

Figure 6. Search operation

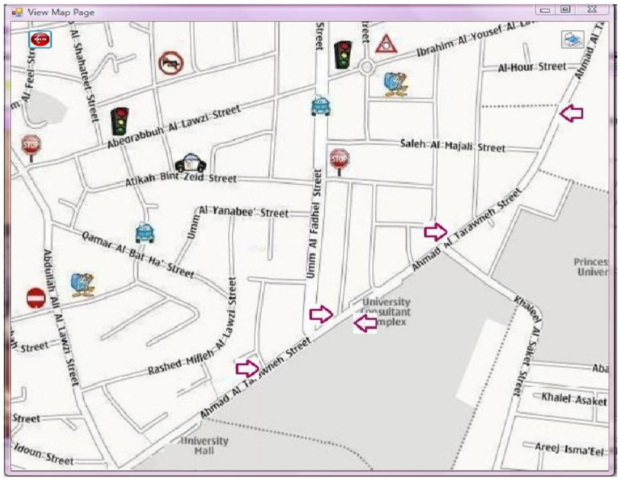


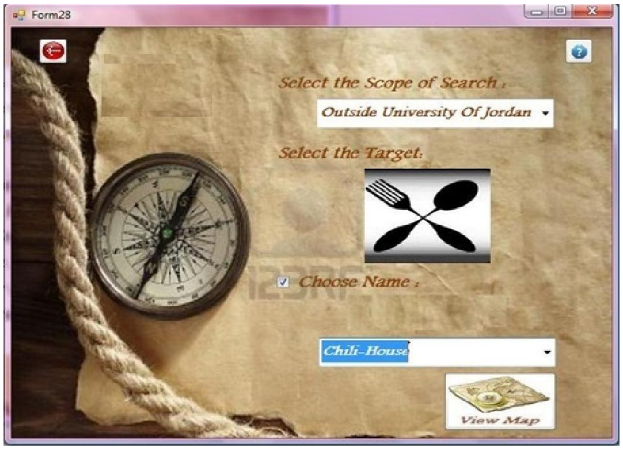
Figure 7. Map result

6.2 Search by Name

This search is a specific search, search by the target name the student must select the target name (see figure 8).

6.3 Finding the Shortest Path

As shown in figures 9 and 10, the student must first determine the region name then the source and the destination. He will get the result as a map or a list.

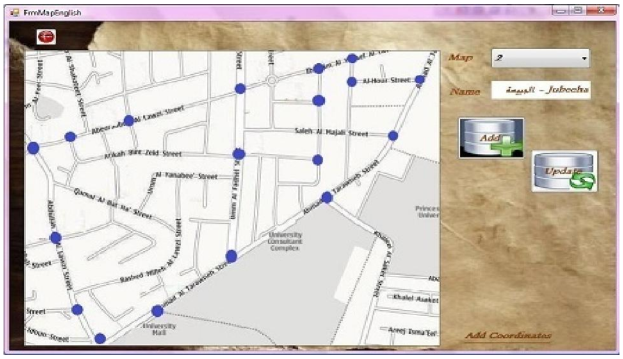


Select the Scope of Search : Outside University Of Jordan

Select the Target

☑ Choose Name : Chili-House

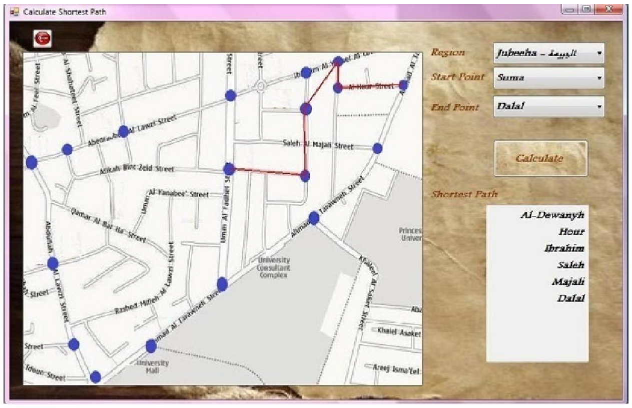
Figure 8. Search by name operation



Map: 2

Name: الجبيهة – Jubeeha

Figure 9. Select region, source and destination.



Region: Jubeeha – الجبيهة

Start Point: Suma

End Point: Dalal

Shortest Path:
Al-Dewanyh
Hour
Ibrahim
Saleh
Majali
Dalal

Figure 10. Result as a map or a list.

7. Conclusion and future work

In this paper we introduced an Innovative E-Guide application for students in order to help and guide them inside and outside the university campus finding needed locations. Student can use any of the services the system provides such as searching, querying and shortest path. The application is user friendly and handy because it is presented in both Arabic and English.

As a future work, we will modify the application by customizing it to work on mobile phones and smart devices such as IPad. Also, we have a plan to cover all regions surrounded the university not just regions mentioned above (Jubeeha and Queen Ranya Street) to guarantee that all regions needed by students.

References

1. Kuhn, P.J. Location-based services in mobile communication infrastructures. AEU-International Journal of Electronics and Communications 2004; 58(3):159-164.
2. Jiang B. and Yao X.. Location-based services and GIS in perspective. Computers Environment Urban System 2006; 30(6):712-725.
3. Zheng, Jianghua; Ciepłuch, Bła ej; Mooney, Peter; Winstanley, Adam. Location Based Services of University Town Based on OpenStreetMap. NUI Maynooth as an example Proceedings of CIICT2009 2009; pp19-22.
4. Michael Kenteris, Damianos Gavalas, Daphne Economou. Mytilene E-guide: a multiplatform mobile application tourist guide exemplar. Multimedia Tools Appl. 2011;54(2): 241-262.
5. Cheverst, K., Davies, N., Mitchell, K., Friday, A. & Efstratiou, C.. Developing a context aware electronic tourist guide. In *Proceedings of the SIGCHI conference on Human factors in computing systems* 2000; pp. 17–24.
6. Simcock, T., Hillenbrand, S.P., Thomas, B.H.: Developing a location based tourist guide

application. In Johnson, C., Montague, P., Steketee, C., eds.: ACSW Frontiers '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003; Volume 21 of CRPIT., Darlinghurst, Australia, Australian Computer Society, Inc. pp. 177–183.

7. Cambria Happ, Dustin Pusch. The George Washington University. International Student Guide. Master of Science in Project Management 2009.
8. Michael Kenteris, Damianos Gavalas, and Daphne Economou. An innovative mobile electronic tourist guide application. PUC: Personal and Ubiquitous Computing 2009; 13(2):103–118.
9. Merin Puthuparampil, Report Dijksra's algorithm. http://cs.nyu.edu/courses/summer07/G22.2340-001/Presentations/ Puthuparampil.pdf (February 2011).
10. Thomas H. Cormen , Clifford Stein , Ronald L. Rivest , Charles E. Leiserson. Introduction to Algorithms. McGraw-Hill Higher Education, U.S.A. 2001: pp. 595–601.
11. Dijkstra, E. W. A note on two problems in connexion with graphs (http://www.m3.ma.tum.de/twiki/pub/MN0506/ WebHome/dijkstra.pdf). Numerische Mathematik 1959; 1: 269–271. doi:10.1007/BF01386390.

9/9/2012