DEVELOPMENT OF WEBSITE FOR CHOOSING A TOURIST ROUTE IN THE ARAB REPUBLIC OF EGYPT

GRADUATE QUALIFICATION WORK
SUSU–02.04.02.2018.308-578.GQW

Supervisor
Cand. Sci., Assoc. Prof.
__________ O.N. Ivanova

Author,
student of the group CE-219
__________ N.A.M. Abo-Dalla

Normative control
__________ O.N. Ivanova
“___”__________ 2018

Chelyabinsk–2018
TASK of the master graduate qualification work
for the student of the group CE-219
Nabil Abo-Elela Mohammed Abo-Dalla
in master direction 02.04.02
“Fundamental Informatics and Information Technologies”
(master program “Database Technologies")

1. The topic (approved by the rector’s order from 04.04.2018 No. 580)
Development of a website for the choosing of a tourist route in the Arab Republic of Egypt

2. The deadline for the completion of the work: 01.06.2018.

3. The source data for the work
3.1. Chanel “Clever Techi” on youtube [Electronic resource] URL: https://www.youtube.com/channel/UC1WxZFhq56x1oxXH-XveSQ (date of access: 03.01.2018).
3.3. Chanel “Traversy Media” on youtube [Electronic resource] URL: https://www.youtube.com/channel/UC29ju8bIPH5as8OGnQzwJyA (date of access: 03.11.2018).

4. The list of the development issues
4.1. To define the problem statement
4.2. To make a comparative analysis of analogues
4.3. To choose development tools
4.4. To determine functional and non-functional requirements
4.5. To design the database
4.6. To design and to implement the web-application
4.7. To test the system.

5. **Issuance date of the task**: 09.02.2018.

**Supervisor**

Cand. Sci, Assoc. Prof. O.N. Ivanova

The task is taken to perform N.A.M. Abo-Dalla
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ......................................................................................... 5

INTRODUCTION ........................................................................................................ 6

1. THE ANALYSIS OF THE SUBJECT AREA ......................................................... 8

  1.1. The problem statement ............................................................................... 8

  1.2. Comparative analysis of analogues ............................................................... 11

  1.3. Choice of development tools ...................................................................... 15

2. DESIGN OF WEB-APPLICATION FOR ONLINE TRIP MAKER. ............. 19

  2.1. Functional and non-functional requirements .............................................. 19

  2.2. Use case diagram ......................................................................................... 19

  2.3. Database description .................................................................................... 20

  2.4. Development of user interfaces ................................................................. 21

3. IMPLEMENTATION OF WEB-APPLICATION FOR ONLINE TRIP MAKER ........................................ 30

  3.1. Components of the project ......................................................................... 30

  3.2. Some examples of code .............................................................................. 31

  3.3. Interfaces of the application ...................................................................... 43

4. TESTING OF A WEB-APPLICATION .............................................................. 54

  4.1. Methods of testing ..................................................................................... 54

  4.2. Functional testing ....................................................................................... 56

  4.3. User interface testing .................................................................................. 57

CONCLUSION ........................................................................................................... 59

REFERENCES ........................................................................................................... 60
ACKNOWLEDGEMENTS

I want to acknowledge and express my gratitude to the South Ural State University and Lagos State University Nigeria, which gave me the opportunity to obtain a master's degree from Russia.

My special appreciation goes to my project supervisor PhD, associate professor, O.N. Ivanova for her supports and guidance during this scientific work. My warm and sincere appreciation also goes out to the lecturers in the department for instilling quality knowledge, academic and moral disciplines into me.

I’d like to thank senior lecturer K.V. Borodulin and PhD, associate professor M.L. Zymbler for their supporting, helping and knowledge.
INTRODUCTION

Topicality

Almost of tourists have a problem to find places, which they can visit in Egypt. In Egypt, we have different structures for building even if you use any of navigations applications it is hard to find the place you looking for beside that in Egypt so many places which is unknown for many; but it has really great places to visit. We understand that each person has his own interest and different conditions. It is really too hard to find a place which matches your situation, so I made a website to help tourists finding where and how they can enjoy their trip.

Research goal and objectives

The goal of the research is the development of a website for the choosing of a tourist route in the Arab Republic of Egypt.

For the reaching this goal we must solve the following objectives:
1) to identify the problem statement;
2) to make comparative analysis of analogues;
3) to choose development tools;
4) to design a web-application and a database;
5) to implement the web-application;
6) to test the system.

The practical significance

This project is useful to help any tourist to enjoy his vacation as possible as he can.

This project can be useful because it contains:
1) secure access for the private data about the users, including the individual identification at the site for all users;
2) saving the time and efforts while working at site;
3) saving money for the users by hiring guide;
4) give many options for the user to choose his trips by his own wishes and conditions;
5) to share a new location with others but not directly;
6) easily check information about place before he visits;
7) ability for the following development of the site.

**Structure of the thesis**

The thesis consists of four chapters, introduction, conclusion and reference list.

In the first chapter, the problem statement is given, as well as the overview and comparative analysis of analogues. Additionally, we describe the chosen development tools here.

In chapter two, there is a description of functional requirements, use case diagram, database scheme and the design of the application interfaces.

In chapter three, there is a components of the project, some examples of SQL-queries and interfaces of the application.

Chapter Four is devoted to the testing of the application.

The thesis has 61 pages; the list of references contains 23 resources.
1. THE ANALYSIS OF THE SUBJECT AREA

1.1. The problem statement

Each country has a lot of places to visit. Egypt has got a lot of places to be seen from foreigners but foreigners don’t know how to go or when those places work. It’s extremely hard for foreigners to know information about places which they never visited without guide.

Last year, 14.7 million tourists visited Egypt, and tourism generated $11 billion in revenue, according to the Egyptian Tourist Authority in New York. Tourism in Egypt makes up about 11 percent of the gross domestic product, the tourism organization said.

The most popular objects you can visit in Egypt it depends on your interest, but let’s show the most popular places which you can visit in Egypt.

Pyramids of Giza

The last surviving of the Seven Wonders of the Ancient World, the Pyramids of Giza are one of the world's most recognizable landmarks. Built as tombs for the mighty Pharaohs and guarded by the enigmatic Sphinx, Giza's pyramid complex has awed travelers down through the ages and had archaeologists (and a fair few conspiracy theorists) scratching their heads over how they were built for centuries. Today, these megalithic memorials to dead kings are still as wondrous a sight as they ever were. An undeniable highlight of any Egypt trip, Giza's pyramids should not be missed [1].

Luxor’s Karnak temple and valley of kings

Famed for the Valley of the Kings, Karnak Temple, and the Memorial Temple of Hatshepsut, the Nile-side town of Luxor in Upper Egypt has a glut of tourist attractions. This is ancient Thebes, powerbase of the New Kingdom pharaohs, and home to more sights than most can see on one visit. While the East Bank brims with vibrant souk action, the quieter West Bank is home to a bundle of tombs and temples that has been called the biggest open air museum in the world. Spend a few days here exploring the colorful wall art of the tombs and gazing in awe at
the colossal columns in the temples, and you will see why Luxor continues to fascinate historians and archaeologists [1].

Islamic Cairo

The atmospheric, narrow lanes of the capital's Islamic Cairo district are crammed full of mosques, madrassas (Islamic schools of learning), and monuments dating from the Fatimid through to the Mameluke eras. This is where you will find the labyrinth shopping souk of Khan el-Khalili where coppersmiths and artisans still have their tiny workshops, and stalls are laden with ceramics, textiles, spice, and perfume. Surrounding the market is a muddle of roads, home to some of the most beautiful preserved architecture of the old Islamic empires. There is a wealth of history here to explore. Visit Al-Azhar Mosque and the dazzling Sultan Hassan Mosque, and make sure you climb to the roof of the ancient medieval gate of Bab Zuweila for the best minaret-speckled panoramas across the district [1].

Aswan

Egypt's most tranquil town is Aswan, set upon the winding curves of the Nile. Backed by orange-hued dunes this is the perfect place to stop and unwind for a few days and soak up the chilled-out atmosphere. Take the river ferry across to Elephantine Island and stroll the colorful streets of the Nubian villages. Ride a camel to the desert monastery of St. Simeon on the East Bank. Or just drink endless cups of tea on one of the riverboat restaurants, while watching the lateen-sailed feluccas drift past. There are plenty of historic sites here and numerous temples nearby, but one of Aswan's biggest highlights is simply kicking back and watching the river life go by [1].

Abu Simble

Even in a country festooned with temples, Abu Simbel is something special. This is Ramses II's great temple, adorned with colossal statuary standing guard outside, and with an interior sumptuously decorated with wall paintings.

Justly famous for its megalithic proportions, Abu Simbel is also known for the incredible feat, which saw the entire temple moved from its original setting –
set to disappear under the water because of the Aswan dam - during the 1960s in a massive UNESCO operation that took four years [1].

There are many other places for sure but it was the top 5 places.

If we started to talk about places that tourists rarely visit and incredibly amazing they are many but we will talk about top 5 [2]:

- Beni Hasan;
- Gebelein;
- Elkab;
- North Saqqara;
- Medium.

Tourism is one of the most important sources of income for Egypt and accounts for around 12% of GDP. This page provides - Egypt Tourism Revenues - actual values, historical data, forecast, chart, statistics, economic calendar and news. Egypt Tourism Revenues - actual data, historical chart and calendar of releases was last updated on May of 2018.

Figure 1 shows tourism’s incoming for many countries including Egypt [3].

<table>
<thead>
<tr>
<th>Country</th>
<th>Last</th>
<th>Previous</th>
<th>Highest</th>
<th>Lowest</th>
<th>Last</th>
<th>Previous</th>
<th>Highest</th>
<th>Lowest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algeria</td>
<td>303.70</td>
<td>Dec/15</td>
<td>258</td>
<td>324</td>
<td>196</td>
<td>USD Million</td>
<td>Yearly</td>
<td></td>
</tr>
<tr>
<td>Bhutan</td>
<td>11.90</td>
<td>Nov/17</td>
<td>14.2</td>
<td>16.7</td>
<td>0.5</td>
<td>USD Million</td>
<td>Monthly</td>
<td></td>
</tr>
<tr>
<td>Cyprus</td>
<td>38400.00</td>
<td>Jan/18</td>
<td>55800</td>
<td>425700</td>
<td>283000</td>
<td>EUR Thousand</td>
<td>Monthly</td>
<td></td>
</tr>
<tr>
<td>Egypt</td>
<td>7.60</td>
<td>Dec/17</td>
<td>3.8</td>
<td>12.5</td>
<td>3.8</td>
<td>USD Million</td>
<td>Yearly</td>
<td></td>
</tr>
<tr>
<td>Fiji</td>
<td>482.50</td>
<td>Dec/17</td>
<td>531</td>
<td>531</td>
<td>272</td>
<td>FJD Million</td>
<td>Quarterly</td>
<td></td>
</tr>
<tr>
<td>France</td>
<td>3524.00</td>
<td>Mar/18</td>
<td>3196</td>
<td>6170</td>
<td>1426</td>
<td>EUR Million</td>
<td>Monthly</td>
<td></td>
</tr>
<tr>
<td>Greece</td>
<td>151.90</td>
<td>Feb/18</td>
<td>153</td>
<td>3522</td>
<td>109</td>
<td>EUR Million</td>
<td>Monthly</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>1804.44</td>
<td>Feb/18</td>
<td>1987</td>
<td>5471</td>
<td>1128</td>
<td>EUR Million</td>
<td>Monthly</td>
<td></td>
</tr>
<tr>
<td>Japan</td>
<td>1404000.00</td>
<td>Jan/18</td>
<td>19713000</td>
<td>23192000</td>
<td>921000</td>
<td>JPY Thousand</td>
<td>Monthly</td>
<td></td>
</tr>
<tr>
<td>Kuwait</td>
<td>831.00</td>
<td>Dec/16</td>
<td>931</td>
<td>931</td>
<td>266</td>
<td>USD Million</td>
<td>Yearly</td>
<td></td>
</tr>
<tr>
<td>Macau</td>
<td>18188.00</td>
<td>Dec/17</td>
<td>15931</td>
<td>18186</td>
<td>11640</td>
<td>MOP Million</td>
<td>Quarterly</td>
<td></td>
</tr>
<tr>
<td>Malaysia</td>
<td>82096.20</td>
<td>Dec/16</td>
<td>69120</td>
<td>92096</td>
<td>6590</td>
<td>MYR Million</td>
<td>Yearly</td>
<td></td>
</tr>
<tr>
<td>Mauritius</td>
<td>6615.00</td>
<td>Jan/18</td>
<td>5511</td>
<td>44376</td>
<td>3056</td>
<td>MUR Million</td>
<td>Monthly</td>
<td></td>
</tr>
<tr>
<td>Mexico</td>
<td>1913874.10</td>
<td>Feb/18</td>
<td>1940696</td>
<td>2242780</td>
<td>168919</td>
<td>USD Thousand</td>
<td>Monthly</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1. Tourism’s incoming
The main object of the website is to guide the users and tell them how and where they can enjoy their holiday. The system will have the following advantages:

- It will ask the user about his conditions
- It will help the user to create trip for him depends on those conditions
- It will keep his trips history in case he wanted to repeat it

So, it is obvious that many countries need such services to find trips for tourists on the base on their own wishes and peculiarities.

1.2. Comparative analysis of analogues

Here a comparative analysis of four analogical systems in different countries is discussed.

TripAdvisor [4] is a website and mobile application giving you a lot of options about choosing the country, hotel, etc. The main page of website is shown in figure 2. The main page of mobile application is shown in figure 3.

![Fig. 2. The main page of the website “TripAdvisor”](image)

The disadvantages of this application are the following: the application is limited with places, application just shows the user country and price of hotel,
restaurant or flight, the user can’t make trip depends on your wishes or conditions and he can’t see his trips history.

MapsMe [5] is a mobile application like any navigator shows you places, which is close to you and shows you your current place. It also can work offline and it’s good if you how to go so you don’t need navigate.

Figure 3 shows the main page of the application.

The disadvantages of this application are the following: the user cannot save his previous trips in case he needed one of them and cannot make trip depends on his wishes or conditions.

Google Map [6] is the most popular navigating mobile application but it also shows the places that is close to you. In addition, you can search about specific place or even category like (café, restaurant, etc.).

Figure 4 shows the main page of Google map mobile application.

The main disadvantage of this application is the same, which is described for the previous service.
Fig. 3. The main page of application “MapsMe”

Fig. 4. Google map main page
2GIS [7] is a mobile application that has many options but it is just for Russia. You can search offline, see all contacts and information about place, know transportation route and navigate to your place.

Figure 5 shows the application.

Fig. 5. Mobile application “2Gis”

The disadvantages of this application are the following: the application is so limited with countries, the application doesn’t work correctly sometimes, the user can’t save his previous trips in case he needed one of them and can’t make trip depends on his wishes or conditions.

Yandex Search [8]. It is a mobile application and website. It contains many stuff like “Yandex Map”, “Yandex Transport”, etc.

This application is also so popular in Russia and many Russian people using it. It shows all details about the place (photos, telephone number, etc.). It also has navigator and search.

Figure 6 shows the main page for the application.

The disadvantages of this application are the following: the application is so limited with countries, the application doesn’t work correctly sometimes, the
user can’t save his previous trips in case he needed one of them and can’t make trip depends on his wishes or conditions.

Fig. 6. The main page of “Yandex Search”

Therefore, we can say that all existing services have some disadvantages, and it would useful to develop a web-service without them.

1.3. Choice of development tools

Database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified -- and the database schema, which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures.

Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery.
Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

We use DBMS because it has:

1) multiple logical tables;
2) transactions;
3) primary keys;
4) referential integrity;
5) column constraints;
6) multithreaded multi user access to your data;
7) indexes and multiple ways to have your data access faster than with your file system;
8) stored procedures.

In addition, all of the above without having to write a single line of code.

Unless you are creating a single-user simple key-value storage. Chances are that you are quite better off WITH a DBMS rather than without.

Some DBMS examples are: MySQL, SQL Server, Oracle, dBASE, and FoxPro [9].

MySQL (and forks), PostgreSQL, MongoDB and SQLite (mobile apps matter!) are the most popular databases by far, they are generic enough to handle 100% of small use cases, which are probably 99% of the entire web and some of them, MySQL for example, is used in big data scenarios. Oracle and SQLServer are also very popular inside enterprises but they represent an insignificant part in the web scenario. There is a vast number of databases out there which are quite famous inside specific areas (or between developers) like Redis, Cassandra, Neo4j and others but they represent too little in worldwide use cases [10].

I select MS SQL because the installation is streamlined, security features are better, security features are better and its lower cost of ownership.

PHP language stands for "Hypertext Preprocessor." It is a recursive acronym, if you can understand what that means. PHP is an HTML-embedded Web scripting language. This means PHP code can be inserted into the HTML of a
Web page. When a PHP page is accessed, the PHP code is read or "parsed" by the server the page resides on.

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere [11].

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices [12].

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document [12].

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets [12].

JavaScript language often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.
Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

We usually use (Java, Python, JavaScript, CSS with HTML, C++, PHP, C, SQL, Ruby and Go) [13].

PHP (Hypertext Pre-Processor) is a server-side web programming language that is widely used for web development. However, there are many languages, which are used for web development or web programming. However, among all of them PHP is the most popular web scripting language.

My web site not so complex to use any of CMS or CMF. I should use such platforms with big website, which contains a lot of taps and menus.

In the first chapter I have described the problem statement, comparative analysis of analogues and choice of development tools [11].
2. DESIGN OF WEB-APPLICATION FOR ONLINE TRIP MAKER

2.1. Functional and non-functional requirements

These are functional requirements available for users for the future application:

The system will be able to function with three types of users: Administrator and Registered User.

The Registered User have registered earlier, he has user IDs and password. The Unregistered User can only register or login. The Registered User in any section of the system must be able to work with the following information:

1) send a new location;
2) check his trip history (three previous trips);
3) answer the questions;
4) get a new trip;
5) update information about himself;
6) see timetable of museums.

List of functions for admin:

1) confirm a new location;
2) remove confirmation from any location;
3) add/edit/delete information about contacts of transportation;
4) search and remove users;
5) remove timetable of museums.

2.2. Use case diagram

Use case diagram contains the basic functions for the user and admin (fig. 7).

The user has use cases such “answer questions about trip” its extend on “show the best variant of trips for the user” use case which is belong to system. He also has “send a new location”, “see or edit his personal information” and “see his trips history”.

19
The admin has use cases such as “delete location”, “confirm a new location” and “remove a location confirmation”, they all included on “add a new location to database” use case. He also has “search for a user”, “delete a user”, “check any user’s profile” and “add information about transportation”.

### 2.3. Database description

Database contains 6 tables named “Markers”, “Location_transport”, “Questions”, “Transport”, “Type” and “Users”. Some of them connected to each other.

Table “User” connected to table “Questions” through cascade relationship with delete and update. The relationship is one “User” to many “Questions”.

Table “Type” connected to table “Markers” through cascade relationship with delete and update. The relationship is one “Type” to many “Location”.

Tables “Location” and “Transport” connected to each other through table “Location_transport” with cascade relationship many “Location” and “Transport” to one “Location_transport”.

Figure 8 shows the DB scheme.
2.4. Development of user interfaces

The figure 9 shows the main page for the user. The user can login, register or reset his password.

The figure 10 shows the register for all users.
The figure 11 shows how to reset password.
The figure 12 shows the admin main page. It contains all the admin’s taps which is “Museum”, “Transportation”, “Location” and “Users”.

Fig. 12. The admin main page

The figure 13 shows the “Museum” tap for the admin. Here the admin will be able to check or delete all museums location.

Fig. 13. “Museum” tap for the admin
The figure 8 shows “Transportation” tap for the admin. Here the admin will be able to add, check or delete transportation companies.

![Diagram of Transportation tap for the admin]

**Fig. 13.** “Transportation” tap for the admin

The figure 14 shows “Confirm Location” tap and its sub tap from “Location” tap for the admin. Here the admin will be able to confirm, remove confirmation or delete locations.

![Diagram of Confirm Location tap from Location tap for the admin]

**Fig. 14.** “Confirm a location” tap from “Location” tap for the admin
The figure 15 shows “Add Location” tap and its sub tap from “Location” tap for the admin. Here the admin will be able to add a new location.

The figure 16 shows “Users” tap for the admin. Here the admin will be able to check profile or delete user. He can also search for any user.

The figure 17 shows the user’s main page. It contains “Trips”, “Send Location”, “Profile” and “Transportation” taps.
The figure 18 shows “Send Location” tap for the user. Here the user will be able to send a new location which is not exist in website’s database.

The figure 19 shows “Edit” tap under “Profile” tap for the user. Here the user will be able to see and edit his profile.
The “Questions” tap under “Profile” tap for the user. Here the user will be able to his previous trips (fig. 20) beside he can add a new one by pressing “New” and answering the questions again (fig. 21).
The figure 22 shows “Trip” tap for the user. Here the user will be able to see his trip and it will work automatically after answering all questions.

The figure 23 shows “Transportation” tap for the user. Here the user will be able to see all transportation companies which the admin added it.
Chapter 2 contains functional and non-functional requirements, use case diagram with the description of all use cases, DB scheme, and design of all interfaces.
3. IMPLEMENTATION OF WEB-APPLICATION FOR ONLINE TRIP MAKER

3.1. Components of the project

3.1.1. Class diagram

Class diagram shows which objects we will use in program.

Class “Usual user” shows which attributes and functions the normal user can use. Normal user can enter his name, password. He can login and add a personal information about himself. In addition, he can send location for system.

Class “Admin” shows what admin can do more than usual user. Admin has the same attributes and functions, which user has and several other functions. Admin can confirm a new location and add it. Admin also can add timetable for museums and malls on database, contact with different transportation companies.

Class “Trip” shows attributes of a trip, which we have. Among them there are location, type of a trip and duration. There are two types of a trip: family and personal trips. Duration could be for one day or more.

Class “Location” shows the location of a user: name of place, coordination, description and type of place. There are such types of places: mall, museum, park, etc.

You can see “Class diagram” in figure 24.

![Class diagram](image-url)

**Fig. 24. Class diagram**
3.1.2. Deployment diagram

It shows the relation between client, system and database.

There are two nodes: client and application.

Node “Client” contains the standard browser used by all actors (user and admin).

Node “Application” contains the system and the database and it shows the relation that system can use database. Node “Application” can be used by node “Client” (fig. 25).

Fig. 25. Deployment diagram

3.2. Some examples of code

3.2.1. Database main connection

The main SQL-query which connect all website with database; it contains “Host name”, “Database name”, “User name” and “Password” (fig. 26).

```php
<?php

// Database connection settings */
$host = 'localhost';
$user = 'root';
$pass = '';
$db = 'accounts';
$conn = new mysqli($host,$user,$pass,$db) or die($mysqli->error);
```

Fig. 26. Database main connection
3.2.2. Signup

When the user tries to create a new account the website connects to the database to save all his information in order to use it when he needs to login.

The website saves all data into “Users” table inside database. The main information which is required to finish signup “First name”, “Last name”, “Email” and “Password”.

PHP-codes starts check if there are any empty fields and if there is website will show error message “URL Empty fields.

Website will check characters of first and last name. the user can only use from “a” to “z” either capital or small letters.

Website will check email and it should be written in right way “Example@example.com” (fig. 27).

```php
<?php
if(isset($_POST['submit'])){
    include_once 'db.php';
    $first = mysqli_real_escape_string($conn, $_POST['first']);
    $last = mysqli_real_escape_string($conn, $_POST['last']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $uid = mysqli_real_escape_string($conn, $_POST['uid']);
    $pwd = mysqli_real_escape_string($conn, $_POST['pwd']);
    
    //errors
    //empty filed
    if(empty($first) || empty($last) || empty($email) || empty($uid) || empty($pwd)) {
        header("Location: ../signup.php?signup=empty");
        exit();
    }else{
        //check characters
        if(!preg_match("/^[a-zA-Z]*$/", $first))
            header("Location: ../signup.php?signup=invalid");
        exit();
    }
    //check email
    if(filter_var($email, FILTER_VALIDATE_EMAIL)) {
        header("Location: ../signup.php?signup=validemail");
        exit();
    }
}
```

Fig. 27. Sign-up form (empty field, characters and email)

Website will send a SQL-Query to database in order to check if there is such user or not. Website will check “Username” and “Email” columns and if the result bigger than “0” it means “Username” or “Email” is already exist and it will show the user that “Username” or “Email” has been already taken.
If the user has entered everything well without repetition, bad characters and real email PHP-code will start saving the user’s data into database.

PHP-code uses SQL-Query insert into database by choosing the table “users” and starts to insert user’s first name into column “First_name”, user’s last name into column “Last_name”, user’s email into column “Email”, user’s username into column “User_uid” and user’s password into column “Password”.

After register website shows to the user that he signed-up successfully (fig. 28).

```php
else{
    $sql = "SELECT * FROM users WHERE user_uid='$uid' || email='$email'";
    $result = mysqli_query($conn, $sql);
    $resultcheck = mysqli_num_row($result);

    if($resultcheck > 0){
        header("Location: ../signup.php?signup=usertaken");
        exit();
    }else{
        //hash pwd
        $hashedPwd = password_hash($pwd, PASSWORD_DEFAULT);
        //insert data to db
        $sql="INSERT INTO users (first_name, last_name, email, user_uid, password) VALUES ('$first', '$last', '$email', '$uid', '$hashedPwd');";
        $result = mysqli_query($conn, $sql);
        header("Location: ../signup.php?signup=succes");
        exit();
    }
}
```

Fig. 28. Sign-up form (check email, username and insert the data)

3.2.3. Login

When the user presses a button “Login” PHP-code starts to make connection between login form and database. Code selects from database table “Users” columns “Username” and “Password” and check if they are existing or not. At first, the program checks whether username or password are empty.

If they are, it will show empty-URL. Otherwise, the program sends SQL-query to DB and receives table with users whose username was asked. If that list is empty, then error is shown “URL Empty field”. If there are no empty fields code starts checking username itself.
Code calls username column from database and if this column bigger than 1, it means that user exist if it not website will show that user isn’t exist. If username is already existing code starts checking password by de-hashing it and comparing with user’s input.

If the user entered wrong password website will show him message with error password (fig. 29).

```php
<?php
session_start();

if(isset($_POST['submit'])){
    include 'db.php';
    $uid = mysql_real_escape_string($conn, $_POST['uid']);
    $pwd = mysql_real_escape_string($conn, $_POST['pwd']);
    //errors
    if(empty($uid) || empty($pwd)){
        header("Location: ../index.php?login=empty");
        exit();
    }else{
        $sql = "SELECT * FROM users WHERE user_uid='$uid';
        $result = mysql_query($conn, $sql);
        $resultCheck = mysql_num_rows($result);
        if($resultCheck < 1){
            header("Location: ../index.php?login=error");
            exit();
        }else{
            if($row = mysql_fetch_assoc($result)){
                //de hash the password
                $hashedPwdCheck = password_verify($pwd, $row['password']);
                if($hashedPwdCheck == false){
                    header("Location: ../index.php?login=error");
                    exit();
                }
            }
        }
    }
}
```

Fig. 29. PHP-code for checking login

If the user has entered existing username and correct password code starts checking type of the user.

Code starts call type column and check it if it equal “1” it means this user is admin and it’ll drag him to admin control panel.

If type of user equal “0” it means this user is normal user and it’ll drag him to logged user main page.

If user tried to enter any page without login website will show him an error and will drag him again to the main page to login (fig. 30).
3.3.4. Send Location

When the user points in map to send a new location the website automatically gets the latitude and longitude.

After coordinate input the user should enter what time this place starts working and what time it ends. If this place works for 24 hours the user can check 24 hours’ box. Automatically the website will write start working time as “12:00 AM” and end work time as “11:59 PM”. Then the user has to choose week working days by checking the day boxes input.

User should write place name and if he wants he can write description of the place.

User should press button “Send location” in the bottom of the form to send a new location.
Function “Save data” is responsible to send all variables to file “Form.php” in order to save them inside database (fig. 31).

```php
function saveData() {
    var name = escape(document.getElementById("name").value);
    var address = escape(document.getElementById("address").value);
    var stime = document.getElementById("stime").value;
    var etime = document.getElementById("etime").value;
    var des = escape(document.getElementById("des").value);
    var type = document.getElementById("type").value;
    var latlng = marker.getPosition();

    var url = "form.php\name=" + name + "\address=" + address + "\stime=" + stime + "\etime=" + etime + "\des=" + des + "\type=" + type + "\lat=" + latlng.lat() + "\lng=" + latlng.lng();
    downloadUrl(url, function(data, responseCode) {
        if (responseCode == 200 && data.length >= 1) {
            infowindow.close();
            document.getElementById("message").innerHTML = "Location has been sent.";
        }
    });
}
```

Fig. 31. Function “Save data”

After PHP-code starts to check all variables (fig. 32).

```php
<?php
require("db.php");

// Gets data from URL parameters
$name = @$_GET['name'];
$address = @$_GET['address'];
$stime = @$_GET['stime'];
$etime = @$_GET['etime'];
$su = @$_GET['su'];
$sno = @$_GET['sno'];
$mo = @$_GET['mo'];
$tu = @$_GET['tu'];
$we = @$_GET['we'];
$th = @$_GET['th'];
$fr = @$_GET['fr'];
$sa = @$_GET['sa'];
$des = @$_GET['des'];
$lat = @$_GET['lat'];
$lng = @$_GET['lng'];
$type = @$_GET['type'];

// Opens a connection to a MySQL server
$connection = mysql_connect("localhost", $username, $password);
if (!$connection) {
    die("Not connected : ". mysql_error());
}

// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : '. mysql_error());
}
```

Fig. 32. Check variables and connection
PHP-code starts to insert “name, address, start time, end time, all week days, latitude, longitude and description” into database inside “Location” table (fig. 33).

```php
// Insert new row with user data
$query = sprintf("INSERT INTO location " , " (id_location, id_type, name, address, start_time, end_time, su, mo, tu, we, th, fr, sa, lat, lng, des ) " , " VALUES (NULL, 'NULL', 'name', 'address', 'startime', 'endtime', 'su', 'mo', 'tu', 'we', 'th', 'fr', 'sa', 'lat', 'lng', 'des')" );
mysql_real_escape_string($name),
mysql_real_escape_string($address),
mysql_real_escape_string($startime),
mysql_real_escape_string($endtime),
mysql_real_escape_string($su),
mysql_real_escape_string($mo),
mysql_real_escape_string($tu),
mysql_real_escape_string($we),
mysql_real_escape_string($th),
mysql_real_escape_string($fr),
mysql_real_escape_string($sa),
mysql_real_escape_string($lat),
mysql_real_escape_string($lng),
mysql_real_escape_string($des));

$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
```

Fig. 33. Inserting data into “Location” table

There is a different table for type to avoid duplication and PHP-code to insert data into “Type” table (fig. 34).

```php
$query = sprintf("INSERT INTO type " , " (id_type, type ) " , " VALUES (NULL, '$type')" );
mysql_real_escape_string($type));

$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
```

Fig. 34. Insert type into “Type” table

3.3.5. Questions

The user can answer questions to get his trip unlimited time and all this answers will be saved into database inside table “questions” through function called “Add Questions” (fig. 35).
Also the last previous 3 answers which contains his trips will be shown to him through this php-code (fig. 36).

```php
<?php
session_start();

include_once 'header.php';
$conn = mysqli_connect('localhost', 'root', '', 'accounts');
if (!$conn) {
    die('Failed to connect to MySQL: ' . mysqli_connect_error());
}

$sql = 'SELECT * FROM questions limit 3';
$query = mysqli_query($conn, $sql);
if (!$query) {
    die('SQL Error: ' . mysqli_error($conn));
}?>
```

Fig. 36. The user last 3 trips

3.3.6. Making trip for the user

Depending on the users’ answers, the website starts to check some conditions, which are already written inside the code (fig. 37). These conditions are about amount of travelers, age, gender, and some others.
Then the website starts getting all information about the places (latitude, longitude, address, description, etc.) which the user can visit from database through PHP-code (fig. 38).

```php
function get_question_results($post_data) {
    $con = mysqli_connect("localhost", 'root', '', 'accounts');
    if (!$con) {
        die('Not connected : ' . mysqli_connect_error());
    }
    $conditions = array();
    if (!empty($post_data['trip_one'])) {
        $trip_one = $post_data['trip_one'];
        $conditions[] = " $trip_one ";
    }
    if (!empty($post_data['trip_two'])) {
        $trip_two = $post_data['trip_two'];
        $conditions[] = " $trip_two ";
    }
    if (!empty($post_data['trip_three'])) {
        $trip_three = $post_data['trip_three'];
        $conditions[] = " $trip_three ";
    }
    if (!empty($post_data['trip_four'])) {
        $trip_four = $post_data['trip_four'];
        $conditions[] = " $trip_four ";
    }
}
```

Fig. 37. Check conditions

3.3.6. Get all confirmed location for the user

```sql
$sqlStatement = 'SELECT markers.id, lat, lng, username, address, description, stime, etime, su, mo, tu, we, th, fr, sa, id_type, confirm as isconfirmed FROM markers WHERE confirm = 1 AND id_type IN (' . implode(' , ', $conditions) . ')

Syntax: Insert new row with place data.
$sqldata = $sqlStatement
$rows = array();
while ($r = mysqli_fetch_assoc($sqldata)) {
    $rows[] = $r;
}
indexed = array_map('array_values', $rows);
// $array = array_filter($indexed);
echo json_encode($indexed);
if (!$rows) {
    return null;
}
```

Fig. 38. Get the user trip PHP-code
When user press “Send location” website shows him all locations which is already saved in order to save the users time and avoid to enter place which is already entered before.

Function “Get_all_location” is responsible to get all location for user which is already saved inside database with confirmation from admin (fig. 39).

```php
function get_all_locations()
{
    $con = mysqli_connect("localhost", 'root', '', 'accounts');
    if (!$con) {
        die('Not connected : ' . mysqli_connect_error());
    }

    $sql_data = mysqli_query($con,"
        SELECT markers.id ,lat,lng,
        username,address,description,stime,etime,
        su,mo,tu,we,th,fr,sa,id_type,confirm as isconfirmed
        FROM markers
    ");
}
```

Fig. 39. Get all confirmed location

3.3.7. Delete and Confirm place

Only admin can delete or confirm new place. When admin opens the map all places either confirmed or not appear to him.

Confirmed places take the red color and unconfirmed places take blue one, when the admin clicks on confirmed place appears for him table with place information.

If the admin wants to remove confirmation of the place but anyway keep it in database, he can simply remove confirmation check mark and the place will not be confirmed but if he wants to completely delete from the database, he can just press button delete.

If the place already not confirmed the admin can check in box confirm and save, then it’ll be confirmed.

Functions and SQL-queries are shown in figure 40.

These function contains the handler both for updating a location and deleting a location.
3.3.8. Add transportation company

The admin can add transportation company on special database table called “Transport”. There are only 3 columns “Name”, “Website” and “Telephone”. When the admin adds a new company, PHP-code first checks if he left any empty field, just one empty field end all saving process. PHP-code also checks the existing of company by checking the “Name”, “Website” and “Telephone” and if the result more the 1 it means that this company is already exist, but if the result is 0 so inserting process starts with inserting all data inside table “Transport” (fig. 41).
3.3.9. list of users for admin

The admin can see all users except admins (fig. 42).

```php
<?php
    require 'includes/db.php';
    $userquery = "SELECT * FROM users Where type=0";
    $records = mysqli_query($sconnection, $userquery);
?>
```

Fig. 42. Get all users except admins

The admin also can see their “User name”, “First name”, “Last name”, “Email”, “Country” and “City”, he can even check their profile for more information or delete the user through this PHP-code (fig. 43).

```php
<?php
    while($row = mysqli_fetch_assoc($records)){
        $email = $row["email"]; //as ["email"]
        $first_name = $row["first_name"]; //as ["first_name"]
        $last_name = $row["last_name"]; //as ["last_name"]
        $user_uid = $row["user_uid"]; //as ["user_uid"]
        $country = $row["country"]; //as ["country"]
        $city = $row["city"]; //as ["city"]
        echo "<tr>";
        echo "<td>".$row["user_uid"]."</td>";
        echo "<td>".$first_name."</td>";
        echo "<td>".$last_name."</td>";
        echo "<td>".$email."</td>";
        echo "<td>".$country."</td>";
        echo "<td>".$city."</td>";
        echo '<td><input type="button" value="Profile" id="submit" name="submit"></tdwhereIn:1.2.$row["id"]."Delete"></a></td>";
        echo '</tr>";
    }
?>
```

Fig. 43. Users information

3.3.10. Search for a user

Admin also can search a user if he has his “User name” and he’ll get all information about this user like (“First name”, “Last name”, “User name”, etc.) in the table (fig. 44).
3.3. Interfaces of the application

On the main page an unlogged user can’t see or do anything except “Login” or “Signup” (fig. 45).

![Fig. 45. The main page](image-url)
The unlogged user can find “Login” form up to the right. It contains 2 input fields: the first one is the username and second one is the password.

Also he can reset his password if he has an account but he doesn’t remember the password or create a new account if he doesn’t have an account at all (fig. 46).

![Login form](image)

Fig. 46. Login form

The user can reset his password if he doesn’t remember it by pressing button “Forger password” (fig. 47).

![Reset password button](image)

Fig. 47. Reset password button

Reset password page will be opened for the user and it requires only email. after that the website will send the new password to the user’s email (fig. 48).

![Reset password page](image)

Fig. 48. Reset password page

If the user doesn’t have account he can use button “Signup”. It will drag him to another page, which contains signup form (fig. 49).

![Sign-up button](image)

Fig. 49. Sign-up button
Signup form contains 5 inputs: first name, last name, email, user name and password (fig. 50).

![Sign-up form]

Fig. 50. Sign-up form

After login the main page will change to show the user what the website contains (fig. 51).

![Logged in user main page]

Fig. 51. Logged in user main page

After the authorization, all taps will appear to the user (fig. 52).
Logged user can use all tabs in website. One of them is “Send location”.

Send location tap contains map with markers. These markers inform users about places which already stored in database in order to avoid a repetition. If the user wants to add a new place he can point this place on map and a new form will appear containing fields “Name”, “Address”, “Start work time”, “End work time”, “Week days”, “Type” and “Description” (fig. 53).

The user can’t leave this input without data. In case the user entered not existing latitude and longitude, it is not shown on a map as well as a website will not accept the place.
After coordinate input the user should enter what time this place starts working and what time it ends. If this place works for 24 hours the user can check 24 hours’ box. Automatically the website will write start working time as “12:00 AM” and end work time as “11:59 PM”. Then the user has to choose week working days by checking the day boxes input.

User should write place name and if he wants he can write description of the place.

User should press button “Send location” in the bottom of the form to send a new location.

When the user press “Save” button the window info will change window info with message “Waiting for admin confirmation” and the marker color will change to blue (fig. 54).

![Fig. 54. Message](image)

If the user refreshed this page this marker will disappear because it’s not confirmed yet (fig. 55).

![Fig. 55. Unconfirmed location](image)

The user can start a new trip by pointing on “Profile” menu and another menu will appear to him contain “Edit” and “Questions” (fig. 56).
After the user click on “Questions” it will open for him new page contains list of his last answers and hyperlink to answer new questions by clicking on “Add Questions” if he never answered any questions before (fig. 57).

When the user clicks on “Add Questions” the questions form will appear to him so he can answer and add new (fig. 58).
When the user press “Submit Answers” web site will show the trip for him through map and marker depends on his previous answers (fig. 59).

![Fig. 59. the user’s trip](image)

The user also can check and edit his own profile from tap “edit” (fig. 60).

![Fig. 60. Display and edit profile](image)

The user can check all transportation list from tap “Transportation” and it’ll appear to him list with all transportation companies (fig. 61).
When the admin login the website shows him his control page (fig. 62).

The admin’s main tap contains “Museum”, “Transportation”, “Location” and “User” (fig. 63).

Tap “Museum” shows to the admin all museums list with start and end work time. He also can delete any of them (fig. 64).
Tap “Transportation” shows the admin all transportation companies with “Name”, “Telephone” and “Website”. He also can edit or delete any of them (fig. 65).

<table>
<thead>
<tr>
<th>Name</th>
<th>Telephone</th>
<th>Web-site</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>nabil</td>
<td>+79320155394</td>
<td>nabil.com</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>asdasfasd</td>
<td>36675456435645</td>
<td>asdasfasdasfasaf.com</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>ahmed</td>
<td>+79320155396</td>
<td>ahmed.com</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>olga</td>
<td>25465312314</td>
<td>olga.com</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>mahmoud</td>
<td>0219221651</td>
<td>mahmoud.ru</td>
<td>Edit</td>
<td>Delete</td>
</tr>
</tbody>
</table>

Fig. 65. Transportation list for the admin

Also he can add a new company through this page (fig. 66).

The “Location” tap contains two more are “Confirm a location” and “Add a location” (fig. 67).

Fig. 66. Add transportation company

Fig. 67. “Location” tap
When the admin press “Confirm Location” the website shows him all locations with red color and the locations which is not confirmed it takes blue color (fig. 68).

![Confirm a location map](image)

**Fig. 68. Confirm a location map**

When the admin press “Add a Location” the website shows him all locations which is already confirmed and he can add any new one (fig. 69).

![Add a location for admin](image)

**Fig. 69. Add a location for admin**
The admin also can see all user in database through “user” tap. He also can delete or check any user’s profile (fig. 70).

![User’s list]

Fig. 70. User’s list

The admin can also search for any user by his username (fig. 71).

![Search for a user]

Fig. 71. Search for a user

The third chapter contains information about Components of the project, Some examples of SQL-queries and Interfaces of the application.
4. TESTING OF A WEB-APPLICATION

4.1. Methods of testing

Software Testing is the process of executing a program or application with the intent of identifying bugs. Testing as the process of validating that a piece of software meets its business and technical requirements. Testing is the primary avenue to check that the built product meets requirements adequately [14].

Whatever the methodology, you need to plan for adequate testing of your product. Testing helps you ensure that the end product works as expected, and helps avoid live defects that can cause financial, reputational and sometimes regulatory damage to your product organization.

Here are the different methods used to judge product behavior and performance.

Black box and white box testing are the two fundamental methods.

**Black box testing** - Also called functional or specification-based testing, this method focuses on output. Testers aren’t concerned with the internal mechanisms. They only check the software does what it’s supposed to. Knowledge of coding isn’t necessary, and testers work at user interface level [15].

**White box testing** - This method uses coding know-how as part of the test procedure. When a product fails, testers go as deep into the code as necessary to find the cause. The software developers do this themselves since they determine how the product should work. Structure-based and glass box testing are other names for this method [16].

**Static testing** - Testers examine the software’s code and documentation but don’t execute the program. Static tests start early in the product’s development during the verification process [16].

**Dynamic testing** - The software is executed with various inputs, and testers compare outputs with expected behavior with this method [16].

**GUI testing** tests GUI characteristics – text formatting, text boxes, buttons, lists, layout, colors, fonts, font sizes, and so on. GUI testing is time-consuming, and third-party companies often take on the task instead of developers [16].
**Functional testing** is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing usually describes what the system does.

Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system [16].

Functional testing differs from system testing in that functional testing "verifies a program by checking it against ... design document(s) or specification(s)", while system testing "validates a program by checking it against the published user or system requirements" [17].

Functional testing has many types:

- the identification of functions that the software is expected to perform
- the creation of input data based on the function's specifications
- the determination of output based on the function's specifications
- the execution of the test case
- the comparison of actual and expected outputs
- to check whether the application works as per the customer need.

Interface Testing is the testing done on application under test (AUT) which actually verifies whether the communication between two different software systems are done correctly [18].

A connection that integrates two components are called interface. This interface in a computer world could be anything like API's, web services, etc. Testing of these connecting services or interface is referred as Interface Testing [18].

Interface is actually software that consists of sets of commands, messages, and other attributes that enable communication between a device and a user [18].

Interface Testing include testing of two main segments.

1. Web server and application server interface.
2. Application server and Database server interface.
For above mentioned scenarios, the interface testing is done to
- check servers are executed properly or not;
- errors are handled properly or returns an error message for any query made by application;
- check the outcomes when connection to web server is reset.

### 4.2. Functional testing

Table 1 shows the results of functional testing of a web-application.

Table 1. Functional testing results

<table>
<thead>
<tr>
<th>No.</th>
<th>Function</th>
<th>Expected result</th>
<th>Obtained result</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>To show the main page</td>
<td>Any visitor can watch this page with the list of the main sections</td>
<td>Any visitor can only watch this page with the list of the main sections</td>
<td>The function works</td>
</tr>
<tr>
<td>2.</td>
<td>To give all users the permission to see the registration page</td>
<td>Any user can see the registration page</td>
<td>Any user can see the registration page</td>
<td>The function works</td>
</tr>
<tr>
<td>3.</td>
<td>To give all users the permission to see the login page at all sections</td>
<td>Any user can see the login page at all sections</td>
<td>Any user can see the login page at all sections</td>
<td>The function works</td>
</tr>
<tr>
<td>4.</td>
<td>To give all visitors the permission to see the page “About us”</td>
<td>Any visitor can see the page “About us”</td>
<td>Any visitor can see the page “About us”</td>
<td>The function works</td>
</tr>
<tr>
<td>5.</td>
<td>To give all visitors the permission to see the page “Contact us”</td>
<td>Any visitor can see the page “Contact us”</td>
<td>Any visitor can see the page “Contact us”</td>
<td>The function works</td>
</tr>
<tr>
<td>6.</td>
<td>To give the registered user permission to login</td>
<td>Any registered user can login</td>
<td>Any registered user can login</td>
<td>The function works</td>
</tr>
<tr>
<td>7.</td>
<td>To give the registered user permission to see send location page</td>
<td>Any registered user can see send location page</td>
<td>Any registered user can see send location page</td>
<td>The function works</td>
</tr>
<tr>
<td>8.</td>
<td>To give the registered user permission to send location</td>
<td>Any registered user can send location</td>
<td>Any registered user can send location</td>
<td>The function works</td>
</tr>
<tr>
<td>9.</td>
<td>To give the registered user permission to answer question</td>
<td>Any registered user can answer questions many time</td>
<td>Any registered user can answer questions many time</td>
<td>The function works</td>
</tr>
<tr>
<td>10.</td>
<td>To give the registered user permission to make trip</td>
<td>Any registered user can see the page to make a trip</td>
<td>Any registered user can see the page to make a trip</td>
<td>The function works</td>
</tr>
<tr>
<td>11.</td>
<td>To give the registered user permission to see the his trip history</td>
<td>Any registered user can see his trips history</td>
<td>Any registered user can see his trips history</td>
<td>The function works</td>
</tr>
<tr>
<td>12.</td>
<td>To give the registered user permission to see the profile page</td>
<td>Any registered user can see his profile page</td>
<td>Any registered user can see his profile page</td>
<td>The function works</td>
</tr>
<tr>
<td>13.</td>
<td>To give the registered user permission to edit his profile</td>
<td>Any registered user can edit his profile</td>
<td>Any registered user can edit his profile</td>
<td>The function works</td>
</tr>
<tr>
<td>No.</td>
<td>Function</td>
<td>Expected result</td>
<td>Obtained result</td>
<td>Conclusion</td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td>14.</td>
<td>Register a new user in the database of the registration page “Register”</td>
<td>The user can insert his name, national number in order to enter his e-mail, login, password and register them in database</td>
<td>The user can insert his name, national number in order to enter his e-mail, login, password and register them in database</td>
<td>The function works</td>
</tr>
<tr>
<td>15.</td>
<td>To give the admin permission to see the user list page</td>
<td>The admin can see user list page</td>
<td>The admin can see user list page</td>
<td>The function works</td>
</tr>
<tr>
<td>16.</td>
<td>To give the admin permission to search for a user</td>
<td>The admin can search for a user</td>
<td>The admin can search for a user</td>
<td>The function works</td>
</tr>
<tr>
<td>17.</td>
<td>To give the admin permission to see the user’s profile page</td>
<td>The admin can see the user’s profile page</td>
<td>The admin can see the user’s profile page</td>
<td>The function works</td>
</tr>
<tr>
<td>18.</td>
<td>To give the admin permission to delete the user</td>
<td>The admin can delete the user</td>
<td>The admin can delete the user</td>
<td>The function works</td>
</tr>
<tr>
<td>19.</td>
<td>To give the admin permission to see the new places page</td>
<td>The admin can see the new places page</td>
<td>The admin can see the new places page</td>
<td>The function works</td>
</tr>
<tr>
<td>20.</td>
<td>To give the admin permission to confirm the new places page</td>
<td>The admin can confirm the new places</td>
<td>The admin can confirm the new places</td>
<td>The function works</td>
</tr>
<tr>
<td>21.</td>
<td>To give the admin permission to see transportation timetable page</td>
<td>The admin can see transportation timetable page</td>
<td>The admin can see transportation timetable page</td>
<td>The function works</td>
</tr>
<tr>
<td>22.</td>
<td>To give the admin permission to add a new transportation timetable page</td>
<td>The admin can add a new transportation timetable</td>
<td>The admin can add a new transportation timetable</td>
<td>The function works</td>
</tr>
<tr>
<td>23.</td>
<td>To give the permission admin permission to see museum timetable page</td>
<td>The admin can see museum timetable page</td>
<td>The admin can see museum timetable page</td>
<td>The function works</td>
</tr>
</tbody>
</table>

All functions work, all tests passed.

4.3. User interface testing

I gave 8 people tasks to test the web-site and give me their feedback and the tasks was:

1) register;
2) login;
3) answer all questions;
4) find your trip;
5) send a new location;
6) check your trips history;
7) check your profile;
8) edit your profile.
9) Reset your password.

When I asked them about each task duration. The average answer duration is shown in table 2.

Table 2. Average answer duration

<table>
<thead>
<tr>
<th>No</th>
<th>Task</th>
<th>Duration</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Register</td>
<td>1 min 30 sec</td>
<td>Good</td>
</tr>
<tr>
<td>2.</td>
<td>Login</td>
<td>10 sec</td>
<td>Perfect</td>
</tr>
<tr>
<td>3.</td>
<td>Answer all questions</td>
<td>1 min</td>
<td>Good</td>
</tr>
<tr>
<td>4.</td>
<td>Find your trip</td>
<td>7 sec</td>
<td>Perfect</td>
</tr>
<tr>
<td>5.</td>
<td>Send a new location</td>
<td>1 min</td>
<td>Good</td>
</tr>
<tr>
<td>6.</td>
<td>Check your trips history</td>
<td>30 sec</td>
<td>Perfect</td>
</tr>
<tr>
<td>7.</td>
<td>Check your profile</td>
<td>20 sec</td>
<td>Perfect</td>
</tr>
<tr>
<td>8.</td>
<td>Edit your profile</td>
<td>1 min</td>
<td>Normal</td>
</tr>
<tr>
<td>9.</td>
<td>Reset your password</td>
<td>10 sec</td>
<td>Perfect</td>
</tr>
</tbody>
</table>

After user interface testing I asked the all volunteers about what the tasks and what is the most thing they didn’t like in order to improve it.

“Register” task was easy but there is no confirmation for password, “Login” task was easy and fast, “Answer all questions” task need to be shown in better place, “Find trip” was easy because it’s automatically, “Send a new location” task was easy to find but has many details, “Check your profile” task was easy and fast, “Edit profile” tasks the can’t upload avatar and “Reset password” was easy because it’s only require email.

The forth chapter contains information about methods of testing, functional testing, and user interface testing.
CONCLUSION

If anyone decided to visit any country without tourist guide for the first time he will get lost. Egypt one of the oldest country and it has a lot of tourist every year. It has a lot of manifest places which almost of tourist never seen.

The building structure in Egypt is so complicated not like other country so it’s not easy to find places in map. Cairo is one of the biggest city on over the world from area and population. In order to make the tourist enjoy his vacation without guides who might be can control his wishes we created a website the tourist can use it easy to make a trips for himself depends on his interests and wishes.

During the developing of the web application, we solved the following tasks:

1) the problem statement was identified;
2) comparative analysis of analogues;
3) choice of development tools;
4) functional and non-functional requirements;
5) development of user interfaces.

The perspectives for the developed application. Always we can think out some function, which we can implement in future for example, are the following:

1) to change the user interface of the system;
2) to make the users can share locations with each other;
3) to give user ability to upload videos and picture about the place;
4) to make all locations appeared with pictures and videos on map;
5) to navigate the user to his location not just show it to him;
6) to translate the website to many languages;
7) to make a mobile application;
8) to give user access to leave comment about the place which he visited;
9) to give user access to but transportation tickets throw the website;
10) to give the user access to check in to any hotel throw website.
REFERENCES


2. Wonderlust. [Electronic resource] URL: https://www.wanderlust.co.uk/content/5-little-known-sites-on-the-nile-you-must-visit/ (date of access: 10.04.218).


7. 2GIS. [Electronic resource] URL: https://2gis.ru/chelyabinsk (date of access: 15.03.2018).


22. Chanel “Clever Techi” on youtube. [Electronic resource] URL: https://www.youtube.com/channel/UC1WxZFhq56xs1oxXH-XveSQ (date of access: 01.03.2018).

23. Chanel “Traversy Media” on youtube. [Electronic resource] URL: https://www.youtube.com/channel/UC29ju8bIPH5as8OGnQzwJyA (date of access: 11.03.2018).