chapter

ONE

INTRODUCTION

1.1 My Map Online

In our project we concentrate on finding driving directions and computing best route based on shortest path leeds to the destination .

1.1.1 What is Map Online ?

Is an web application that calculates the shortest path to the desired destination

DISTANCE AND TIME SPENT

TURN-BY-TURN SHORTEST DRIVEN DIRECTION

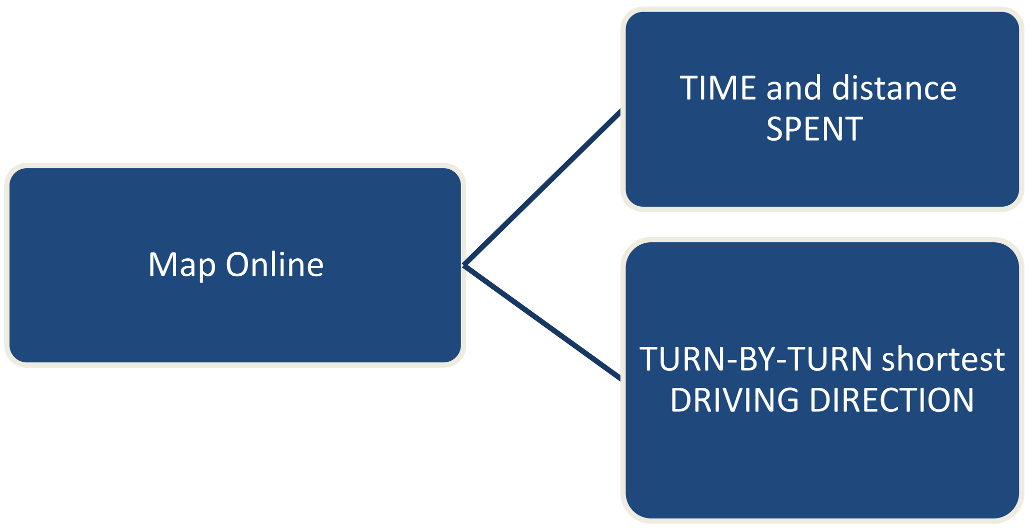


Figure 1.1 objectives of the project

- **DISTANCE AND TIME SPENT**

  Calculate the distance between two cities and time spent to reach the desired destination .

  For example :- If you need to travel from zagazig to Ismailia pathing through Abo hammad , El-Qasasin and Abo sower .

  The total distance 80.2 km and **time spent** is 82minutes .

- **TURN-BY-TURN SHORTEST DRIVING DIRECTION**

  Gives you the shortest path to the desired destination.

  For example:- If you need to travel from zagazig to cairo

  There are multiple routes leeds to cairo :-
  First:-Zagazig,Minia El-kmh, Banha,Qalub,Cairo.
  Total Distance is 82.9km & **Time spent** is 1 hour 24 minutes
  Second:-Zagazig,Belbes,El-salam,Cairo
  Total Distance is 81.6km & **Time spent** is 1 hour 24 minutes
  Third:-Zagazig,Abo hammad,El-Qasain,Abo sower,Ismailia,El-asher, El-salam,Cairo
  Total Distance is 81.1km & **Time spent** is 1 hour 25 minutes

You need to know which the best route ??

The application find the best route with lowest distance .

In our example:- The best route is third route

1.2 Problem Definition:-

- Do you map multiple addresses each day ?
- Do you spend hours mapping your routes ?
- Do you want to save time, fuel and money ?

1.2.1 Disadvantages traditional technique:-

- Time consuming.
- Money consuming
- Fuel consuming

1.3 Problem Solving:-

We can solve problems of traditional technique by:

Converting traditional technique into computer system by using electronic map, our project aims to do this. So all things can connect to the computer without need to the the large calculation.

1.3.1 Advantages of map online:-

- Save time and effort that wasted in doing routine operations.
- Saving fuel.
- Less money cosuming .
- Finding the direction and shortest route easily
- Increase performance in finding the shortest route.
- Make work less tedious.
- minimize errors in finding the direction and shortest route to destination.

1.4 History of maps

Maps have an exciting history of their own. Over time, they evolved from being rough sketches, often based on travellers' tales and stories passed on through word of mouth that may or may not have been true.Nowadays maps are accurate scientific instruments Their development runs in parallel with the development of civilisation.

1.4.1 Early Maps

Early attempts at creating maps were extremely limited by a lack of knowledge of areas other than very local surroundings of the map maker`s.

The earliest known maps are from Babylon (in modern Iraq), where they were produced on clay tablets. These date from about 2500 BC.

The Turin Papyrus is an Ancient Egyptian map. it dates from around 1200 BC. It was created for Ramses IV, an Egyptian king.

Like the kings who had gone before him, Ramses IV had ambitious plans to build great temples and statues.

The map was created as a guide to the area where the huge stones used to build the temples and statues were quarried. Click on the image to get a closer look.

The Turin Papyrus is an Ancient Egyptian map. it dates from around 1200 BC. It was created for Ramses IV, an Egyptian king. Like the kings who had gone before him, Ramses IV had ambitious plans to build great temples and statues. The map was created as a guide to the area where the huge stones used to build the temples and statues were quarried.

1.4.2 Ptolemy's Map Creations

Ptolemy's work was lost to the West until the Renaissance when his writings, rather than his maps, survived. People who constructed maps in the fifteenth and sixteenth centuries recreated the maps of Ptolemy by following the notes and writings in his books. Look at the map below from 1503 which was based on Ptolemy's writings.

1.4.3 Medieval Maps

Between 500 and 1100 AD, or the Early Middle Ages, Europe descended into the Dark Ages. During this time the Arabs led the mapping world in terms of accuracy and detail. See the image below to take a closer look at a Moroccan map from 1154.

During the middle ages, or medieval period, it was traditional to draw maps with a religious perspective - Jerusalem was often shown as the centre of the world. As the Vikings explored in the Atlantic, discovering Iceland, Greenland and America, some of their discoveries were incorporated.

1.4.4 Mercator's Projection

Gerardus Mercator (1512-1594), the famous Flemish cartographer, created the first effective way of showing the world as a sphere on a flat surface in the mid-16th century. This was known as the Mercator projection. His world map of 1569 was the best to date. He produced an atlas in 1578 and this was regulary updated to include new information in later issues. We still use The Mercator projection today.
Mecator is famous for what is called his projection. He devised a way of making globes of the Earth. Until then there was no way of showing a map of the world other than on a flat sheet of paper. He knew that the Earth was a sphere - but how would you show that using a map? Have you ever tried to wrap up a football? That is what it would be like trying to wrap a map of the world around a sphere. The ends would be all wrinkled and information would be lost.

He made the globes from papier-mache spheres, coated in thin plaster. Next he divided the map of the world into 12 different pieces, like segments of an orange that were narrow and each end near the North & South Poles and wider in the middle. He then put all the pieces together (like a 3D Jigsaw) and stuck them in the correct places on the globe.

1.4.5 18th & 19th Century Maps

Maps became increasingly accurate in the 18th, 19th and 20th centuries. As the European powers became stronger in the Americas, Africa and Asia, maps were essential for managing the new empires. The first and second world wars required detailed, accurate maps on a scale never before created.

1.4.6 Aerial Photography

Aerial photography is the taking of photographs from the air. The French photographer and balloonist Gaspar Felix Tournachon, also known as "Nadar" was the first individual known to have taken an aerial photograph in 1858. The practice of aerial photography expanded during World War 1 where they were used when planning attacks and defenses. It is widely used in cartography today due to the wealth of information it provides.

1.4.7 SATELLITE MAPS

Satellite technology has revolutionised the map-making process and has taken the field of aerial photography to new 'heights'.

The year of 1957 saw the beginning of satellite imagery with the launch of the Russian satellite, Sputnik. With the launch of the Landsat satellites by NASA in 1972, satellite imagery began to be sold commercially.

Nowadays satellites can take very detailed digital images of the earth's surface from over 650 kilometres away. They can identify objects on the earth as small as one square metre and circle the earth fourteen times a day.

1.4.8 GOOGLE MAPS

An exciting recent development in mapping is the availability of detailed maps on the Internet.

Google Maps is an excellent example. This program allows you to select a country and zoom in and get very detailed maps, often down to small street level. While coverage of Ireland is not as good as it might be, it is improving all the time.

A sister product, Google Earth, provides a way to zoom in on particular parts of Ireland, using a huge database of satellite images. Depending on the level of coverage available for a particular location, you can often see your own town, village or even house !

Map-making has come a long way.

chapter Two

Information System Concepts and System Development Life Cycle

2.1 OVERVIEW[3]

This chapter provides different concepts and definitions of information technology development.

System analysis and design methods are applied by system analysts to facilitate the development of information technology and computer applications.

►►*What is an Information technology?*

- An Information technology (IT) is a technology which uses computers to gather, process, store, protect, and transfer information. Today, it is common to use the term Information and communications technology (ICT) because it is unimaginable to work on a computer which is not connected to the network.

►►What is a Computer Application System?

- A **computer application** is computer-based solution to one or more business problems and needs. One or more computer applications are typically contained within an information system.
- One specialist plays a special role in systems and applications development, the systems analyst.
- A **systems analyst(s)** facilitates the:-
  Development of information technology and computer applications. By facilitating the study of business problems and needs to determine how the business system and information technology can solve the problem and accomplish improvements for the business well.

- The systems analyst performs **systems analysis and design.**
  - **Systems analysis** is the study of a business problem domain for the purpose of recommending improvements and specifying the business requirements for the solution.
  - **Systems design** is the specification or construction of a technical, computer-based solution for the business requirements identified in a systems analysis. (Note: Increasingly, the design takes the form of a working prototype.)

- **Most systems analysts use some variation of a system problem solving approach called a system development life cycle.**

  - **A *systems development life cycle* (SDLC) is a systematic and orderly approach to solve system problems.**

- **The SDLC usually incorporates the following general-purpose problem solving steps:**

  - ***Planning*:-** identify the scope and boundary of the problem, and plan the development strategy and goals.
  - ***Analysis*: -** study and analyze the problems, causes, and effects. Then, identify and analyze the requirements that must be fulfilled by *any* successful solution.
  - ***Design*: -** if necessary, design the solution not all solutions require design.
  - ***Implementation*: -** implement the solution.
  - ***Support*: -** analyze the implemented solution, refine the design, and implement improvements to the solution. Different support situations can thread back into the previous steps.

►►Modern Business Trends and Implications for the System Analyst:-

System analysts are being significantly influenced and affected by several business trends such as Business process redesign and Continuous process improvement

- **Business process redesign (BPR)** is the study, analysis, and redesign of fundamental business processes to reduce costs and improve value added to the business.
- **Continuous process improvement (CPI)** is the continuous monitoring of business processes to affect small but measurable improvements to cost reduction and value added.

►►A review fundamental of information systems:-

Most experts agree on the fundamental difference between data and information.

- **Data** are raw facts about the organization and its business transactions. Most data items have little meaning and use by themselves.

- **Information** is data that has been refined and organized by processing and human intelligence.

2.2 Information System Development CONCEPTS[4]

This section introduces a system development life cycle-based methodology as the process used to develop information systems.

2.2.1 System Development Life Cycles and Methodologies

- The process used to develop information systems is calleda *methodology***.**
- All methodologies are derived from a logical system problem-solving process that is sometimes called a *system development life cycle*.

- **A system development life cycle (SDLC):-**
  Is a logical process by which systems analysts, software engineers, programmers, and end-users build information systems and

- Computer applications to solve business problems and needs. It is sometimes called an application development life cycle.

- **Amethodology** :
  Is the physical implementation of the logical life cycle that incorporates (1) step-by-step activities for each phase, (2) individual and group roles to be played in each activity, (3) deliverables and quality standards for each activity, and (4) tools and techniques to be used for each activity?

- A true methodology should encompass the entire system's development life cycle.
- Most modern methodologies incorporate the use of several development tools and techniques.
- Methodologies ensure that a consistent, reproducible approach is applied to all projects.
- Methodologies reduce the risk associated with shortcuts and mistakes.
- Methodologies produce complete and consistent documentation from one project to the next.

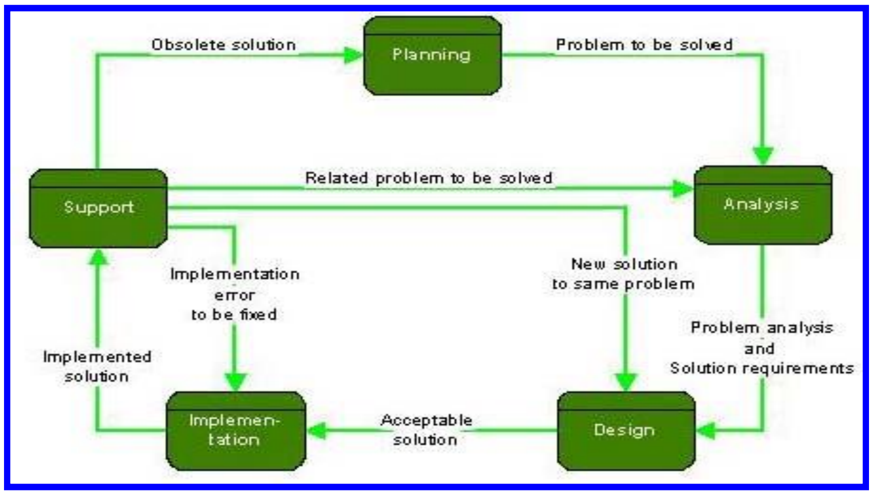

Obsolete solution
Planning
Problem to be solved
Related problem to be solved
Support
Analysis
Implementation error to be fixed
New solution to same problem
Problem analysis and Solution requirements
Implemented solution
Implemen-tation
Acceptable solution
Design

Figure 2.1A System Development Life Cycle.

2.2.2 Underlying Principles of Systems Development

There are some general principal that should underlie all systems development methodologies.

- Get the Owners and Users Involved
- Use a Problem-Solving Approach
- Establish Phases and Activities
- Establish Standards for Consistent Development and Documentation
- Justify Systems as Capital Investments
- Don't Be Afraid to Cancel or Revise Scope
- Divide and Conquer
- Design Systems for Growth and Change

2.2.3 Cross Life Cycle Activities[5]

- **Cross life cycle activities** are activities that overlap many or all phases of the methodology, in fact; they are normally performed in conjunction with several phases of the methodology.

- Cross life cycle activities include:

  - Fact Finding
  - Documentation and Presentation
  - Estimation and Measurement
  - Feasibility Analysis
  - Project Management
  - Process Management.

►►Fact Finding:-

- **Fact finding** – also called information gathering or data collection is the formal process of using research, interviews, meetings, questionnaires, sampling, and other techniques to collect information about systems, requirements, and preferences.

►►Documentation and Presentation:-

- Two forms of communication that are common to systems development projects are documentation and presentation.

- **Documentation** is the activity of recording facts and specifications for a system.
- **Presentation** is the related activity of formally packaging documentation for review by interested users and managers. Presentations may be either written or verbal.

►► Estimation and Measurement:-

- Information systems are significant capital investments. For this reason, estimation and measurement activities are commonly performed to address the quality and productivity of systems.

- **Estimation** is the activity of approximating the time, effort, costs, and benefits of developing systems. The term estimation (as in "make a guess") is used to describe the same activity in the absence of reliable data.

- **Measurement** is the activity of measuring and analyzing developer productivity and quality (and sometimes costs).

►►Feasibility Analysis:-

- **Feasibility** is a measure of how beneficial the development of an information system would be to an organization.
- **Feasibility analysis** is the activity by which feasibility is measured.

►►Project Management and Process Management:-

- Systems development projects may involve a team of analysts, programmers, users, and other IS professionals who work together.

- ***Project management*** is the ongoing activity by which an analyst plans, delegates, directs, and controls progress to develop an acceptable system within the allotted time and budget.
- ***Process management's*** intent is to standardize both the way we approach projects, and the deliverables we produce during projects.
- ***Process management*** is an ongoing activity that establishes standards for activities, methods, tools, and deliverables of the life cycle.

2.3 Basic components Of Information System CONCEPTS[6]

Every system can be considered a black box, which has input, process, files and output. The basic difference between a business system and any other is that the majority of the business system outputs are in the form of information (formatted data) printed in reports, invoices, screens, and so on, while most other systems' outputs are not printed information

➢ Data input:-

Data input to a computer must be structured (organized) so that they can be accepted by the computer data are normally grouped in to sets or chains called records.

Each record has a primary key (i.e., the data element or elements that identify it).

➢ Data stored:-

Data can be stored in a variety of ways and in a variety of structures. From the user's viewpoint the data are either in *random access storage* such as disk files or in *batch storage* such as tape files. Records in a file can be *independent* or *dependent* on one another in a hierarchy or a network.

➢ Output data:-

Output data are that are read by people. As such, they must be presented to the reader in a format and structure that he can use. Any output record may have the same structure as a record in the file, be part of a file record, or could result from data combined from two or more records in the file.

➢ File maintenance:

File maintenance is the processing of input data to update the data stored in the file. (The term *file* includes data base management system.) The programs or modules maintain each record in the file or file individually. Each input record's content and structure are read and matched with the record definition in the program or data base management system. If it does not match the record definition, it will be rejected. If it matches the record definition, it could be further validated with criteria established in the program or module. Assuming that it is a valid record, it may then update a record in file or combine with other input data and / or record in file and update a particular record. Human readable output data obtained during File maintenance should relate only to information regarding acceptance or rejection of the input data. (These output data are not to be confused with the data obtained during output production.)

The basic functions performed during file maintenance are:

- Addition of records
- Deletion of records
- Modification of records (the addition deletion or the updating of data attributes)
- Reading (accessing)data in a record without updating them

➢ Output production:-

In output production, programs or modules take data from a file or files and produce “reports,” such as video screens, invoices checks, control reports, and files. It is worth emphasizing that by dividing the data processed into file maintenance and output production, the updating of the file has been completely separated from the production of output. Also, the production of a single report or a set of reports during one output production process is completely independent of every similar output production process.

➢ File / data base content and structure:-

It also follows that changing the file in content or organization has a direct impact on file maintenance and output production. So if a file is well designed in terms of the user’s functional needs, it should not be necessary to change it unless the user’s functions change.

➢ Systems, programs, and modules:-

We may appear to be putting the cart before the horse the terms *system, program,* and *module* have been used freely to this point without discussing what they mean.

System: a set of processes that meet specific needs. These processes require input and provide output to meet these need system exist everywhere and the world would collapse if nature did not operate very effective systems.

Program: a physical entity in a computer system which can stand alone. It has a specific name by which it is called or accessed from a library.

Module: a set of computer instructions that executes instructions on data input and produces data as output.

2.4 Application System Development Methodologies

➢ Generalized model of an ASDM:-

The major activities that make up an application system development methodology are as follows:

- Feasibility study
- Business specification
- System specification
- System design
- System development and testing
- System implementation
- System review

➢ Feasibility study:-

Project scope User’s system objectives Performance requirements interfacing systems General description of system to be developed with alternate choices Impact on the organization Impact on the computer environment Development cost operating cost Benefits and risks.

➢ Business specification:-

Definition of the business objectives or the functions Definition of the data required to the objectives or the functions Definition of the logical records and files Definition of the data input Definition of the outputs (if required) Identification of when output or file data are required Identification of the need for centralized or decentralized files or databases Definitions of the input process logic Definitions of the output process logic (if required).

➢ System specification:-

Logical system divided into computerized and manual processes possible implementation options such as on-line and batch update, on-line data Access, and so on.

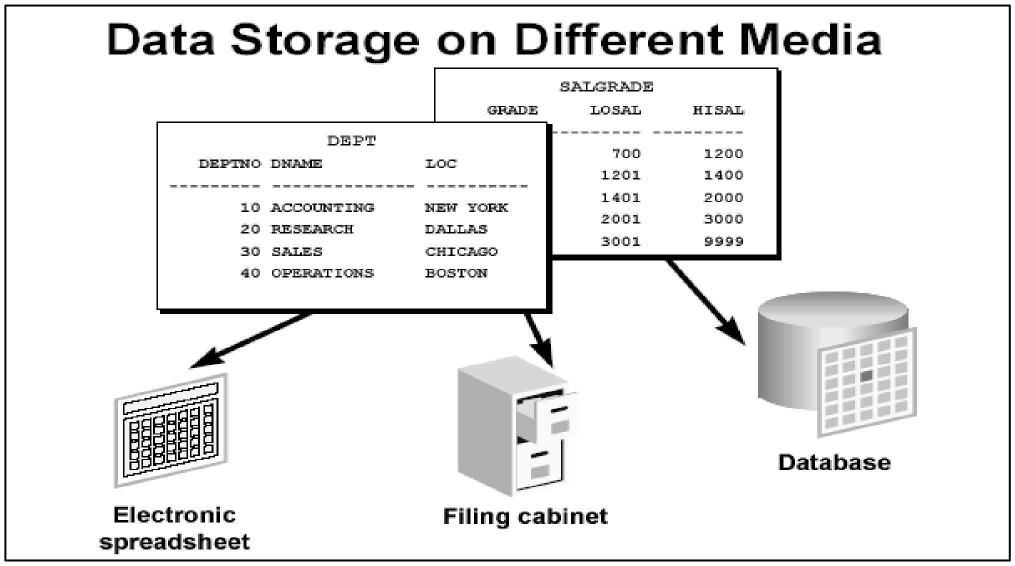
- **Strategy and Analysis System design:-**

Physical file or data-base design Network design Physical architecture of subsystems and programs Detailed program and model logic Test plans File conversion plans Hardware and software acquisition and installation plans Implementation strategy.

2.5 A System Development Life Cycle

From concept to production, you can develop a database by using the system development life cycle, which contains multiple stages of development? This top-down, systematic approach to database development transoms business information requirements into an operational database.

- Study and analyze the business requirements. Interview users and managers to identify the information requirements. Incorporate the enterprise and application mission statements as well as any future system specifications.
- Build models of the system. Transfer the business narrative into graphical representation of business information needs and rules.
- Experts.
- Design
- Design the database based on the model developed in the strategy and analysis phase.
- Build and Document
- Build the prototype system. Write and execute the commands to create the tables and supporting objects for the database.



SALGRADE

GRADE	LOSAL	HISAL
	700	1200
	1201	1400
	1401	2000
	2001	3000
	3001	9999

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Figure 2.2 Data Storage on Different Media

Every organization has some information needs. A library keeps a list of members, books, due dates, and fines.

An organization needs to save information about employees, departments, and salaries. These pieces of information are called *data*.

Organizations can store data on various media and in different formats, such as a hard-copy document in a filing cabinet or data stored in electronic spreadsheets or in databases.

A ***database***is an organized collection of information.

To manage databases, you need database management systems (DBMS). A DBMS is a program that stores, retrieves, and modifies data in the database on request. There are four main types of databases: *hierarchical*, *network*, *relational*, and more recently *object relational*

2.6 Components of the Relational Model

- Collections of objects or relations that store the data
- A set of operators that can act on relations to produce other relation
- Data integrity for accuracy and consistency

Models are a cornerstone of design. Engineers build a model of a car to work out any details before putting it into production. In the same manner, system designers develop models to explore ideas and improve the understanding of the database design.

Objective is to produce a model that fits a multitude of these uses, can be understood by an end user, and contains sufficient detail for a developer to build a database system.

In an effective system, data is divided into discrete categories or entities. An entity relationship (ER)

Model is an illustration of various entities in a business and the relationships between them. An ER model is derived from business specifications or narratives and built during the analysis phase of the system development life cycle. ER models separate the information required by a business from the activities performed within a business. Although businesses can change their activities, the type of information tends to remain constant. Therefore, the data structures also tend to be constant.

- **Benefits of ER Modeling**
  - Documents information for the organization in a clear precise format.
  - Provides a clear picture of the scope of the information requirement.
  - Provides an easily understood pictorial map for the database design.
  - Offers an effective framework for integrating multiple applications

chapter THREE

SYSTEM ANALYSIS AND DESIGN

System Analysis

3.1 Introduction

In business, System Analysis and Design refers to the process of examining a business situation with the intent of improving it through better procedures and methods.
System analysis and design relates to shaping organizations, improving performance andachieving objectives for profitability and growth. The emphasis is on systems in action,the relationships among subsystems and their contribution to meeting a common goal.

Looking at a system and determining how adequately it functions, the changes to be made and the quality of the output are parts of system analysis.

Organizations are complex systems that consist of interrelated and interlocking subsystems.

Changes in one part of the system have both anticipated and unanticipated consequences in other parts of the system.
The systems approval is a way of thinking about the analysis and design of computer based applications. It provides a framework for visualizing the organizational and environmental factors that operate on a system.
When a computer is introduced into an organization, various functions' and dysfunction'soperate on the user as well as on the organization. Among the positive consequences are improved performance and a feeling of achievement with quality information.

Among theunanticipated consequences might be a possible threat to employees job, decreased morale of personnel due to back of involvement and a feeling of intimidation by users due to computer illiteracy. The analyst's role is to remove such fears and make the system a success.
System analysis and design focus on systems, processes and technology.

⮑ **Formally,** *System analysis*is the dissection of a system into its component pieces for purposes of studying how those component pieces interact and work. With respect to information system development, System analysis is the survey and planning of the system and project, the study and analysis of the existing business and information system, and the definition of business requirements and priorities for a new or improved system.

⮑ **The *Purpose of this phase*** is to draw a whole view for the activities of different administrations, and the tasks which were charged by these administrations. This makes us imagine the initial vision for the applications which are possible to be constructed in order to mimic the reality.

➢ *Output of this stage:-*

- **There are many deliverables of this phase as the following:-**

1. A data store (form) displays the data for different administrations and includes organizational structure, organizational level, and supervision kind.
2. A data store clarifies the tasks and the responsibilities charged by each organizational unit.
3. A data store clarifies a description and analysis of tasks for each organizational unit.
4. A data store displays the relations analysis among different organizational units.
5. A data store display a common data stores, tables, and reports that are used by different organizational units with a general form to clarify the coding of different data stores, and another general form to clarify the coding of different organizational units.
6. A block diagram (box figure) clarifying the relations among different administrations through a common data stores.
7. Gathering the different applications for each organizational unit level.
8. A block diagram for each application individually.

- There are various techniques that can be used at this phase, here we will explain and clarify how this phase can be accomplished using different techniques as follows:-

- The main technique which was used at this phase is **Fact-Finding technique**.

- ***Fact-Finding*** **technique** (it is also called *data collection* or *informationgathering* technique.) is the formal process of using research, interviews, questionnaires, sampling, and other technique to collect information about systems and their requirements

  - ***The common Fact-Finding technique are :-***

    - Research and site visits.
    - Sampling of existing documentation, forms, and databases.
    - Observation of the work environment.
    - Questionnaires.
    - Interviews.

3.2 STRATEGIES AND TECHNIQUES FOR SYSTEM ANALYSIS

There are several popular or emerging strategies for system analysis. These techniques can be used in combination with one another. Some of these strategies such as follow:

- Structured Analysis and Design Technique (SADT)
- Information Engineering (IE)
- Structured Requirement Definition (SRD)
- Jackson System Development (JSD)
- Higher-Order Software (HOS)
- Prototyping

3.3 DATA MODELING

- One way to structure unstructured problems is to draw models.
  - **A model** is a representation of reality. Just as a picture is worth a thousand words, most system models are pictorial representations of reality.
- Models can be built for existing systems as a way to better understand those systems, or for proposed systems as a way to document business requirements or technical designs.

►►What are Logical Models?:-

- **Logical models** show what a system 'is' or 'does'. They are implementation-independent; that is, they depict the system independent of any technical implementation. As such, logical models illustrate the *essence* of the system.

►►What are Physical Models?

- **Physical models** show not only *what* a system 'is' or 'does', but also how the system is physically and technically implemented. They are implementation-dependent because they reflect technology choices, and the limitations of those technology choices.

- Systems analysts use logical system models to depict business requirements, and physical system models to depict technical designs.
- Systems analysis activities tend to focus on the logical system models for the following reasons:

- Logical models remove biases that are the result of the way the current system is implemented or the way that any one person thinks the system might be implemented.
- Logical models reduce the risk of missing business requirements because we are too preoccupied with technical details.
- Logical models allow us to communicate with end-users in non-technical or less technical languages.

- Data modeling is a technique for defining business requirements for a database.

- **Data modeling** is a technique for organizing and documenting a system's DATA. Data modeling is sometimes called database modeling because a data model is usually implemented as a database. It is sometimes called *information modeling*.

- There are several notations for data modeling, but the actual model is frequently called an **entity relationship diagram** (ERD).

- An **ERD** depicts data in terms of the entities and relationships described by the data.

3.4 PROCESS MODELING

- **Process modeling** is a technique for organizing and documenting the structure and flow of data through a system's PROCESSES and\ or the logic, policies, and procedures to be implemented by a system's PROCESSES.
  Process modeling originated in classical software engineering methods.

3.4.1 Data Flow Diagram (DFD):-

A systems analysis process model consists of *data flow diagrams* (DFDs).

- A **data flow diagram (DFD)** is a tool that depicts the flow of data through a system and the work or processing performed by that system. Synonyms include bubble chart, transformation graph, and **process model**.

►►Data Flow Diagrams Versus Flowcharts:

- Processes on a data flow diagram can operate in parallel. Processes on flowcharts can only execute one at a time.
- Data flow diagrams show the flow of data through the system.
- Their arrows represent paths down which data can flow. Looping and branching are not typically shown.

- Flowcharts show the sequence of processes or operations in an algorithm or program.
- Their arrows represent pointers to the next process or operation. This may include looping and branching.
- Data flow diagrams can show processes that have dramatically different timing and flowcharts don't.

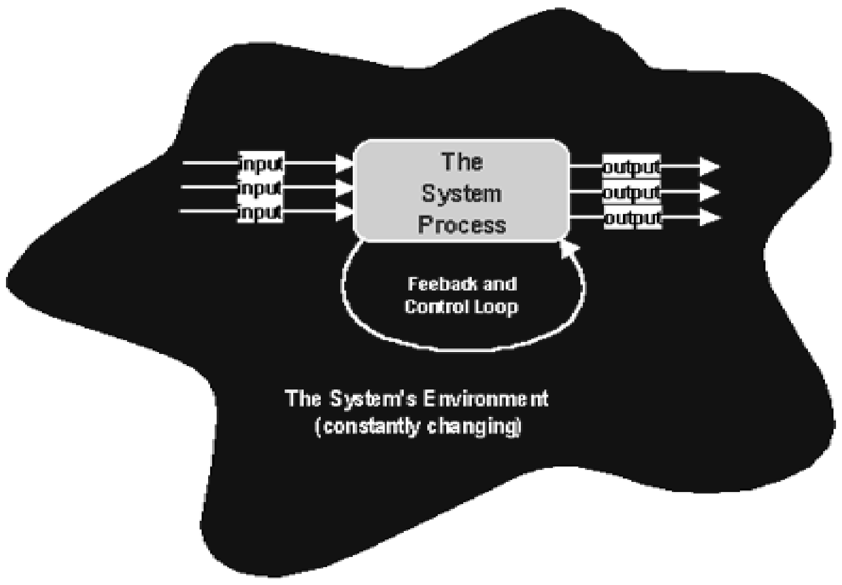


Figure 3.1 the classical process model of the system.

3.4.2 Process Decomposition

- What do you do when a complex system is too difficult to fully understand when viewed as a whole (meaning, *as a single process*)?
- In systems analysis we separate a system into its component subsystems, which in turn are decomposed into smaller subsystems, until such a time as we have identified manageable subsets of the overall system.
- This technique is called *decomposition*.

- **Decomposition** is the act of breaking a system into its component subsystems, processes, and subprocesses. Each 'level' of *abstraction* reveals more or less detail (as desired) about the overall system or a subset of that system.

- A *decomposition diagram* (also called a *hierarchy chart* or *Task Tree*, discussed in chapter5) is a popular tool to illustrate system decomposition.

  - A **decomposition diagram** shows the top down functional decomposition and structure of a system.

- A decomposition diagram is essentially a planning tool for more detailed processes models, namely, data flow diagrams.

►►Data Flows:-

- The flow of data between a system and its environment, or between two processes inside a system an relationship between a system and its environment, or between two processes is *communication.*

  - A **data flow** represents an input of data to a process or the output of data (or information) from a process. A data flow is also used to represent the creation, deletion, or update of data in a file or database (called a *data store* on the DFD).

3.5 NETWORK MODELING

- Computer Networks have become the nervous system of today’s information systems.
- The computer network is a *physical* component of an information system.
- Must be created to support the *logical* distribution of data, processes, and interfaces of an information system.

►►Network modeling:-is a technique for documenting the geographic structure of a system. Synonyms include distribution modeling and *geographic modeling*

- The need for network modeling is being driven by a technical trend – distributed computing.

  - **Distributed computing** is the assignment of specific information system elements to different computers which cooperate and interoperate across computer network. A synonym is **client/server computing**; however, client/server is actually one style of distributed computing.

►►A network modeling tool is needed to document what we learn about a business system's geography and requirements.

- **Network modeling** is a diagrammatic technique used to document the shape of a business or information system in terms of its business locations.

- The concept of geography is based on locations.

- A **location** is any place at which users exist to use or interact with the information system or application. It is also any place where business can be transacted or work performed.

3.6 OBJECT MODELING

- The approach of using object modeling during systems analysis and design is called *object-oriented analysis*.

►►Object-oriented analysis (OOA):-

Techniques are used to (1) study existing objects to see if they can be reused or adapted for new uses, and to (2) define new or modified objects that will be combined with existing objects into a useful business computing application.

- The object-oriented approach is centered on a technique referred to as *object modeling*.

►►Object modeling:- is a technique for identifying objects within the systems environment, and the relationships between those objects.

- The object-oriented approach to system development is based on the concept of objects that exist within a system's environment. In object-oriented approaches to systems development the definition of an object is as follows:

- An **object** is something that is or is capable of being seen, touched, or otherwise sensed, and about which users store data and associate behavior.

- The types of objects may include a person, place, thing, or event. In the object-oriented circles, the term “data” refers to what are called attributes.

  ►►**Attributes:-** are the data that represent characteristics of interest about an object.

- Each individual customer object is referred to as an object instance.

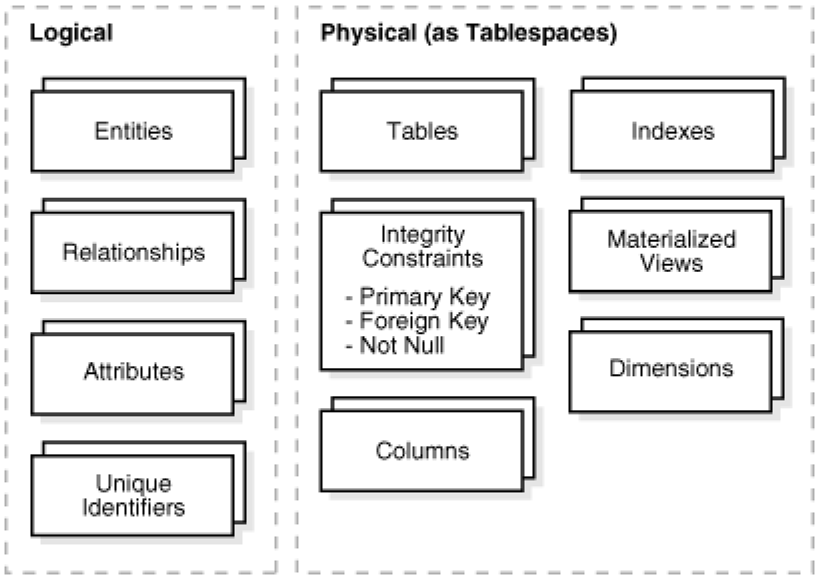
  - An **instance** (or *object instance*) of an object consists of the values for the attributes that describe a specific person, place, thing, or event.

►►What is the “behavior” of an object?

- **Behavior** refers to those things that the object can do and which correspond to functions that act on the object’s data (or attributes).

- An important concept of object modeling is the concept of categorizing objects into *classes*.
  - A **class** is a set of objects that share common attributes and behavior. A class is sometimes referred to as an *object class*.

3.7 SYSTEM DESIGN



Are the evaluationof alternative solutions and the specification of a detailed computer-based solution. The design stage is responsible for describing the functional requirement of the system (called *Logical design*). It specifies the characteristics of the system components necessary to put the logical design into action (called *Physical design*). In other words the design stage determines how the new system will work to meet the business needs defined during system analysis.

3.8 STRATEGIES FOR SYSTEM DESIGN

There are many popular strategies or techniques for performing system design. These techniques can be used in combination with one another.

- **Modern Structured Design:** a technique that focuses on processes.
- **Information Engineering (IE):** a technique that focuses on data and strategic planning to produce application projects**.**
- **Prototyping:**a technique that is an iterative process involving a close working relationship between designers and users to produce a model of the new system.
- **Joint Application Development (JAD):** techniques that emphasizes participative development among system owners, users, designers, and builders. During JAD sessions for system design, the system designer takes on the role of the facilitator.
- **Rapid Application Development (RAD):** a technique that represents a merger of various structured techniques with prototyping and JAD to accelerate systems development.

- **Object-Oriented Design (OOD):** a new system strategy that follows up object-oriented analysis to refine object requirement definitions and to define new design specific objects.

3.9 APPLICATION ARCHITECTURE AND PROCESS DESIGN

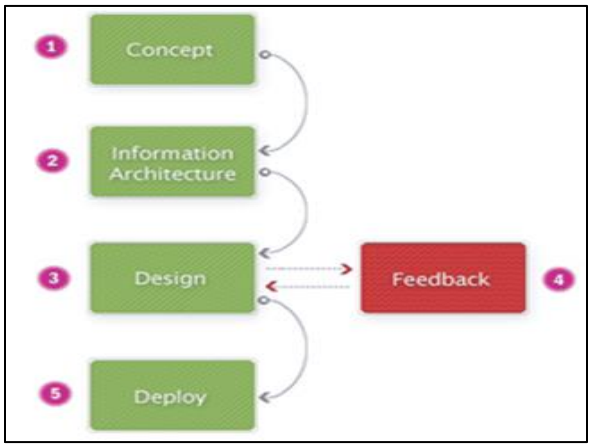


Figure 3.3Application Architecture and Process Design

Information application architecture and process design include techniques for distributing data, processes, and interfaces to network locations in a distributed computing environment.

An application architecture defines the technologies to be used by (and to build) one, more, or all information systems in terms of its data, processes, interfaces, and network components.

The prevailing computing model is currently **client / server** wherein networks of clients, single-user computers, are connected to and interoperates with servers, multiple-user computers that share their services. This is also called distributed computing.

Centralized computing, distributed presentation, distributed data, distributed data and logic, and Internet/intranet computing are flavors of client/server computing.

Client/server computing can be based on different network topologies including bus, ring, star, and hierarchical networks.

Data storage is typically implemented using distributed relational database technology that either partitions data to different servers or replicates data on multiple servers.

Processes are implemented using highly integrated tool kits called software development environments.

Physical **data flows diagrams** model an information system's application architectures and processes. Because they show the planned implementation of all processes, data stores, and data flows, they serve as a general design or blueprint for subsequent detailed design, prototyping, and construction.

System flowcharts are a lesser used diagram to show the same implementation features as a physical data flow diagrams. Although they are rarely drawn today, many older, legacy systems use them for documentation.

3. 10 THE DESIGN STAGE PHASES

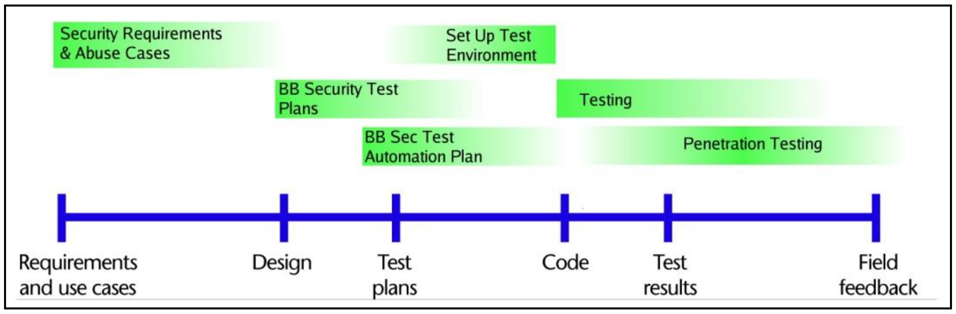


Figure 3.4Design Stage Phases

- **The logical database design**
- **The physical database design**.
- **The physical application or software design**.

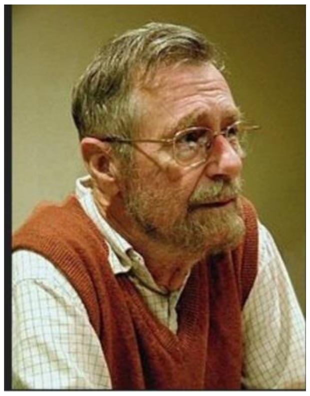
chapter

FOUR

Dijkstra's algorithm

4.1 conceived by

Dutch computer scientistEdsgerDijkstra in 1956 and published in 1959,[1][2] is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing as a subroutine in other graph algorithms, or in GPS technology.



4.2 The purpose of Dijkstra's algorithm

For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path first is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First).

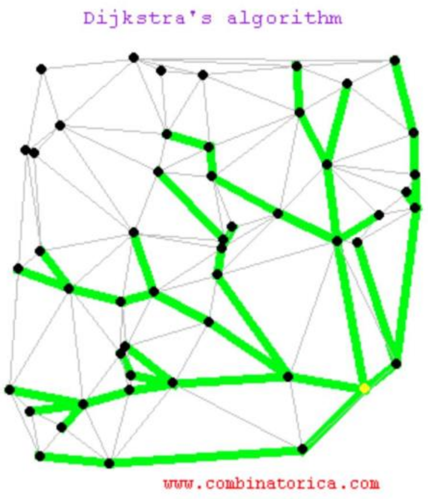
4.3 Single-Source Shortest Path Problem and solution

4.3.1 Single-Source Shortest Path Problem

a The problem of finding shortest paths from a source vertex v to all other vertices in the graph.

a Weighted graph G = (E,V)

a Source vertex s ^ V to all vertices v ^ V



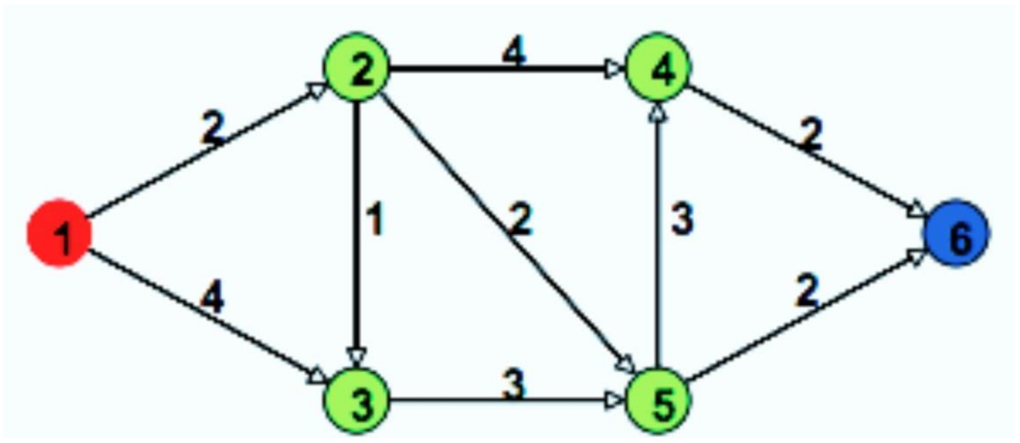
4.3.2 Solution to the single-source shortest path problem in graph theory

Using Dijkstra’s Algorithm

¤ Both directed and undirected graphs

¤ All edges must have nonnegative weights

¤ Graph must be connected



4.4 Algorithm

let the node at which we are starting be called the **initial node**. Let the **distance of node *Y*** be the distance from the **initial node** to *Y*. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Mark all nodes unvisited. Set the initial node as current. Create a set of the unvisited nodes called the *unvisited set* consisting of all the nodes except the initial node.
3. For the current node, consider all of its unvisited neighbors and calculate their *tentative* distances. For example, if the current node *A* is marked with a distance of 6, and the edge connecting it with a neighbor *B* has length 2, then the distance to *B* (through *A*) will be 6 + 2 = 8. If this distance is less than the previously recorded tentative distance of *B*, then overwrite that distance. Even though a neighbor has been examined, it is not marked as "visited" at this time, and it remains in the *unvisited set*.
4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the *unvisited set*. A visited node will never be checked again.

5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the *unvisited set* is infinity (when planning a complete traversal), then stop. The algorithm has finished.
6. Select the unvisited node that is marked with the smallest tentative distance, and set it as the new "current node" then go back to step 3.

4.5 Description

> ***Note:*** *For ease of understanding, this discussion uses the terms* ***intersection****,* ***road*** *and* ***map*** *— however, formally these terms are* ***vertex****,* ***edge*** *and* ***graph****, respectively.*

Suppose you would like to find the shortest path between two intersections on a city map, a starting point and a destination. The order is conceptually simple: to start, mark the distance to every intersection on the map with infinity. This is done not to imply there is an infinite distance, but to note that that intersection has not yet been *visited*; some variants of this method simply leave the intersection unlabeled. Now, at each iteration, select a *current* intersection. For the first iteration the current intersection will be the starting point and the distance to it (the intersection's label) will be zero. For subsequent iterations (after the first) the current intersection will be the closest unvisited intersection to the starting point—this will be easy to find.

From the current intersection, update the distance to every unvisited intersection that is directly connected to it. This is done by determining the sum of the distance between an unvisited intersection and the value of the current intersection, and relabeling the unvisited intersection with this value if it is less than its current value. In effect, the intersection is relabeled if the path to it through the current intersection is shorter than the previously known paths. To facilitate shortest path identification, in pencil, mark the road with an arrow pointing to the relabeled intersection if you label/relabel it, and erase all others pointing to it. After you have updated the distances to each neighboring intersection, mark the current intersection as *visited* and select the unvisited intersection with lowest distance (from the starting point) -- or lowest label—as the current intersection. Nodes marked as visited are labeled with the shortest path from the starting point to it and will not be revisited or returned to.

Continue this process of updating the neighboring intersections with the shortest distances, then marking the current intersection as visited and moving onto the closest unvisited intersection until you have marked the destination as visited. Once you have marked the destination as visited (as is the case with any visited intersection) you have determined the shortest path to it, from the starting point, and can trace your way back, following the arrows in reverse.

Of note is the fact that this algorithm makes no attempt to direct "exploration" towards the destination as one might expect. Rather, the sole consideration in determining the next "current" intersection is its distance from the starting point. In some sense, this algorithm "expands outward" from the starting point, iteratively considering every node that is closer in terms of shortest path distance until it reaches the destination. When understood in this way, it is clear how the algorithm necessarily finds the shortest path, however it may also reveal one of the algorithm's weaknesses: its relative slowness in some topologies.

4.6 PSEUDOCODE

In the following algorithm, the code u := vertex in *Q* with smallest dist[], searches for the vertex *u* in the vertex set *Q* that has the least dist[*u*] value. That vertex is removed from the set *Q* and returned to the user. dist_between(*u*, *v*) calculates the length between the two neighbor-nodes *u* and *v*. The variable *alt* on lines 20 & 22 is the length of the path from the root node to the neighbor node *v* if it were to go through *u*. If this path is shorter than the current shortest path recorded for *v*, that current path is replaced with this *alt* path. The previous array is populated with a pointer to the "next-hop" node on the source graph to get the shortest route to the source.

```
1  functionDijkstra(Graph, source):
 2     for each vertex v in Graph:                     // Initializations
 3        dist[v] := infinity ;                   // Unknown distance function from
 4                                              // source to v
 5         previous[v] := undefined ;                   // Previous node in optimal
path
 6     end for// from source
 7
 8     dist[source] := 0 ;                       // Distance from source to source
 9     Q := the set of all nodes in Graph ;                  // All nodes in the graph are
10                                                    // unoptimized - thus are in Q
11     while Q is not empty:                            // The main loop
12         u := vertex in Q with smallest distance in dist[] ;    // Source node in first
case
13         remove u from Q ;
14         if dist[u] = infinity:
15             break ;// all remaining vertices are
16         end if// inaccessible from source
17
18         for each neighbor v of u:                    // where v has not yet been
19                                                    // removed from Q.
20             alt :=dist[u] + dist_between(u, v) ;
21             if alt<dist[v]:                        // Relax (u,v,a)
22                 dist[v] := alt ;
23                 previous[v] := u ;
24                 decrease-key v in Q;                // Reorder v in the Queue
25             end if
26         end for
27     end while
28     return dist;
29  endfunction
```

If we are only interested in a shortest path between vertices *source* and *target*, we can terminate the search at line 13 if *u* = *target*. Now we can read the shortest path from *source* to *target* by reverse iteration:

```
1  S := empty sequence
2  u := target
3  while previous[u] is defined:                       // Construct the shortest path
with a stack S
4      insert u at the beginning of S// Push the vertex into the stack
5      u := previous[u]                                // Traverse from target to source
6  end while ;
```

Now sequence *S* is the list of vertices constituting one of the shortest paths from *source* to *target*, or the empty sequence if no path exists.

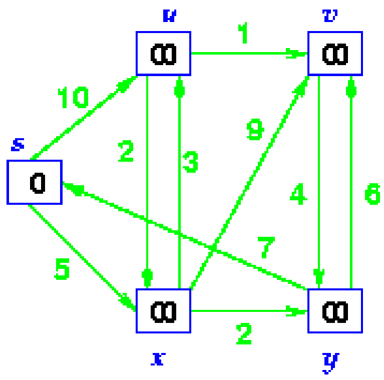
A more general problem would be to find all the shortest paths between *source* and *target* (there might be several different ones of the same length). Then instead of storing only a single node in each entry of previous[] we would store all nodes satisfying the relaxation condition. For example, if both *r* and *source* connect to *target* and both of them lie on different shortest paths through *target* (because the edge cost is the same in both cases), then we would add both *r* and *source* to previous[*target*]. When the algorithm completes, previous[] data structure will actually describe a graph that is a subset of the original graph with some edges removed. Its key property will be that if the algorithm was run with some starting node, then every path from that node to any other node in the new graph will be the shortest path between those nodes in the original graph, and all paths of that length from the original graph will be present in the new graph. Then to actually find all these shortest paths between two given nodes we would use a path finding algorithm on the new graph, such as depth-first search

4.7 Analysis

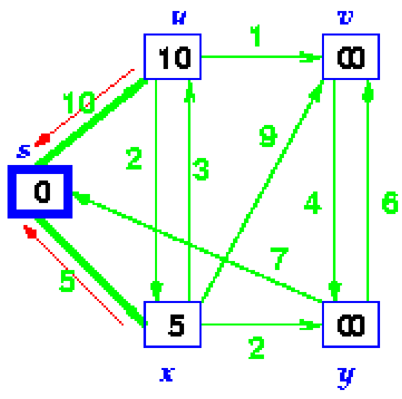
Like Prim's algorithm, Dijkstra's algorithm runs in O(|E|lg|V|) time.

EXAMPLE: STEP BY STEP OPERATION OF DIJKSTRA ALGORITHM.

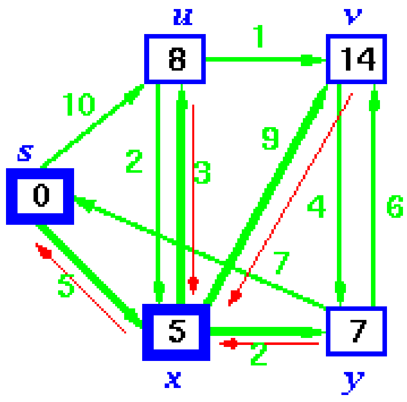
Step1. Given initial graph G=(V, E). All nodes nodes have infinite cost except the source node, s, which has 0 cost.



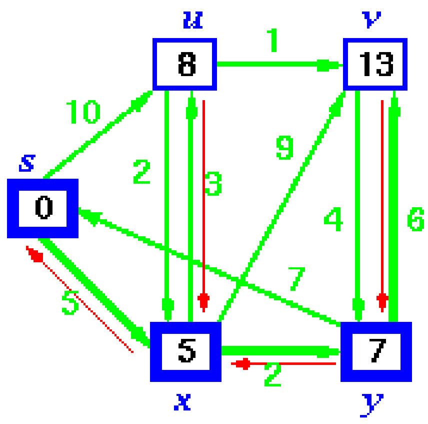
Step 2. First we choose the node, which is closest to the source node, s. We initialize d[s] to 0. Add it to S. Relax all nodes adjacent to source, s. Update predecessor (see red arrow in diagram below) for all nodes updated.



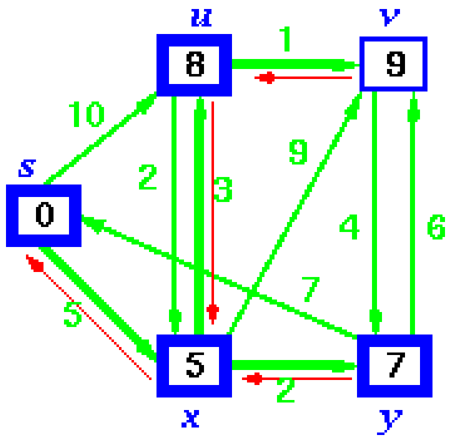
Step 3. Choose the closest node, x. Relax all nodes adjacent to node x. Update predecessors for nodes u, v and y (again notice red arrows in diagram below).



Step 4. Now, node y is the closest node, so add it to S. Relax node v and adjust its predecessor (red arrows remember!).



Step 5. Now we have node u that is closest. Choose this node and adjust its neighbor node v.



Step 6. Finally, add node v. The predecessor list now defines the shortest path from each node to the source node, s.

chapter FIVE

Forms

5.1The run of web application

Main Window

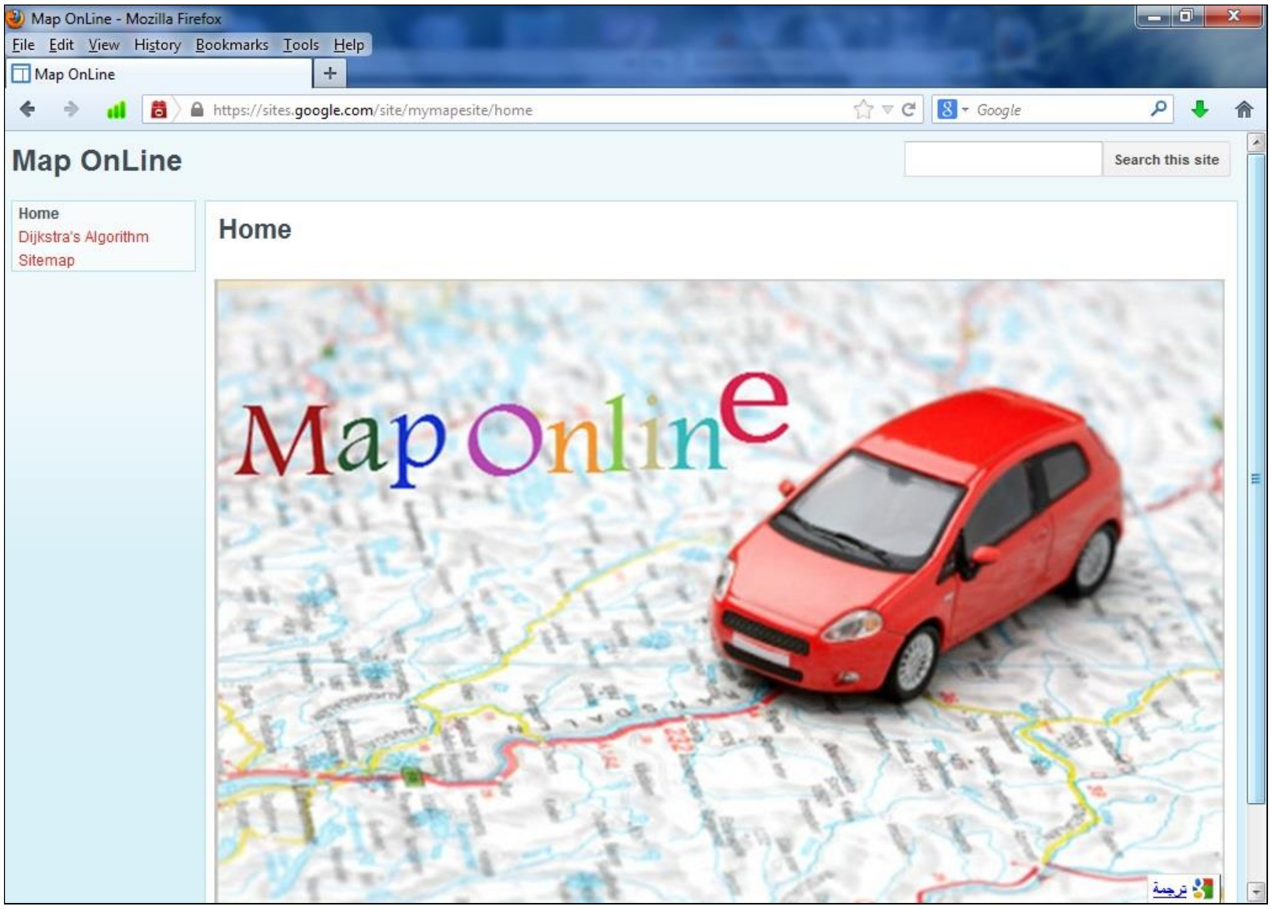
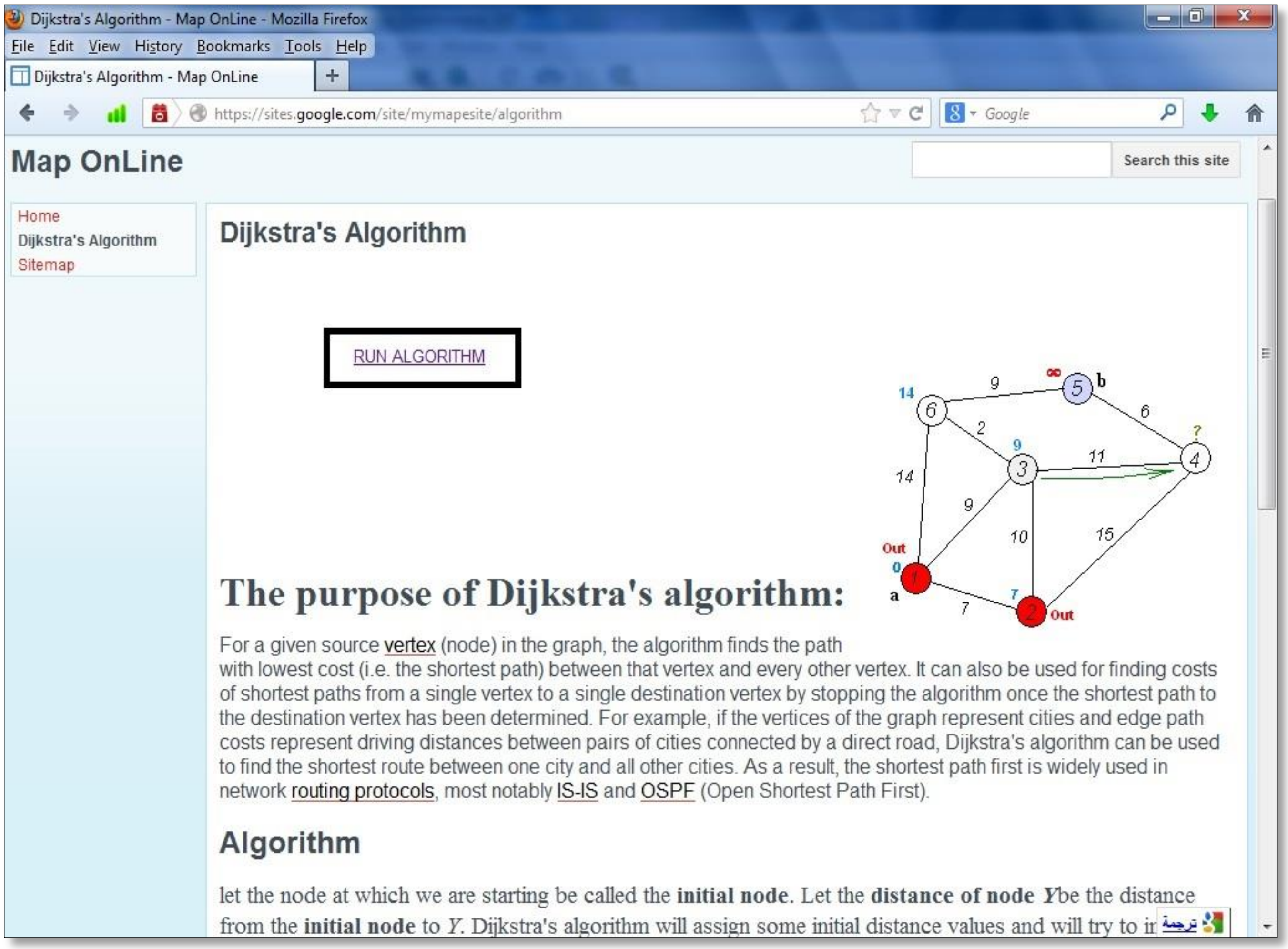

Map OnLine - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Map OnLine
https://sites.google.com/site/mymapesite/home
Google
Map OnLine
Search this site
Home
Dijkstra's Algorithm
Sitemap
Home
Map Online
ترجمة

Figure 5.1   home page

When press Dijikstra ‘s Algorithm Display this Page


Dijkstra's Algorithm - Map OnLine - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://sites.google.com/site/mymapesite/algorithm
Search this site
Home
Dijkstra's Algorithm
Sitemap
RUN ALGORITHM

Map OnLine

Dijkstra's Algorithm

The purpose of Dijkstra's algorithm:

For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path first is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First).

Algorithm

let the node at which we are starting be called the **initial node**. Let the **distance of node *Y*** be the distance from the **initial node** to *Y*. Dijkstra's algorithm will assign some initial distance values and will try to ir

Figure 5.1.1 algorithm page

To Display This Page Press On Run Algorithm From the previous Page

In this page you can choose the city you start from and the distination City

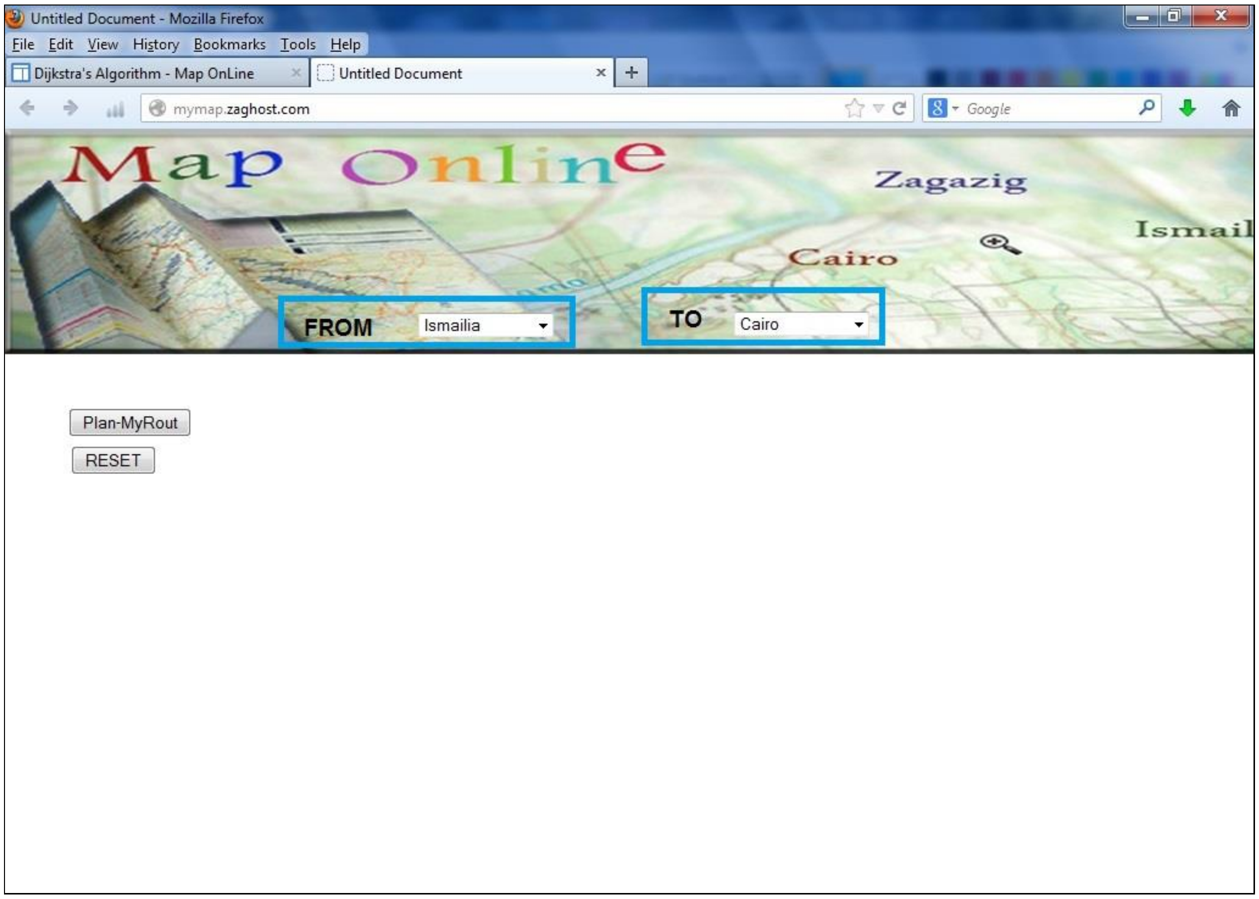

Untitled Document - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Dijkstra's Algorithm - Map OnLine
Untitled Document
mymap.zaghost.com
Google
Map Online
Zagazig
Ismail
Cairo
FROM
Ismailia
TO
Cairo
Plan-MyRout
RESET

Figure 5.1.2   choose of location and destination

To display result Press On Plane-My Rout

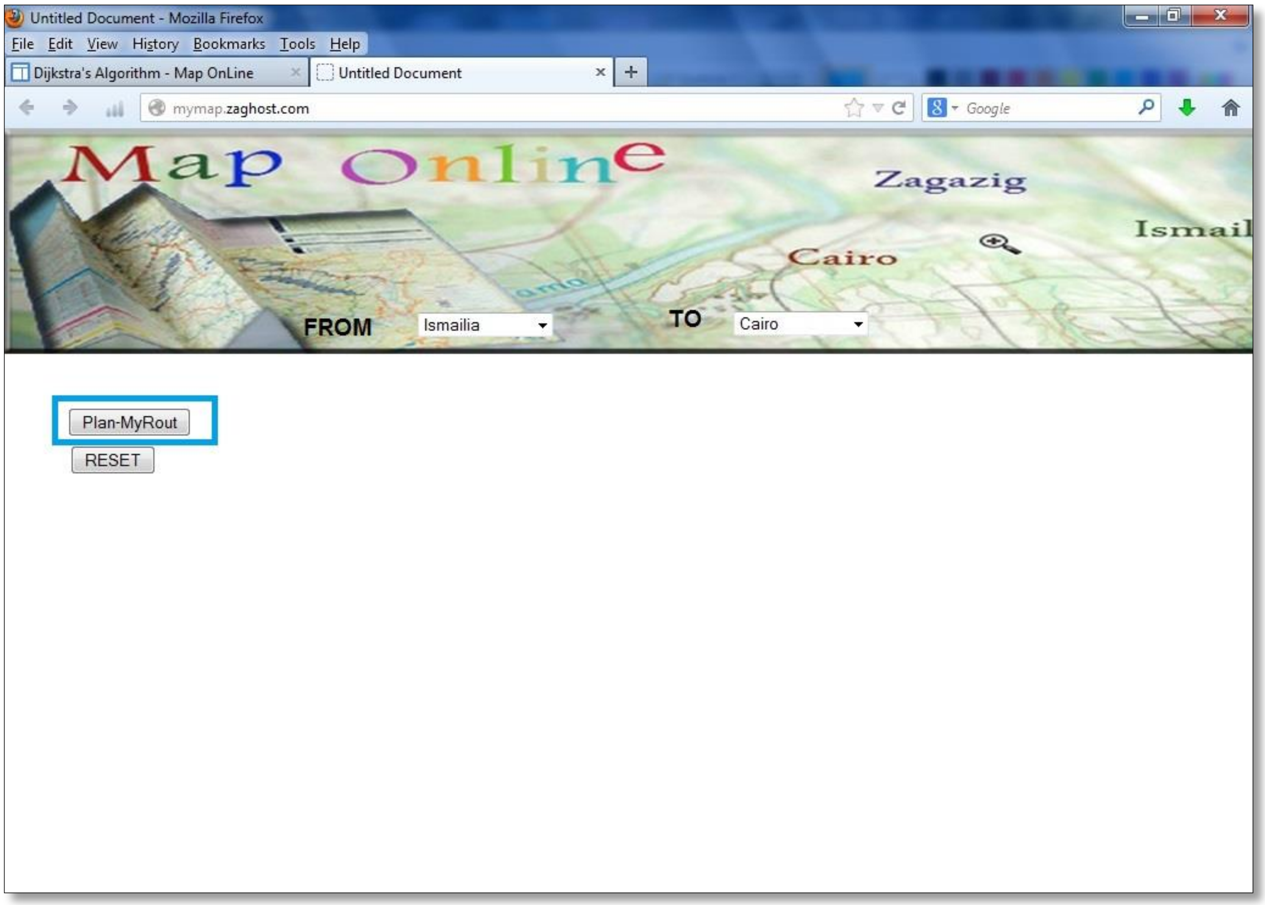
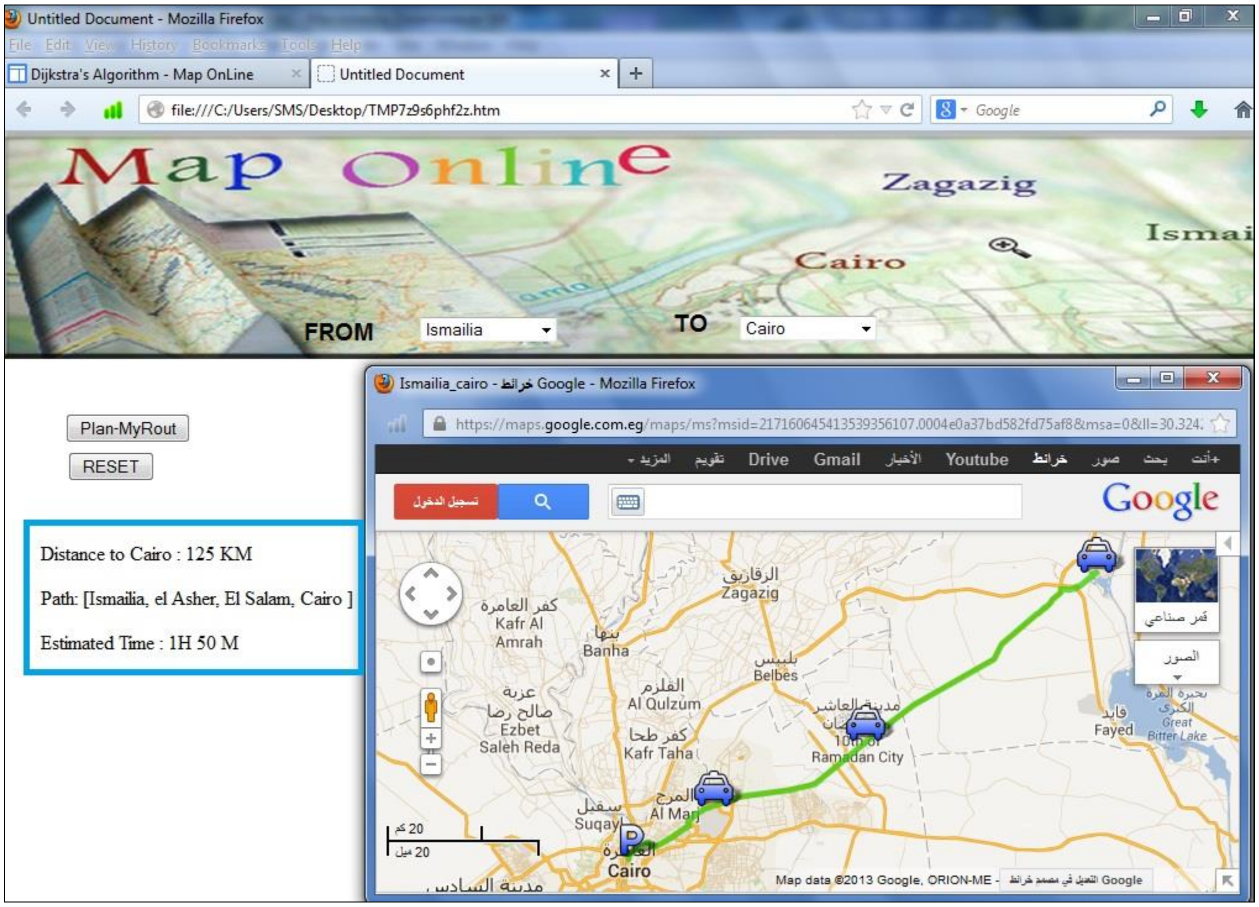

Untitled Document - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Dijkstra's Algorithm - Map OnLine
Untitled Document
mymap.zaghost.com
Google
Map Online
Zagazig
Ismail
Cairo
FROM
Ismailia
TO
Cairo
Plan-MyRout
RESET

Figure 5.1.3 press plane my route to run the algorithm

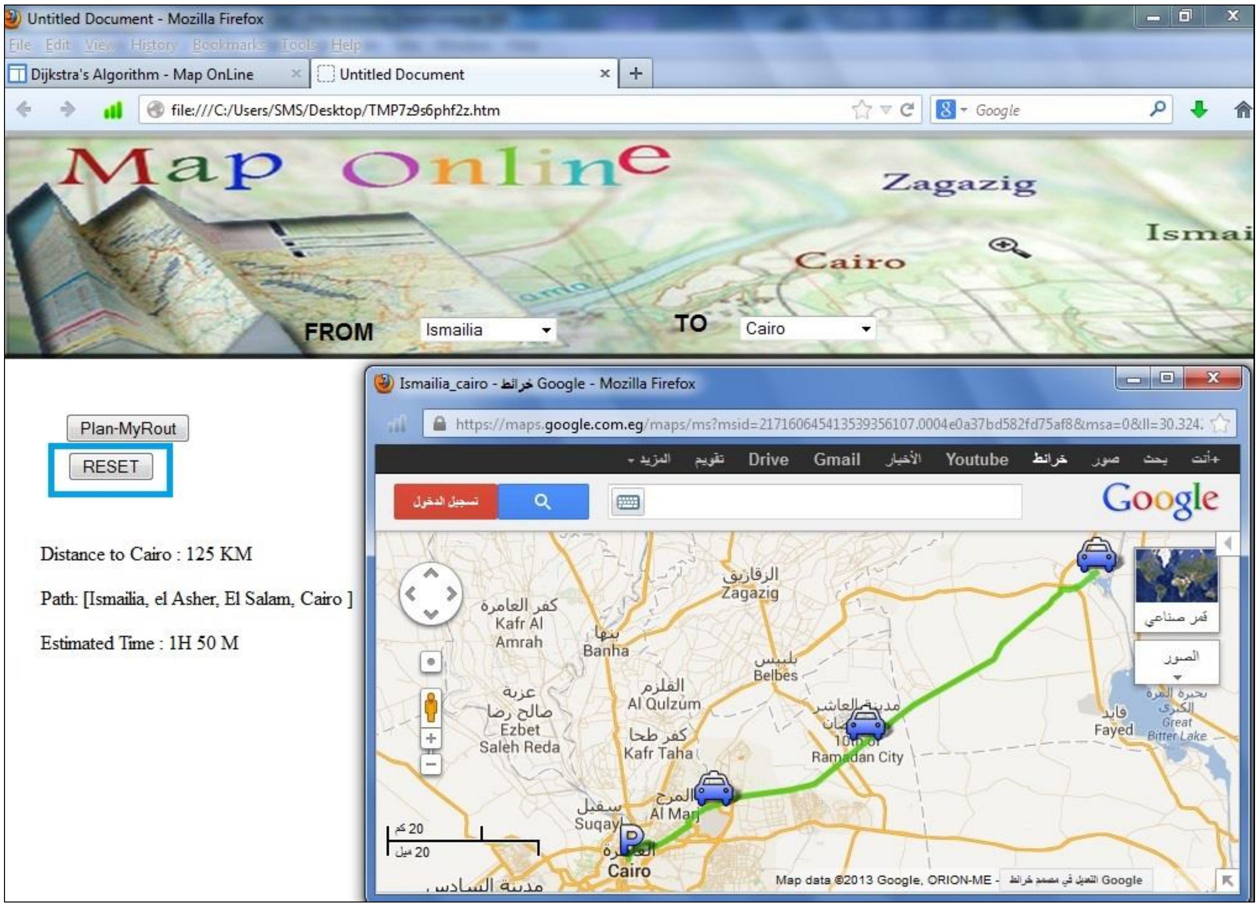


Distance to Cairo : 125 KM

Path: [Ismailia, el Asher, El Salam, Cairo]

Estimated Time : 1H 50 M

Figure 5.1.4 result of running algorithm


Untitled Document - Mozilla Firefox
Dijkstra's Algorithm - Map OnLine
Untitled Document
file:///C:/Users/SMS/Desktop/TMP7z9s6phf2z.htm
Map Online
FROM
Ismailia
TO
Cairo
Plan-MyRout
RESET
Ismailia_cairo - خرائط Google - Mozilla Firefox

Distance to Cairo : 125 KM

Path: [Ismailia, el Asher, El Salam, Cairo]

Estimated Time : 1H 50 M

Figure 5.1.5 pressing Reset button

press on reset to empty the web page

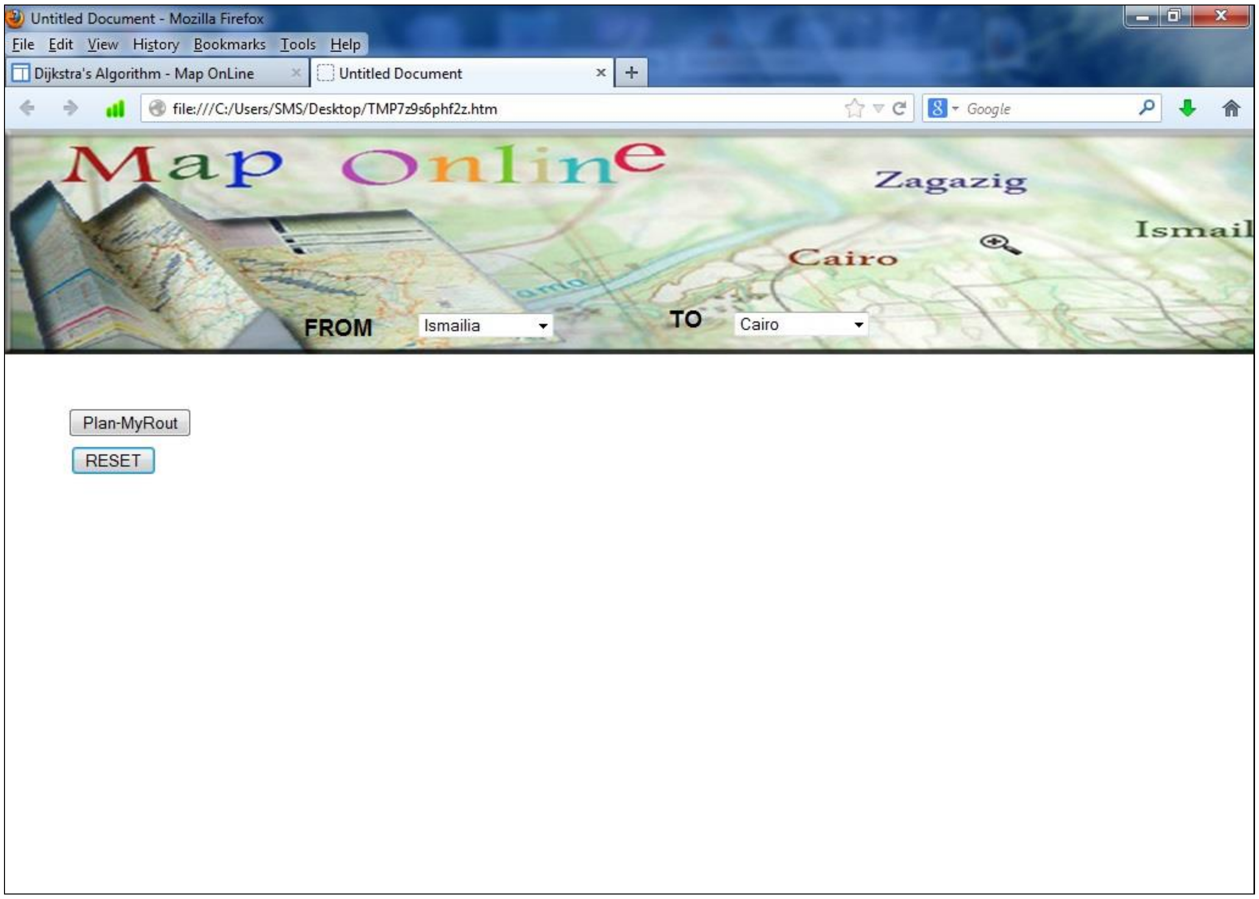
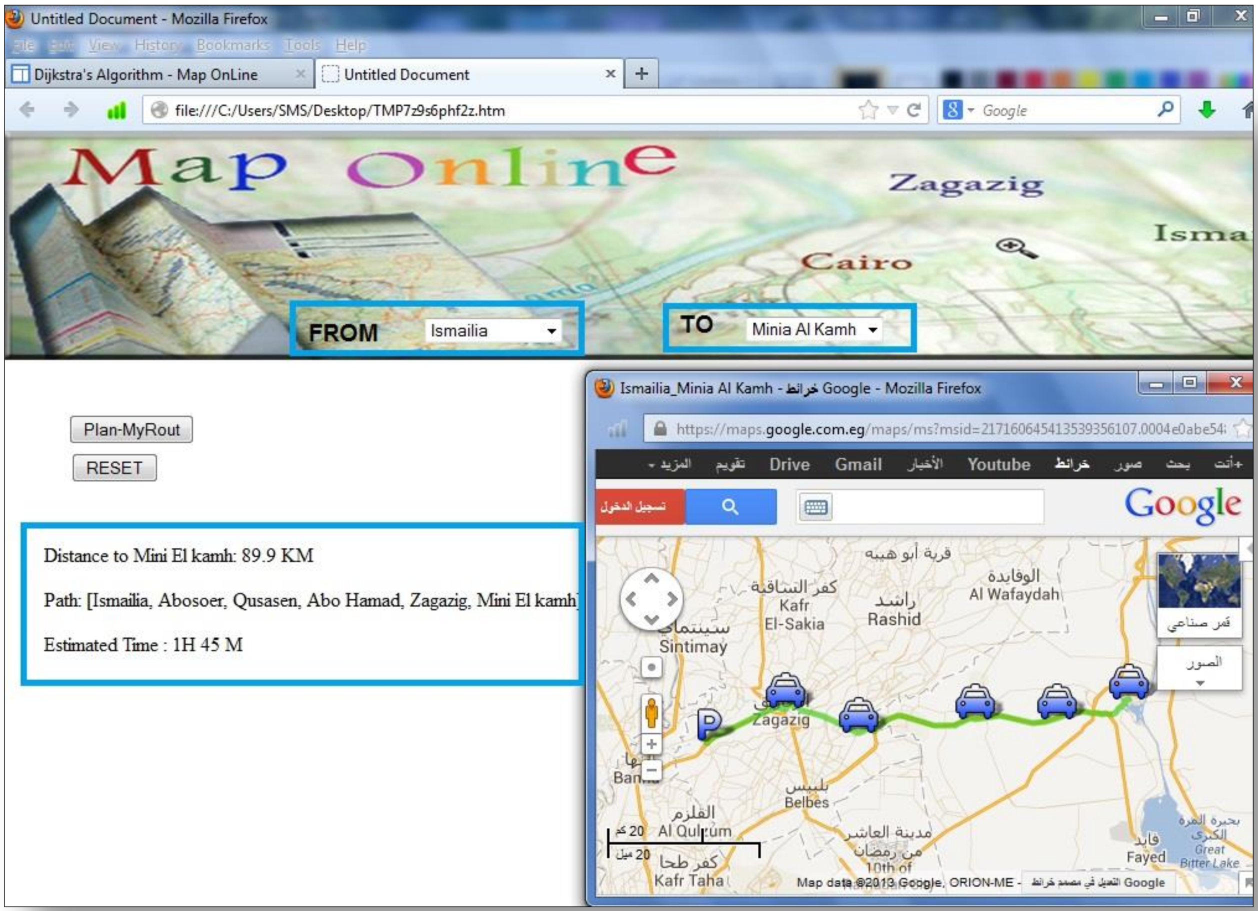

Untitled Document - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Dijkstra's Algorithm - Map OnLine
Untitled Document
file:///C:/Users/SMS/Desktop/TMP7z9s6phf2z.htm
Google
Map Online
Zagazig
Ismail
Cairo
FROM
Ismailia
TO
Cairo
Plan-MyRout
RESET

Figure 5.1.6 result of using reset button

Another Example to go from cairo to minia el kamh


Untitled Document - Mozilla Firefox
Dijkstra's Algorithm - Map OnLine
Untitled Document
file:///C:/Users/SMS/Desktop/TMP7z9s6phf2z.htm
Map Online
Zagazig
Cairo
FROM
Ismailia
TO
Minia Al Kamh
Plan-MyRout
RESET
Ismailia_Minia Al Kamh - خرائط Google - Mozilla Firefox
Google

Distance to Mini El kamh: 89.9 KM

Path: [Ismailia, Abosoer, Qusasen, Abo Hamad, Zagazig, Mini El kamh]

Estimated Time : 1H 45 M

Figure 5.1.7 choosing another location and destination

5.2 The run of mobile application:

Choosing application icone to run application

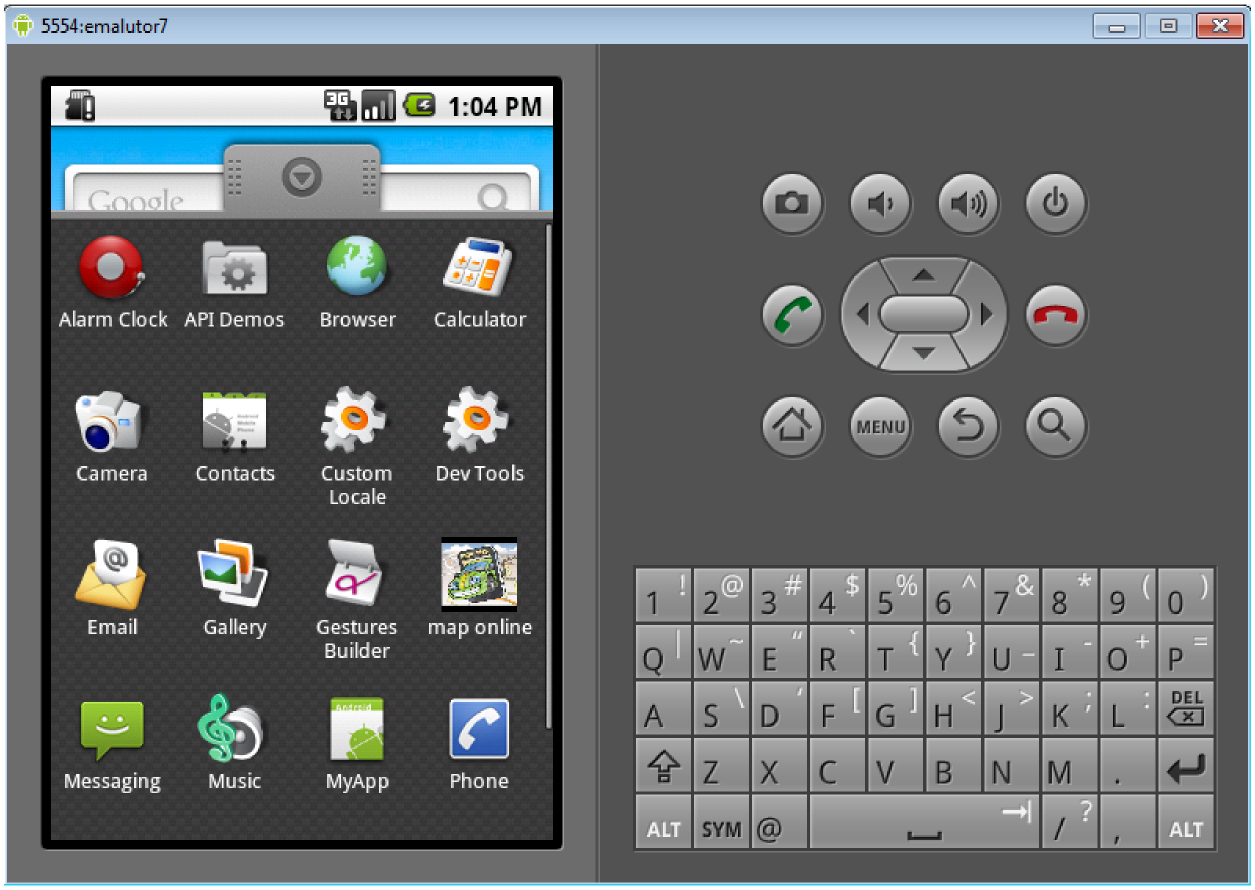


Figure 5.2.1 the application icon

The main window that appear to the user

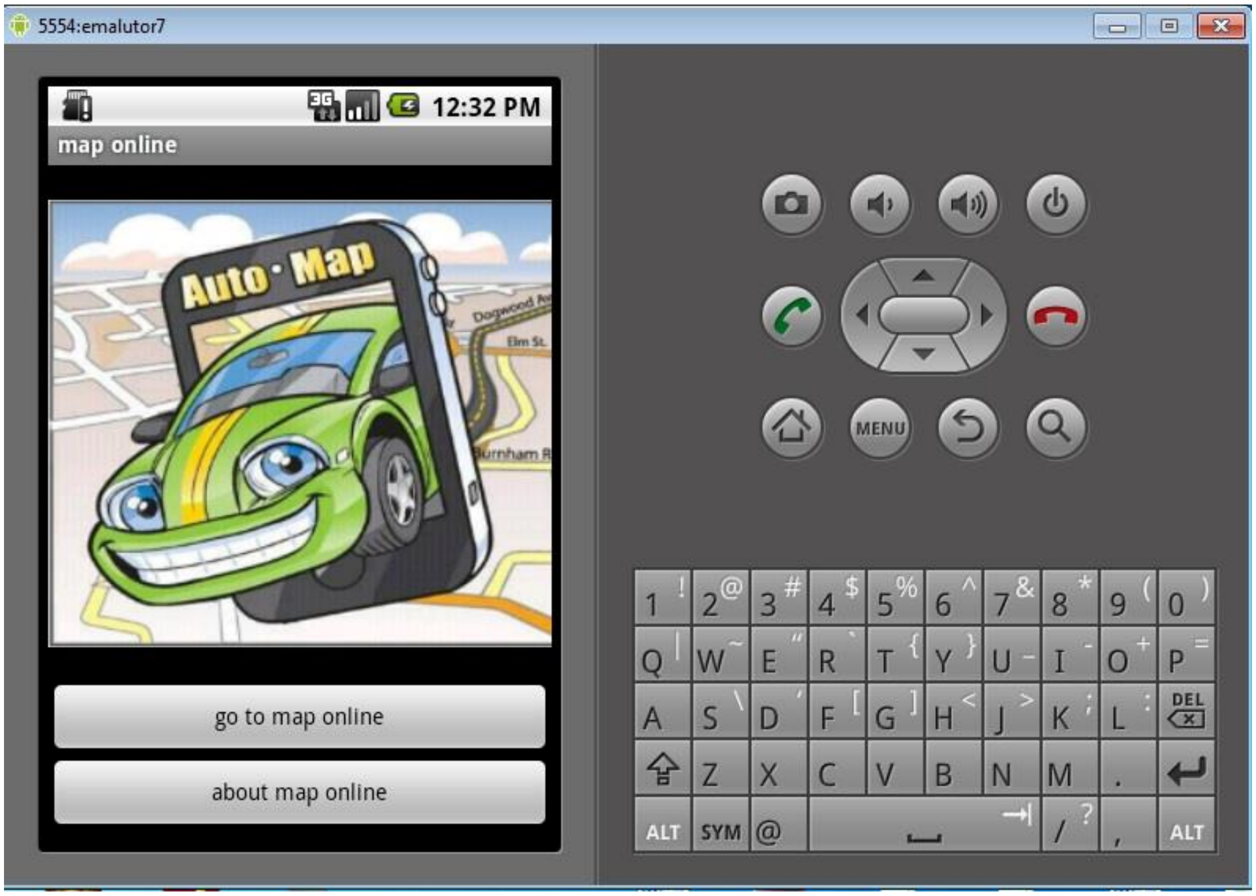


Figure 5.2.2 the main page

Download of road data

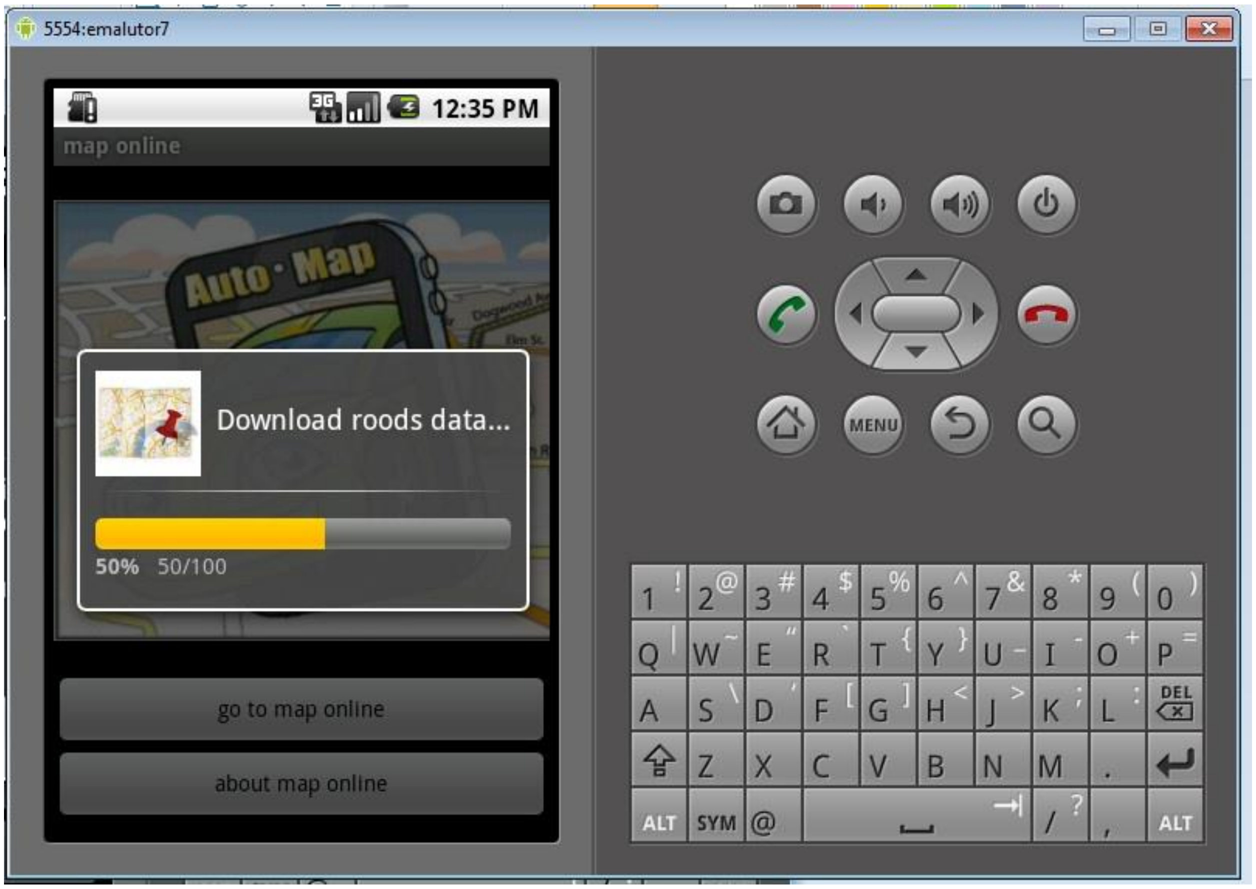


Figure 5.2.3 loading data

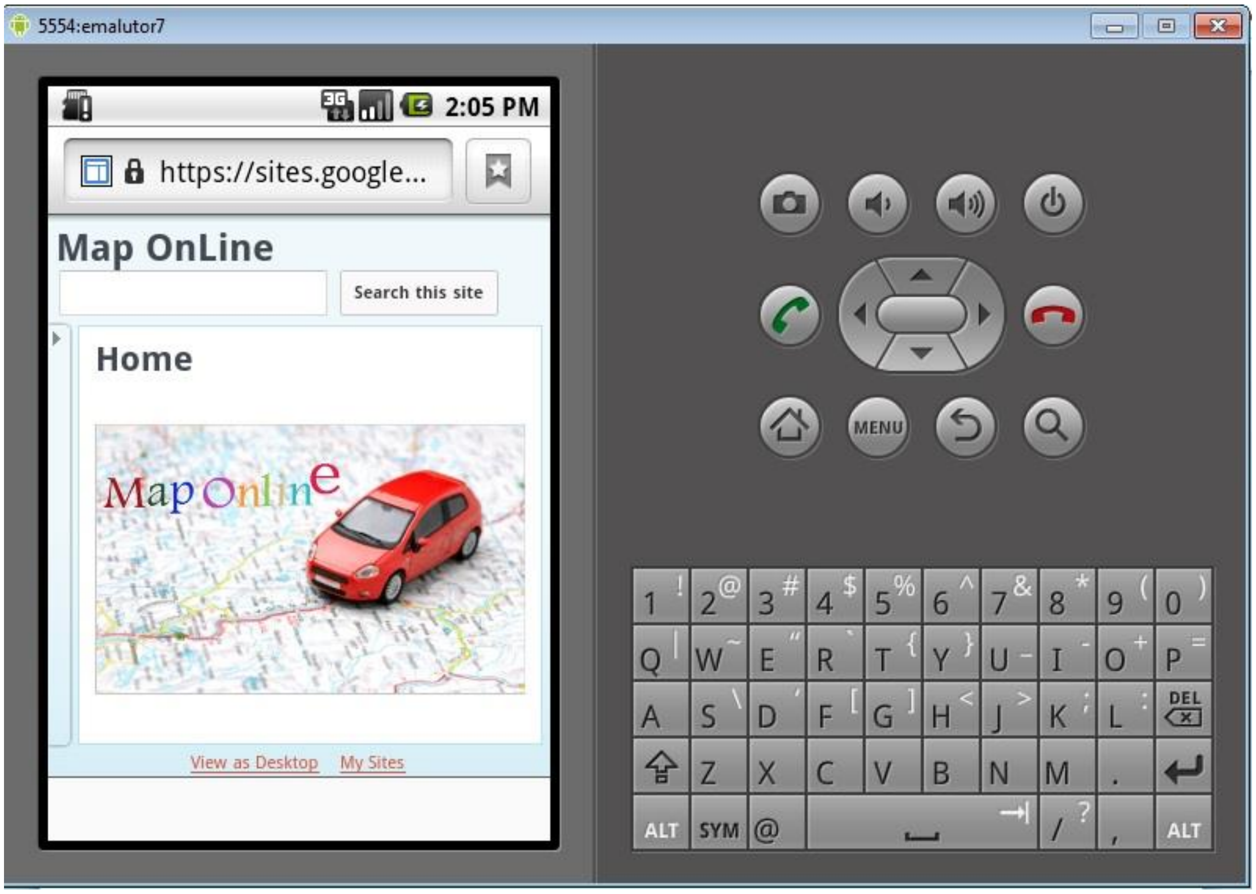


Figure 5.2.4 home page

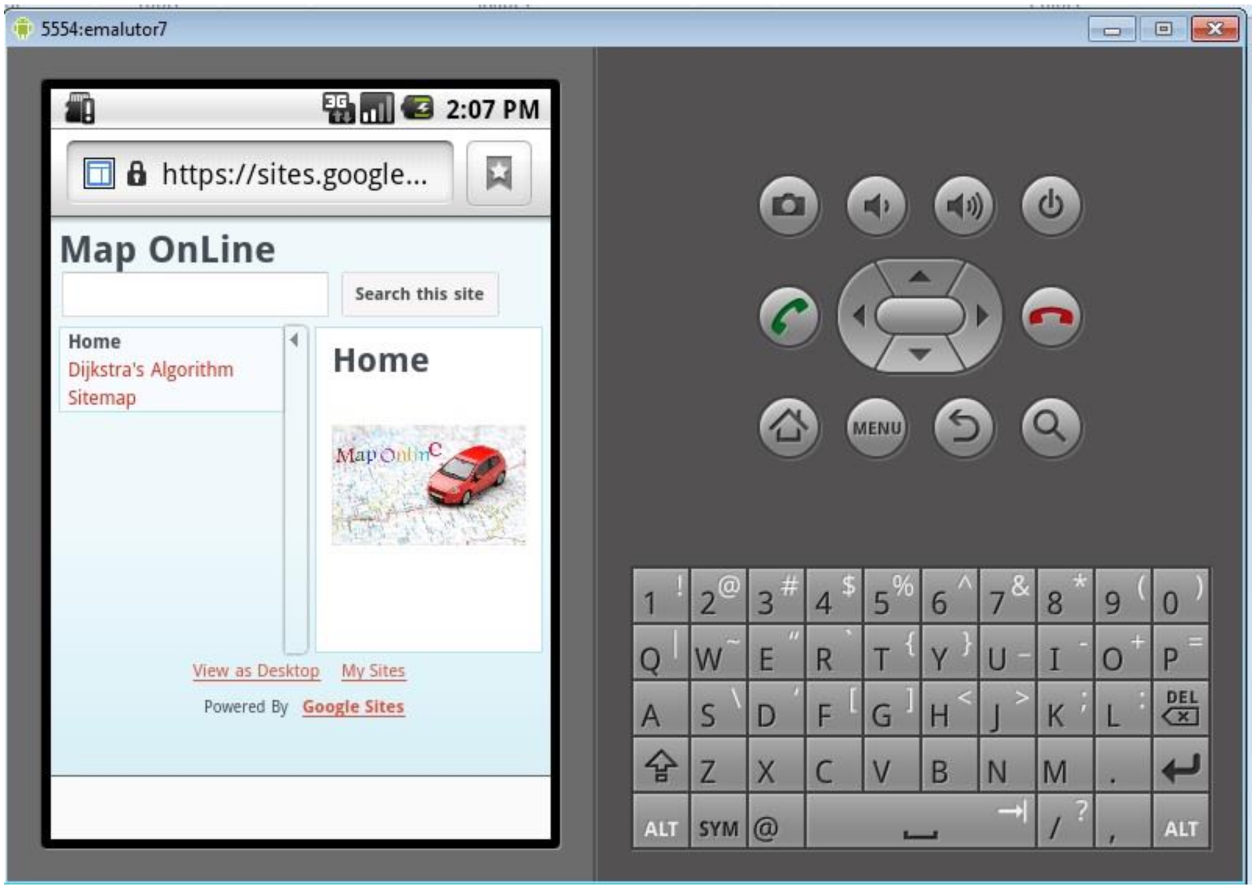


Figure 5.2.5 home page data

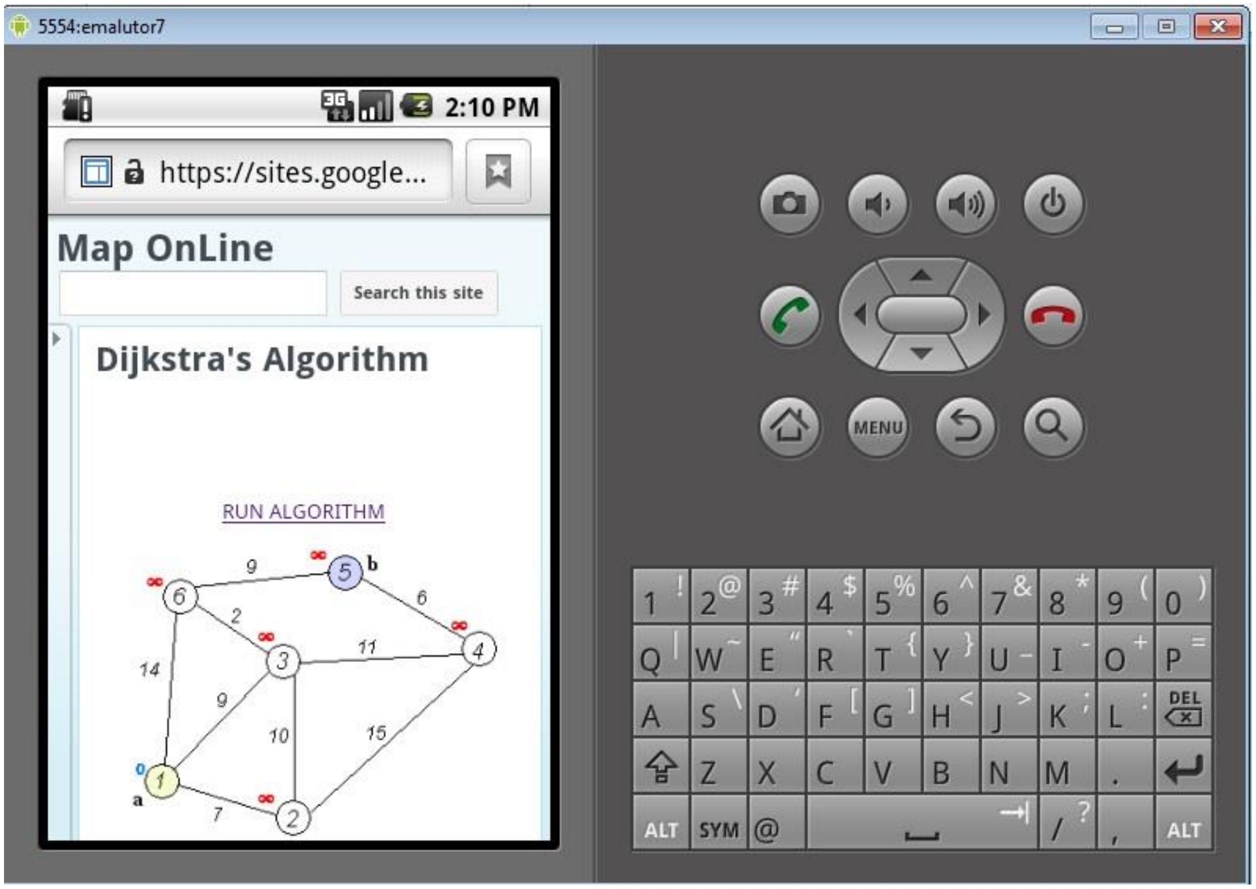


Figure 5.2.6 dijkstra algorithm page

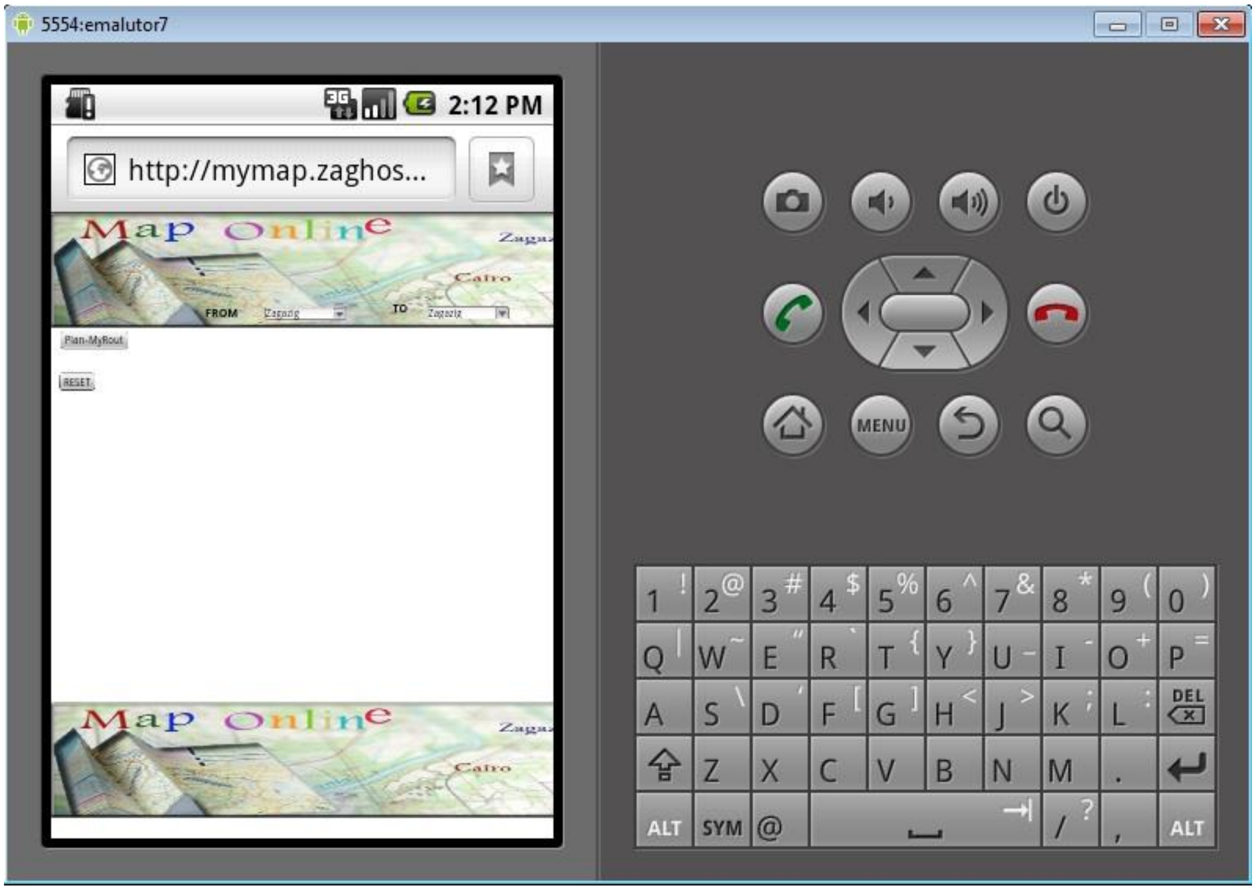


Figure 5.2.7 choosing location and distance and the run

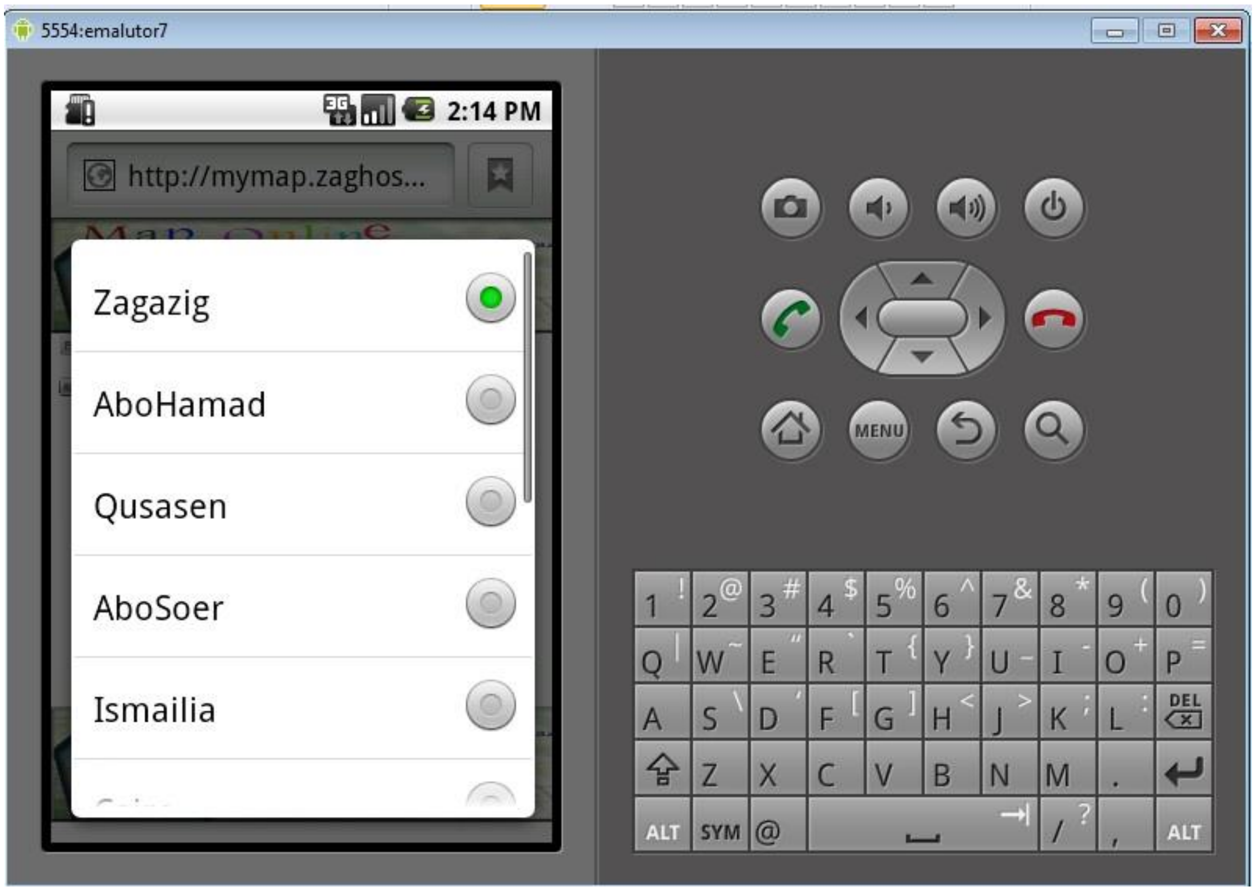


Figure 5.2.8 choosing cities

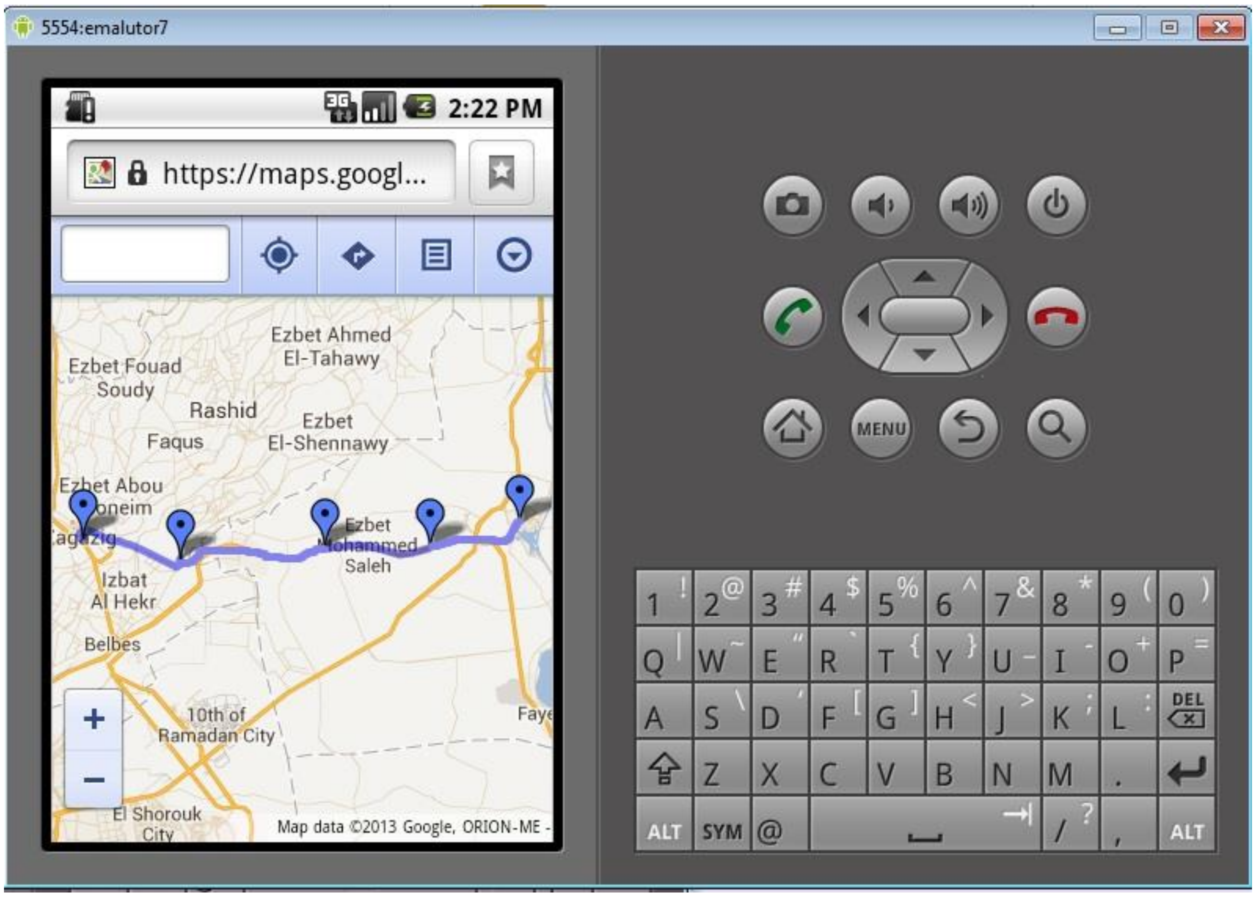
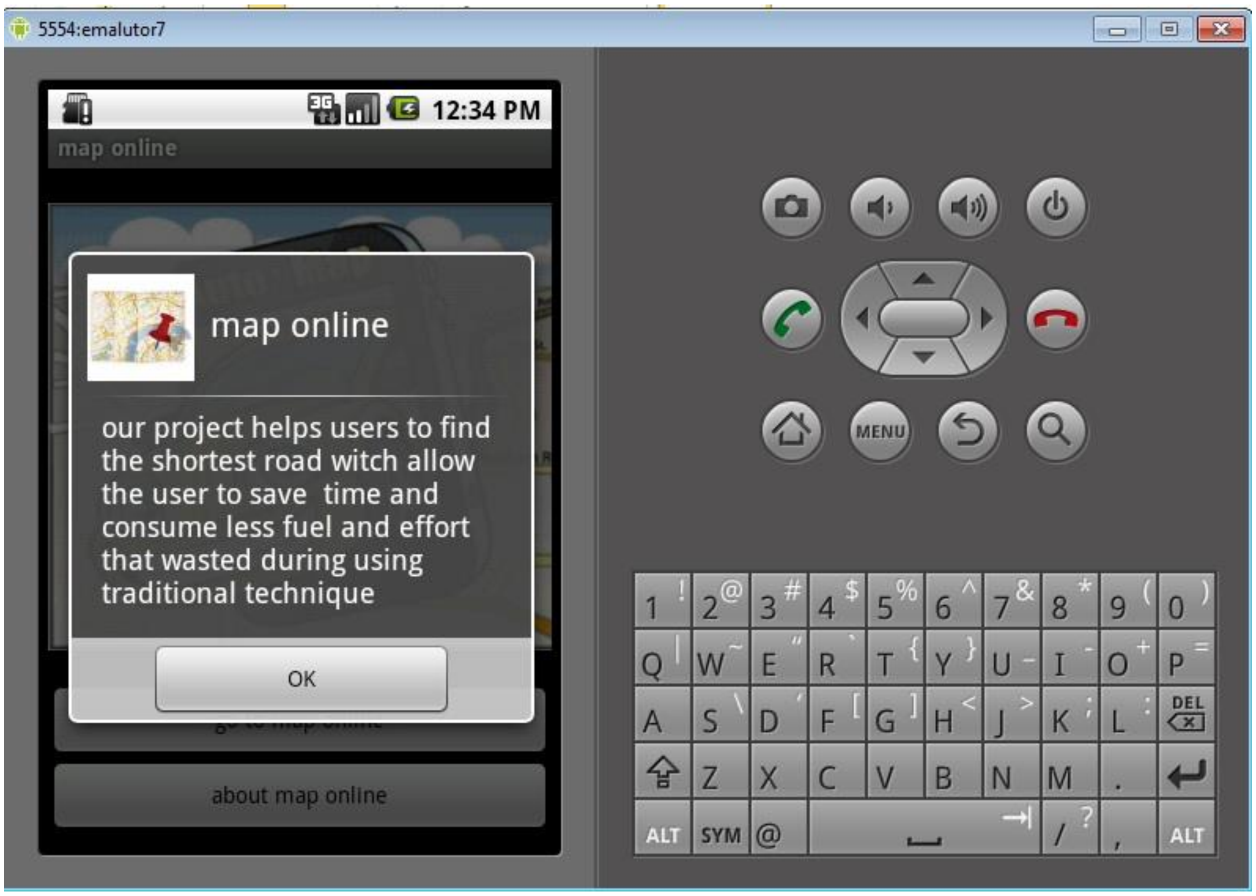


Figure 5.2.9 map of shortest bath

Overview about map online project



our project helps users to find the shortest road witch allow the user to save time and consume less fuel and effort that wasted during using traditional technique

Figure 5.2.10 result of pressing about map online reset button

chapter SIX

EVALUATION AND FUTURE WORK

6.1 EVALUATION PHASE

This our evaluation about the project that we have finished this will contain:-

- Analysis and survey phase.
- Design phase.
- Coding phase.
- Implement phase.

6.1.1 Evaluating of project member team

Before the last phases there are some aspects that must be in our consider is that how this team work. This team was very perfectly organized as the tasks have been divided about the team member during the all year. There are things that effect on our team and make success to accomplish the project **from these things**:-

I. Members of Team understand each other's.
II. Team Spirit encourages us to complete this project.
III. Although everyone of team lives in different places, but they cooperate to finish this project and they take the working on the project as challenge and there were feeling with the responsibility and duties about working on group.
IV. Patience and Persistence

6.1.2 Evaluating survey and analysis phase

In this phase we can evaluate our effort that there was an organized session of working to the organization that we make the project for it .we think that we must make a new project for new place that really need such a projects for using IT in his field so we make our decision to make the project for it this after take the permission from our supervising doctor and the management of our faculty members.

6.1.3 Evaluating design phase

We can say that is the evaluation of alternative solutions and the specification of a detailed computer-based solution. as we aren't experience at the first time it take very much time to end it as the final framework.

6.1.4 Coding phase

That was the simplest phase for our team work group as we have finished a good analysis for the work at the organization so it was an easy way to code what can see, also as we are so good in using the JAVA language.

6.1.5 Evaluating implement phase

From our opinion, this was the hardest phase we phase because we didn't know anything about how to deal with PHP language. But we persisted to learn it to complete the project. And we thank our supervisor.
There was many updating for the screen that we have implemented after revising the users who will use this program as the user is suitable one to give you an information about how will the program will be.

6.1.6 Evaluating the time

As the time is important element at any project. The time in our project is in our side to complete the project before determinate time. Sometime we take all days of week for the project to produce it in better image.

6.1.7 Evaluating the project My Map Online

Project My Route Online was very helpful and cooperative from the beginning of our project to the finishing it. They support us with information we need and this facilitate our mission.

6.1.8 Evaluating of the supervisor

We take the opinion of our supervisor in all phases of the project, he solve any problem that faced us with giving the solutions and recommendations. He guides us with better things. He follows all phases of the project and satisfied with our work. We thank him for his cooperation.

6.2 Conclusion

6.2.1 WE ACCOMPLISH IN OUR PROJECT

we make mobile web application with Andorid OS:

• This an application that calculates all the possible routes and gives you the best route .

• Concentrate on getting driving directions and computing best route based on shortest path leeds to the destination

If user need to travel from place to anthor and there are multiple routes leeds to your desired distination ,the used algorithm gives you the shortest path to the desired destination and Time spent .

When user use this application and want to travel from zagazig to cairo:
Gives all direction leads to same destination.

There are multiple routes leeds to cairo :-
First:- zagazig , mnia el kmh , banha,qalub , cairo
Second:- zagazig , belbes , el salam , cairo
Third:- zagazig,abo hammad , el qsasen,abo soir , Ismailia,el asher , el salam , cairo
, and find the shortest path from zagazig to cairo with lowest distance and lowest time.
Finally show map and direction that views the shortest route to user .

Plan your routes using Map Online with directions for driving and road maps to save the planning time and create efficient routes.

6.3 Future Work

Support or add new multiple road maps to project for avaliable to any user to enable to travel to desired loaction.(increase road maps)

Add web capability to supports Multiple Locations (more than 2 locations)

Add services that view these services on best route on map to help user such as Gas station,Hospitals,…etc .

Calculates all the possible routes and gives you the best route according crowedd path .