DEVELOPMENT OF ONLINE BUS TICKET BOOKING SYSTEM

GRADUATE QUALIFICATION WORK
SUSU–02.04.02.2019.308-658.GQW

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

Author,
student of the group CE-229
__________ O.A. Adeladja

Normative control
__________ O.N. Ivanova
“___”___________ 2019
TASK

of the master graduate qualification work
for the student of the group CE-229
Adelaja Oluwaseun Adebayo
in master direction 02.04.02
“Fundamental Informatics and Information Technologies”
(master program “Database Technologies”)

1. The topic (approved by the order of the rector from 25.04.2019 No. 899)
Development of online bus ticket booking system.

2. The deadline for the completion of the work: 05.06.2019.

3. The source data for the work

4. The list of the development issues
4.1. Make a comparative analysis of analogical sites.
4.2. Choose development tools for implementation.
4.3. Declare functional and non-functional requirements for the system.
4.4. Design the project.
4.5. Design DB scheme.
4.6. Implement the project.
4.7. Test the project.

5. **Issuance date of the task:** 08.02.2019.

**Supervisor**

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

**The task is taken to perform**

O.A. Adelaja
# TABLE OF CONTENTS

INTRODUCTION ................................................................. 5

1. THE ANALYSIS OF THE SUBJECT AREA ...................... 8
   1.1. The problem statement ........................................... 8
   1.2. Importance and uniqueness of system design ............... 8
   1.3. Comparative analysis of analogues ......................... 9
   1.5. Choice of development tools .................................. 21

2. DESIGN OF WEB-APPLICATION FOR ONLINE TRANSPORT BOOKING
   (E-TICKETING) SYSTEM .................................................. 26
   2.1. Functional and non-functional requirements ............... 26
   2.2. Use case diagram ................................................ 27
   2.3. Database scheme ................................................ 28
   2.4. Interface design ................................................ 30

3. IMPLEMENTATION OF THE WEB-APPLICATION ............... 32
   3.1. Component Diagram ............................................. 32
   3.2. File Structure .................................................. 33
   3.3. SQL queries and code samples ............................... 36

4. TESTING OF THE WEB-APPLICATION ............................ 44
   4.1. Methods of testing ............................................. 44
   4.2. Functional testing ............................................. 44
   4.3. Integration testing ............................................ 49
   4.4. Usability testing .............................................. 50

CONCLUSION ................................................................. 54

REFERENCE ................................................................. 56

APPENDIX ................................................................. 59
INTRODUCTION

Topicality of Research

Constant usage of road transportation majorly the bus travelling intra and inter districts is a large growing business in several countries; the manual use of bus reservation is presently consumes a lot of time by having to stay on long queue and also very strenuous. For this reason, an effective system is to be designed to make ease of the issue of bus reservation amongst indigenes within the country.

The online bus reservation system is web-based application that permits users to check the available bus tickets, purchase the bus tickets and pay the bus ticket online [1]. Online Bus Reservation System provides bus transportation system, a facility to reserved seats, cancellation of seats and different types of enquiry which need an instant and quick reservation. This system can be used by the users in performing online reservation via internet for their all business purposes. Users can use this program directly on their websites [4]. Nowadays, information system has integrated their locality as a catalyst to successful management at organization. In this project development, a computerize system is implemented to also record all information about the country’s transport booking system systematically.

The online transport booking system will be developed to make sure that the user can make necessary booking processes of the transport online with facilities provided by this system. It also helps customers to have easier access for reservation of their tickets. The use of bus traveling is a large growing business in Nigeria and other countries; hence bus reservation system deals with maintenance of records of each passenger who had reserved a seat for a journey. It also includes maintenance of information like schedule and details of each bus [22]. Also, we get to know that there are many operations, which they have to do manually.

It takes a lot of time and causes many errors. Due to this, sometimes a lot of problems occur and they were facing many disputes with customers. To solve the above problem, and further maintaining records of items, seat availability for customers, price of per seat, bill generation and other things, we are offering this
proposal of reservation system. E-ticketing could be extended to major entertainment and touristic sites and thus facilitate access to major points of interest within cities, making e-ticketing also interesting for travellers. Urban tourism is the fastest growing tourism sector in the world [15]. Public transport operators have been trying to replace paper-based tickets with electronic media, and many countries have implemented or are about to introduce e-ticketing systems. The main characteristic of e-ticketing is that tickets are sold and stored in electronic devices. However, the benefits of a comprehensive e-ticketing system for public transport operators are hard to quantify, as the main targets of e-ticketing is an improved service quality. In monetary aspects, e-ticketing could reduce administrative costs as fewer cashiers or attendants are needed, fare processing times could be reduced and a better throughput of passengers could be allowed [6].

Aim and objectives of this project

The aim of the project is to design and to develop Online Bus Ticket Booking and Rental system.

The objectives of this project are:
1) to make a comparative analysis of the analogical sites;
2) to design database and project due to the functional requirements;
3) to implement the project;
4) to perform functional, integration, and usability testing.

Practical significance of the project is considered to be the following:
• create a web-site that enables the purchase of bus ticket functions. Customers can book ticket through the online system and need not to queue to make ticket transaction at the cashier counter;
• management of seating and reservations efficiently and effectively;
• give access to customer to check available bus ticket online and also know the time departure and arrival for each bus in the system;
• generate an accurate and detailed report of sales details;
• manage various trip inter and intra states, the cost and types;
• improve working conditions and help to build a secure system.
This web-application is to be bought and used by the Ministry of Works and Transportation in Nigeria. This project will help the staffs in this ministry to improve the customer’s satisfaction and reduce the workload of traveller spending extra time in booking transport service in Nigeria.

**Structure of the thesis**

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

In the first chapter, the analysis of the subject area, as well as the problem statement, Importance and Uniqueness of this system design, Comparative analysis of analogous systems, Choice of development tools which includes the DBMS type used (SQL server management studio 2014), the Content Management System (CMS), ASP.NET Web Form and the C# programming language and the advantages of using Content Management System.

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

In chapter three, we discussed the implementation of the web-application which includes the component diagram, File Structures, we showed several fragments of ASP.NET C# codes and SQL queries for the implementation of the basic functionality of the system.

Chapter four is devoted to the testing of the application. It contains the results of functional, integration and usability testing.

The thesis has 79 pages; the list of references contains 23 resources.
1. THE ANALYSIS OF THE SUBJECT AREA

1.1. The problem statement

Prior to the development of Online Transport Booking System, transportation by road never used any system to record all information about booking, rather the use of a hand written system on paper to make book, record and retrieve all information about the transport and driver by using a manual filing system. Based on these reasons, there resulted to the inadequate maintenance system which involves the improper handling of booking forms, repetition of identical data records due to the staffs not being able to retrieve the list of documents and transaction made in past.

Manual hand storing of data on paper in the cabinets can consume too much space and making difficult for past records to be retrieved easily. The level of security for the hand written manual paper is not guaranteed because the ticket bookings stored in the filing cabinets can be freely accessible to anyone; thereby if this information gets into inappropriate hands it can be used as an allegation against the company or blackmail the organization.

1.2. Importance and uniqueness of system design

Online based bus ticket system is an important project to be implemented in Nigeria for an improvement in business strategies and services being rendered to enhance the organization’s performance. Customers can enjoy the importance because it will permit them to check, book available bus tickets and to get receipts for payment online.

Electronic tickets are safer, reliable, faster and also cheaper compared to the hand written manual paper ticket being issued by a cashier [8]. In area of profit making by any transportation company, the use of this system can increase the profit drastically because more customers are attracted online than hiring many staffs at the counter to sell bus tickets, thereby reducing the high rate of the company paying salaries to numerous transport workers [7]. Online bus tickets
system assists every transport branches or units coordinator in calculating daily, weekly and annual collection and generation of reports.

There are several online international transportation sites such as TripMeAdvisor, Busindia, MapsMe, Rentalcars, Kayak, CoachUSA, Megabus, VisitBritainShop and many others, but some of these sites have only air flight and railway means of transportation to convey travellers from one place to another. In addition to these features provided by the sites, some cities, provinces and districts within Nigeria are not provided and accessible to users when they input the name in the search drop – down platform for the source and destination points. This project will have provinces and districts also that will be accessible to travellers who want to navigate via bus to various place mainly states in Nigeria which are not provided in the international sites. It will be majorly designed for bus ticketing to cover regions which are not accessible and this will cheaper transportation mean compare to others that is strictly restricted to only the airplanes and train that are more expensive.

1.3. Comparative analysis of analogues

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

In Great Britain, there is an analogical site with the same functionality. It is named the VisitBritainShop [23]. It has on the home page the following sections: Travel and Transport; Attractions; Sightseeing passes; Trip and Tour; and other features. When the user clicks on “trip and tour” section, they can see all means of transportation such as britrail passes, helicopter and cruises, sightseeing buses and cruises, airport travel that convey passengers on a tour. We are particular about the bus tour as it is the main analogues analysis, we should consider with this project. Figure 1 shows the home page of the visitbritainshop site.

When the user clicks on the “Trips and tour” section, they will see the available means of transportation for their trips and tours, especially the bus tours
features where they can view and book buses to their various destinations. Figure 2 describes the trips and tours page with the available features.

Fig. 1. Homepage of VisitBritainTour

Fig. 2. Trips and Tours Page with the available features
When the users click on the option for the bus tours, the platform for the list of prices for each trip is displayed in figures 3–5.

![UK Trips & Tours](image)

**Fig. 3. Price for Trips of the tours**

Figure 4 shows also shows the prices for different tour within different cities available to consumer on the website platform.

![Various Prices available for Customers](image)

**Fig. 4. Various Prices available for Customers**

In figure 5, we can also see the prices available to customers for each trip and tour they choose to embark on within various cities.
Apart from users being given prices available, they can also view the categories of age ranges between the adult, children, senior, families, and groups also input the number of people who are interested in these tours and trip. Figures 6 and 7 shows the price and details available for a customer to sightseeing within the cities Manchester and Cardiff.
Figure 7 shows the booking order available for customers travelling within the city Cardiff.

Fig. 7. Booking Ticket page for Sightseeing in Cardiff

Rentalcars is another analogues site that can be compared to this project design. The site gives the user’s platform to book cars and buses for their respective journey and in additional the users can also hire these vehicles in for their respective purpose on the desired date the wish. The purpose of rental can be for a trip within cities on the basis on business or leisure. It also give the customers to choose the particular drivers age, they want to drive them on such trip and tour. Figure 8 shows the page of the rentalcars page.

Fig. 8. Rentalcars Home page
Figure 9 shows a user input the city Lazio also can select the option of the driver’s age range, with the pick-up date and drop-off date for a leisure purpose.

![Fig. 9. Booking page for User](image)

The list of prices available for the user’s booking is provided with the various cars available for the trip.

![Fig. 10. Various Cars available and their prices page](image)

In India, there is a site of the same functionality which is available for customers to book bus and cabs online. The site is named busindia [3]. The user can book bus ticket for one way, round trips and also multiple trips within cities in
When the user wants to cancel the ticket either full or partially, they can click on the cancelation on the home page, provide all the necessary details which includes the Passenger Name Record P.N.R number being generated at their booking process, email ID and their mobile number then submit. Figure 12 shows the page for the cancelation of tickets.
Users can view tickets about buses to be booked to know the various prices. This platform page is shown in figure 13.

The same procedure in which the user inputs their PNR number, email ID and mobile number is also used in viewing the bus tickets. The site also provides for the user sign in page for existing user and can create a new account page for new users. Figure 14 shows the sign-in page available for existing users.
Figure 15 shows the Create New Account page for user on the busindia web application.

![Create New Account page BusIndia](image)

**Fig. 15. Create New Account page BusIndia**

In the United State of America, there is a bus online ticket system named megabus. Consumers can book bus ticket for both one way and round trip. It also provides the platform to input the numbers of travelers for such trips and tours. The user can search with the name of the town, city, or zip-code. Users can check their buses, can also sign up as a new user or log in as an existing user. Figure 16 shows the home page of the megabus site.

![Megabus Homepage](image)

**Fig. 16. Megabus Homepage.**
Figure 17 shows the price of a user who has booked a one way ticket.

Fig.17. One way ticket Bus booking

Figure 18 shows the booking ticket for a round trip by a user.

Fig. 18. Round Trip with price page

As described above, the user can login as an existing user or sign up as a new user on the megabus web application. Figure 19-20 shows these pages.
Table 1 shows some features available in the four sites considered and how they differ in their functions with their respective features.

Table 2 shows other features different between the sites taken into consideration.
Tab. 1. Some Criteria that differs between the VisitBritain, BusIndia, MegaBus and RentalCars websites.

<table>
<thead>
<tr>
<th>No</th>
<th>Criteria</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>VisitBritainTour</td>
</tr>
<tr>
<td>1.</td>
<td>Real cost provided</td>
<td>+</td>
</tr>
<tr>
<td>2.</td>
<td>Other means of transport</td>
<td>+</td>
</tr>
<tr>
<td>3.</td>
<td>Type of trip</td>
<td>-</td>
</tr>
<tr>
<td>4.</td>
<td>Information about tickets</td>
<td>+</td>
</tr>
<tr>
<td>5.</td>
<td>Advanced settings</td>
<td>-</td>
</tr>
<tr>
<td>6.</td>
<td>Service for renting</td>
<td>-</td>
</tr>
<tr>
<td>7.</td>
<td>Service for booking tickets</td>
<td>+</td>
</tr>
<tr>
<td>8.</td>
<td>Popularity</td>
<td>5/5</td>
</tr>
<tr>
<td>9.</td>
<td>Working on the territory of Nigeria</td>
<td>-</td>
</tr>
<tr>
<td>10.</td>
<td>Ability to select a seat</td>
<td>-</td>
</tr>
</tbody>
</table>

Tab. 2. Some Features that differs between the VisitBritain, BusIndia, MegaBus and RentalCars websites.

<table>
<thead>
<tr>
<th>No.</th>
<th>Sites</th>
<th>Features that Differ</th>
</tr>
</thead>
</table>
| 1.  | VisitBritainTour| 1. The site provides the cost of children, adults, family for every trip or tour. These prices can be sorted from either the lowest to highest or vice versa, when user click on the sort by drop box located on the Bus tour page.  
2. The site provides other means of transport system but from the survey made the bus booking for ticket is more affordable, especially for children.  
3. The site has a Login and Sign up page for customers. |
| 2.  | BusIndia        | 1. This system provide the user a platform to select the type of trip either one way, round trip and multi trip for their source to destination.  
2. The site also provides the user to enable them cancel and view bus ticket after they have filled necessary details.  
3. The site has a Login and Sign up page for customers. |
<p>| 3.  | MegaBus         | 1. This site has a similar feature like the BusIndia site in the aspect of providing user to make an option to one way or round trip but the multi trip in not included in the site page. |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>It also provide the users the platform to input number of travelers but did not categorize the travelers into children or adult like the visitbritaintour site.</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>The site has a Login and Sign up page for customers.</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>RentalCars</td>
<td>1. This site provides to function the bus ticket for trip and also vehicle rental or hire by customers.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. A distinct feature about this site; it permit the user to select the age range of the driver they want for the trip booking or hire.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. The trip or rental service provided by this site is on the leisure or business basis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. The site has a Login and Sign up page for customers.</td>
</tr>
</tbody>
</table>

From the comparative analysis for these four sites made, we could deduce that some sites had some additional features; one had combined features which made it distinct over the others. The common feature that existed was the site providing a sign up and login page for customers. The rentalcar site had an additional feature for the customers to also rent vehicles.

In this work, the booking of bus ticket and rental of bus by distributors will be implemented. Some other features such as the customers being able to give feedback of the journey, the administer and distributor being able to provide the details of the vehicle like the bus ticket package for customers, the administer being able to view feedbacks, the administer also being able to know his employee such as the accountant and drivers, thereby knowing the profit details. The customers can also confirm booking package and rent bus to their various destination either within or outside the region. The project can provide the accountant and administer access to knowing the profit made for proper report of the sales as stated in the aim and objectives of this project in section 1.2.

1.5. Choice of development tools

Most online web applications, for instance the Online Bus Ticket Booking System, have a platform to manage and store information about bus ticket. Obviously important development tools such as database will be needed to serves the purpose of an archive [16]. Database is a collection of data that is usually
managed by DBMS. Database is a computerized record keeping system, which maintains a database as a collection of computerized data files [16].

Database system enables user to some perform basic operations such the “adding of new files; inserting of data into existing files; retrieving data from existing files; deleting of data from exiting files and changing which may be termed as updating data in the existing files” to the DB. Database management system (DBMS) is a powerful tool for creating and managing large amount of data efficiently and allowing it to persist over long periods of time, safely [16]. DBMS can also be referred to as a computer software application that interacts with the user, other applications, and the database to capture and analyze data. Modern DBMS popularly used today are Sybase, Oracle, PostgreSQL, MySQL, SQLite, SQL server management studio, IBM DB2 [12].

I choose to use the DBMS SQL server management studio because it is a supported database for a platform of ASP.NET (Web Form, MVC) in C1 CMS which is an Open Source CMS [12].

SQL server management studio (SSMS) is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools which work with objects and features of the server. SSMS has five components which are the Object Explorer, Template Explorer, Solution Explorer, Visual Database Tools and query and text editors (SQL server management studio) [12]. SSMS is used to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database, and SQL Data Warehouse. SSMS provides a single comprehensive utility that combines a broad group of graphical tools with a number of rich script editors to provide access to SQL Server for developers and database administrators of all skill levels.

A Content Management System (CMS) is the computer software application that supports the creation, management and modification of the content of web information pages. Most CMS typically support multiple users in a collaborative
The main reason for choosing the Content Management System (CMS) over the Content Management Framework (CMF) in the design is that most CMS can be operated single or simply integrated with other possible applications. CMS are basically installed directly on a network, the Internet, or could run locally on a personal computer. In the world today, CMS is used to create websites that do not involve more programming language high-levelled knowledge to set up, customize, and maintain the web pages. CMS typically provides the following features:

- SEO (Search Engine Optimization) – friendly URLs;
- Integrated and Online help;
- Modularity and extensibility;
- User and group functionality;
- Templates support for changing designs [13].

The advantages of CMS is that there is reduced need to code from scratch; easy to create a unified look; version control which is also known as revision or source control which the ability to revert a document to a previous revision, which is critical for allowing editors to track each other's edits, correct mistakes, and defend against vandalism and spamming [13].

C1 CMS could be regarded as a CMS that migrate its data store to Microsoft SQL server database also as the CMS without database it uses XML files for its data store. C1 CMS (formerly Composite C1 & Orckestra CMS) is a free open source .NET-based web content management system. The C1 CMS has its license for Mozilla Public License (MPL), which is a free and open source software license [10].

The next meaningful reason to choose C1 CMS is that it is a free product. As it is an Open Source Software it has the following pros [13, 10]:

software environment (application software designed to help people involved in a common task to achieve their goals).
- **Customizations** - an open source system can be built and customized to your specific needs, both in initial setup and in the future; as those needs change, so can the system;

- **Flexibility** - built to easily integrate with other technologies and systems;

- **Supported by a community, not a company** - open source developers are able to pick up and learn a system to maintain and extend it, you would not be locked into one provider for support or further customization--could also be fully supported and maintained by an in-house team;

- **Existing frameworks** - while customization is at the core of open source, so is reusability of code assets, including features and systems that can easily be dropped in to create a complete system, so you’re not starting from scratch every time;

- **You own it** - when it is built for you, you own that instance of the software, thus you do not have to pay to use it, only to maintain it (in-house or by a provider);

- **Portability** - If you end up ever wanting to move off of the platform you are currently on, exporting your data should be a breeze unlike many custom content management systems. Many times we simply export out of Movable Type or Drupal right into WordPress in one fell swoop.

As all other Open Source Software C1 CMS have the following cons [10]:

- **Upfront cost** - since they are highly customized, there is more upfront effort to get it off the ground;

- **Less "out of the box" features** - some features that come with proprietary systems might be expensive to create with open source;

- **Less predictable support** - support and maintenance only happens as needed, instead of regularly, and is generally less predictable.

I have decided to work with Active Server Pages (ASP.NET) because it is more flexible and easy to work with. ASP.NET is an Open Source web application
to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web application and web services [2].

ASP.NET Web pages, known officially as Web Forms are the main building blocks for application development in ASP.NET. There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while web sites structures allow the user to copy the files directly to the server without prior compilation. Web forms are contained in files with a ".aspx" extension; these files typically contain static (X)HTML markup or component markup. The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page [2].
2. DESIGN OF WEB-APPLICATION FOR ONLINE TRANSPORT BOOKING (E-TICKETING) SYSTEM

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 four types of users: Customers (usual users), Administrator, Distributor, and Accountant who is an employee.

Customers (usual user) will be able to login with their username and password detail if they are registered users but unregistered customers will have to sign up as new users.

The Unregistered Customer can only register. He can also read general information about the system at the main web-page.

The Registered Customer in any section of the system must be able to work with the following information:

- can book tickets for a trip through the online system with all required settings (place in a bus, date and time of departure, etc.)
- can change booking details
- can rent a bus
- see available buses for renting
- Give feedback about the trip and drivers
- Change his/her login detail such as password

Accountant can:
- generate reports of sales details;
- calculate profits for distributors and public buses

Administrator (manager) will:
- manage the routes, types of buses, bus stops;
- RUD information about booked tickets and requests for renting of cars
- See feedback

Distributers will:
- CRUD information about his transport for rental
- See his profits and loses
- System
- send receipt for renting buses or buying tickets

2.2. Use case diagram

Figure 21 shows the use case diagram for the system which gives a description of the functional requirements each actor is expected to implement. In this use case diagram, there are four actors which include the “Administer”, “Accountant”, “Customer” and “Distributer”. The use case “view feedback” which is a function executed by the administer is an extend by the “send feedback” by the customer. The use case “view profit” is also an extend of the use case “to calculate the profit” done by the accountant. The system has ten use cases.

![Use Case Diagram](image-url)

Fig. 21. Use Case Diagram
2.3. Database scheme

I developed Database scheme, which is represented in fig. 22

![Database Scheme Diagram](image)

The Database schema consists of 7 tables (fig. 22). These tables are `VehicleType`, `VehicleForTicket`, `VehicleForRentage`, `Feedback`, `OrderTicket`, `User`, and `OrderVehicle`. These tables are connected to each other with the relationship “one to many” and also some tables have the foreign and primary key connection established.

The internal structure of the tables is shown below with their data types, columns names and the null restrictions for all columns (fig. 23–29). These internal structures were created in Microsoft SQL server management studio, DBMS software.

![Internal Structure Table](image)

Fig. 22. Database Scheme

Fig. 23. Internal structure of the table “Feedback”
Fig. 24. Internal structure of the table “OrderTicket”

Fig. 25. Internal structure of table “OrderVehicle”

Fig. 26. Internal structure of table “User”
2.4. Interface design

The online booking system consists of many interfaces for four basic roles of users: customer, administrator, distributor and accountant. These interfaces are classified into 6 levels as the figure below depicts. The available interfaces for all of them are shown in fig. 30.
Fig. 30. Schema of available interfaces for various users
3. IMPLEMENTATION OF THE WEB-APPLICATION

3.1. Component Diagram

Component diagram is a special kind of UML diagram. Component diagrams are basically used in modeling physical aspects of the object-oriented systems. It shows the organization and dependencies among a set of components. It is used to model the static implementation view of a system [5]. Component diagrams commonly contain Components, Interfaces and Dependency, generalization, association, and realization relationships. It may also contain notes and constraints [21].

The component diagram for this system has two nodes: “Client: User’s Computer” and “Server” (fig. 31).

It depicts that many user’s computers can be connected to the server. The server contains Database Component, ASP.NET component and the website. The component “Database SQL Server” is connected to the website. The component website depends on ASP.NET. The component diagram of this system is shown in fig. 31.

Fig. 31. Component Diagram
3.2. File Structure

The file structure contains 34 files with the extensions “.aspx” (which has the User Interfaces and is usually rendered in HTML tags) also with its respective “.aspx.cs” written in C#, in addition there are two main master pages. These main master pages are responsible for the login web form and registration web form. The files of the web forms are located inside 4 folders which are the “accountant folder”, “customer folder”, “distributor folder” and the “admin folder”. Each file is distinctly defined in the entire folders based on the role assigned to the system for the purpose of flexibility in the coding. A separate main folder which contains a subfolder named images folder (where the images that displayed on the web pages are saved) and also a backup file to the database with extension “.bak”. The description of the aspx-files of project and their functions are shown in table 2.

Tab. 2. File Structure and functions

<table>
<thead>
<tr>
<th>No</th>
<th>File</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Login</td>
<td>To login the users of the system to their respective pages.</td>
</tr>
<tr>
<td>2.</td>
<td>CustomerHome</td>
<td>To view the web home platform available for only the customer on the system.</td>
</tr>
<tr>
<td>3.</td>
<td>CustomerOrderPackage</td>
<td>To view bus order details being inputted by customers.</td>
</tr>
<tr>
<td>4.</td>
<td>CustomerViewPackage</td>
<td>To view the table that displays all the necessary information needed by the customers to select their booking packages.</td>
</tr>
<tr>
<td>5.</td>
<td>ConformOrder</td>
<td>To confirm order details and proceed to booking confirmation also.</td>
</tr>
<tr>
<td>6.</td>
<td>RentVechicle</td>
<td>To view the web platform available for customers rental of vehicle.</td>
</tr>
<tr>
<td>7.</td>
<td>ViewRentedVechicles</td>
<td>To view records in the table of all customers who have rented vehicle.</td>
</tr>
</tbody>
</table>

Continuation of tab. 2
<table>
<thead>
<tr>
<th></th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td>CustomerViewBooking</td>
<td>To view records of the table with all booked buses by the customer.</td>
</tr>
<tr>
<td>9.</td>
<td>ViewVehicle</td>
<td>To view table for the vehicle details available for customer rentals and booking which has been provided by the administrator.</td>
</tr>
<tr>
<td>10.</td>
<td>CustomerFeedback</td>
<td>To add customer feedback, this can only be viewed in the record of the administrator platform.</td>
</tr>
<tr>
<td>11.</td>
<td>CustomerChangePassword</td>
<td>To provide a platform for the customers to modify or update their passwords.</td>
</tr>
<tr>
<td>12.</td>
<td>Customer.MasterPage</td>
<td>To create and provide shared layout and functionality to the other pages related to the customer. It is the standard content page responsible for the templates and defines the placeholders for the content, which can be overridden by content pages (the page containing the content to be displayed on the browser page).</td>
</tr>
<tr>
<td>13.</td>
<td>AdminHome</td>
<td>To view the web home platform available for only the administrator on the system.</td>
</tr>
<tr>
<td>14.</td>
<td>Admin.MasterPage</td>
<td>To create and provide shared layout and functionality to the other pages related to the administrator.</td>
</tr>
<tr>
<td>15.</td>
<td>AddPackage</td>
<td>To add record to the table for booking package. This is done by the Administrator.</td>
</tr>
<tr>
<td>16.</td>
<td>ViewPackages</td>
<td>To view table for bus booking package added by the administrator. It also allows the admin to edit and delete records.</td>
</tr>
<tr>
<td>17.</td>
<td>AdminChangePassword</td>
<td>To provide a platform for the administrator to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>modify or update their passwords.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>18. AdminViewBooking</strong></td>
<td>To view the records of the table with all bus being booked by the customers. This can be viewed by the admin also.</td>
<td></td>
</tr>
<tr>
<td><strong>19. ViewFeedback</strong></td>
<td>To view the feedbacks sent by the customers and the distributors. This is only available for the administrator.</td>
<td></td>
</tr>
<tr>
<td><strong>20. ViewUsers</strong></td>
<td>To view all records about the users (customers, distributor and the accountant) details of the system and also delete records of user. This is available for the admin only.</td>
<td></td>
</tr>
<tr>
<td><strong>21. AdminAddVehicle</strong></td>
<td>To add details (which includes the vehicle number, vehicle name, vehicle type) about the vehicles which can be rented and booked.</td>
<td></td>
</tr>
<tr>
<td><strong>22. AdminViewVehicle</strong></td>
<td>To view record of the table for the vehicle details. The administrator can edit and delete record also.</td>
<td></td>
</tr>
<tr>
<td><strong>23. AccountHome</strong></td>
<td>To view the web home platform available for only the accountant on the system.</td>
<td></td>
</tr>
<tr>
<td><strong>24. CalculateProfit</strong></td>
<td>To calculate and view the profit for every bus booking and rental made by the customer. This is available for only the accountant.</td>
<td></td>
</tr>
<tr>
<td><strong>25. ViewSalesReport</strong></td>
<td>To view a graphical picture of the sales report of the amount of the buses booked by the customers.</td>
<td></td>
</tr>
<tr>
<td><strong>26. AccountantChangePassword</strong></td>
<td>To provide a platform for the accountant to modify or update their passwords.</td>
<td></td>
</tr>
<tr>
<td><strong>27. Account.masterpage</strong></td>
<td>To create and provide shared layout and functionality to the other pages related to the</td>
<td></td>
</tr>
</tbody>
</table>
3.3. SQL queries and code samples

In this work, I have written 25 different SQL-queries which works with different tables in the database. These queries perform mainly the selection, update, delete and also inserting values to the tables written in the aspx.cs form which is ASP.NET C# framework. Fig. 32 shows the code for inserting values into the field of the table "Registrationn" and this is for the non-existing users (customer, distributor, and accountant). It has a dropdown list tagged as "dropdownlist1" which depicts the user type.
Fig. 32. Insert SQL queries for the "Registration"

Fig. 33 shows the queries that selects the values in the fields (PackageId and the Referenceno) from the table named "ConfirmPackageBooking" and group by PackageId and Referenceno to calculate the SUM of the total profit made with specific condition (where Date between "Textbox1.txt" and "Textbox2.txt"). Textbox used in the queries depict the "From-Date" and the "To-Date" respectively. The "group by" clause is a SQL command that is used to group records that have the same values and it is used in conjunction with the aggregate function to produce a summarize output from the database [20].

Fig. 33. Select SQL queries for the "CalculateProfit"

Fig. 34 shows the SQL query for updating the customer's password. In this query, the update statement is just being used in modifying the existing value in the records which is the "TextBox5.Text" and set this modified value to
"TextBox3.Text" as a new password in the table "Registrationn". The same SQL query is also used for the administrator, accountant and the distributor but the partial class name differs.

```csharp
public partial class Customer_ChangePassword : System.Web.UI.Page
{
    Class cl = new Class1();
    Customer c = new Customer();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (cl.IpLogin())
        {
            cl.UpdateRegistrationn(set_password: "" + TextBox3.Text + "" where Password="" + TextBox5.Text + "");
            if (c.IsInProfile(1))
            {
                TextBox3.Text = TextBox4.Text = TextBox5.Text = ""
                Response.Write("<script>alert('Customer Password updated Successfully!');</script>");
            }
            else
            {
                Response.Write("<script>alert('Customer Password Not Yet Updated!');</script>");
            }
        }
        else
        {
            throw new Exception();
        }
    }
}
```

**Fig. 34. Update SQL queries for "CustomerChangePassword"**

Fig. 35 shows the SQL query for updating the distributor's password but with a different partial class name (Distributor_ChangePassword).

Partial Class is just a special feature of C#. It provides a special ability to implement the functionality of a single class into multiple files and all these files are combined into a single class file when the application is compiled.

A partial class is created by using a *partial* keyword. This keyword is also useful to split the functionality of methods, interfaces, or structure into multiple files [14].

Fig. 36, fig. 37, fig. 38 and fig. 39 show the SQL queries which selects the user type for the "Login" from the tables "Admin"; "Accountant"; and "registrationn" respectively. The “if – else if”, and “else ”statement was used in the Login file structure for the four user type with dropdown list. When the admin login successfully, it redirects to the web form "Adminhome.aspx". This is also applied to the "Accounthome.aspx" for the accountant; "Customerhome.aspx" for the customer; and "DistributorHome.aspx" for the distributor.
Fig. 35. Update SQL queries for "DistributorChangePassword"

```csharp
public partial class DistributorChangePassword : System.Web.UI.Page
{
    Class c1 = new Class();
    
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    
    protected void button_Click(object sender, EventArgs e)
    {
        try
        {
            string qry = "UPDATE Registrations set Password = " + TextBox.Text + " where Password = " + TextBox2.Text + " ;";
            int i = c1 исполняет(qry);
            if (i > 0)
            {
                TextBox2.Text = TextBox.Text = TextBox2.Text = "";
                Response.Write("<script>alert('Distributor Password Updated Successfully....!')</script>");
            }
            else
            {
                Response.Write("<script>alert('Distributor Password Not Yet Updated....!')</script>");
            }
        }
        catch (Exception)
        {
            throw;
        }
    }
}
```

Fig. 36. Select SQL query Login (Admin)

```csharp
public partial class Main_login : System.Web.UI.Page
{
    Class c1 = new Class();
    
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    
    protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
    {
        if (DropBoxList.SelectedItem.Text == "Admin")
        {
            try
            {
                string qry = "select * from Admin where Username = " + TextBox1.Text + " and Password = " + TextBox2.Text + " ;";
                DataSet ds = c1.select(qry);
                if (ds.Tables[0].Rows.Count > 0)
                {
                    Response.Write("<script>alert('Admin Login Successfully....!')</script>");
                    Response.Redirect("~/Admin/Adminhome.aspx");
                }
            }
            catch (Exception)
            {
                throw;
            }
        }
    }
}
```
else if (DropDownList.SelectedValue == "Accountant")
{
    try
    {
        string qry = "select * from accountant where Employeename='" + TextBox1.Text + "' and Password='" + Textbox2.Text + ";"
        DataSet ds = cl.select(qry);
        if (ds.Tables[0].Rows.Count > 0)
        {
            Session["UserID"] = ds.Tables[0].Rows[0][0].ToString();
            Session["userRealId"] = ds.Tables[0].Rows[0][8].ToString();
            Session["username"] = TextBox1.Text;
            Response.Write("<script>alert('Accountant Login Successfully....!');</script>");
            Response.Redirect("~/Accountant/AccountHome.aspx");
        }
        else
        {
            Response.Write("<script>alert('Accountant Not Yet Login....!');</script>");
            TextBox1.Text = Textbox2.Text = "";
        }
    }
    catch (Exception)
    {
        throw;
    }
}

Fig. 37. Select SQL query Login (accountant)

else if (DropdownList.SelectedValue == "Customer")
{
    try
    {
        string qry = "select * from Registrations where Username='" + TextBox1.Text + "' and Password='" + Textbox2.Text + ";"
        DataSet ds = cl.select(qry);
        if (ds.Tables[0].Rows.Count > 0)
        {
            Session["UserID"] = ds.Tables[0].Rows[0][0].ToString();
            Session["userRealId"] = ds.Tables[0].Rows[8][0].ToString();
            Session["username"] = TextBox1.Text;
            Response.Write("<script>alert('Customer Login Successfully....!');</script>");
            Response.Redirect("~/Customer/CustomerHome.aspx");
        }
        else
        {
            Response.Write("<script>alert('Customer Not Yet Login....!');</script>");
        }
    }
    catch (Exception)
    {
        throw;
    }
}

Fig. 38. Select SQL query Login (Customer)
Fig. 39. Select SQL query Login (Distributor)

Fig. 40 shows the SQL query for selecting all columns from the table "Orderdetail" and also inserting values into the record of the table. It is shown in the figure that two strings ("B" and an integer with an increment by one) are concatenated in the "TextBox1.Text" which depicts the OrderId and it is a unique ID and changes as the administrator save the details.

Fig. 40. Select and insert SQL query to AddPackage
Fig 41, fig 42 and fig 43 shows the SQL queries which select, update and delete records respectively from the table "orderdetail". It is also responsible to view also the packages to the Administrator from the back-end (database).

Fig. 41. Select SQL queries for the ViewPackage

```csharp
public void bind()
{
    try
    {
        string qry = " select * from Orderdetail";
        DataSet ds = c1.select(qry);
        if (ds.Tables[0].Rows.Count > 0)
        {
            GridView.DataSource = ds;
            GridView.DataBind();
        }
        else
        {
            Response.Write("<script>alert('There Is No such Information ')</script>");
        }
    }
    catch (Exception)
    {
        Response.Write("<script>alert('There Is No such Information ')</script>");
    }
}
```

Fig. 42. Update SQL queries for the ViewPackage

```csharp
protected void GridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    int index = e.RowIndex;
    GridViewRow row = (GridViewRow)GridView.Rows[index];
    Label lb1 = (Label)row.FindControl("label1");
    TextBox txt1 = (TextBox)row.FindControl("textBox1");
    TextBox txt2 = (TextBox)row.FindControl("textBox2");
    TextBox txt3 = (TextBox)row.FindControl("textBox3");
    TextBox txt5 = (TextBox)row.FindControl("textBox5");
    TextBox txt6 = (TextBox)row.FindControl("textBox6");
    TextBox txt10 = (TextBox)row.FindControl("textBox10");
    TextBox txt11 = (TextBox)row.FindControl("textBox11");
    TextBox txt12 = (TextBox)row.FindControl("textBox12");


    int i = c1.update(qry);
    if (i > 0)
    {
        Response.Write("<script>alert('Updated Packages Successfully...')?></script>");
        GridView.EditIndex = -1;
        bind();
    }
    else
    {
        Response.Write("<script>alert('Packages not yet Updated...')?></script>");
    }
}
```
Fig. 43. Delete SQL queries for the ViewPackage

protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
    GridView1.EditIndex = e.NewEditIndex;
    bind();
}

protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
{
    GridView1.EditIndex = -1;
    bind();
}

protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    int index = e.RowIndex;
    GridViewRow row = (GridViewRow)GridView1.Rows[index];
    Label lbl = (Label)row.FindControl("Label19");

    string qry = "DELETE FROM OrderDetail WHERE OrderId='" + lbl.Text + "'";
    int i = e1.sqldelete(qry);
    if (i > 0)
    {
        Response.Write("<script>alert('Deletion done!')</script>\n");
        bind();
    }
    else
    {
        Response.Write("<script>alert('Not Yet Deletion!')</script>\n");
    }
}
4. TESTING OF THE WEB-APPLICATION

4.1. Methods of testing

Web-application testing is a software testing practice to test the websites or web-applications for potential bugs. It is a complete testing of web-based applications before making live (production environment). A web-based system needs to be checked completely from end-to-end before it goes live for end users. Web-application is performed to make sure that the system is functioning properly and can be accepted by real-time users [18]. Testing could help to address the issues in web application before exposed to public like the Functional issues, web application security, web services issues, integrations issues, environment issues and its ability to handle traffic is checked [19]. There are several methods for testing software [18, 19]. These methods include: functionality testing; usability testing; integration testing; compatibility testing; performance testing; crowd testing; database testing and the security testing [19]. In this session of the work, we chose to focus only on three testing types of this system which are the functional, integration and the usability.

4.2. Functional testing

Functional testing involves the testing of each component to know if they are functioning as expected or not. It is also known as the component testing [18]. This test is for all the links in web pages, database connection, forms used for getting and submitting information from the user in the web pages. The testing of the forms which is an integral part of any website is one of the functionality tests. Forms are used for receiving information from users and to interact with them. Form testing involves: (1) checking the all validation of the fields; (2) check for default values of the fields; (3) wrong inputs in the forms to the field in the forms; (4) testing all internal link; (5) optional and mandatory fields [19].
Within the functional testing I have tested all functional requirements declared in section 2.1. The results of functional testing are given in table 3.

Table 3. The test case for the system and how it responds to the function

<table>
<thead>
<tr>
<th>ID</th>
<th>Test Case</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>To show main page.</td>
<td>Any user can access this page and view the list of main sections provided on the platform.</td>
<td>User can access this page and view the list of main sections provided on the platform properly.</td>
<td>This function works effectively.</td>
</tr>
<tr>
<td>2</td>
<td>To give users the permission to view the login page and enter their details.</td>
<td>Any user can view the login page and input their details based on their user type/role.</td>
<td>Users can view the login page and input their details based on their user type/role properly.</td>
<td>This function works effectively.</td>
</tr>
<tr>
<td>3</td>
<td>To give all the user the permission to view the registration page and enter their details.</td>
<td>Any user can view the registration page.</td>
<td>Users can view the registration page and input their details properly.</td>
<td>This function works effectively.</td>
</tr>
<tr>
<td>4</td>
<td>To give the user permission to see the “About Us” page.</td>
<td>Any user can view the page “About Us”</td>
<td>Any user can view the “About Us” page properly</td>
<td>This function works effectively.</td>
</tr>
<tr>
<td>5</td>
<td>To give the user permission to see the “Contact Us” page.</td>
<td>Any user can view the page “Contact Us”</td>
<td>Any user can view the “Contact Us” page properly</td>
<td>This function works effectively.</td>
</tr>
<tr>
<td>6</td>
<td>Showing message with an error while registering a new user</td>
<td>If the user while writing his/her details fails to enter some required field. The system shows the message with the mistake.</td>
<td>If the user while writing his/her details fails to enter some required field. The system shows the message with the mistake properly.</td>
<td>This function work effectively</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>7.</strong></td>
<td>Showing message with the password detail capability while the existing user on the login page.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the users password is not so strong, it shows a message “Strength:-poor”; if the password is medium level, it shows a message “Strength:-average”; and when it is strong enough, it show the message “Strength:-Excellent”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the users password is not so strong, it shows a message “Strength:-poor”; if the password is medium level, it shows a message “Strength:-average”; and when it is strong enough, it show the message “Strength:-Excellent”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>This function works effectively.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>8.</strong></td>
<td>At the login page, if the user fails to select his user type on drop down list and try to log into the system. It prompts the user to choose the user type.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>It shows the verification message for selecting the user type “user type required”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>It shows the verification message for selecting the user type “user type required”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>It works effectively.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>9.</strong></td>
<td>At the “Add Booking Package” page, the administrator can insert necessary information and save.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The administrator can view the “Add Booking Package”, insert necessary details except the order ID which is uniquely generated from the database and cannot be altered in the page.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The administrator can view the “Add Booking Package”, insert necessary details except the order ID which is uniquely generated from the database and cannot be altered in the page.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>This function works properly.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10.</strong></td>
<td>At the page “View Booking Package” the Administrator can view the detail of the information of the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The administrator can view the table containing the booking package added by him/her.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The administrator can view the table containing the booking package added by him/her properly.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>This function works effectively.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>booking package added.</td>
<td>When the administrator click “edit”, he/she can modify the value and the record is “updated” also when the “delete” is clicked, a record is also deleted successfully.</td>
<td>When the administrator click “edit”, he/she can modify the value and the record is “updated” also when the “delete” is clicked, a record is also deleted successfully.</td>
<td>The function works properly.</td>
<td></td>
</tr>
<tr>
<td>11. At the page “View Booking Package”, the administrator can click “Edit” and “delete” section.</td>
<td>The administrator can view the bus packages that have been booked by different customers with their details.</td>
<td>The administrator can view the bus packages that have been booked by different customers with their details.</td>
<td>This function works properly.</td>
<td></td>
</tr>
<tr>
<td>12. At the page “View Booking”, the administrator can view table showing the booked bus packages.</td>
<td>The administrator can view the table showing all the feedbacks.</td>
<td>The administrator can view the table with the feedbacks given by the distributors and customers.</td>
<td>This function works properly.</td>
<td></td>
</tr>
<tr>
<td>13. At the page “View Feedback”, the administrator can view the table showing all the feedbacks.</td>
<td>The administrator can view the table showing all the user of the system (accountant, distributor, and customer) also “delete” any user.</td>
<td>The administrator can view the table showing all the user of the system (accountant, distributor, and customer) also “delete” any user.</td>
<td>The function works properly.</td>
<td></td>
</tr>
<tr>
<td>14. At the page “View All Users”, the administrator can view the table with the information of all the users.</td>
<td>All the users (admin, customers, distributor, and accountant) can update their passwords successfully.</td>
<td>All the users (admin, customers, distributor, and accountant) can update their passwords successfully.</td>
<td>The function works properly.</td>
<td></td>
</tr>
<tr>
<td>15. At the page “change password”, the users can update their passwords.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>password and it is changed in the database.</td>
<td>passwords successfully.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>16.</strong> At the “View Package” Page, the customers can view the details of the booking packages.</td>
<td>The customers can view table that allows them to “ConfirmBooking”, “view orders” and “confirmOrder” successfully.</td>
<td>The customers can view table that allows them to “ConfirmBooking”, “view orders” and “confirmOrder” successfully.</td>
<td>This function works properly.</td>
<td></td>
</tr>
<tr>
<td><strong>17.</strong> At the “Rent Vechicle” page. The customers view information need to be inputted for the rentage.</td>
<td>The customer can rent vehicles and input their information which will be saved in the database.</td>
<td>The customer can rent vehicles and input their information which will be saved in the database.</td>
<td>The function works properly.</td>
<td></td>
</tr>
<tr>
<td><strong>18.</strong> At the “View Rented Vehicle” page. The customer can view the table with the information of rented vehicles.</td>
<td>The customer can view the table with the information of rented vehicles with other details.</td>
<td>The customer can view the table with the information of rented vehicles with other details.</td>
<td>The function works properly.</td>
<td></td>
</tr>
<tr>
<td><strong>19.</strong> At the “Send Feedback” page, the users (except the admin and accountant) can provide feedbacks.</td>
<td>The customers and distributors can be able to send feedback about the services which the administrator can only view.</td>
<td>The customers and distributors can be able to send feedback about the services which the administrator can only view.</td>
<td>The function works properly.</td>
<td></td>
</tr>
<tr>
<td><strong>20.</strong> At the “Add vehicle” page available for the</td>
<td>The distributor can add vehicle for rentage and the information is stored</td>
<td>The distributor can add vehicle for rentage and the information is stored</td>
<td>The function works properly.</td>
<td></td>
</tr>
</tbody>
</table>
### 21. View Profit Page

At the “view profit” page, the accountant can view the profits made within a specified date range. The accountant can input a FromDate and also a ToDate, when the search is made. The total profit is being retrieved and calculated based on the bookings and dates stored in the database.

The function works properly.

### 22. Sale Report Details

At the “Sale Report Details”, the accountant can view the sales report of all bookings. The accountant can view the sales report of the booking package details.

The function works properly.

The screenshots of the project are represented in the appendix section.

#### 4.3. Integration testing

Integration testing refers to the level of software testing where individual units are combined as group and tested. The purpose of this level of testing is to review faults in the interaction between integrated units [9]. It is a phase in software testing where individual software modules are combined and tested as a group. It is important to conduct this test in every online web based application because it helps to evaluate the compliance of the system or component with the specified functional requirements [11]. It focuses on determining the correctness of the interfaces of the online system, once all modules have been unit tested,
integration testing has been implemented [17]. The results of the integration testing are given below.

The major result of the integration testing occurs in the connection between the project work in visual studio and the database. When the project is built in visual studio and the SQL server database is not active, a server error is being displayed in the browser due to inability to access and connect to the database. Also such an error can occur when Services menu for the local computer is not activated for the application “SQL Server 2014 Management studio”. The screenshot of such error is shown in fig. 44.

![Server Error Image](image)

Fig. 44. Sample of Integration testing showing the SQL connectivity errors

4.4. Usability testing

Usability testing in most web-based system plays an important role. This testing is basically implemented by testers to make sure all possible test cases which are aimed to the audience of the application are efficiently and effectively working [19]. It is process by which the human-computer interaction features of the project are measured, and weaknesses are pin pointed/identified for correction purposes. Usability testing includes the following [19]:

- ease of learning, which means the website should be easy for use.
• navigation, which means the information and instructions provided, should be clear enough for the user to easily navigate via the web page.
• general appearance, the main menus should be provided and consistent enough on each page.
• content checking and subjective user satisfactions.

The results of the usability testing are given below based on the tester’s assessments:

Information about tasks performed

User 1 played the role as an administrator; User 2 played the role as a customer; User 3 played the role as distributer and User 4 played the role as accountant.

Functional tasks perform by the respondents:

• Procedure 1: Users were to access the main page of the site.
  Task was to start the project via visual studio and view the main page in any browser (such as google chrome, Mozilla Firefox, opera and internet explorer). Most of them used google chrome because it was the default browser for the start page of the web based application.
• Procedure 2: Users were to access the login page and enter their details after they have registered.
  Task involves going through the authorization procedure to login into their respective platform based on the user.
• Procedure 3: Users were to make pages review and perform the functions on their respective platform/profile pages.
  Task involves: User 1 who serves as the admin able to “Add a package”, to “view the package”, “to view the list about users” and to possibly change his/her password if they feel it is necessary. User2 who serves as the customers able to “register to the system because he/she does not exist in the system database”, “View package order added by the admin”, “provide his/her feedback for the User 1 to view”, “rent available vehicles”, “Confirm Booking”, “Order Booking”, “View Rented vehicles”, “View Booking Buses”, “Send Feedbacks” and “Change
Password”. **User 3** who serves as the distributor able to “access the page for the distributor by login”, “add vehicle and view the vehicle he/she added as the user 3”, “provide his/her feedback for user 1 to be able to view” and also “changes his/her if he/she wishes”. **User 4** who serves as the accountant able to “view the profit by provided the date range he/she views in the database” and also “change password if he/she wish”.

**Respondent’s Names.**

**User 1**: Ekaterina Maximova, a bachelor student of the Chelyabinsk State University (CSU) in law department.

**User 2**: Clinton Ojukwu (a student in the language school of SUSU (SIGMA); Levis Tresor Petiho (a master student in economics and management department South Ural State University SUSU) and Abdulraheem Saltan (a master student of system programming department South Ural State University (SUSU)).

**User 3**: Fidele Benimani, a master student of biochemistry department in South Ural State University (SUSU).

**User 4**: Olga Kim, a bachelor student of South Ural State University (SUSU).

Table. 4. Rating of the Respondents Results

<table>
<thead>
<tr>
<th>Name of Respondents</th>
<th>Roles</th>
<th>Task Completed (shared and individual ideas)</th>
<th>Time Taken to performance the task</th>
<th>Mistakes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ekaterina Maximova</td>
<td>Admin</td>
<td>100%</td>
<td>10 mins</td>
<td>3</td>
</tr>
<tr>
<td>Clinton Ojukwu</td>
<td>Customer (1)</td>
<td>100%</td>
<td>12 mins</td>
<td>2</td>
</tr>
<tr>
<td>Levis Tresor</td>
<td>Customer (2)</td>
<td>100%</td>
<td>9 mins</td>
<td>2</td>
</tr>
<tr>
<td>Abdulraheem Saltan</td>
<td>Customer (3)</td>
<td>100%</td>
<td>15 mins</td>
<td>5</td>
</tr>
<tr>
<td>Fidele Benimani</td>
<td>Distributor</td>
<td>100%</td>
<td>10 mins</td>
<td>3</td>
</tr>
<tr>
<td>Olga Kim</td>
<td>Accountant</td>
<td>100%</td>
<td>8 mins</td>
<td>-</td>
</tr>
</tbody>
</table>

**Deficiencies Identification**

The study revealed the following deficiencies that occurred when the respondents who used this system to perform the test. Most of them tried to login
into their respective platforms without the registration process (that stores their
details into the database and makes them an existing user to easily login) and it
returned them page to the main page. I had to explain to them, they had to register
at first before they could access their profile. Some found it difficult to easily
understand the words on the menu page because their fluency in understanding the
English letters was on the intermediate level but explained the meanings by sharing
ideas with them on how to navigate the system. The user 1 when she tried to “add
booking package”, attempted to change the “order ID” several times manually (she
even refreshed the page continuously), until I explained to her that it is unique
generated and increases as you add a booking package. I also shared ideas with her
that the concept of the primary key implementation on the “Order ID” even though
it was not in her field of study, but she learnt something new during her testing of
the system. User 3 which is the distributor attempted to change his “User ID which
was 3014” to another ID while giving a feedback but the feedback didn’t send, I
had to explain that the “User ID could not be changed that it was uniquely
generated for each user that logs in”. Customer (3) which is also a User 2 made a
mistake on the login page because he never selected his user type on the dropdown
menu after entering both username and password, so the page didn’t open the
customer page for him to perform any function, but I shared ideas with him on how
to perform the task accurately.
CONCLUSION

In most countries, the use of road transportation is of high demand by the citizens but some problems are associated to this mean of transportation. Typical problems such as the manual booking of tickets (intra and inter trips) from the cashier at bus terminals by long queue; the strenuous and time consumption factor; the tendency of misplacing bus tickets of customer and difficulty in the retrieval of this records. These problems can be solved by making life easier for the customers through the creation of an online booking system at their convenience and comfortability; this is the main reason for developing such a system.

The system is practically of great importance because it reduces the stress of queue to make ticket for booking and rental at the cashier counter; to generate an accurate and detailed profits details; and also management of booking seats effectively.

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

1) the modern tools of web sites developing were selected and the comparative analysis of the content management system for web-development are performed;

2) the structure of the required database for online bus booking and rental system is developed;

3) the web-application was designed, implemented and tested. The testing including the functionality testing also known as component testing; the usability testing which was performed by multiple users and also the integration testing.

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

1) the system should send messages to the users to verify a confirmation link via SMS or email whenever they successfully book the ticket or rent;

2) also, to implement authentication mechanism for an electronic payment system via token encryption (e-cash), Online credit cards (PIN, digital signature), e-cheque via (PIN and digital signature). This was not possible to implement
because customer’s bank details has to be confidential and secured also it will required several registration processes to convince banks officials to implement such due to safety purpose such as the fear of hacking by intruders or fraudsters;

3) the system must show the secret question on the page of recovering password in order to help user who forgot his password to remember his secret answer;

4) the system must give several routes automatically to the customer based on the linkage with some geographical region on the map thus he/she must not be the one to input it manually by hands;

5) to develop mobile application, since it is more easier to use and everyone has a mobile phone but to everyone walks free with their notepad or laptop;

6) to improve the user interface.
REFERENCE


20. SQL GROUP BY Statement, w3schools.com, [Electronic resource] URL: https://www.w3schools.com/sql/sql_groupby.asp/ (the date of access: 01.02.2019).


APPENDIX

Fig. 1: Home Page on the Main Platform of the Website.

Fig. 2: Login Page on the Main Platform of the Website
Fig. 3: Login Platform with the Users (Admin, Accountant, Customer, Distributor)

Fig. 4: Registration Page for the non-existing Users (Customer, Distributor, Accountant)
Fig. 5: “About Us” page on the Main Platform

Fig. 6: “Contact Us” page on the Main Platform
Administrator Module

Fig. 7: Administrator Login details

Fig. 8: Administrator Page Login successfully.
Fig. 9: Administrator Add Booking Package/Order Page

Fig. 10: Administrator View Booking Bus Package.
Fig. 10a: Administrator updating a record in the field (no of days) in the View Booking Bus Page.

Fig. 10b: Administrator View Booking Bus package updated successfully.

Fig. 11: Admin About User Page with dropdown menu (View Booking, View Feedback, View all users of the system).
Fig. 12: Admin View Booked Bus page.

Fig. 13: Admin View Feedback page.

Fig 14: Admin View All User Page.
Fig. 15: Admin Change Password Page.

Customer Module

Fig. 16: Customer Login detail.

Fig. 17: Customer Page Login successfully.

Customer Bus Booking Platform
Fig. 18: View Bus Booking Package Page available for Customer.

Fig. 19: Customer Order details after the confirm booking has been clicked.

Fig. 20: Customer Confirm Booking Page after the Order has been clicked
Customer Rental Platform.

Fig. 21: Rent Vehicle Page

Fig. 22: View Rented Vehicle Page

Fig. 23: View Booked Vehicle Page by the Customer
Fig. 24: Customer Feedback page.

Fig. 25: Customer Change Password Page

Distributor Module

Fig. 26: Distributor Login detail
Fig. 27: Distributor Page Login Successfully.

Fig. 28: Add Vehicle page by the Distributor
Fig. 29: Distributor sends feedback page

Fig. 30: Distributor Change Password Page
Accountant Module

Fig. 31: Accountant Login details Page

Fig. 32: Accountant Page Login Successfully

Fig. 33: ViewProfit page available for the Accountant.
Fig. 34: Total Profit viewed by Accountant

Fig. 35: Accountant Change Password page.