WEB BASED RESERVATION INFORMATION MANAGEMENT SYSTEM AND ONLINE VEHICLE RESERVATION SYSTEM FOR UNION CABS SERVICE

Y.G. WELANDAWE

BIT registration number: R051953

Index number: 0519537

Supervisor: A. S. Kalansooriya

2009/2010

BIT

UCSC

This dissertation is submitted in partial fulfilment of the requirement of the Degree of Bachelor of Information Technology (External) of the University of Colombo School of Computing

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text.

I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for inter-library loans, and for the title and summary to be made available to outside organizations.

.... Date: 03-08-2010

Y. G. Welandawe

(Name of Candidate)

Countersigned by:

.... Date: 03-08-2010

A. S. Kalansooriya

(Name of Supervisor)

Abstract

Union Cabs service is a rapidly growing company which has been serving the customers in Kottowa area and has earned the trust of their customers through more than 10 years of quality service. At present Union cab service is the most popular cab service in the Kottawa, Maharagama, Homagama and Piliyandala areas. And also lot of vehicle owners now encourage to joining with them. Union Cab Service currently does their work manually and has no proper documentation procedure. As a result of this they are facing various difficulties. Therefore with this project an effort has been made to improve the system quality and efficiency.

The entire system is divided in to two sub systems, namely reservation information management system and online vehicle reservation system. The reservation information management system will handle the Union Cabs information like Vehicle, Drivers, Officers, reservations, and customer details and generate required reports for the management. Online service to reserve vehicles for online customers will be provided through the online vehicle reservation system, and also better information about the Union cabs service will be provided through this. As additional features SMS gateway and payment gateway have been integrated to get the advantage of new technology.

The system has been developed according to the three-tier architecture and also some Object Oriented techniques have been employed. Unified Modeling Language was used in the analysis and design stages of the project. WAMP (Windows, Apache, MySQL, PHP) and Prototype Javascript framework are the main technology selection for this system. HTML, PHP, Javascript, CSS and SQL are the programming language used for this system. UMLet was used as a tool to design UML diagrams, and RISE Editor to design database.

This dissertation comprises of all the functions that were carried out during this project in fulfilling the required objectives. This is divided in to chapters for the purpose of clearly defining and explaining the work carried out during each phase of this project. Each chapter consist of a detailed explanation of each phase, and uses clear diagrams, charts and graphs for the purpose of further explaining these things.

Acknowledgment

First I wish to give my gratitude to my supervisor, Mr. A. S. Kalansooriya for his enormous support and assistance in completing all the stages of the project successfully.

I am most grateful to the Union Cabs owner Ravishan Gamage and his staff for providing me with this opportunity and for their whole hearted support.

And also I would like to thank Onvi Debug Solution (Pvt) Ltd staff, especially Mr. Patrick, Dulitha, Achintha and Himal for the support that they gave me.

I would like to thank University of Colombo school of Computing for the valuable opportunity they have provided.

Moreover each and everybody who gave me a helping hand in building this system is remembered with gratitude for everything they have done for the success of this final outcome.

Table of Contents

Page No

Table of Figures

Table of Tables

List of Acronyms

WAMP	- Windows Apache MySQL PHP
PHP	- PHP Hypertext Preprocessor
FAQ	- Frequently Ask Questions
SL	- Sri Lanka
US	- United States
SMS	- Short Message Service
NIC	- National Identity Card
A/C	- Air Condition
km	- Kilo meters
hrs	- Hours
PDF	- Portable Document Format
RDBMS	- Relational Database Management System
PC	- Personal Computer
OO	- Object Oriented
UML	- Unified Modeling Language
HTML	- hypertext markup language
CSS	- Cascading Style Sheets
XP	- experience points
VGA	- Video Graphics Array
CS	- Adobe Creative Suite
GHz	- gigahertz
MB	- Megabytes
RAM	- Random Access Memory
GB	- Gigabytes
KB	- Kilobytes
SQL	- Structured Query Language
FPDF	- Free PDF
JPEG	- Joint Photographic Experts Group
PNG	- Portable Network Graphics
GIF	- Graphics Interchange Format
PHPlot	- PHP Plots
WWW	- World Wide Web

Chapter 1 – Introduction

1.1 Motivation

Today information technology plays an important part in the business and web application technology. There are many reasons to use web based applications, but the single most important reason is to save time and money.

Union Cabs service is one of the rapidly growing cab services and has earned the trust of their customers through more than 10 years of quality service. At present Union cab service is the most popular cab service in the Kottawa area. Therefore a lot of vehicle owners are now interested in joining them. The company provides a wide range of travel solutions and special packages to address the varied needs of their customers. They have two vehicle parking station in Kottawa and Siddamulla. They have a large number of vehicles like cars, vans, wedding cars, lorries and busses for hire.

Union Cabs service currently does their work manually. The company owns some of the vehicles that are available for hire. But when the company needs specific types of vehicles they register privately owned vehicles with them. As mentioned before a lot of vehicle owners are now interested in joining them, therefore the number of registered vehicles in the cab service rapidly changes (average 5 changes per month). Documenting those changes and managing the files are not easy. Using a web application system to manage that information will save time, reduce work complexity, and will also improve data security and data reliability.

As was mentioned before this cab service has two parking stations in Siddamulla and Kottawa. And they are hoping to establish more parking stations around the area. When a customer wants to reserve a vehicle, they have to either go to one of the two parking stations or do the reservation through telephone calls. Normally they need to communicate with other parking station to make a reservation, and they currently use telephone call for this. It takes too much time and money to make most of these reservations. These things waste the customer’s valuable time. So, it is better to use computer system to manage those information flows between parking station. By using a web application system all parking station will be able to access a centralised information base as their communication centre, rather than using phone calls and wasting their time.

In the internet we can find many different vehicle reservation web applications from different countries. But it is not easy to find local vehicle reservation systems from the internet. Now local people are interested in various kinds of online services and so it is a huge opportunity for the businesses to market their service across the country. So providing an online vehicle reservation system for local people is a good business strategy for Union Cabs service. And they can have a lot of benefits by providing an online service while providing some benefits to the customers.

Some customers wish to see the specific vehicle before they hire it. Most of the vehicles are not usually parked at the parking stations, so the only way to achieve this now is to go to the parking station when the vehicle is parked there and see it. This must also be coordinated through a number of telephone calls. By adding photographs and details of vehicles to the online system these kinds of situations can be minimised.

Some benefits for Union Cabs:

- Speed up the business process
- Easy communication
- Reach many more customers
- Provide 24 hour information service
- Get customer attraction by using modern technology
- Use of information technology is advantageous with the competitors

Some benefits for customers:

- Can get information and reserve any time, any place
- Can get information without reaching the cab service office
- Easy communication

1.2 Objectives and scope

Develop a web based reservation information management system to increase the communication, documentation and information management and availability while decreasing the work complexity. And this part will be accessed by Union Cabs officers at the parking stations. And Union cabs Boss will also be able to access the system from any place he travels.

The web based reservation information management system improves the information flow between parking stations with better centralised information storage. It means Union cabs officers, boss and customers (through the online vehicle reservation system) can access the information from a single place without wasting time to communicate with each other.

Developing clear data previews for the office users with the data manipulation facilities improves the information flows. So it will provide better data preview structure to access data and manage with secure environment.

Designing a better database system to handle all the information that is required gives the advantage of having a well designed database for future maintenance.

Providing monitoring pages for the Manager is also an objective of this project which includes generating required reports like current status of the vehicles, reservations details, employees work. This will generate PDF files which contain proper data taken from the database.

This will also provide a user management section with functionalities like create user, add permissions to user, edit, delete and etc. to improve the security of the data.

Develop an online reservation system to provide all information about the services and vehicles available to the customer. So that he/she may able to get quick information and reserve cabs according to their requirements.

Developing clear data previews for the customer with the reservation cost calculation facilities improves the information flows. So it is needed to design complex algorithms to calculate the cost and guess to preview other useful option to the customers.

In the online reservation system it provides view vehicles, view packages, make reservation, FAQ, add comment and etc. In addition this provides helpful information to the customers like maps, tour guides and so on.

Provide online payment facility to the online vehicle reservation system to pay advances through the internet. For this it will integrate online payment gateway with the online vehicle reservation system. (And also it will us a web service to convert SL Rs. to US $)

Provide SMS notification system to inform drivers and customer about the hire turns. It will do by integrating an online SMS gateway to the Union Cabs system.

Chapter 2 – Analysis

2.1 Description of current system

Union Cabs service has been using a manual system to manage all the information regarding the business process. A separate file system is used to keep the data about vehicles, drivers, packages, reservations, attendance and so on.

The vehicle file is used to manage vehicle and driver information. The vehicle licence copy, insurance copy, drivers NIC copy and two forms about vehicle owner's and driver's personal details are collected.

A log book is used to keep reservations data. And the receiving reservations are simply added to the log book in the receiving order. It is not necessary most of the time to assign vehicles when the reservations are made. But sometimes very rarely some customers request for specific vehicles to hire. In such a case the information about the vehicle is noted down in the log book where the reservation is made. So basically making a reservation sometimes becomes difficult to handle without making several calls to drivers and other parking stations.

Daily an attendance sheet is marked to be informed about which vehicles are available on that particular day. The attendance is notified by the drivers through calling the cab service office in each morning. In the attendance sheet the vehicle numbers and drivers names are simply written row by row. Normally a unique name is used for each driver and it helps to remember the driver and his vehicle easily rather than remembering the vehicle number. And these vehicles are used for quick reservations in the order of the attendance sheet each day. A separate file is kept for package details and rates.

2.2 Outline of existing similar solutions

Most of the local cab services do not use any online systems or computerised information systems for their business. They still do their business manually and only use the internet to add advertisements like in the case of [WWW1], [WWW2] and [WWW3].

There are only a very few online cab services in our country that provide the opportunity of reserving vehicles online. But they are only limited to reserving special types of vehicles and special travelling packages that are above a certain price range. This only appeal to people above a certain economic status and because of that has a limited scope and does not have an appeal to the masses of normal people who hire vehicles. These are some example for that [WWW4], [WWW5] and [WWW6].

Some similar solutions for online vehicle reservation systems can be found from other countries. Different countries provide different solutions according to the requirements of the people of that country. But these have similar characterises like gathering reservation period, date and time details, customer details and contact details. And all these reservation systems have been categorized according to the hiring options. Following are some of those examples [WWW7], [WWW8], [WWW9], [WWW10] and [WWW11].

2.3 Requirements

Understanding the detailed functionalities of a manual system is very important to develop accurate software for that specific manual system, and it is the major consideration in this section. Therefore requirement analysis was allocated with a significant amount of time with the purpose of clear identification of all functional and non-functional requirements.

Good domain knowledge is important in clarifying accurate requirements of the current manual system. Because domain knowledge always helps figure out the most suitable fact gathering technique and carry out the techniques in effective manner with saving time.

2.3.1 Fact gathering techniques

Several fact gathering techniques like interviewing, monitoring ongoing business and studying relevant system documents that are currently being used were followed. The most important factor involved in interviewing was selecting the suitable people to be interviewed. Four kinds of people which involved Union cabs owner, Union cabs officers, cab drivers and customers were identified as these suitable people. The Union cabs owner and some experienced officers were interviewed to get some detailed information about all the related business areas that the Union cabs operates and it was very helpful in identifying the business functional requirement (critical requirements). Some Union cabs officers and some drivers were interviewed to identify all the business processes and administrative functions of the current system. Monitoring the ongoing business process was used as a technique in getting a better knowledge about the business processes and the document review was helpful in identifying administrative functions.

2.3.2 Requirements regarding to vehicle

As was mentioned earlier currently Union Cabs document all the vehicle details with the driver's details. That means each vehicle has a specific driver. There are a lot of documentations in registering a vehicle with the union cabs. But those details are not discussed because in this system the main objective is to handle the reservation information. But there is vehicle information that is required to make reservations. Those are,

- Vehicle number
- Vehicle type
- Number of seats
- Driver's name
- Driver's NIC
- Driver's licence
- Driver's contact

This information must be managed properly because they are directly related to the reservations.

The main functionalities required by the system through vehicle are listed below.

- Add new vehicle to the system when a new vehicle registers with the Union cabs. The above mentioned information about each vehicle should be gathered.
- Facilitate edit detail function for already added vehicles to be flexible with the information changes of the vehicle or driver.
- Delete function to remove vehicles from the system when the vehicle is leaving the company. In the delete function it should be verified that the vehicle is not assigned for future reservations.
- Should provide a complete preview of the list of vehicle that is available in the system.
- Should provide preview vehicle group by the vehicle type.

The vehicle type data are very important because, the prices of the reservations vary with it. The following vehicle types have been defined;

- Car - Non A/C, One way A/C, Two way A/C
- Van - Non A/C, One way A/C, Two way A/C
- Bus
- Lorry

There are three different air condition facilities as Non – A/C, Oneway A/C and Twoway A/C for car and van vehicle types. Keeping those air condition characterises are also important, because the charging rates are differ with air condition level.

2.3.3 Requirements regarding to packages

Union Cabs provide reservation options for customers. These options have different vehicle types and pricing rates. Following are the details of these options;

“Short tour up and down“ is a famous option which provide cars and vans. And this option is used for hires which are less than 150 km. The cost rates differ according to travelled distance in km and waiting hours. The rates also differ according to the three different A/C facilities. Following table gives the complete description about calculating the charges according to the travelled km and waiting hours. These rates are similar for cars and vans.

Table 2.1 Short tour up and down charges

	Non A/C	One way A/C	Two way A/C	Charge for waiting hour
First 10 km	Rs. 450	Rs. 475	Rs. 500	Rs.50 per ½hr
Rate for next 90km (10km-100km)	Rs. 25	Rs. 27	Rs. 30	Rs.50 per ½hr First 1/2hr is free
Rate for next 50km (100km-150km)	Rs. 22	Rs. 24	Rs. 27	Rs.100 per 1hr First 2hr is free

“Long tour up and down” is another option which is provided for cars and vans for long distant travels. And this option is available for hires which travel more than 150 km. The first 12 hours are free for the tour and chargers only for travelled distance. But if a tour uses more than 12 hours but travels less than 300km, then it will charge every single extra hour according to the following table. The rates for extra hours differ for three different A/C facilities as explained earlier and in here there is another consideration which changes the charging rates. That is about up country trips. For up country trips the charges are higher than for normal trips. If a tour uses more than 12 hours and travels more than 300km the extra hours are not charged but the extra distance is charged according to the following table. These rates are similar for cars and vans.

Table 2.2 Long tour up and down charges

		Non A/C	One way A/C	Two way A/C	Charge per extra hour (first 12hrs are free)
Rate for next km	Non up country	Rs. 20	Rs. 22	Rs. 24	If tour<300km then Rs. 100 If tour>300km then Rs. 0
	Up country	Rs. 21	Rs. 23	Rs. 25	

The next option is “Drop” which is used for one way travel using car or van. Charges for this option also depend on travelled km and waiting hours. And also the rates change with the air condition level. Following table gives the complete description about calculating the charges according to the travelled km and waiting hours. Those rates are also same for both cars and vans.

Table 2.3 Short tour drop charges

		Non A/C	One way A/C	Two way A/C	Charge for waiting hour
First 10 km		Rs. 440	Rs. 460	Rs. 490	Rs.50 per ½hr
Rate for next 20km (10km-30km)		Rs. 44	Rs. 46	Rs. 49	Rs.50 per ½hr
Rate for next km	Non up country	Rs. 39	Rs. 41	Rs. 44	Rs.100 per 1hr
	Up country	Rs. 40	Rs. 42	Rs. 45	

Union cabs provide two special package as 50km package and 100km package. These packages are offered only for up and down tours. Following two tables gives the complete description about calculating the charges according to the extra km and extra hours.

Table 2.4 50km package (only for up and down)

	Non A/C	One way A/C	Two way A/C
price	Rs. 1500	Rs. 1700	Rs. 1850
duration	5hr	5hr	5hr
length	50km	50km	50km
Charge per extra hour	Rs. 100	Rs. 100	Rs. 100
Charge per extra km	Rs. 25	Rs. 27	Rs. 30

Table 2.5 One day package (only for up and down)

	Non A/C	One way A/C	Two way A/C
price	Rs. 2500	Rs. 2700	Rs. 3000
duration	8hr	8hr	8hr
length	100km	100km	100km
Charge per extra hour	Rs. 100	Rs. 100	Rs. 100
Charge per extra km	Rs. 22	Rs. 24	Rs. 27

Union cabs also provide special reservation packages for airport travellers. Mainly this package is divided in to two packages as airport drop package and airport up and down package. Kottawa to Katunayaka have around 55km of distance. So currently the charging rates are only calculated for Kottawa to Katunayaka and the package is fixed to 55km travelling distance. But there are additional charges for extra kilo meters and has no chargers for waiting hours. Following table gives the complete description about calculating the charges according to the extra km.

Table 2.6 Airport package chargers

		Non A/C	One way A/C	Two way A/C
Drop 55km	price	Rs. 2100	Rs. 2300	Rs. 2600
	Charge per extra km	Rs. 39	Rs. 41	Rs. 44
Up and down 110km	price	Rs. 2400	Rs. 2600	Rs. 2900
	Charge per extra km	Rs. 22	Rs. 24	Rs. 27

There is also a bus reservation option for trips. And this one also offers up and down tours. Following two tables give the complete description about calculating the charges according to the km and waiting hours.

Table 2.7 Bus chargers

First 5km	Rs. 500
Rate per km	Rs. 50
Charge per waiting hour	Rs. 100

Union cabs also have lorries for hire. For lorries they have two main packages as up and down package and drop package. Following two tables gives the complete description about calculating the charges according to the km and extra km.

Table 2.8 Lorry chargers

	First 5km	Rate per additional km
Up and down (110km)	Rs. 500/=	Rs.50/=
Drop (55km)	Rs. 500/=	Rs. 45/=

The final reservation option is "Wedding" vehicle reservation. In this reservation option customer can travel maximum 100km in 8hrs just for Rs. 8000/=. For every additional hour they charges Rs. 250/= and for every additional kilometre they charges Rs. 35/=.

Those are the current vehicle reservation options that are provided by the Union Cabs to the customers. So the main requirement of the reservation package part is to calculate the price of the reservations. So there should be forms that are needed to be filled by inserting the required data to calculate prices.

Some customers need to check the prices of reservations before making a reservation. In the manual system the Union Cabs officers calculate these prices using calculators. The new system should provide the facility of calculating prices of reservations by simply entering some form of values. By providing this facility more accurate information can be provided to the customer without struggling with the calculator. And also a detailed clear preview of charge rates of all reservation options must be provided.

2.3.4 Requirements regarding to system users

At the parking stations there are office personnel to help customers with all the information needed to make reservations. Ms. Ravi is the Manager of Union Cabs service and a busy person. So it is hard to meet him at the Union Cabs office. So the office management is carried out by some trustworthy Union cabs officers.

Therefore the new system should manage system uses and protect the system from outsiders. Usually Mr. Ravi is handles the most important processors like registering vehicles, change vehicle details, manage office users and etc. So he needs two different user categories as admins and offices. Admin type users can do all the stuff that new system provides. But there are some restricted functions for officer type users as follows;

- Office users could not add vehicles
- Office users could not edit vehicles
- Office users could not delete vehicles
- Office users could not view attendance
- Office users could not add attendance
- Office users could not delete attendance
- Office users could not generate any report
- Office users could not add users
- Office users could not edit users
- Office users could not delete users
- Office users could not view users

2.3.5 Requirements regarding to daily vehicle attendance

Union cabs have a lot of vehicles registered with their system. The daily attendance order is used to allocate vehicles to the reservations of that day. This attendance order system is used only for the normal hires, and is not used for special kinds of reservations like wedding or hires with specially selected vehicles.

All vehicle drivers have to report to the Union Cabs parking stations each day before 8.00 am and say whether they are available in that day or not. According to that Union Cabs office manager uses the attendance sheet to get the information about the availability of vehicle in that day. Therefore the new system should provide an attendance handling system. And also it should provide add, delete attendance facilities and preview daily attendance and monthly attendance.

2.3.6 Requirements regarding to reservations

The most critical requirement of this business process is reservation handling. To make a reservation the reservation information and customer information are gathered as follows;

Reservation information:

- Start date time
- End date time
- Number of seats
- Option and package
- Kilometres
- Hours/Waiting hours
- Specify vehicle (If required)
- Note

Customer information:

- Full name
- Address
- Land phone number
- Mobile phone number

NOTE: If a customer reserves more than one vehicle for their tour, it will make separate reservation records for each vehicle, because the costs (value of km) of the vehicles are not always same. So in this new system it should keep reservation records for each vehicle in this kind of situations.

As an additional improvement for the system it should provide SMS notification facility for inform drivers that they have new reservation.

2.3.7 Requirements regarding to reports

Attendance reports, vehicle lists, reservation reports are the most important reports that Union cabs are currently using. Following list gives the detailed information about these reports.

- Reports regarding to the attendance
  - Daily vehicle attendance reports
  - Monthly vehicle attendance reports
- Reports regarding to the vehicles
  - Vehicle reports categorized by vehicle type
- Reports regarding to reservations
  - Daily reservations
  - Monthly reservations

As a system improvement this system can provide the following additional helpful reports. These new reports are,

- Reservation frequency over the year
- Reservation frequency over selected month
- Compare number of reservation for with vehicle type in a given period
- Vehicle attendance frequency charts
- Package charge rates report

All Union Cabs procedures are carried out manually, so difficulties like time consumption, lack of security and difficulty in handling and managing records have to be faced. Therefore Union Cabs has the necessity of improving system. Using the information technology Union Cabs can improve and expand the business while overcoming these problems.

The cab service has two parking stations in Siddamulla and Kottawa but is hoping to establish more parking stations around the area. Now, when a customer wants to reserve a vehicle, they have to either go to one of these parking stations or do the reservation through telephone calls. The union cabs officers have to communicate with other parking station before making a reservation, and this communication is done through telephone calls and consumes too much time and money to make a reservation. So a computer system is a better option to manage those information flows between parking station. By using a web application system all parking stations will be able to access a centralised information base as their communication centre, rather than using phone call and waste time.

After the whole business process was studied it was decided to build the main system as Web based reservation information management system and online vehicle reservation system. This will help to reduce the complexity of the system and hence gives better understanding. The online reservation system is an additional improvement of the business process. Online payment system is an optional feature for the online reservation system.

By analysis the following requirements were categorized as functional and non-functional requirements of the new two systems. They are expanded in the next two sub sections.

2.4 Web based reservation information management system

2.4.1 Functional requirements

The main functionality of this system is handling the data documented by the company on each day. This system can be further divided into sub systems as follows;

- Vehicle module
- Attendance module
- Package module
- Reservation module
- System user module
- Report module
- Online reservation notification module

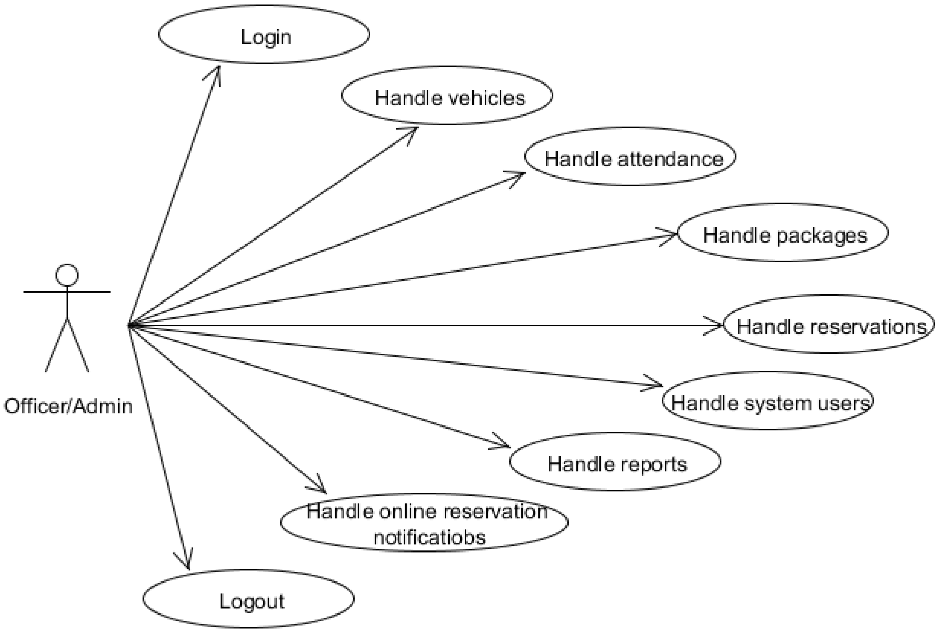


Figure 2.1 The Use Case diagram for vehicle reservation management system

In the Vehicle module there are functions for handling all the information related to vehicles and drivers and provide administration facilities. Following list gives the detailed functions that are provided in the vehicle module;

- Add new vehicle
  - In this function a form is provided for inserting the data that is related to the new vehicle and the driver who handles that vehicle.
- Edit existing vehicle
  - In this function edit facility is provide to be flexible with changes of existing vehicle's details. By using this Union cabs officers can change the status of the vehicles.
- Delete vehicle
  - By using this function it is possible to unregister a vehicle from the system. But it can only be done for unreserved vehicle.
- Preview vehicle list
  - This function can be used to list down all the vehicles that are registered with the system. And there is an option for listing only a certain vehicle type.

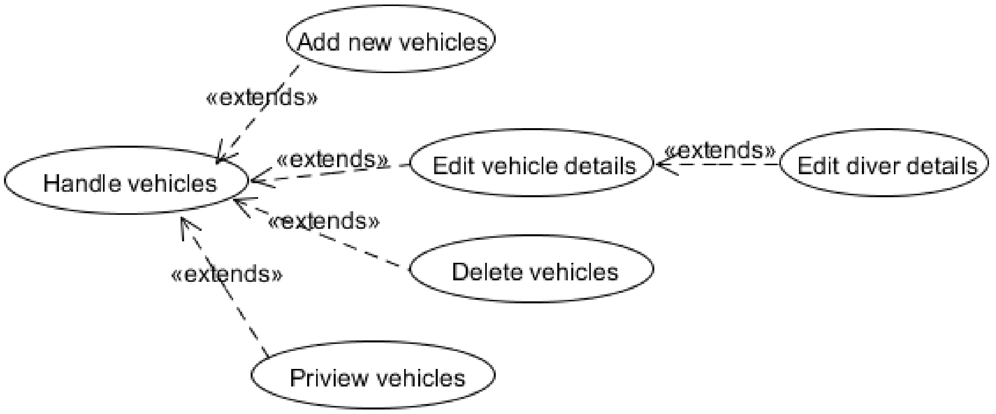


Figure 2.2 The Use Case diagram for vehicle handle module

In the attendance module there are functions for handling all information about attendance. Following list gives the detailed functions that are provided from this module;

- Add attendance
    - In this function a form is provided for inserting the data that is related to the vehicle or driver which useful for identifying vehicle's attendance.
- Delete attendance
    - Using this function an attendance data can be removed from the system.
- Preview attendance list
    - By using this function it is possible to list down all the vehicles that attend in a given date or period.

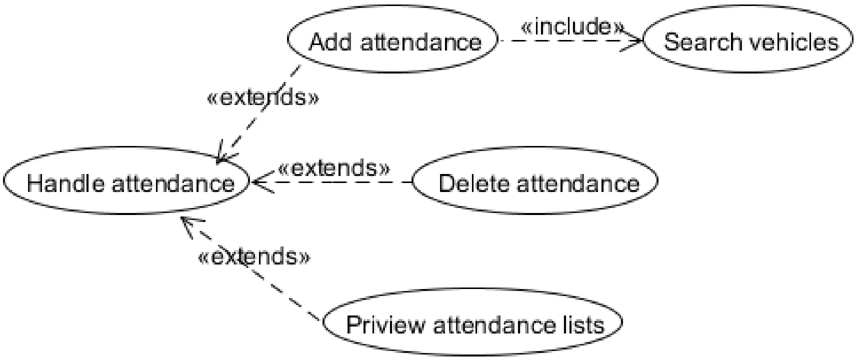


Figure 2.3 The Use Case diagram for attendance handle module

In the package module there are few functions that support calculating the charges of reservations. Following list provides the detail functionalities that are done by this module;

- Preview detailed list of packages with rates and other costs
- Charges calculates for every reservation options

The reservation module handles all the functions related to the vehicle reservations. The main function is to make reservations according to the selected reservation option. And it maintains a reservation calendar to get waiting reservations. Following list gives detail information about the function that is provided by the reservation model;

- Add reservation
    - This function provides a form to insert the reservations details and customer details.
    - It also provides optional vehicle selection facility according to the reservation option in the add reservation form.
    - In the add reservation processes it will check if the selected vehicle has any reservation time period conflicts with other reservation of that vehicle.
    - It should send a SMS to the driver if the reservation is specified to a vehicle

- Edit reservation
    - This will provide edit facility to become flexible with changing customer requirement and changing reservation dates.
    - In here it is also necessary to check if the selected vehicle has any reservation time period conflicts with other reservations of that vehicle.
    - If any changes were with the vehicle in that reservation, then it also necessary to notify the old driver and the new driver.

- Delete reservation
    - This function can be used to remove the reservation if a customer cancels the reservation.
    - And if the reservation was made for a specific vehicle then it is necessary to notify the driver about the cancelation.

- Preview reservation
    - There should be a function tom give a list of all waiting reservations.
    - There should also be a function to list all reservations in a give time period.

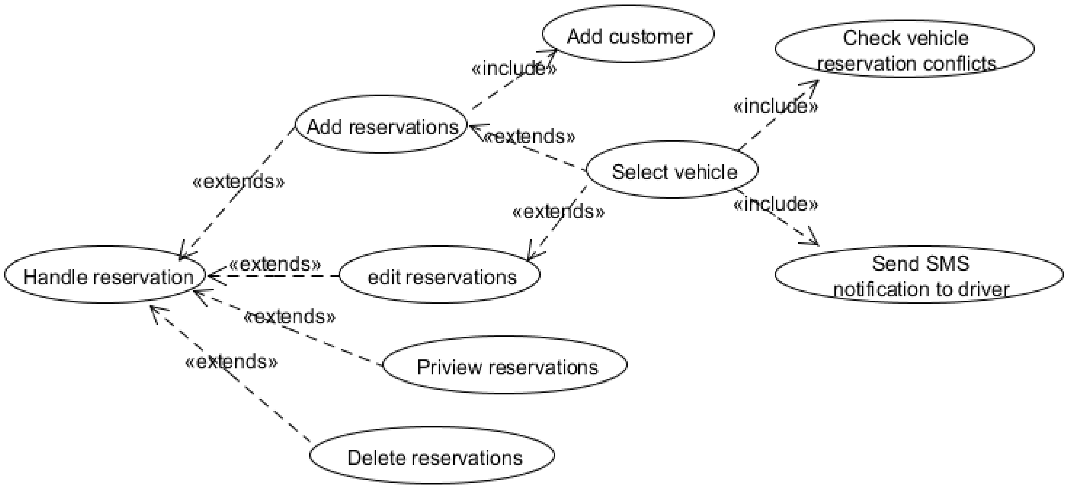


Figure 2.4 The Use Case diagram for reservation handle module

In the system user module there are functions for handling all the information related to system users and provide administration facilities. It handles the user accounts to use the system in a secure environment. Following list gives the detail functions that are provides from the system user module;

- Add new user
  - In this function it provides a form for inserting the data that is related to the new user.

- Edit existing user
  - In this function the edit facility is provided so as to be flexible with changes of existing user’s details.

- Delete user
  - This function can be used to unregister a user from the system.

- Preview user list
  - All the users that registered with the system can be listed down through this function.

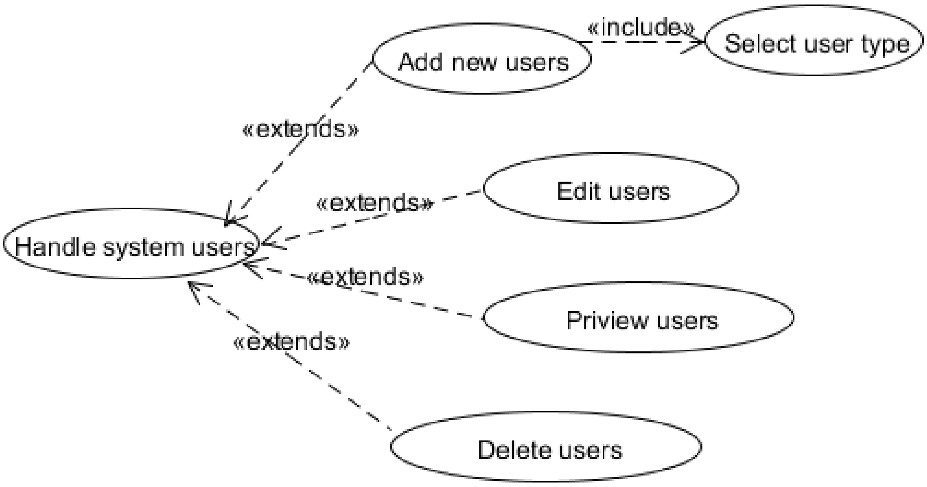


Figure 2.5 The Use Case diagram for system users handle module

When adding a user to the system, it is required to select if this new user is an admin or officer. By this module it has restricted functions for officer type users as follows;

- Office users could not add vehicles
- Office users could not edit vehicles
- Office users could not delete vehicles
- Office users could not view attendance
- Office users could not add attendance
- Office users could not delete attendance
- Office users could not generate any report
- Office users could not add users
- Office users could not edit users
- Office users could not delete users
- Office users could not view users

The report module provides required report creating facilities for the management. So user can generate reports of vehicles, drivers, packages, users separately. Users who have the report generating privileges can generate database reports of required specified data fields. Following list provides details of functional requirements from report module;

- Daily vehicle attendance report
- Monthly vehicle attendance report
- Registered vehicle list categorized by vehicle type
- Daily reservation
- Monthly reservation
- Reservation frequency over the year
- Reservation frequency over selected time period
- Compare number of reservation for with vehicle type in a given period
- Vehicle attendance frequency charts
- Package charge rates report

Online reservation notification module has to handle the online reservation requests that customers requests through online reservation system. So Union cabs office has to accept or reject the reservation through the system and inform the customer if the reservation is accepted or explain the reason for rejecting the request.

2.4.2 Non-functional requirements

- Security: System should protect from accessing unauthorized persons or systems.
- Availability: System should be available any time.
- Portability: System should be easy to move at different places for authorized persons or systems.
- Response and processing time requirement: System should have speed processing.
- Usability: System should be user friendly.

2.5 Online vehicle reservation system

2.5.1 Functional requirements

This is an additional feature to the Union Cabs. It provides information about available vehicles, packages and other helpful information to the customer and makes it easy to reserve vehicles. Therefore the customers can simply browse the site through internet and get required information.

Basically this will provide a clear view of Union Cabs vehicles and vehicle packages to the online customer. Customers can select simple reservations through this system and also can make payments through online payment gateway. But this system does not provide advanced reservations for special types of reservations e.g. wedding vehicles. Customers have to contact the Union Cabs office to make special kinds of reservations.

The online reservation requests are sent to the main reservation management system and the union cabs offices have to accept the reservation request.

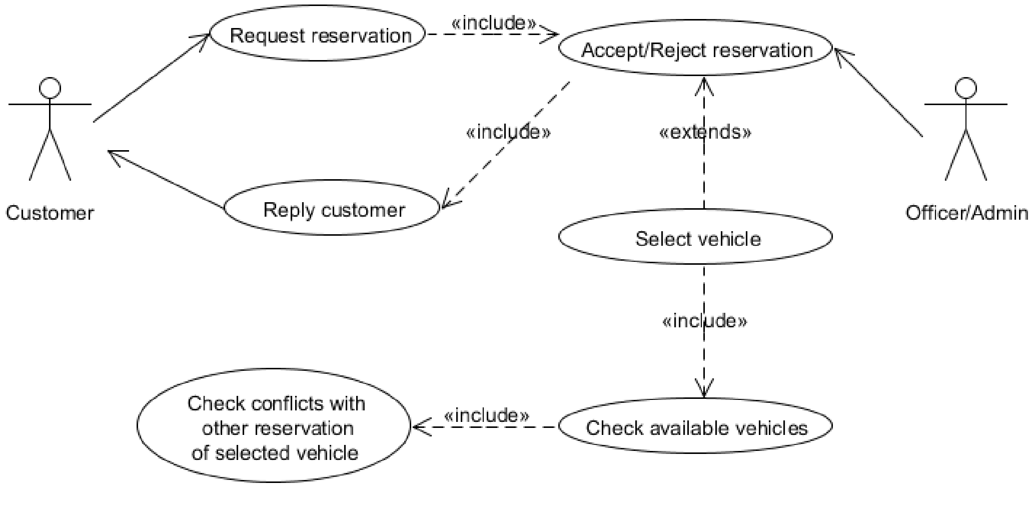


Figure 2.6 The Use Case diagram for online vehicle reservation process

Table 2.9 Description of online vehicle reservation use case diagram

Use Case	Online vehicle reservation
Actors	Customer, Office and admin
Overview	Make quick reservation over internet
Preconditions	Customer should live supported region
Flow of events	1. Customer makes vehicle reservation request 2. Office or admin receives online requests 3. Check available vehicle regarding online requests (if required) 4. Check reservation conflicts with other reservation which have already made with selected vehicle (if required) 5. Accept or reject vehicle reservation request 6. send feedback
Post conditions	Add reservation to the database, send SMS notification to the driver

2.5.2 Non-functional requirements

- Security: System information should protect from accessing from outsiders.
- Availability: System should be available any time.
- Portability: System should be easy to access from any place.
- Response time requirement: System should have speed processing.
- Usability: System should be user friendly.

Before starting to design this system some online systems were studied that are having a similar process as this. Some complex abroad systems were studied, and it was found that they have a little bit different business process than this. But some knowledge about the input information that is required from the online customers was identified. But it was impossible to find a similar dynamic system. Most similar systems give basic information to online customers and require to be contacted through telephone to make reservations. Some of studied online systems;
[WWW6], [WWW7], [WWW8], [WWW9], [WWW10] and [WWW11].

Chapter 3 – Design

3.1 Alternate solution evaluation

Before selecting software architecture for the new Union Cabs system there were some important things that needed to be considered. That is, there were important requirements like the following;

- Connecting two vehicle stations to a single system to improve communication
- The Union cabs data should be shared among those two vehicle stations
- System should provide the required data to the online customers
- Should provide a facility to the Union cabs owner to log in to the system from anywhere he wants to works

Therefore to complete these requirements this system should have a centralized database for sharing information with two vehicle stations, owner and online customers. Therefore stand-alone software development is not suitable for this system.

Since this system has a shared database requirement it is needed to consider about centralized and distributed database systems. Distributed database systems are most useful for large size organizations which use large amounts of data. And also building distributed database systems are expensive. But Union Cabs service is a normal size company and distributed database systems are not suitable for them. Therefore it is better to select a centralized database structure for their system.

There are a lot of software development structures for developing systems which have centralized database systems. Since this system is to be connected with the World Wide Web as an online vehicle reservation system, it is better to develop it as a web based application. Because it is easy to connect customers and other vehicle stations through internet rather than using other networked systems. And it is also a cost effective.

3.2 Selected solution description/justification

Since the selected solution is a web based system it needed software structure which is suitable for web application systems. Three-tire-architecture is the most famous architecture which is used in web based software engineering.

3.2.1 Three-tier-architecture

Three-tier is a client–server architecture in which the user interface, functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. The three-tier model is software architecture and a software design pattern.

Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the user interface code.
Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture").

Three-tier architecture has the following three tiers

Presentation tier:
This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

Application tier (business logic, logic tier, data access tier, or middle tier):
The logic tier is pulled out from the presentation tier and, as its own layer; it controls an application's functionality by performing detailed processing.

Data tier:
This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

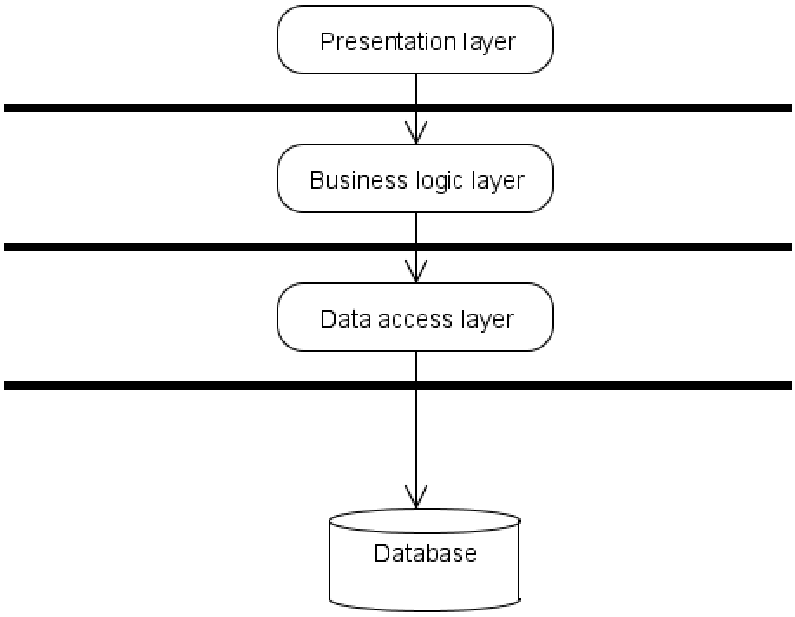


Figure 3.1 Three-tier architecture

The user interfaces are managed in the presentation layer. The business processes is managed at the business logic layer by using functions. The database is handled by the data access layer. This is the main high level structure that is provided as the data flow.

3.2.2 Programming paradigm

Procedural design and object oriented design are the main two design methods that can be used to computer programming. The object oriented design was chosen because there are many advantages this system can obtain from it. Some of the advantages are given below;

- Simplicity: software objects model real world objects, so the complexity is reduced and the program structure is very clear.
- Modularity: each object forms a separate entity whose internal workings are decoupled from other parts of the system.
- Modifiability: it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods.
- Extensibility: adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones.
- Maintainability: objects can be maintained separately, making locating and fixing problems easier.
- Re-usability: objects can be reused in different programs.

To design this system by object oriented design some designing patterns were used to improve the code. These are the design patterns that were used in the design phase;

Singleton pattern:
Some application resources are exclusive in that there is one and only one of these types of resource. For example, the connection to a database through the database handle is exclusive. You want to share the database handle in an application because it's an overhead to keep opening and closing connections, particularly during a single page fetch. The singleton pattern covers this need. An object is a singleton if the application can include one and only one of that objects at a time.

To develop the system’s Object Oriented Design diagrams UMLet tool which is free open-source software way which anyone can download from [WWW12], was used and the RISE Editor was used to design the database which is also a free software way anyone can download from [WWW13].

3.2.3 System structure

According to the requirements the whole Union Cabs system divides to two main systems as Union Cabs Vehicle Reservation Management System and Union Cabs Online Vehicle Reservation System. Each system has sub systems which full fill different functional requirements as mentioned in the requirement analysis chapter. Following diagram shows the main subsystem structure;

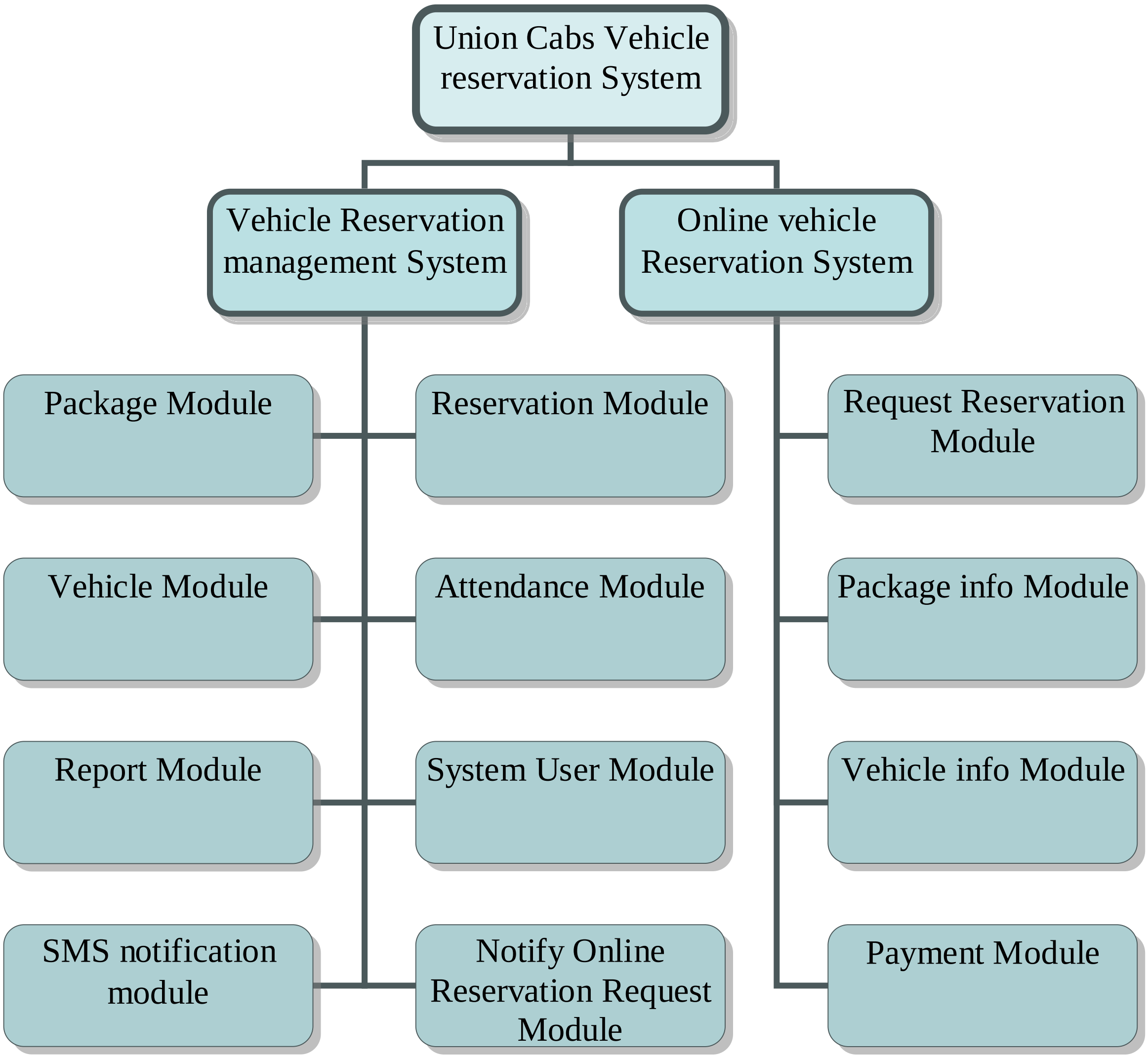

Union Cabs Vehicle reservation System
Vehicle Reservation management System
Online vehicle Reservation System
Package Module
Reservation Module
Request Reservation Module
Vehicle Module
Attendance Module
Package info Module
Report Module
System User Module
Vehicle info Module
SMS notification module
Notify Online Reservation Request Module
Payment Module

Figure 3.2 Systems and subsystems structure

Each subsystem has used three-tier-architecture design and have functions that full filled the system requirements that were mentioned in the analysis chapter. All subsystems use the system database. Therefore some object features were used to improve the system by using the object oriented advantages. The Singleton design pattern was used in the database connection to have reliable database connection for the system.

3.2.4 System module design and object oriented designing

Following diagram gives the overall design of the information management system;

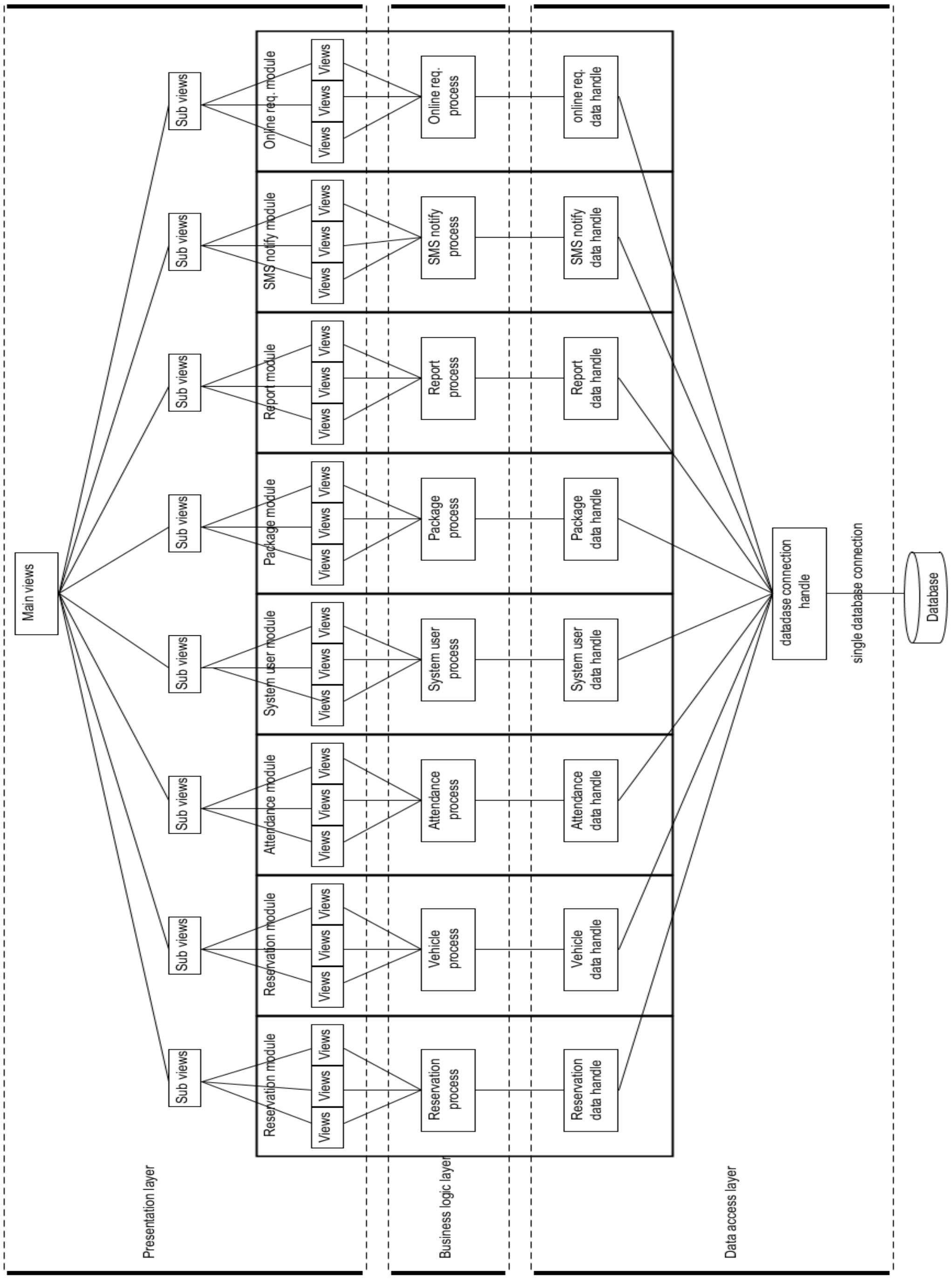


Figure 3.3 Overall system design of the Information Management system

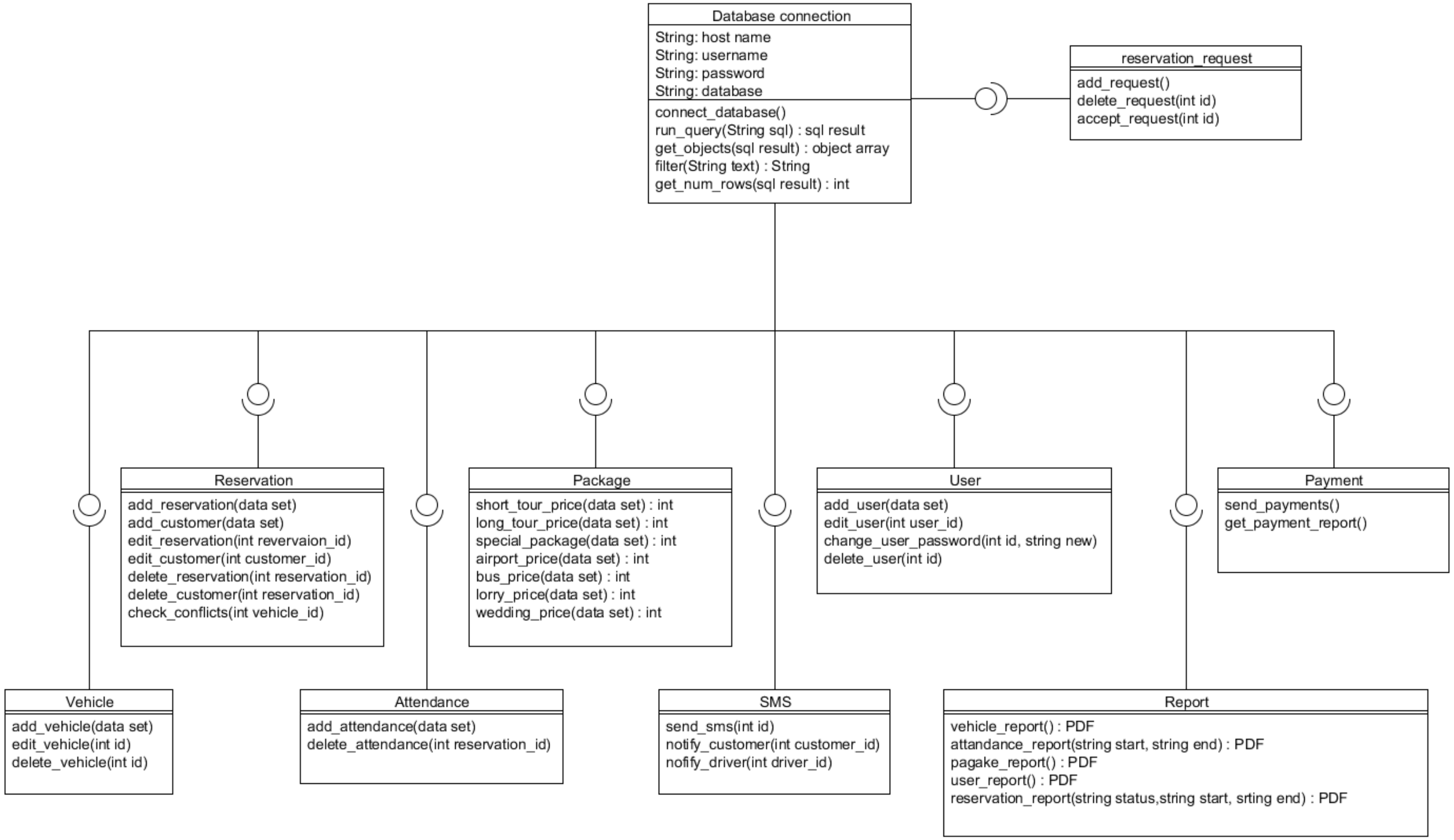

Database connection
String: host name
String: username
String: password
String: database
connect_database()
run_query(String sql) : sql result
get_objects(sql result) : object array
filter(String text) : String
get_num_rows(sql result) : int
reservation_request
add_request()
delete_request(int id)
accept_request(int id)
Reservation
add_reservation(data set)
add_customer(data set)
edit_reservation(int revervaion_id)
edit_customer(int customer_id)
delete_reservation(int reservation_id)
delete_customer(int reservation_id)
check_conflicts(int vehicle_id)
Package
short_tour_price(data set) : int
long_tour_price(data set) : int
special_package(data set) : int
airport_price(data set) : int
bus_price(data set) : int
lorry_price(data set) : int
wedding_price(data set) : int
User
add_user(data set)
edit_user(int user_id)
change_user_password(int id, string new)
delete_user(int id)
Payment
send_payments()
get_payment_report()
Vehicle
add_vehicle(data set)
edit_vehicle(int id)
delete_vehicle(int id)
Attendance
add_attendance(data set)
delete_attendance(int reservation_id)
SMS
send_sms(int id)
notify_customer(int customer_id)
nofify_driver(int driver_id)
Report
vehicle_report() : PDF
attandance_report(string start, string end) : PDF
pagake_report() : PDF
user_report() : PDF
reservation_report(string status,string start, srting end) : PDF

Figure 3.4 Class diagram of the Union cabs reservation system

In object oriented development "including" technique was used to get better code management. For that the following symbol which is normally used for situations of implementing a class using an interface was used.

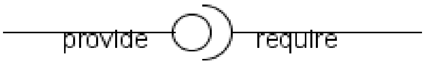


Figure 3.5 Object oriented including technology

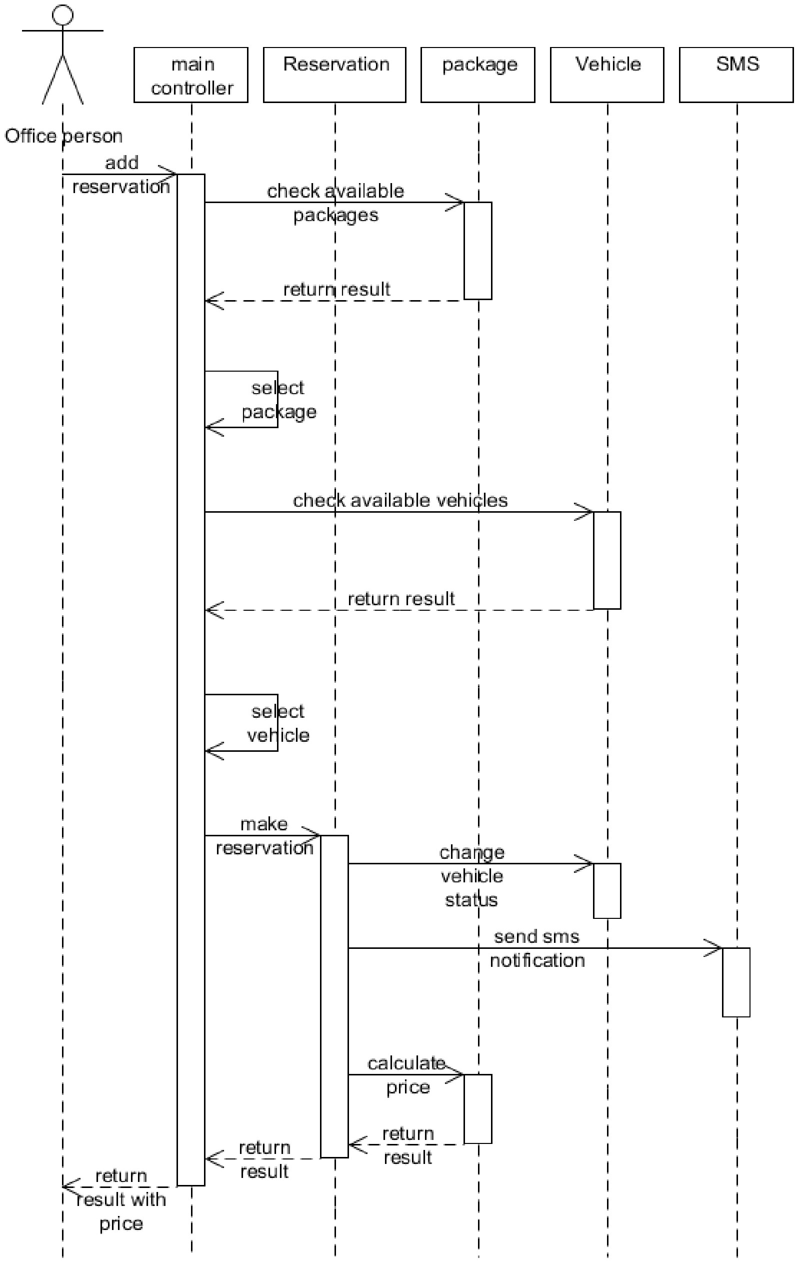


Figure 3.6 Sequence diagram for make vehicle reservation (at office)

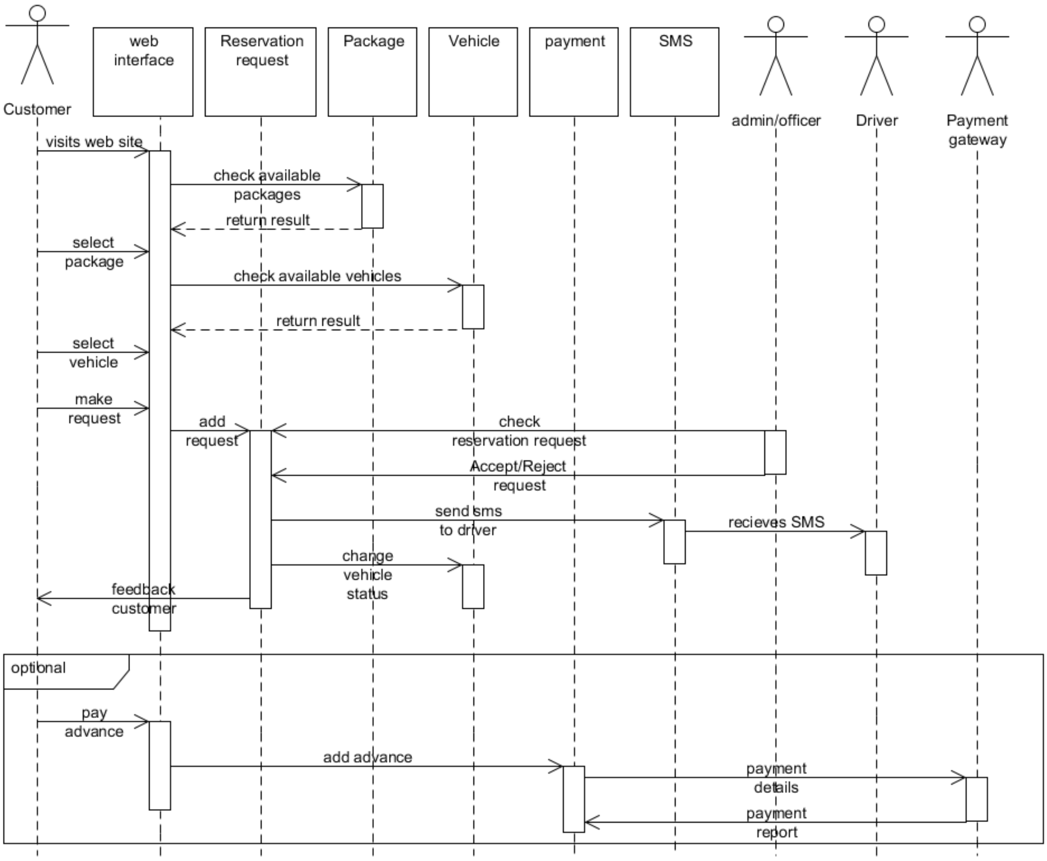


Figure 3.7 Sequence diagram for make vehicle reservation (at online)

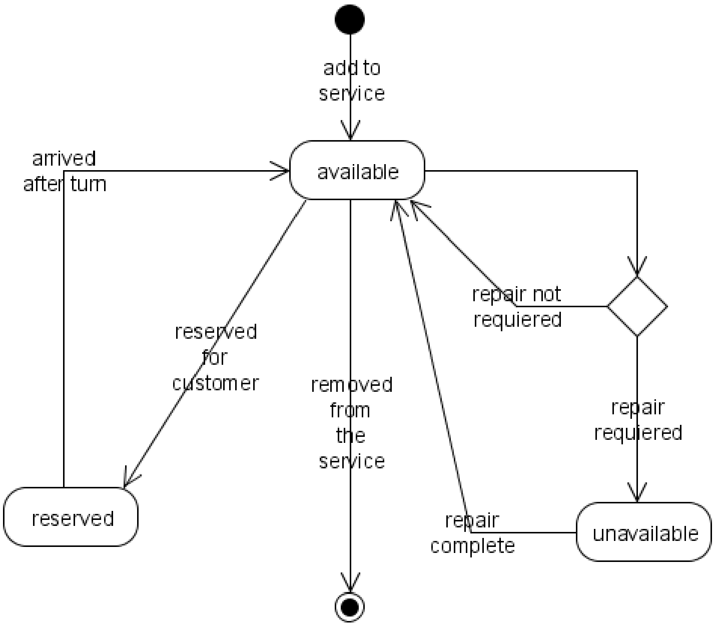


Figure 3.8 State transition diagram of vehicle in the Union Cabs system

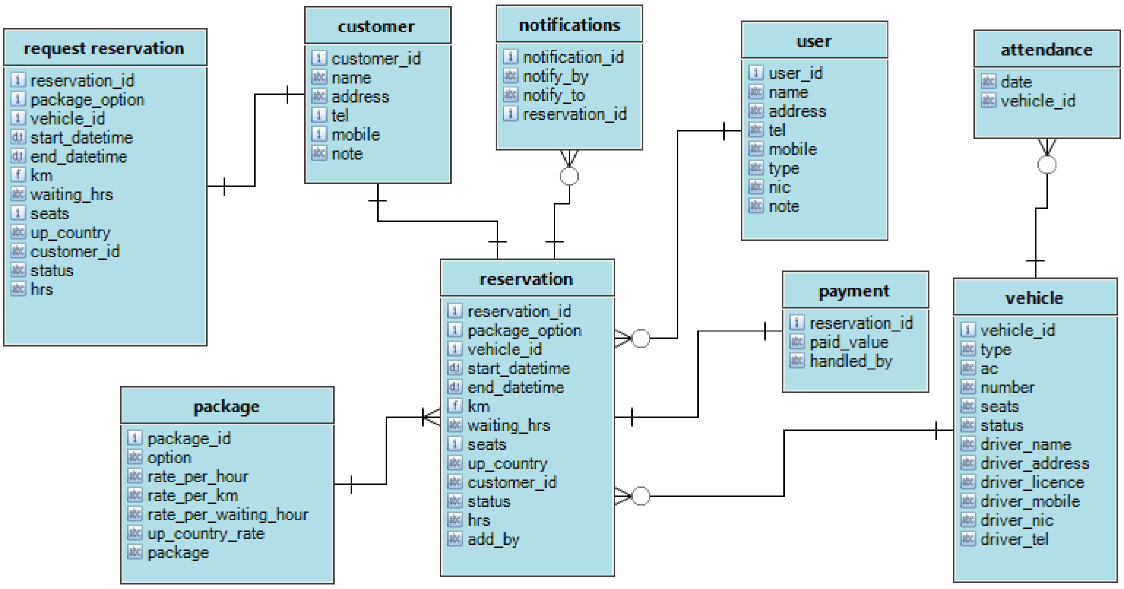
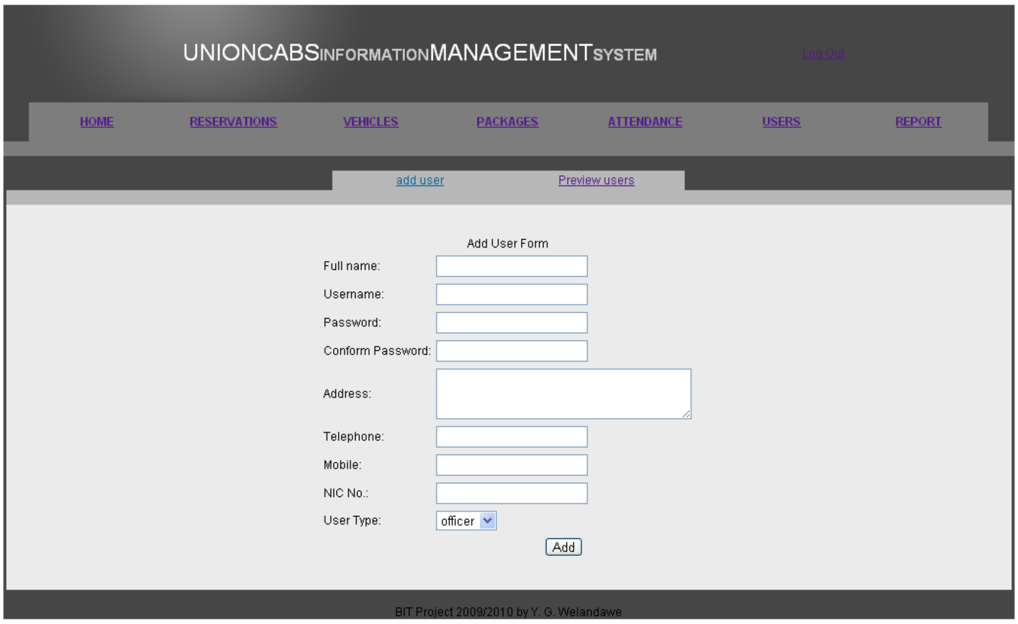


Figure 3.9 Entity relationship diagram of the Union Cabs system database

An algorithm for calculating the charges by automatically selecting an appropriate package option from given package, km, waiting hrs and total hrs inputs were needed. The pseudocode of this algorithm is given under appendix C.

3.2.5 User interface design

Main interface were designed using Adobe Photoshop CS4 application. Page layouts and banners were implemented using this application. Following image shows the designed interface of the Union Cabs Reservation Management System.



Add User Form

Full name:

Username:

Password:

Conform Password:

Address:

Telephone:

Mobile:

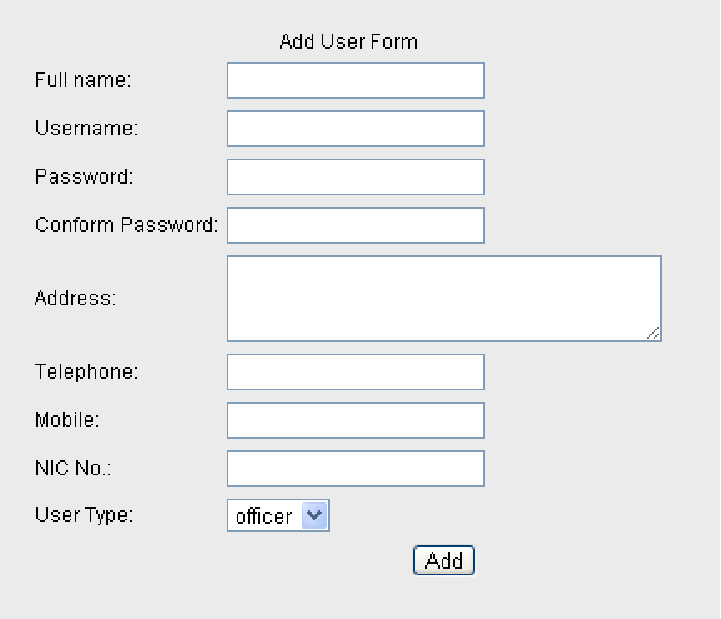
NIC No.:

User Type: officer

Figure 3.10 Main interface design

Macromedia Dreamweaver 8 was used to design data from to provide add and edit use interfaces.

Data form:



Add User Form

Full name:

Username:

Password:

Conform Password:

Address:

Telephone:

Mobile:

NIC No.:

User Type: officer

Figure 3.11 design of data insert forms

See Appendix D for more UI design.

Chapter 4 – Implementation

There were some important factors that had to be considered before selecting software for developing this system. Since the new system is a web based system selecting a web application development tool with compatible database tools is needed. And also the selected software platform should support the technology requirements which are needed to develop designed system. Those requirements are listed below;

- Object oriented development
- Client server technology
- Implement through three-tier-architecture
- Package system

The selected software should support to the nun-functional requirements that are mentioned in the chapter 2.

4.1 Implementation environment (hardware/software)

WAMP (Windows-Apache-MySQL-PHP) was selected as the software development environment. Windows as the platform, Apache as the web server, MySQL as the database server and PHP as the scripting language of server side web development was used. HTML, Javascript and CSS are used for client side web development.

HTML and CSS were used for developing user interfaces. Javascripts were used for developing form validations. PHP was used for developing business logic and handling the MySQL database.

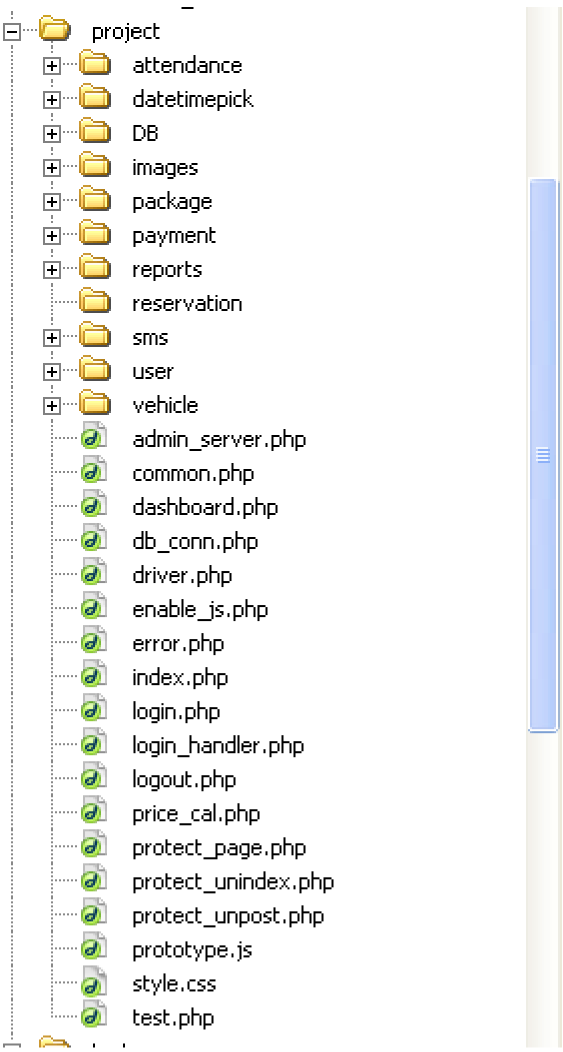
Windows XP was used as the development platform. Windows 7 was sometimes also used . Macromedia Deamweaver 8, Adobe Photoshop CS4 and MySQL query browser 1.2.12 was used as the system development tools while using Mozilla Firefox, Google Chrome, and Internet Explorer to run the application. Photoshop and Dreamweaver were used to create interfaces of the system. Interface layouts and graphic designs are made by using Photoshop. Dynamic layouts, data forms and data preview tables were developed by using Dreamweaver. And also Dreamweaver was used to develop PHP, HTML, CSS and Javascript programs. Databases and tables were developed by using MySQL Query Browser.

When considering the hardware environment, the Photoshop application which was used for the graphic design part required some good performance with processing and VGA. And next the MySQL database server required some processing power. But WAMP can be used on low performance hardware environments like less than 1GHz process and memory less than 128MB RAM. Therefore considering the overall hardware requirements 3.00GHz Dual Core processor, 1GB RAM, 256MB VGA and 80GB space available hard disk was used. Actually this is much higher performance than the required performance to do the main development. But this kind of performance was required to use Adobe Photoshop application. Additionally it required an internet connection to connect with the SMS and Payment gateways. Therefore 64KB/sec connection was used for that.

4.2 Code and module structure description

4.2.1 System file structure

The main action in this part is converting the system design to codes and files. The whole system was modularized and structured each single module in the design chapter. Therefore package system was used to build these modules. Following diagram shows the package system in the implementation.



```
project
    attendance
    datetimepick
    DB
    images
    package
    payment
    reports
    reservation
    sms
    user
    vehicle
    admin_server.php
    common.php
    dashboard.php
    db_conn.php
    driver.php
    enable_js.php
    error.php
    index.php
    login.php
    login_handler.php
    logout.php
    price_cal.php
    protect_page.php
    protect_unindex.php
    protect_unpost.php
    prototype.js
    style.css
    test.php
```

Figure 4.1 System file structure

For each module a package had to be created to keep view, business control, data access php files and javascript validate files. So in the above list each folder contains related php files and js file for each system module.

4.2.2 Database layer implementation

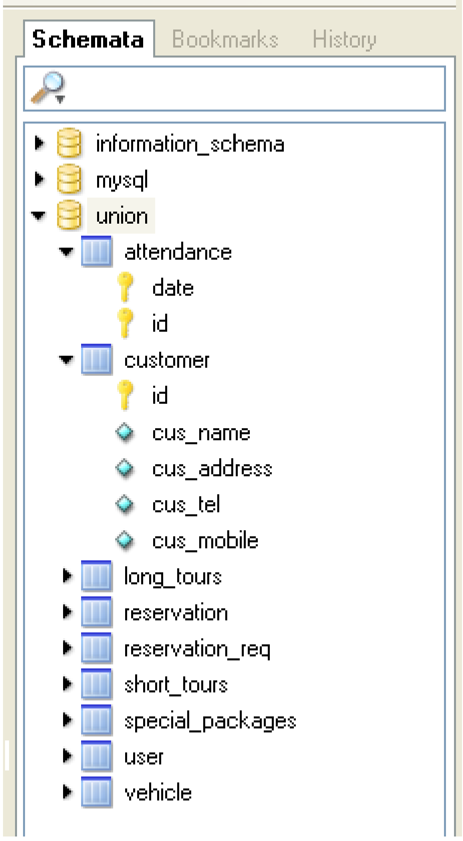
As mention before, the selected MySQL database management system was used for developing the databases to the Union Cabs reservation system. Following SQL queries were used for making databases and tables according to the design.

Create database statement;

```
CREATE DATABASE `union` /*!40100 DEFAULT CHARACTER SET latin1 */;
```

The table create statements are given under appendix C.

The created database and table list was displayed as follow;



```
information_schema
mysql
union
    attendance
        date
        id
    customer
        id
        cus_name
        cus_address
        cus_tel
        cus_mobile
    long_tours
    reservation
    reservation_req
    short_tours
    special_packages
    user
    vehicle
```

Figure 4.2 Created database

In the database connection part, object oriented singleton pattern was used to use only one database connection for database transactions. The reason for using this pattern was to control the number of database connections that were generate from the Union Cabs system.

Conversion of designed database connection to code and other database functions are given in Appendix C.

4.2.3 Business logic layer implementation

Implementing reservation charges calculate functions was a very important part of the system business logic implementation. There the all package algorithms was utilize to three function as short_tour_price(), long_tour_price() and special_pakage_price().

short_tour_price() function calculates up_and_down(car van less than 150km), drop (car van less than 150km), bus (only up and down) packages chargers.

long_tour_price() function calculates only calculates long tours which use car or van with more than 150km.

special_pakage_price() function calculates up_and_down_50, up_and_down_100, airport_up_and_down, airport_drop, lorry_up_and_down, lorry_drop and wedding package chargers.

Following code is the conversion of pre designed reservation chargers calculation algorithm;

```
class price_cal{
#This function use for car van/bus hires which used lower than 150km
/*
$package:               package name {up_and_down/drop/bus_up_and_down}
$ac:                    ac pricing option {non_ac/oneway_ac/twoway_ac}
$up_country:            yes or no {0/1}
$km:                    km count {0 < km <= 150}
$waiting_hrs:           wating time count in hours {in integers}
*/
public static function short_tour_price($package,$ac,$up_country,$km,$waiting_hrs){
        $sql = "SELECT *
                FROM short_tours
                WHERE package_name='$package' AND lower_km<'$km' AND
                '$km'<=upper_km AND up_country='$up_country';";
        $data = db_conn::get_data($sql);
        if(1==count($data)){
                $info = $data[0];
                $price = 0;
                $ac = $ac."_price";

                if($info->pricing_type == "fixed"){
                        $price += $info->$ac;
                }else if($info->pricing_type == "rate"){
                        $price += ($info->$ac) * $km;
                }

                $waiting_charge = ($info->waiting_charge_rate) * (($waiting_hrs / ($info->waiting_hr_per)) - $info->free_waitings);
                $price += $waiting_charge;
                return $price;
        }else{
                return false;
        }
}

#This function use only for car van hires which used higher than 150km
/*
$ac:                    ac pricing option {non_ac_price/oneway_ac_price/twoway_ac_price}
$up_country:            yes or no {1/0}
$km:                    km count {150 < km <= 10000}
$hrs:                   time count in hours {in integers}
*/
public static function long_tour_price($ac,$up_country,$km,$hrs){
        $ac = $ac."_price";
        $sql = "SELECT *
                        FROM long_tours
                        WHERE lower_km<'$km' AND '$km'<=upper_km AND
                        up_country='$up_country';";
        $data = db_conn::get_data($sql);

        if(1==count($data)){
                $info = $data[0];
```

```
			$price = 0;
			$price += ($info->$ac)*$km;
			if($hrs > $info->fixed_hrs){
				$extra_hr_charge = ($info->charge_per_extra_hr) * ($hrs - ($info->fixed_hrs));
				$price += $extra_hr_charge;
			}
			return $price;
			}else{
				return false;
			}
		}

#This function use for car van/lorry/wedding hires which used special packages
/*
$package:	name of the package {up_and_down_50/ up_and_down_100/
		 airport_up_and_down/ airport_drop/ lorry_up_and_down/ lorry_drop/
		wedding}
$ac:		ac pricing option {non_ac/oneway_ac/twoway_ac}
$km:		km count {0 < km}
$hrs:		time count in hours {in integers}
*/
public static function special_package_price($package,$ac,$km,$hrs){
	$sql = "SELECT *
			FROM special_packages
			WHERE package_name='$package';";
	$data = db_conn::get_data($sql);

	if(1==count($data)){
		$info = $data[0];
		$price = 0;
		$ac_price = $ac."_price";
		$ac_km_rate = $ac."_km_charge";

		$price += $info->$ac_price;
		if($km > $info->fixed_km){
			$price += ($km - $info->fixed_km) * $info->$ac_km_rate;
```

```
        }
        if($hrs > $info->fixed_hrs){
            $price += ($info->extra_hr_charge) * ($hrs - ($info->fixed_hrs));
        }
        return $price;
        }else{
            return false;
        }
    }
}
```

User, Vehicle, Attendance, reservation, sms, payments, package and report classes was also developed with all required variables and functions as designed.

4.2.4 Presentation layer implementation

Main interface were designed using Adobe Photoshop CS4 application. Page layouts and banners were implemented using this application.

Macromedia Dreamweaver 8 was used to develop data from to provide add and edit facilities for user, vehicle, package, attendance, report and reservation modules. Those forms are explained in Appendix D.

4.2.5 Reusable components

FPDF free reusable PHP class was used to develop system report generates. Some pre developed scripts are found form the www.fpdf.org to use for print tables and chart diagrams on the PDF files that useful for the required Union Cabs reports.

FPDF is an open-source PHP class which have 1732 lines of codes. It allows generating PDF files with pure PHP that is to say without using the PDFlib library. F from FPDF stands for Free that anyone may can use it for any kind of usage and modify it to suit their needs. FPDF has other advantages, high level functions. Here is a list of its main features:

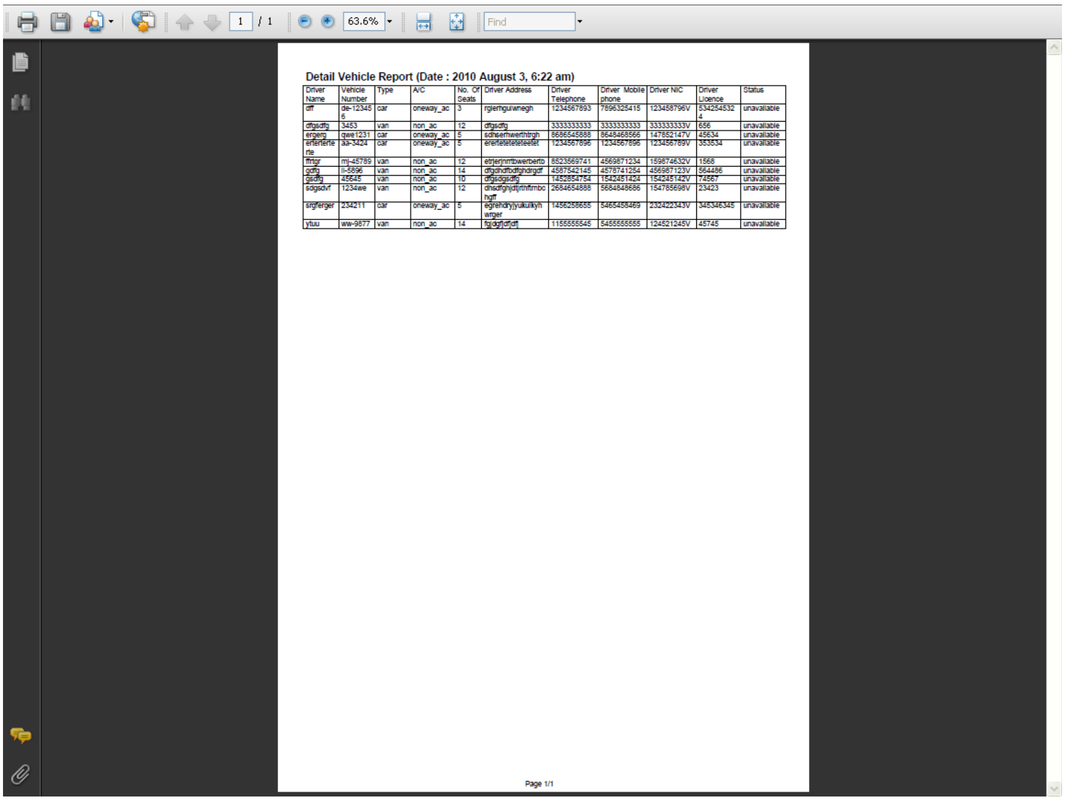
- Choice of measure unit, page format and margins
- Page header and footer management
- Automatic page break
- Automatic line break and text justification
- Image support (JPEG, PNG and GIF)
- Colours
- Links
- TrueType, Type1 and encoding support
- Page compression

In the scripts link at www.fpdf.org lot of extended useful script for different needs can be found. morepagestable.php is one of the useful extended class for fpdf which can generate table with headers and footers in the pdf pages. Following sample code gives a brief idea about this component;

This is a code that is use to generate attendance report;

```
if(isset($_POST)){
        $sy = $_POST["s_y"];
        $sm = $_POST["s_m"];
        $sd = $_POST["s_d"];
        $ey = $_POST["e_y"];
        $em = $_POST["e_m"];
        $ed = $_POST["e_d"];
        $start = $sy."-".$sm."-".$sd;
        $end = $ey."-".$em."-".$ed;
        $sql = "select a.date as date, v.driver_name as name, v.number as number from
union.attendance a, union.vehicle v where a.id=v.id and (date between '$start' and '$end') order
by a.date;";
        $data = db_conn::get_data($sql);
        $today = date("Y F j, g:i a");
        $pdf = new PDF('P','pt');
        $pdf->AddPage();
        $pdf->AliasNbPages();
        $pdf->SetFont('Arial','B',12);
        $pdf->MultiCell(0,20,'Detail Attendance Report (Date : '.$today.')');
        // set the tablewidths like this or write an extra function
        $pdf->tablewidths = array(180,180,180);
        srand(microtime()*1000000);
        $datas[] = array("Date","Name","Register number");
        foreach($data as $obj){
        $datas[] = array($obj->date,$obj->name,$obj->number);
        }
        $pdf->SetFont('Arial','',8);
        $pdf->morepagestable($datas,10);
        $pdf->Output();
}
```

Following image shows the preview of the vehicle detail in generated PDF file.



Detail Vehicle Report (Date : 2010 August 3, 6:22 am)

Driver Name	Vehicle Number	Type	A/C	No. Of Seats	Driver Address	Driver Telephone	Driver Mobile phone	Driver NIC	Driver Licence	Status
dff	de-123456	car	oneway_ac	3	rglerhguiwnegh	1234567893	7896325415	123458796V	5342545324	unavailable
dfgsdfg	3453	van	non_ac	12	dfgsdfg	3333333333	3333333333	333333333V	656	unavailable
ergerg	qwe1231	car	oneway_ac	5	sdhserhwerthtrgh	8686545888	8648468566	147852147V	45634	unavailable
erterterterte	aa-3424	car	oneway_ac	5	erertetetetetеefet	1234567896	1234567896	123456789V	353534	unavailable
ffrtgr	mj-45789	van	non_ac	12	etrjerjnrrtbwerbertb	8523569741	4569871234	159874632V	1568	unavailable
gdfg	ll-5896	van	non_ac	14	dfgdhdfbdfghdrgdf	4587542145	4578741254	456987123V	564486	unavailable
gsdfg	45645	van	non_ac	10	dfgsdgsdfg	1452854754	1542451424	154245142V	74567	unavailable
sdgsdvf	1234we	van	non_ac	12	dhsdfghjdtjrthftmbchgff	2684654888	5684848686	154785698V	23423	unavailable
srgferger	234211	car	oneway_ac	5	egrehdryjyukulkyhwrger	1456258655	5465458469	232422343V	345346345	unavailable
ytuu	ww-9877	van	non_ac	14	fgjdgfjdfjdfj	1155555545	5455555555	124521245V	45745	unavailable

Page 1/1

Figure 4.3 Sample generated PDF

PHPlot PHP class was used for generating statistical charts to full fill the report requirements. PHPlot is a graph library for dynamic scientific, business, and stock-market charts. PHPlot allows PHP developers to create pie charts, bar graphs, line graphs, point graphs, etc. from a PHP application. From http://phplot.sourceforge.net/ PHPlot class files and development documents were found. Three different charts were designed using this PHPlot class.

Single line chart was used to plot the number of vehicle reservations over the year. Following image shows the preview of the chart;

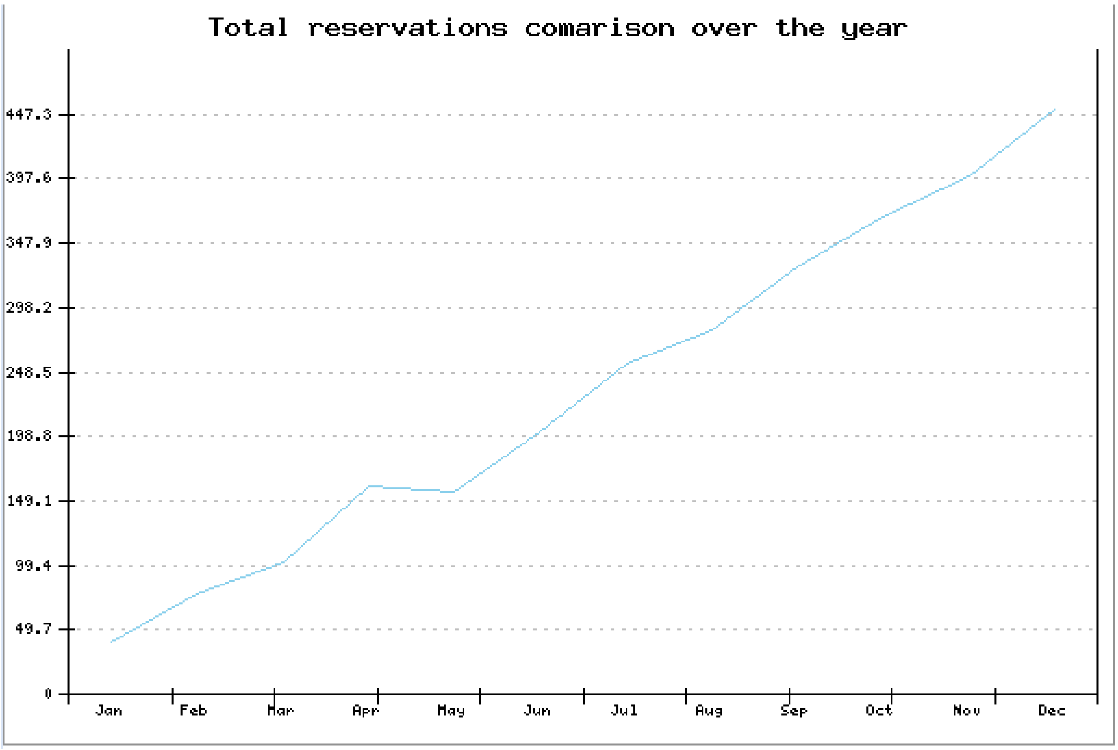


Figure 4.4 Total reservation comparison chart

Following code was used to develop above chart;

```
require_once 'phplot/phplot.php';

$data = array(
  array('Jan', 40),   array('Feb', 78),   array('Mar',102),  array('Apr',161),   array('May',156),
  array('Jun',202),  array('Jul',255),   array('Aug',281),   array('Sep',329),  array('Oct',368),
  array('Nov',399),   array('Dec',451)  );
```

```
$plot = new PHPlot(600, 400);
$plot->SetImageBorderType('plain');
$plot->SetPlotType('lines');
$plot->SetDataType('text-data');
$plot->SetDataValues($data);
# Main plot title:
$plot->SetTitle('Total reservations comarison over the year');
# Make sure Y axis starts at 0:
$plot->SetPlotAreaWorld(NULL, 0, NULL, NULL);
//Turn off X axis ticks and labels because they get in the way:
$plot->SetXTickLabelPos('none');
$plot->SetXTickPos('xaxis');
$plot->DrawGraph();
```

Multi lines chart was used to develop plot chart which shows comparison of reservation options in each month. Following image shows the preview of the chart;

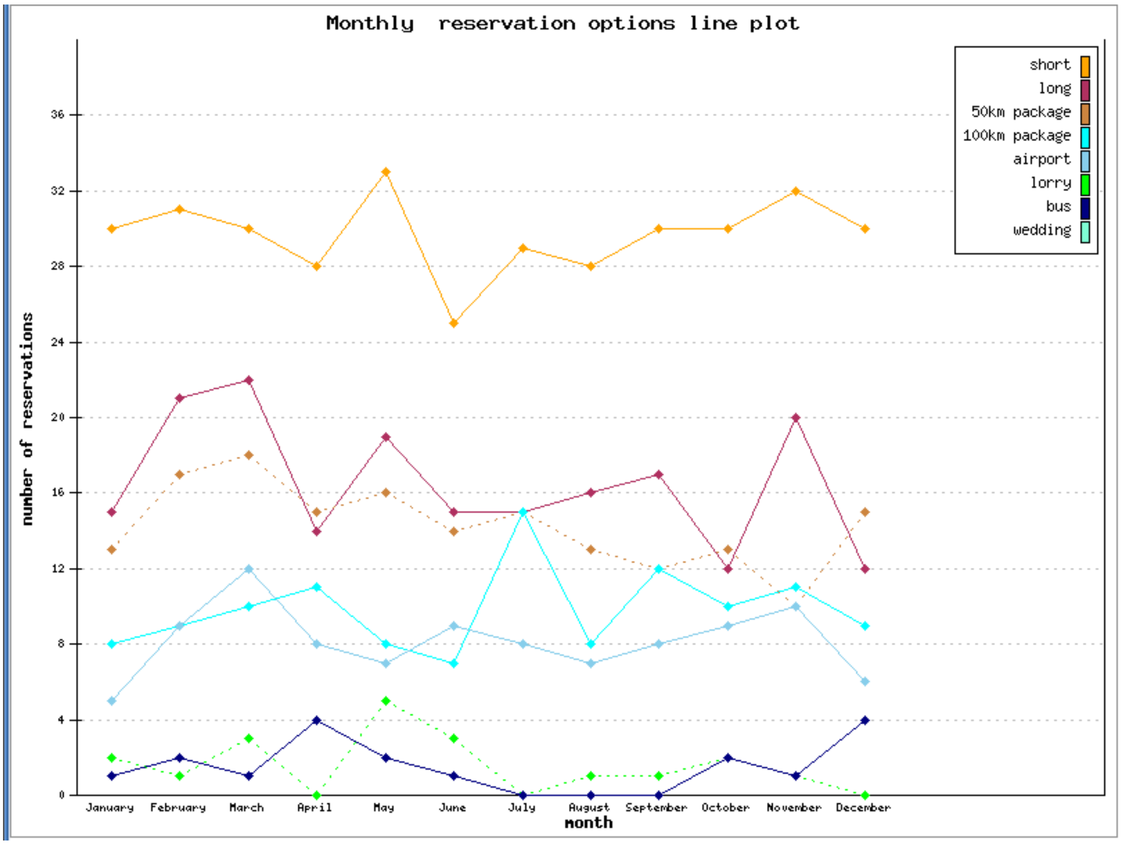


Figure 4.5 Reservation option comparison chart (over the time)

Following code was used to develop above chart;

```
//Include the code
require_once 'phplot/phplot.php';

//Define the object
$plot = new PHPlot(800,600);
$plot->SetImageBorderType('plain');

//Set titles
$plot->SetTitle("Monthly  reservation options line plot");
$plot->SetXTitle('month');
$plot->SetYTitle('number of reservations');

# This array is used for both the point shapes and legend:
$shapes = array('short', 'long', '50km package', '100km package', 'airport', 'lorry', 'bus',
'wedding');

//Define some data
$example_data = array(
    array('January'   ,30,15,13, 8, 5,2,1),
    array('February'  ,31,21,17, 9, 9,1,2),  // here we have a missing data point, that's ok
    array('March'     ,30,22,18,10,12,3,1),
    array('April'     ,28,14,15,11, 8,0,4),
    array('May'       ,33,19,16, 8, 7,5,2),
    array('June'      ,25,15,14, 7, 9,3,1),
    array('July'      ,29,15,15,15, 8,0,0),
    array('August'    ,28,16,13, 8, 7,1,0),
    array('September' ,30,17,12,12, 8,1,0),
    array('October'   ,30,12,13,10, 9,2,2),
    array('November'  ,32,20,10,11,10,1,1),
    array('December'  ,30,12,15, 9, 6,0,4)
);
$plot->SetDataValues($example_data);

# Increase X range to make room for the legend.
$plot->SetPlotAreaWorld(0, 0, 15,40);
```

```
//Turn off X axis ticks and labels because they get in the way:
$plot->SetXTickLabelPos('none');
$plot->SetXTickPos('none');

# Need some different colors;
$plot->SetDataColors(array('orange', 'maroon', 'peru', 'cyan', 'SkyBlue', 'green','navy',
'aquamarine1', 'violet', 'pink'));

# Also show that as the legend:
$plot->SetLegend($shapes);
//Draw it
$plot->DrawGraph();
```

Pie chart was used to develop chart which compare reservation options in each month. Following image shows the preview of the chart;

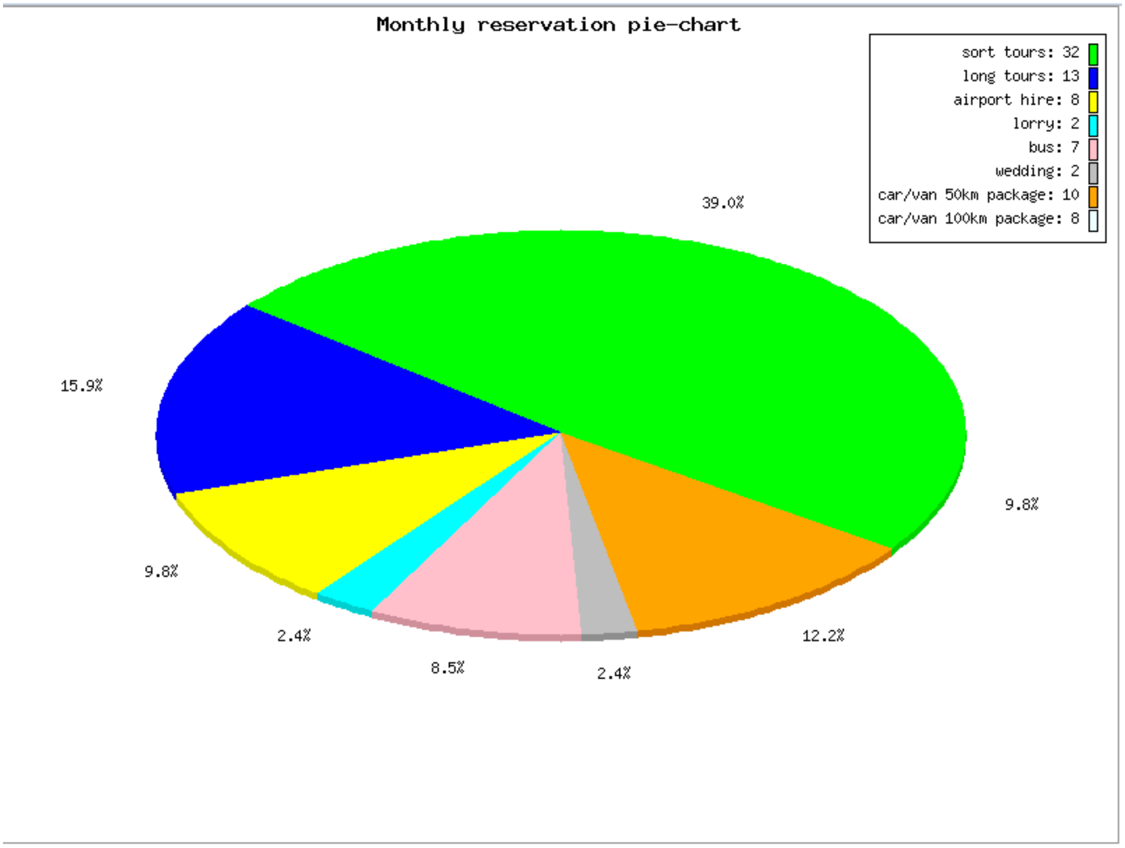


Figure 4.6 Compares number of reservation between reservation options

Following code was used to develop above chart;

```
require_once 'phplot/phplot.php';

# The data labels aren't used directly by PHPlot. They are here for our
# reference, and we copy them to the legend below.
$data = array(
  array('sort tours', 32),
  array('long tours', 13),
  array('airport hire', 8),
  array('lorry', 2),
  array('bus', 7),
  array('wedding', 2),
  array('car/van 50km package', 10),
  array('car/van 100km package', 8)
);

$plot = new PHPlot(800,600);
$plot->SetImageBorderType('plain');

$plot->SetPlotType('pie');
$plot->SetDataType('text-data-single');
$plot->SetDataValues($data);

# Set enough different colors;['red', 'green', 'blue', 'yellow', 'cyan','magenta', 'brown', 'lavender',
'pink','gray', 'orange']
$plot->SetDataColors(array('green', 'blue', 'yellow', 'cyan','pink','gray', 'orange'));

# Main plot title:
$plot->SetTitle("Monthly reservation pie-chart");

# Build a legend from our data array.
# Each call to SetLegend makes one line as "label: value".
foreach ($data as $row)
  $plot->SetLegend(implode(': ', $row));

$plot->DrawGraph();
```

Prototype Javascript framework was used to integrate advanced Ajax components and date time picks. It is an easy way to use advance javascript technology with few coding line. The prototype framework was found as www.prototypejs.org.

Some advanced functions from Prototype Javascript framework was used to implement Ajax communications to get immediate reservations requests that customers request online. Therefore Union Cabs officer and admins can have online reservation requests within seconds of the customer requesting it.

Following coding shows the used function from the prototype framework;

```
<script type="text/javascript" src="prototype.js"></script>
<script>
function checkRequests(){
        new Ajax.PeriodicalUpdater('requests', 'admin_server.php', {
                method: 'get', frequency: 1, decay: 1
        });
}
</script>

<body onLoad="checkRequests();">
<br />
<center><h2>:::Online Customer Requests:::</h2></center>
<br />
<div id="requests" align="center"><input type="button" value="Start Checking Requests"
onClick="checkRequests();"/></div>
<br />
<br />
</body>
```

Got a free user friendly date picker and time picker from the jongsma.org [www.home.jongsma.org/ software/js/datepicker] and integrate it with the data forms which were needed to select dates and time. It was developed by using prototype framework. All information about installing the component from the jongsma.org web site was gathered. The required files were downloaded from www.prototypejs.org. Following code shows the integration of date picked and time picker;

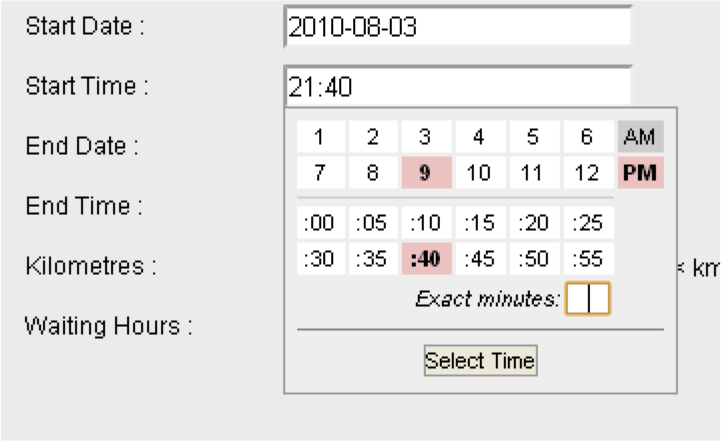
```
<script type="text/javascript" language="javascript" src="datetimepick/prototype.js"></script>
<script type="text/javascript" language="javascript" src="datetimepick/prototype-date-extensions.js"> </script>
<script type="text/javascript" language="javascript" src="datetimepick/behaviour.js"> </script>
<script type="text/javascript" language="javascript" src="datetimepick/datepicker.js"> </script>

<link rel="stylesheet" href="datetimepick/datepicker.css" />

<script type="text/javascript" language="javascript" src="datetimepick/behaviors.js"> </script>

<table>
        <form action="reservation/rese_handler.php" id="customer_form" name="customer_form" method="post">
   <tr><td>Start Date : </td>
       <td><input id="start_date" name="start_date" type="text" class="datepicker"/></td>
   </tr>
   <tr>
         <td>Start Time : </td>
        <td><input id="start_time" name="start_time" type="text" class="timepicker"/></td>
   </tr>
   <tr>
        <td>End Date : </td>
        <td><input id="end_date" name="end_date" type="text" class="datepicker"/>td>
   </tr>
   <tr>
         <td>End Time : td>
        <td>input id="end_time" name="end_time" type="text" class="timepicker"/></td>
   </tr>
```

Following images shows the previews of this javascrip time picker and date picker;



Start Date : 2010-08-03

Start Time : 21:40

End Date :

End Time :

Kilometres :

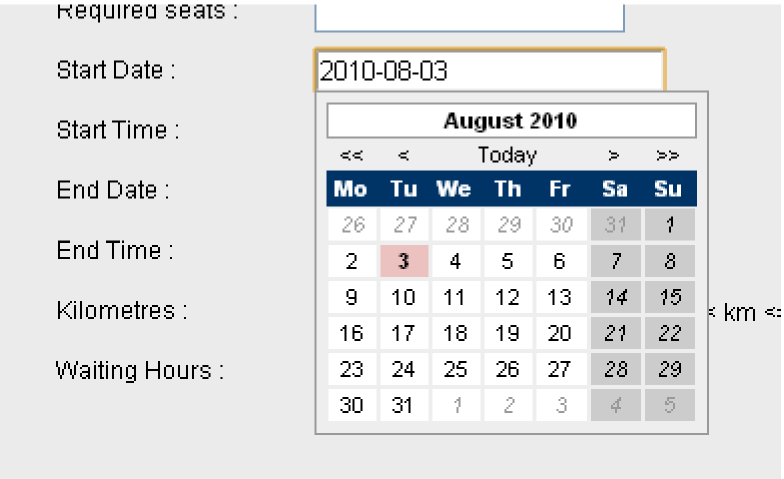
Waiting Hours :

1	2	3	4	5	6	AM
7	8	**9**	10	11	12	**PM**

:00	:05	:10	:15	:20	:25
:30	:35	**:40**	:45	:50	:55

Exact minutes:

Figure 4.7 Time picker



Required seats :

Start Date : 2010-08-03

Start Time :

End Date :

End Time :

Kilometres :

Waiting Hours :

August 2010

Mo	Tu	We	Th	Fr	Sa	Su
26	27	28	29	30	31	1
2	**3**	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Figure 4.8 Date Picker

More coding’s available in Appendix C: Code Listing section of the dissertation.

Chapter 5 – Evaluation

The main objective of software testing is to identify the correctness, completeness and quality of developed computer software. Software testing helps to Verify and Validate the software and to check whether the Software is working as it is intended to work. Software testing part was completed by using test plan, test cases and test data.

5.1 Test plan and results

Making test plan was very easy, since the whole system was a well-structured one. Therefore a simple test plan was enough to accomplish the testing phase of Union Cabs system development. This test plan included test items, test objectives, used test cases, used test data and expected results. Since the system was modularized and 100% uncoupled (expect the reservation and package module) at the business logic layer, those modules were used as the tested items. Therefore the most critical tested items were as follows;

- System user module
- Vehicle module
- Attendance module
- Package module
- Reservation module and SMS module
- Report module
- Notify online reservation request module
- Online payment module

5.1.1 System user module test

Table 5.1 Test plan for system use module

Objective	Test case	Expected result	Status
Check whether add user component works properly for admin user	Enter the add user tab and type valid user details to the add user form	Redirect to main user module page and display green notification as "Add user done"	Pass
Check whether user preview component works properly for admin user	Enter in to the user preview tab	Display list of user details sorted by user names	Pass
Check whether user edit component works properly for admin user	Change edit form data to valid data and submit	Redirect to the user preview page	Pass
Check whether user delete component works properly for admin user	Click the delete link at the user preview page and enter "yes" on the alert massage	Redirect to the user preview page and display the green massage as "user delete done"	Pass
Check whether user logging component works properly for admin user and officer user	Enter accepted login details to the logging page	Redirect to the dashboard page	Pass
Check whether officer user can't access the attendance module	Enter the attendance tab	Redirect to the dashboard and display the red massage as "officer users can access attendance module"	Pass
Check whether officer user can't access the report module	Enter the report tab	Redirect to the dashboard and display the red massage as "officer users can access report module"	Pass
Check whether officer user can't access the add vehicle page	Enter the add vehicle tab in the vehicle module	Redirect to the vehicle main page and display the red massage as "officer users can access add vehicle page"	Pass

Check whether officer user can’t access the add user page	Enter the add user tab in the user module	Redirect to the user main page and display the red massage as “officer users can access add user page”	Pass
Check whether officer user can’t access the edit user page	Enter the edit link at the user preview page	Redirect to the user main page and display the red massage as “officer users can access edit user page”	Pass
Check whether officer user can’t access the edit vehicle page	Enter the edit link at the vehicle preview page	Redirect to the vehicle main page and display the red massage as “officer users can access edit vehicle page”	Pass

5.1.2 Vehicle module test

Table 5.2 Test plan vehicle module

Objective	**Test case**	**Expected result**	**Status**
Check whether add vehicle component works properly	Enter the add vehicle page and enter valid data in to the form	Redirect to the vehicle preview page and display green massage as “add vehicle done!”	Pass
Check whether preview vehicle component works properly	Enter the preview vehicle page	Display list of vehicles	Pass
Check whether edit vehicle component works properly	Enter the edit vehicle link at the vehicle preview page and change form data to another valid data set	Redirect to the vehicle preview page and display green massage as “edit vehicle done”	Pass
Check whether delete vehicle component works properly for non-reserved vehicles	Enter the delete vehicle link at the vehicle preview page and click “yes” to the alert massage	Redirect to the vehicle preview page and display green massage as “delete vehicle done”	Pass
Check whether delete vehicle component not	Enter to the delete vehicle link of	Redirect to the vehicle preview page and display	Pass

works for reserved vehicles	currently reserved vehicle at the vehicle preview page and click “yes” to the alert massage	red massage as “can’t delete reserved vehicles”	

5.1.3 Attendance module test

Table 5.3 Test plan for attendance form

Objective	Test case	Expected result	Status
Check whether attendance add by driver name component works properly	Enter the add attendance tab and enter valid driver name to the name text box and enter	Return to the today attendance preview page and display green massage as “add attendance done”	Pass
Check whether attendance add by vehicle number component works properly	Enter the add attendance tab and enter valid vehicle number to the vehicle text box and enter	Return to the today attendance preview page and display green massage as “add attendance done”	Pass
Check whether preview attendance component works properly	Enter the preview attendance tab and select a date. Then click the “view” button	Preview list of attendance details of selected date	Pass

5.1.4 Package module test

Table 5.4 Test plan for package module

Objective	Test case	Expected result	Status
Check whether package charge calculator component works properly	Enter the package charge calculator page and select reservation option, then enter valid reservation	Display correctly calculated charges	Pass

	details to the forms and click "calculate" button		
Check whether package charges details component works properly	Go to the package detail tab in the package page	Display charges details of all reservation options.	Pass

5.1.5 Reservation module and SMS module test

Table 5.5 Test plan for reservation module

Objective	Test case	Expected result	Status
Check whether add reservation component works properly	In the add reservation page select a reservation option and enter valid reservation details to the form and enter	Return to the reservation preview page and display green massage as "add reservation done", the reservation status is "waiting" and charges should calculate correctly. And also should send SMS to the drive, if a vehicle selected	Pass
Check whether preview reservation component works properly	Enter the preview reservation tab	The first list should show completing reservations, and next list should show waiting reservation	Pass
Check whether edit reservation component works properly	Enter the edit reservation link in the reservation preview page and change data to another valid data set	Return to the reservation preview page and display green massage as "edit reservation done", the reservation status should change according to the changes and charges should calculate correctly. And also should send SMS to the	

		drive, if a vehicle selected	
Check whether delete reservation component works properly	Enter the delete reservation link in the reservation preview page and press “ok” button in the alert massage	Redirect to the reservation preview page and display green massage as “reservation delete done”	

5.1.6 Report module test

Table 5.6 Test plan for report module

Objective	Test case	Expected result	Status
Check whether report of vehicles component works properly	Enter the vehicle report page	Generate PDF file with list of all vehicle details which have registered with the system.	Pass
Check whether attendance report component works properly	Go to the attendance report page and select starting date and the end date from the form and enter	Generate PDF file with all attendance details within the given time period	Pass
Check whether report package detail component works properly	Enter report package tab	Generate PDF which contain all packages details	Pass
Check whether report total reservations chart component works properly	Enter total reservation chart report tab	Generate PDF file that contain single line chart of number of reservation over the time	Pass
Check whether report compare reservation options chart component works properly	Enter compare reservation options and select start time and end time	Generate PDF file that contain multi line chart, and each different colour line represents a different reservation option	Pass

Check whether report compare reservation options pie chart component works properly	Enter compare reservation options pie chart and select start time and end time	Generate PDF file that contain pie chart, and each colour represents a different reservation option	Pass

5.1.7 Notify online reservation request module test

Table 5.7 Test plan for notify online reservation request module

Objective	Test case	Expected result	Status
Check whether online reservation request component works properly	Go to the customer site and select reservation option. Then enter the valid reservation details and submit	Display online reservation requests in the Union cabs dashboard in a second	Pass
Check whether accept online reservation request component works properly	Go to the Union cabs dashboard and accept a reservation request	The accepted reservation request should display on reservation list in the reservation module	Pass

5.1.8 Online payment module test

Table 5.8 Test plan for online payment module

Objective	Test case	Expected result	Status
Check whether online payment component works properly	In the customer web site, select pay online after selecting reservation options and reservation details.	Receive payment report to the Union cabs payment tab	Pass

5.2 Test data and expected results

Some past Union Cabs data were used as test data and expected results. These data were captured from some old documents from the Union Cabs head office. After using these test data, the expected result was obtain through the system.

5.3 Acceptance testing

The user acceptance testing is usually a black box type of testing. In other words, the focus is on the functionality and the usability of the application rather than the technical aspects.

This Vehicle Reservation Information Management System was tested in the read environment using real test data. The acceptance testing was done by experiences end use from the Union Cabs and the test cases was made by him.

The acceptance testing was successfully completed. Any software has newly identified requirements. Therefore the software developers upgrade their software to new versions for answer to those newly identified requirements. As like that, found some new requirements for the Union cabs reservation information management system during the acceptance testing.

5.4 Detected errors

In this system FPDF and PHPlot reusable codes were used to generate PDF and chart images respectively. There were some errors in both these components when the systems run on PHP display all errors mode (this mode changes using php.ini error handling variables). But the system worked fine then on “off” error display mode.

In the FPDF following errors returned when the system was working on error display “On” mode;

```
Warning: Cannot modify header information - headers already sent by (output started at
D:\wamp\www\project\reports\morepagestable.php:12) in D:\wamp\www\project\reports\fpdf
.php on line 1017
FPDF error: Some data has already been output, can't send PDF file
```

But this error could be removed by changing several codes in the extended FPDF file. But it was hard to apply this correction to this system, because the scheduled time period was not enough for that.

In the PHPlot following errors returned when the system was working on error display “On” mode;

```
Deprecated: Function eregi()
Deprecated: Function split()
Deprecated: Function ereg_replace()
```

The reason for this error was that these functions will be removed from PHP in the newer versions. So that was only a warning for PHP developers.

These errors are not harmful for the system. But this may cause difficulties in the future improvements. Therefore PHPlot developers might be able to provide improved component for this.

Chapter 6 – Conclusion

6.1 Critical assessment of project

Since this system covers all requirements in the reservation management, Union Cabs officers can easily manage their vehicle reservations with having clear understanding of the on-going processes. By using this system they can register vehicles and can easily check vehicle information through it. And also can mark attendance of those registered vehicles in a very easy and quick way.

There is fully functional vehicle reservation handling system. By using it officers can make reservations without considering vehicle reservation conflicts. Because it is automatically handle by the system. The attendance is also needed to make quick reservation on the day. The newly generated feature to the Union Cabs is the facility to make reservations online. Customers request vehicle reservations at home and the Union Cabs officers can have that information in a second. And also by using the reports the Union Cabs owner can have useful analysed statistical charts for future business improvements. And also the data is perfectly secured form the outsiders.

The system was developed using three-tier-architecture with having some Object Oriented futures. With these structural strengths the system implemented using free-open-source technologies which have flexible and portable advantages.

This project was sensed to have been a successful one, because it has accomplished all the proposed functional and non-functional requirements with additional features. Creating this much of functionalities in a limited time period was not an easy job. But there are lot of ways to improve this system. Actually this system was developed using basic technologies and techniques. But the topic was very heavy and there were a lot of functionalities to complete. Therefore in this stage it is hard to cover all the reachable levels from this project. Therefore the suggestions for future improvements are giver in the next topic.

6.2 Future work

As mentioned before, the system can be improved in a lot of ways by adding new feature and new technologies. Those improvements that can be used for this system are discussed below.

When the user is going to remove a vehicle from the system which has been reserved, then a feature can be added to remove the vehicle from reservation.

Automated database backup system could be added to the database.

Actually the vehicle reservation charge calculator is not a powerful feature for Union Cabs. Actually this function is required by the driver at the customer's travel destination, because the final charge is calculated at the customer dropping place. Therefore the best solution for this is providing a mobile application for the driver to calculate hire charges.

Establish voice communication between Siddamulla head office, Kottawa branch and to the online customers. It will improve the communication and save the time.

6.3 Lessons learnt

- Get familiar with Prototype JavaScript Framework
- Experience on Ajax usage
- Get familiar with FPDF and PHPlot PHP libraries
- Gaining knowledge and experience on Object Oriented Singleton design patterns
- Get hands on experience on software development process
- Time management
- Gathering actual customer requirements is not an easy job

References

[WWW1] http://www.sriderana.com

[WWW2] http://www.lankaholidays.com

[WWW3] http://www.danweem.com/sri-lanka-classifieds-61.html

[WWW4] http://www.atl.lk

[WWW5] http://www.infotaxi.org

[WWW6] www.rentalcarsrilanka.com

[WWW7] www.thrifty.co.uk

[WWW8] www.nationalcar.co.uk

[WWW9] www.sixt.co.uk

[WWW10] www.vanhire3000.com

[WWW11] www.leisuretours.biz

[WWW12] http://www.umlet.com/changes.htm

[WWW13] http://www.risetobloome.com/

Appendix A - System Documentation

Hardware requirements

Table A.1 Server minimum hardware requirements

	Recommended Hardware Requirements
Processor	Intel Pentium D or equal processor
Memory	256 MB RAM
Hard Disk	10 GB Free Disk Space
Screen Resolution	1280 * 1024
Internet	128Kb/s connection minimum

Table A.2 End user hardware requirements (Union Cabs officers)

	Recommended Hardware Requirements
Processor	Intel Pentium IV or equal processor
Memory	128 MB RAM minimum
Hard Disk	10 GB Free Disk Space (is more than enough)
Screen Resolution	1280 * 1024 minimum
Internet	64Kb/s connection minimum

Software requirements

Table A.3 Server software requirements

	Recommended Software Requirements
Operating system	Windows server 2000
WAMP server	WAMP 2.0

Table A.4 Client software requirements

	Recommended Software Requirements
Operating system	any
Web browser	Google Chrome and Mozilla Firefox3.5.x are tested and

	supported. Internet Explorer is not recommended to Union Cabs end users. Customers may can use.

How to setup server

Step 1: Install windows operating system. (Windows 2000 or newer version)

Step 2: Install the WAMP 2.0 of C driver. (Do not need change any this during the installation)

Step 3: Copy project and customer_site folder to C:\wamp\www\.

Step 4: Go to php.ini file (click WAMPSERVER icon of system tray -> PHP -> php.ini). Go to Error handling and logging part and set “display_errors” to “Off” (display_errors = Off). Then restart the WAMPSERVER (click WAMPSERVER icon of system tray -> Restart All Services).

Step 5: Open a web browser. Type “http://localhost/project/install_db.php” and press enter. Then go to the “C:\wamp\www\project\“ and delete “install_db.php” file.

How to setup client

Download and install Google chrome or Mozilla Firefox 3.5.x or newer version.

How to obtain database backup

First you have to go to system tray and click on the WAMPSERVER icon. Then click on the phpMyAdmin. After that a web browser will open (is your default browser) with opened phpMyAdmin. Then click on “Export” tab in the home page. Then select “union” database in the Export box. After that scroll down and tick the Save as and press “Go” button. Immediately the backup file will be downloaded from the web browser.

Appendix B - User Documentation

This section delivers a guide to use the Union Cabs system with interfaces and appropriate details. Note that this user documentation covers only the main functionalities with selected interfaces of the developed system.

Login to the system

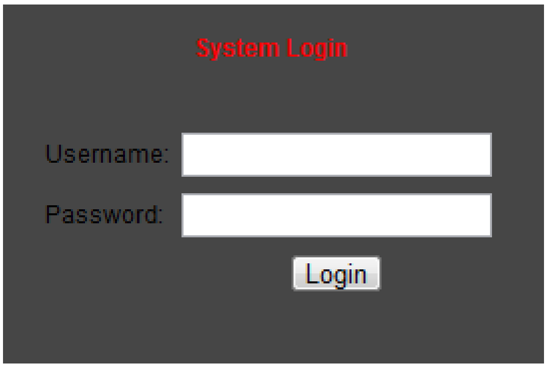
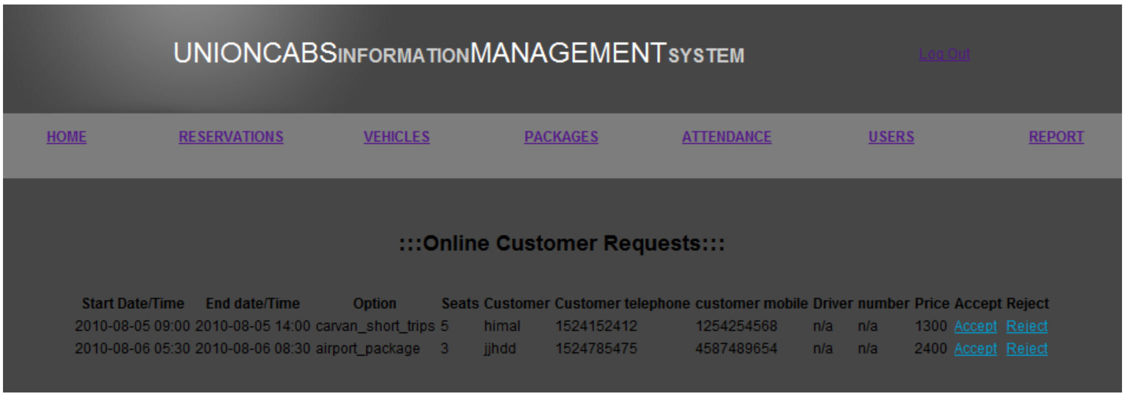


Figure B.1 System login form

This is the entry page to the system. It provides the login facility which allows the users to enter into the system once the correct User Name and Password is entered. Entered values are case sensitive. Note that some modules are in the system access controlled according to the logged user.

Home page (Dashboard)



:::Online Customer Requests:::

Start Date/Time	End date/Time	Option	Seats	Customer	Customer telephone	customer mobile	Driver	number	Price	Accept	Reject
2010-08-05 09:00	2010-08-05 14:00	carvan_short_trips	5	himal	1524152412	1254254568	n/a	n/a	1300	Accept	Reject
2010-08-06 05:30	2010-08-06 08:30	airport_package	3	jjhdd	1524785475	4587489654	n/a	n/a	2400	Accept	Reject

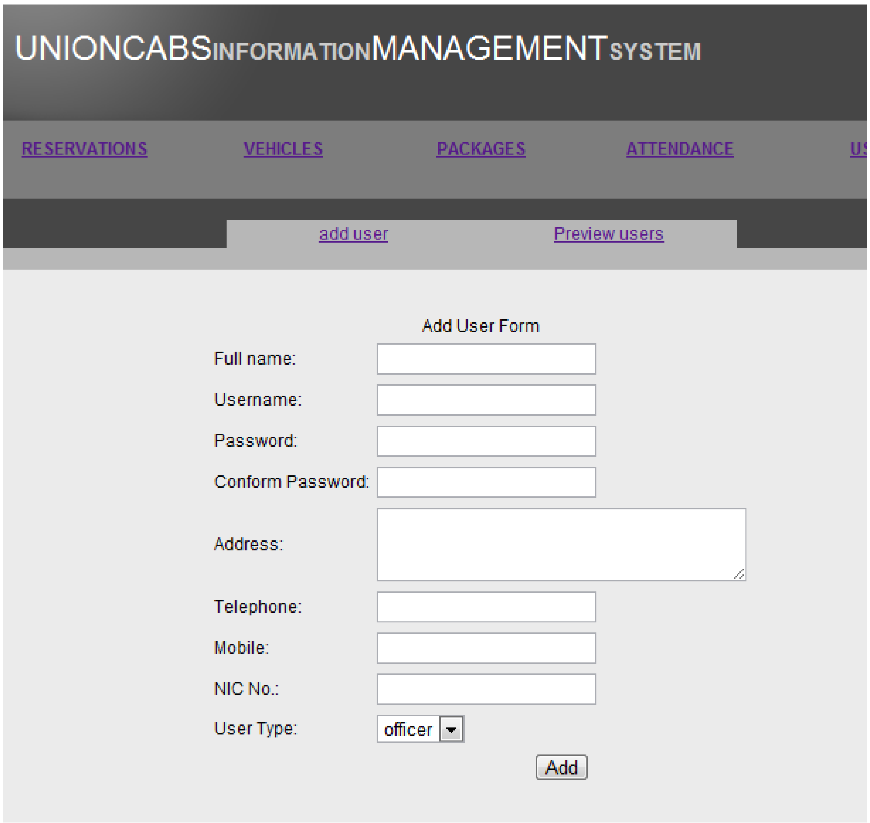
Figure B.2 Home page

Once successfully logged in, the home page of the system is displayed. This includes tabs which perform different tasks of the system. The home page also includes the facility of displaying the list of online vehicle reservation requests.

User tab

Once click on the User tab, it shows two sub tabs called add user and preview user. Only admin users are allowed to access add user page. Preview users page can be accessed by both users (admin and officer).

In order to add new user click on the add user tab in the user tab. Use the empty form to enter the new users details.



Add User Form

Full name:

Username:

Password:

Conform Password:

Address:

Telephone:

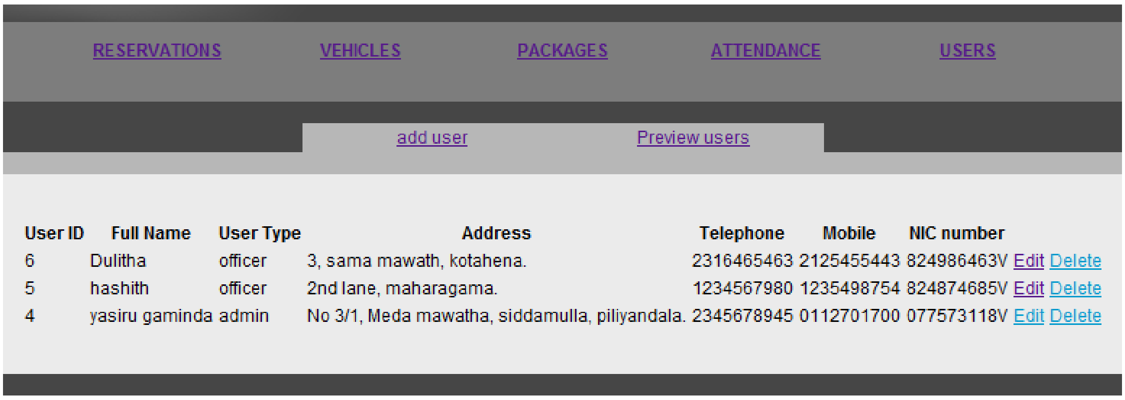
Mobile:

NIC No.:

User Type: officer

Figure B.3 Add new user form

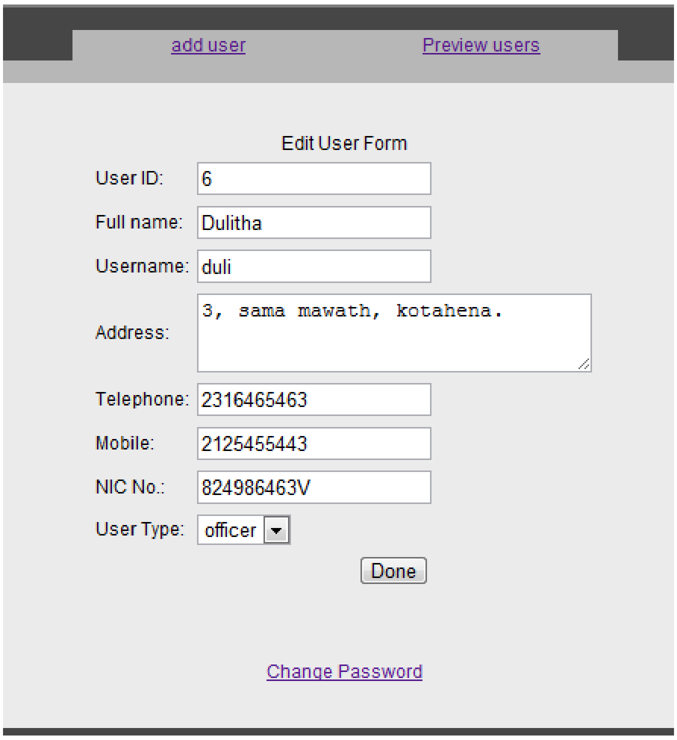
Preview user tab can be used to get the list of users that are currently registered with the system.



User ID	Full Name	User Type	Address	Telephone	Mobile	NIC number		
6	Dulitha	officer	3, sama mawath, kotahena.	2316465463	2125455443	824986463V	Edit	Delete
5	hashith	officer	2nd lane, maharagama.	1234567980	1235498754	824874685V	Edit	Delete
4	yasiru gaminda	admin	No 3/1, Meda mawatha, siddamulla, piliyandala.	2345678945	0112701700	077573118V	Edit	Delete

Figure B.4 Preview users

Edit links in the user preview page can be use to edit each user. Use delete link to remove user from the system. To change password use change password link in the bottom of the edit page.

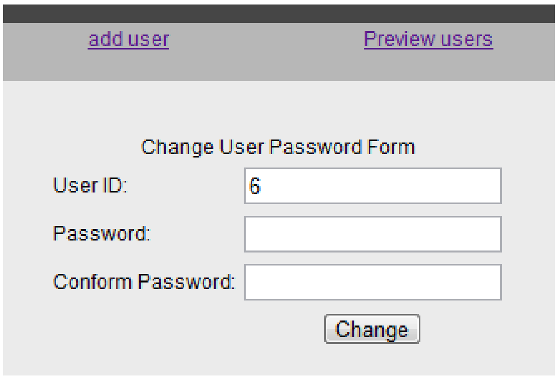


Edit User Form

User ID:	6
Full name:	Dulitha
Username:	duli
Address:	3, sama mawath, kotahena.
Telephone:	2316465463
Mobile:	2125455443
NIC No.:	824986463V
User Type:	officer

Figure B.5 Edit user form

In the change password link it shows the form to change password as follows.



Change User Password Form

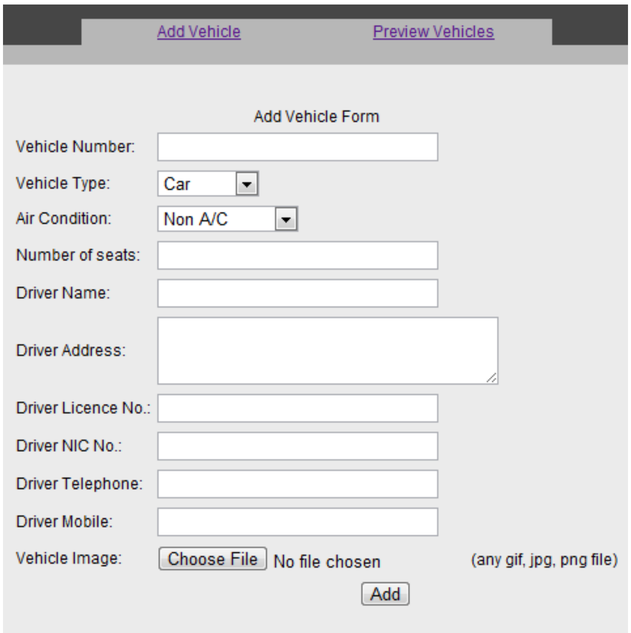
User ID:	6
Password:	
Conform Password:	

Figure B.6 Change user password

Vehicle tab

Once click on the Vehicle tab, it shows two sub tabs called add vehicle and preview vehicles. Only admin users are allowed to access add vehicle page. Preview vehicles page can be accessed by both users (admin and officer).

In order to add new vehicle to the system click on the add vehicle tab in the vehicle tab. Use the empty form to enter the new vehicle details. Vehicle image can also be upload from here.

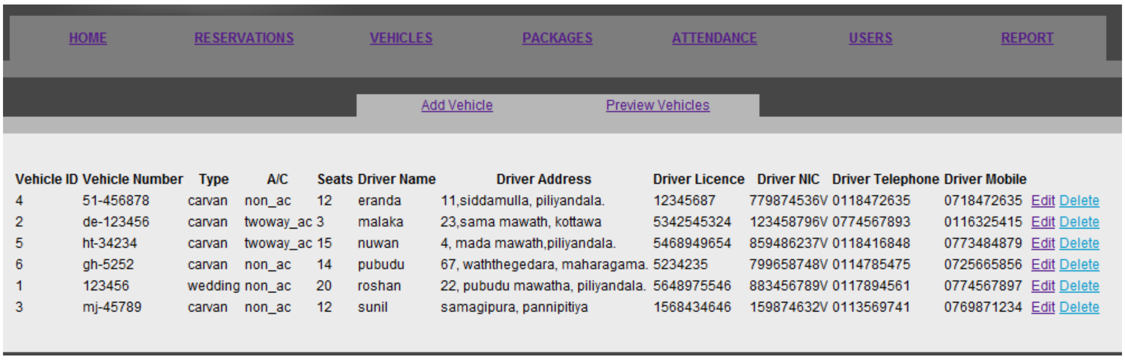


Add Vehicle Form

Vehicle Number:	
Vehicle Type:	Car
Air Condition:	Non A/C
Number of seats:	
Driver Name:	
Driver Address:	
Driver Licence No.:	
Driver NIC No.:	
Driver Telephone:	
Driver Mobile:	
Vehicle Image:	No file chosen (any gif, jpg, png file)

Figure B.7 Add new vehicle form

Using the preview vehicles tab the list of vehicles that are currently registered with the system can be obtained.



Vehicle ID	Vehicle Number	Type	A/C	Seats	Driver Name	Driver Address	Driver Licence	Driver NIC	Driver Telephone	Driver Mobile		
4	51-456878	carvan	non_ac	12	eranda	11,siddamulla, piliyandala.	12345687	779874536V	0118472635	0718472635	Edit	Delete
2	de-123456	carvan	twoway_ac	3	malaka	23,sama mawath, kottawa	5342545324	123458796V	0774567893	0116325415	Edit	Delete
5	ht-34234	carvan	twoway_ac	15	nuwan	4, mada mawath,piliyandala.	5468949654	859486237V	0118416848	0773484879	Edit	Delete
6	gh-5252	carvan	non_ac	14	pubudu	67, waththegedara, maharagama.	5234235	799658748V	0114785475	0725665856	Edit	Delete
1	123456	wedding	non_ac	20	roshan	22, pubudu mawatha, piliyandala.	5648975546	883456789V	0117894561	0774567897	Edit	Delete
3	mj-45789	carvan	non_ac	12	sunil	samagipura, pannipitiya	1568434646	159874632V	0113569741	0769871234	Edit	Delete

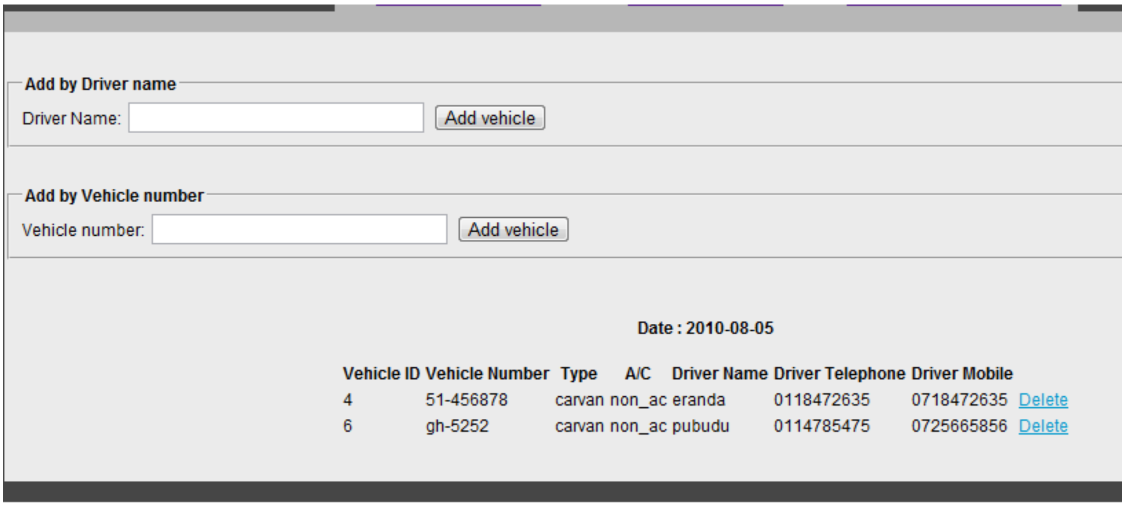
Figure B.8 Preview vehicles

Use edit links in the vehicle preview page to edit each vehicle. Use delete link to remove vehicle from the system.

Attendance tab

Once click on the Attendance tab, it shows three sub tabs called add attendance, view attendance and preview registered vehicles. Only admin users are allowed to access all these tabs.

In order to add an attendance to the system click on the add attendance tab in the attendance tab. Use driver name or vehicle number to add a vehicle to the available vehicle list. To use add by driver name feature, a unique name must be set for the driver's name in each vehicle. The attended vehicle list is shown at the bottom of the add attendance page.



Driver Name:

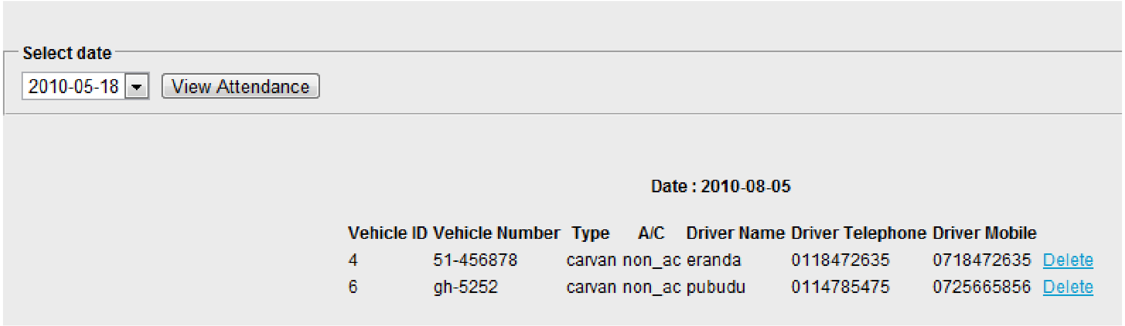
Vehicle number:

Date : 2010-08-05

Vehicle ID	Vehicle Number	Type	A/C	Driver Name	Driver Telephone	Driver Mobile	
4	51-456878	carvan	non_ac	eranda	0118472635	0718472635	Delete
6	gh-5252	carvan	non_ac	pubudu	0114785475	0725665856	Delete

Figure B.9 Add attendance form

In order to check previous attendances, by using preview all attendance tab and selecting the month that is needed to view and pressing enter would by sufficient.



2010-05-18

Date : 2010-08-05

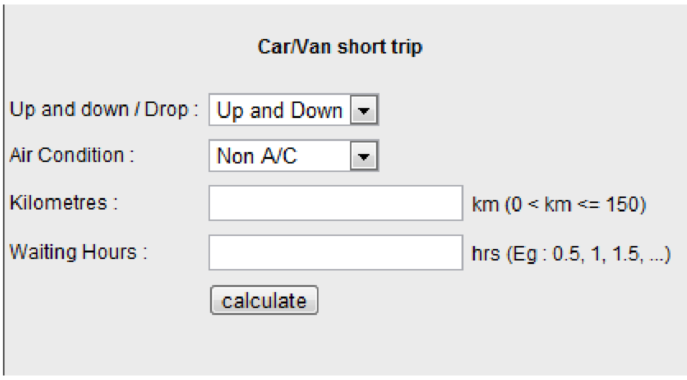
Vehicle ID	Vehicle Number	Type	A/C	Driver Name	Driver Telephone	Driver Mobile	
4	51-456878	carvan	non_ac	eranda	0118472635	0718472635	Delete
6	gh-5252	carvan	non_ac	pubudu	0114785475	0725665856	Delete

Figure B.10 Preview attendance

Package tab

Once click on the Package tab, it shows two sub tabs called price calculators and view packages. Any user can visit these pages.

In order to calculate a reservation price, needs to go to price calculator page and select required vehicle reservation option. Use the empty fields to enter reservation details and press calculate button to get price.



Car/Van short trip

Up and down / Drop : Up and Down

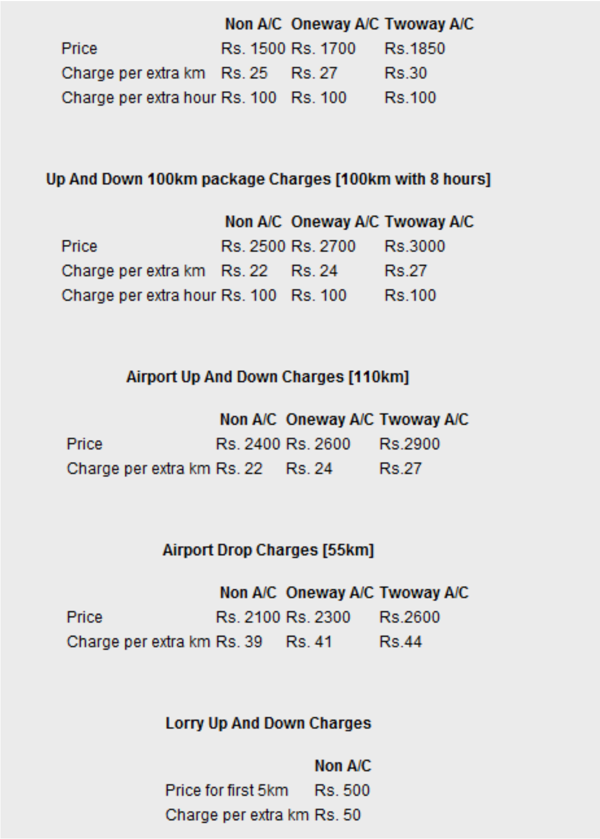
Air Condition : Non A/C

Kilometres : km (0 < km <= 150)

Waiting Hours : hrs (Eg : 0.5, 1, 1.5, ...)

Figure B.11 charges calculator

Use preview package details tab to view list of all reservation option's details.



	Non A/C	Oneway A/C	Twoway A/C
Price	Rs. 1500	Rs. 1700	Rs.1850
Charge per extra km	Rs. 25	Rs. 27	Rs.30
Charge per extra hour	Rs. 100	Rs. 100	Rs.100

Up And Down 100km package Charges [100km with 8 hours]

	Non A/C	Oneway A/C	Twoway A/C
Price	Rs. 2500	Rs. 2700	Rs.3000
Charge per extra km	Rs. 22	Rs. 24	Rs.27
Charge per extra hour	Rs. 100	Rs. 100	Rs.100

Airport Up And Down Charges [110km]

	Non A/C	Oneway A/C	Twoway A/C
Price	Rs. 2400	Rs. 2600	Rs.2900
Charge per extra km	Rs. 22	Rs. 24	Rs.27

Airport Drop Charges [55km]

	Non A/C	Oneway A/C	Twoway A/C
Price	Rs. 2100	Rs. 2300	Rs.2600
Charge per extra km	Rs. 39	Rs. 41	Rs.44

Lorry Up And Down Charges

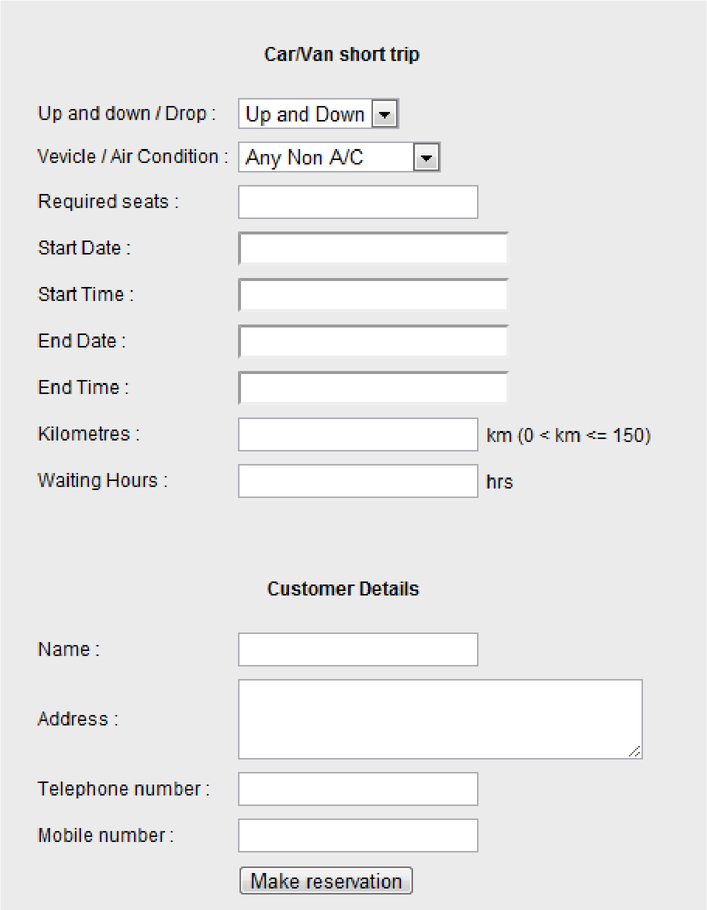
	Non A/C
Price for first 5km	Rs. 500
Charge per extra km	Rs. 50

Figure B.12 Preview reservation charges details

Reservation tab

Once click on the reservation tab, it shows two sub tabs called add reservation and view reservation. Any user can access all these tabs.

In order to add new reservation click on the add reservation tab in the reservation tab. Select required reservation option. Use empty form to enter new reservation details and customer details.



Car/Van short trip

Up and down / Drop : Up and Down

Vevicle / Air Condition : Any Non A/C

Required seats :

Start Date :

Start Time :

End Date :

End Time :

Kilometres : km (0 < km <= 150)

Waiting Hours : hrs

Customer Details

Name :

Address :

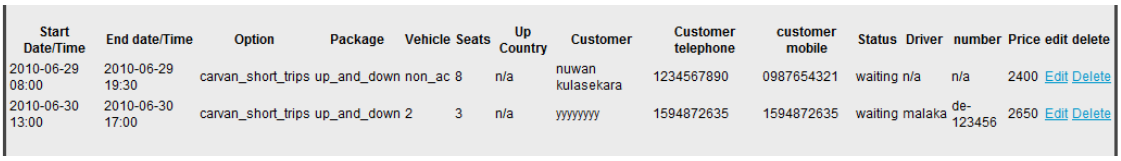
Telephone number :

Mobile number :

Figure B.13 Add reservation form

Adding reservation will not become successful if that entering reservation conflicts with the existing reservations. If the reservation is added successfully, then the system sends a SMS to the driver (if a vehicle was chose for reservation).

In the preview reservation tab it shows the waiting reservations, completing reservations and completed reservations separately.



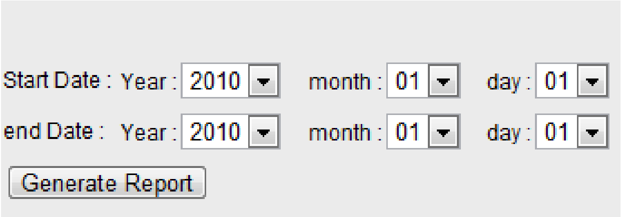
Start Date/Time	End date/Time	Option	Package	Vehicle	Seats	Up Country	Customer	Customer telephone	customer mobile	Status	Driver	number	Price	edit	delete
2010-06-29 08:00	2010-06-29 19:30	carvan_short_trips	up_and_down	non_ac	8	n/a	nuwan kulasekara	1234567890	0987654321	waiting	n/a	n/a	2400	Edit	Delete
2010-06-30 13:00	2010-06-30 17:00	carvan_short_trips	up_and_down	2	3	n/a	yyyyyyy	1594872635	1594872635	waiting	malaka	de-123456	2650	Edit	Delete

Figure B.14 Preview reservations

Use edit and delete links to edit or delete reservation.

Report tab

Once click on the report tab, it shows eight sub tabs called user, vehicle, attendance, package, reservation, total reservation chart, compare reservation option chart and compare reservation option pie chart. Each tab generates PDF formatted reports. Starting and ending dates are required for attendance, reservation and other three chart reports.



Start Date : Year : 2010 month : 01 day : 01

end Date : Year : 2010 month : 01 day : 01

Figure B.15 Attendance report generate form

Appendix C - Code listing

C.1 Chargers calculating algorithm

Variables;

Short tour up and down/ Long tour up and down = up_and_down

Drop = drop

50km package = package_50

100km package = package_100

Airport up and down package = airport_up_and_down

Airport drop package = airport_drop

Bus = bus

Lorry up and down = lorry_up_and_down

Lorry drop = lorry_drop

Short tour up and down/ Long tour up and down chargers calculating algorithm

```
If package = up_and _down then
        If 0< km<=10 then
                Price = (number of waited ½ hours)*50
                if vehicle_type = non_ac then
                        price = price+450
                if vehicle_type = oneway_ac then
                        price = price+475
                if vehicle_type = twoway_ac then
                        price = price+500
        else if 10<km<=100 then
                price = (number of waited ½ hours -1)*50
                if vehicle_type = non_ac then
                        price = price+25*km
                if vehicle_type = oneway_ac then
                        price = price+27*km
                if vehicle_type = twoway_ac then
                        price = price+30*km
        else if 100<km<=150 then
                price = (number of waited hours -2)*100
                if vehicle_type = non_ac then
                        price = price+22*km
                if vehicle_type = oneway_ac then
                        price = price+24*km
                if vehicle_type = twoway_ac then
                        price = price+27*km
        else if 150<km then
                if ip_country = true then
                        if vehicle_type = non_ac then
                                price = 21*km
                        if vehicle_type = oneway_ac then
                                price = 23*km
                        if vehicle_type = twoway_ac then
                                price = 25*km
                else
                        if vehicle_type = non_ac then
                                price = 20*km
                        if vehicle_type = oneway_ac then
                                price = 22*km
                        if vehicle_type = twoway_ac then
                                price = 24*km
                if km <300 then
                        price = price+(number of waited hours)*100
```

Car/van drop chargers calculating algorithm

```
if package = drop then
        If 0< km<=10 then
                Price = (number of waited ½ hours)*50
                if vehicle_type = non_ac then
                        price = price+440
                if vehicle_type = oneway_ac then
                        price = price+460
                if vehicle_type = twoway_ac then
                        price = price+490
        else if 10<km<=30 then
                price = (number of waited ½ hours)*50
                if vehicle_type = non_ac then
                        price = price+44*km
                if vehicle_type = oneway_ac then
                        price = price+46*km
                if vehicle_type = twoway_ac then
                        price = price+49*km
        else if 30<km then
                price = (number of waited hours)*100
                if up_country = true
                        if vehicle_type = non_ac then
                                price = price+40*km
                        if vehicle_type = oneway_ac then
                                price = price+42*km
                        if vehicle_type = twoway_ac then
                                price = price+45*km
                else
                        if vehicle_type = non_ac then
                                price = price+39*km
                        if vehicle_type = oneway_ac then
                                price = price+41*km
                        if vehicle_type = twoway_ac then
        price = price+44*km
```

50km package chargers calculating algorithm

```
if package = package_50 then
    Price = 0
    if vehicle_type = non_ac then
        price = price+1500
    if vehicle_type = oneway_ac then
        price = price+1700
    if vehicle_type = twoway_ac then
        price = price+1850
    if hours>5 then
        extra_hours = hours-50
        price = price+extra_hours*100
    if 50<km then
        extra_km = km-5
        if vehicle_type = non_ac then
            price = price+extra_km*25
        if vehicle_type = oneway_ac then
            price = price+extra_km*27
        if vehicle_type = twoway_ac then
            price = price+extra_km*30
```

100km package chargers calculating algorithm

```
if package = package_100 then
    Price = 0
    if vehicle_type = non_ac then
        price = price+2500
    if vehicle_type = oneway_ac then
        price = price+2700
    if vehicle_type = twoway_ac then
        price = price+3000
    if hours>8 then
        extra_hours = hours-8
        price = price+extra_hours*100
    if 80<km then
        extra_km = km-80
        if vehicle_type = non_ac then
            price = price+extra_km*22
        if vehicle_type = oneway_ac then
            price = price+extra_km*24
        if vehicle_type = twoway_ac then
            price = price+extra_km*27
```

Airport up and down package calculating algorithm

```
if package = airport_up_and_down then
        Price = 0
        if vehicle_type = non_ac then
                price = price+2400
        if vehicle_type = oneway_ac then
                price = price+2600
        if vehicle_type = twoway_ac then
                price = price+2900
        if 110<km then
                extra_km = km-110
                if vehicle_type = non_ac then
                        price = price+extra_km*22
                if vehicle_type = oneway_ac then
                        price = price+extra_km*24
                if vehicle_type = twoway_ac then
                        price = price+extra_km*27
```

Airport drop package chargers calculating algorithm

```
if package = airport_drop then
        Price = 0
        if vehicle_type = non_ac then
                price = price+2100
        if vehicle_type = oneway_ac then
                price = price+2300
        if vehicle_type = twoway_ac then
                price = price+2600
        if 110<km then
                extra_km = km-110
                if vehicle_type = non_ac then
                        price = price+extra_km*39
                if vehicle_type = oneway_ac then
                        price = price+extra_km*41
                if vehicle_type = twoway_ac then
                        price = price+extra_km*44
```

Bus chargers calculating algorithm

```
if package = bus then
        Price = 500
        Price = price+waiting_hours*100
        if km>5 then
                extra_km = km-5
                price = price+extra_km*50
```

Lorry up nad down package chargers calculating algorithm

```
if package = lorry_up_and_down then
        Price = 500
        if km>5 then
                extra_km = km-5
                price = price+extra_km*50
```

Lorry drop package chargers calculating algorithm

```
if package = lorry_drop then
        Price = 500
        if km>5 then
                extra_km = km-5
                price = price+extra_km*45
```

C.2 Create database, tables statements and database connection

Attendance table:

```
                DROP TABLE IF EXISTS `union`.`attendance`;
                CREATE TABLE  `union`.`attendance` (
                        `date` varchar(45) NOT NULL,
                        `id` varchar(45) NOT NULL,
                        PRIMARY KEY (`date`,`id`) USING BTREE
                        ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Customer table:

```
                DROP TABLE IF EXISTS `union`.`customer`;
                CREATE TABLE  `union`.`customer` (
                        `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
                        `cus_name` varchar(45) NOT NULL,
                        `cus_address` varchar(60) NOT NULL,
                        `cus_tel` varchar(45) NOT NULL,
                        `cus_mobile` varchar(45) NOT NULL,
                        PRIMARY KEY (`id`)
                        )     ENGINE=InnoDB     AUTO_INCREMENT=19     DEFAULT
                CHARSET=latin1;
```

Reservation table:

```
DROP TABLE IF EXISTS `union`.`reservation`;
CREATE TABLE  `union`.`reservation` (
        `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
        `option` varchar(45) NOT NULL,
        `package` varchar(45) NOT NULL,
        `vehicle` varchar(45) NOT NULL,
        `seats` varchar(45) NOT NULL,
        `up_country` varchar(45) NOT NULL,
         `start_date` varchar(45) NOT NULL,
        `start_time` varchar(45) NOT NULL,
        `end_date` varchar(45) NOT NULL,
        `end_time` varchar(45) NOT NULL,
        `km` varchar(45) NOT NULL,
        `hrs` varchar(45) NOT NULL,
        `waiting_hrs` varchar(45) NOT NULL,
        `cus_id` varchar(45) NOT NULL,
        `status` varchar(45) NOT NULL,
        PRIMARY KEY (`id`)
        ) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT
CHARSET=latin1;
```

Reservation_req table:

```
DROP TABLE IF EXISTS `union`.`reservation_req`;
CREATE TABLE  `union`.`reservation_req` (
         `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
        `option` varchar(45) NOT NULL,
        `package` varchar(45) NOT NULL,
        `vehicle` varchar(45) NOT NULL,
        `seats` varchar(45) NOT NULL,
        `up_country` varchar(45) NOT NULL,
        `start_date` varchar(45) NOT NULL,
         `start_time` varchar(45) NOT NULL,
        `end_date` varchar(45) NOT NULL,
        `end_time` varchar(45) NOT NULL,
        `km` varchar(45) NOT NULL,
             `hrs` varchar(45) NOT NULL,
            `waiting_hrs` varchar(45) NOT NULL,
            `cus_id` varchar(45) NOT NULL,
            `status` varchar(45) NOT NULL,
             PRIMARY KEY (`id`)
            )    ENGINE=InnoDB    AUTO_INCREMENT=19    DEFAULT
      CHARSET=latin1;
```

User table:

```
      DROP TABLE IF EXISTS `union`.`user`;
      CREATE TABLE  `union`.`user` (
             `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
            `name` varchar(45) NOT NULL,
            `username` varchar(45) NOT NULL,
            `password` varchar(45) NOT NULL,
            `address` varchar(60) NOT NULL DEFAULT 'n/a',
            `tel` varchar(10) NOT NULL DEFAULT '0',
            `mobile` varchar(10) NOT NULL DEFAULT '0',
            `nic` varchar(20) NOT NULL,
            `type` varchar(20) NOT NULL,
            PRIMARY KEY (`id`)
            )    ENGINE=InnoDB    AUTO_INCREMENT=11    DEFAULT
      CHARSET=latin1;
```

Vehicle table:

```
      DROP TABLE IF EXISTS `union`.`vehicle`;
      CREATE TABLE  `union`.`vehicle` (
             `id` int(11) NOT NULL AUTO_INCREMENT,
            `number` varchar(15) NOT NULL,
            `type` varchar(20) NOT NULL,
             `ac` varchar(10) NOT NULL,
            `seats` int(11) NOT NULL,
            `driver_name` varchar(50) NOT NULL,
            `driver_address` varchar(50) NOT NULL,
             `driver_licence` varchar(20) NOT NULL,
            `driver_nic` varchar(20) NOT NULL,
          `driver_tel` varchar(15) NOT NULL,
          `driver_mobile` varchar(15) NOT NULL,
          `status` varchar(15) NOT NULL,
           PRIMARY KEY (`id`)
          )    ENGINE=MyISAM    AUTO_INCREMENT=21    DEFAULT
      CHARSET=latin1;
```

Table that keeps reservation chargers details:

```
      DROP TABLE IF EXISTS `union`.`long_tours`;
      CREATE TABLE  `union`.`long_tours` (
           `package_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
           `lower_km` int(10) unsigned NOT NULL,
           `upper_km` int(10) unsigned NOT NULL,
           `pricing_type` varchar(45) NOT NULL,
           `non_ac_price` int(10) unsigned NOT NULL DEFAULT '0',
           `oneway_ac_price` int(10) unsigned NOT NULL DEFAULT '0',
           `twoway_ac_price` int(10) unsigned NOT NULL DEFAULT '0',
           `fixed_hrs` int(10) unsigned NOT NULL,
           `charge_per_extra_hr` int(10) unsigned NOT NULL,
           `up_country` varchar(45) NOT NULL,
           PRIMARY KEY (`package_id`)
          )    ENGINE=InnoDB    AUTO_INCREMENT=5    DEFAULT
          CHARSET=latin1;

      DROP TABLE IF EXISTS `union`.`short_tours`;
      CREATE TABLE  `union`.`short_tours` (
           `package_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
           `package_name` varchar(45) NOT NULL,
           `lower_km` int(10) unsigned NOT NULL,
           `upper_km` int(10) unsigned NOT NULL,
           `pricing_type` varchar(45) NOT NULL,
           `non_ac_price` int(10) unsigned NOT NULL,
           `oneway_ac_price` int(10) unsigned NOT NULL,
           `twoway_ac_price` int(10) unsigned NOT NULL,
           `waiting_hr_per` double NOT NULL,
           `waiting_charge_rate` int(10) unsigned NOT NULL,
           `free_waitings` int(10) unsigned NOT NULL,
           `up_country` varchar(45) NOT NULL,
           PRIMARY KEY (`package_id`)
          ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;

      DROP TABLE IF EXISTS `union`.`special_packages`;
      CREATE TABLE  `union`.`special_packages` (
           `package_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
           `package_name` varchar(45) NOT NULL,
           `fixed_hrs` int(10) unsigned NOT NULL,
           `fixed_km` int(10) unsigned NOT NULL,
           `non_ac_price` int(10) unsigned NOT NULL DEFAULT '0',
           `oneway_ac_price` int(10) unsigned NOT NULL DEFAULT '0',
           `twoway_ac_price` int(10) unsigned NOT NULL DEFAULT '0',
           `extra_hr_charge` int(10) unsigned NOT NULL,
           `non_ac_km_charge` int(10) unsigned NOT NULL,
           `oneway_ac_km_charge` int(10) unsigned NOT NULL,
           `twoway_ac_km_charge` int(10) unsigned NOT NULL,
           PRIMARY KEY (`package_id`)
          ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;
```

By using object oriented singleton pattern, the database connection was designed. Following pseudocode explains the scenario;

```
Class Database_connecting {
    Private Static String host;  Private Static String username;
    Private Static string password;  Private static string database;
    Private Static db_connection;
    Private static function  db_connection(){
        If( is_not_set(db_connection)){
            This . Db_connection = connect_db(host,username,password);
}       }       }
```

Following coding shows the conversion of designed database connection function;

```
class db_conn{
    private static $db_connection;
    private static function connect_db(){
        if(!isset(self::$db_connection)){
            self::$db_connection = mysql_connect("localhost","root","") or die("Could not connect ".mysql_error());
        }
        mysql_select_db("union",self::$db_connection) or die("Could not select db ".mysql_error());
    }

    private static function run_query($sql){
        self::connect_db();
        $result = mysql_query($sql) or die ("Query failed: " . mysql_error() . " Actual query: " . $sql);
        return $result;
    }
}
```

Several functions are implemented to return data as object arrays. Following coding shows the developed functions for return data;

```
    private static function run_query($sql){
        self::connect_db();
        $result = mysql_query($sql) or die ("Query failed: " . mysql_error() . " Actual query: " . $sql);
        return $result;
    }

    public static function send_query($sql){
        $result = self::run_query($sql);
        return $result;
    }
```

```
public static function get_query($sql){
        $result = self::run_query($sql);
        return $result;
}

public static function get_data($sql){
        $result = self::run_query($sql);
        if(!(mysql_num_rows($result)>0)){
                $array = array();
                $array[0] = "error";
                return $array;
        }else{
                $array = array();
                while($row = mysql_fetch_object($result)){
                        array_push($array,$row);
                }
                return $array;
        }
}
```

A function was developed to prevent SQL injection attacks to protect the data of the Union Cabs system. Following coding shows the developed function to prevent SQL injection attacks.

```
public static function filter($text){
        self::connect_db();
        $output = mysql_real_escape_string($text);
        return $output;
}
```

C.3 Security functions

Code for check the system runs on JavaScript enabled browser;

```
<body>
<noscript>
        <meta http-equiv="refresh" content="0;URL=enable_js.php" />
</noscript>

        <h2 id="nojs" style="font-family:Verdana, Arial, Helvetica, sans-serif; color:#CC0000;">JavaScript is turned off in your web browser. Turn it ON (Enable) to access the system, then refresh the page.</h2>

        <script>
                document.getElementById("nojs").style.display="none";
                alert("Javascript enabled successfully! Press ok to continue");
                window.location = "index.php";
        </script>

</body>
```

Login handling code

```
<?php
session_start();
require_once("protect_unpost.php");
require_once("db_conn.php");
require_once("common.php");

if(isset($_POST["login"])){
//for protect from sql injection
        $un = db_conn::filter($_POST["username"]);
        $pw = db_conn::filter($_POST["password"]);
        $pass = sha1($pw);
```

```
	$sql = "SELECT * FROM union.user WHERE username='$un' AND
password='$pass';";
	$result = db_conn::get_query($sql);
	$num_rows = db_conn::get_num_rows($result);
	$result1 = db_conn::get_data($sql);
	$result2 = $result1[0];
	if($num_rows>0){
		$_SESSION["user_id"] = $result2->id;
		$_SESSION["user_type"] = $result2->type;
		$_SESSION["user_name"] = $result2->name;
		$_SESSION["login"] = true;
		common::redirect("?path=dashboard");
	}else{
		$_SESSION["error"] = "Wrong Username or Password!";
		common::redirect("?path=login");
	}
}else{
	common::redirect("?path=login");
}
?>
```

Logout code

```
<?php
session_start();
require_once("common.php");
session_destroy();
common::redirect("?path=login");
?>
```

Common class code

```
<?php
require_once("protect_page.php");
```

```
class common{
#user for redicet to a nother page (eg:?path=filname&subpath=subfilename)
        public static function redirect($path){
                if (!headers_sent()) {
                header('Location: /project/'.$path);
                exit;
                }else {
                echo "Headers already sent. Cannot redirect, for now please click this <a
href=\"/project/".$path."\">link</a> instead\n";
                        exit;
                }
        }
}
?>
```

Add user JavaScript validation code

```
function validate() {

     if(document.add_user_form.name.value==''){
        alert("Enter user's full name!");
        document.add_user_form.name.focus();
        return false;
     }
     if(document.add_user_form.username.value==''){
        alert("Enter user's username!");
        document.add_user_form.username.focus();
        return false;
     }
     if(document.add_user_form.password.value==''){
        alert("Enter user's password!");
        document.add_user_form.password.focus();
        return false;
     }
     if(document.add_user_form.confpassword.value==''){
```

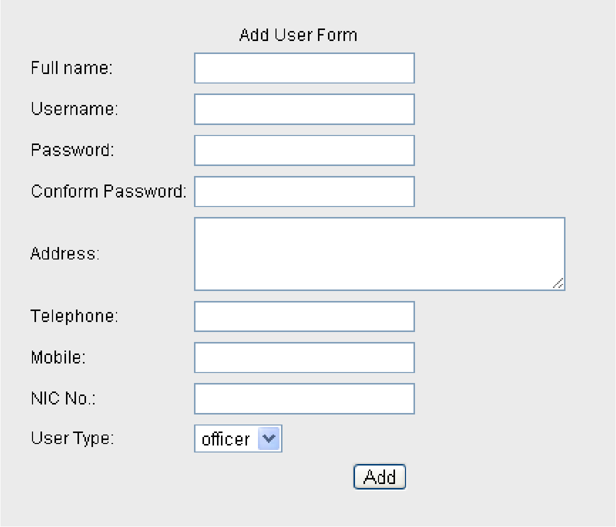
```
    alert("Enter user's conform password!");
    document.add_user_form.confpassword.focus();
    return false;
 }
 if(document.add_user_form.password.value!=
    document.add_user_form.confpassword.value){

    alert("Password and conform password are not matching!");
    document.add_user_form.password.focus();
    return false;
 }
 if(document.add_user_form.address.value==''){
    alert("Enter user's address!");
    document.add_user_form.address.focus();
    return false;
 }
 if(document.add_user_form.tel.value==''){
    alert("Enter user's telephone number!");
    document.add_user_form.tel.focus();
    return false;
 }
if(isNaN(document.add_user_form.tel.value) ||
                (document.add_user_form.tel.value<=0) ||
                (document.add_user_form.tel.value.length!=10)){
    alert("Invalid telephone number! (eg: 0112700800)");
    document.add_user_form.tel.focus();
    return false;
}
 if(document.add_user_form.mobile.value==''){
    alert("Enter user's mobile number!");
    document.add_user_form.mobile.focus();
    return false;
 }
 if(isNaN(document.add_user_form.mobile.value) ||
                (document.add_user_form.mobile.value<=0) ||
                (document.add_user_form.mobile.value.length!=10)){
    alert("Invalid mobile number! (eg: 0123456789)");
```

```
            document.add_user_form.mobile.focus();
            return false;
        }
        if(document.add_user_form.nic.value=="){
            alert("Enter user's NIC number (national identy card number)!");
            document.add_user_form.nic.focus();
            return false;
        }
if((document.add_user_form.nic.value.length!=10)||
    (isNaN(document.add_user_form.nic.value.substr(0,9)))||
    (document.add_user_form.nic.value.substr(9)!="V")){
            alert("Invalid NIC number! (eg: 123456789V)");
            document.add_user_form.nic.focus();
            return false;
                        }
}
```

Appendix D – UI implementations

Add user form:



Add User Form

Full name:

Username:

Password:

Conform Password:

Address:

Telephone:

Mobile:

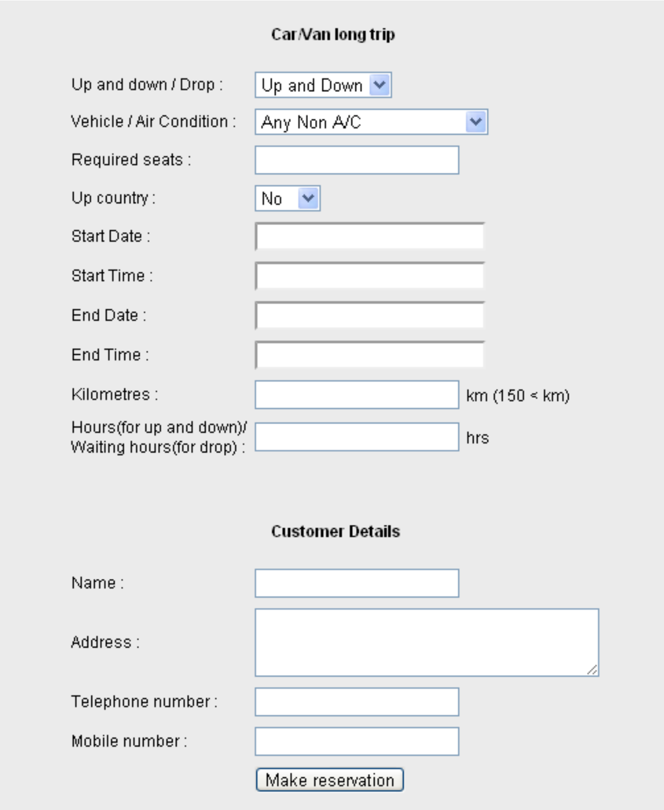
NIC No.:

User Type: officer

Figure D.1 User add form

Add reservation form:

There are seven different add reservation forms for each reservation option. All add reservation forms are similar to following form.



Car/Van long trip

Up and down / Drop : Up and Down

Vehicle / Air Condition : Any Non A/C

Required seats :

Up country : No

Start Date :

Start Time :

End Date :

End Time :

Kilometres : km (150 < km)

Hours(for up and down)/ Waiting hours(for drop) : hrs

Customer Details

Name :

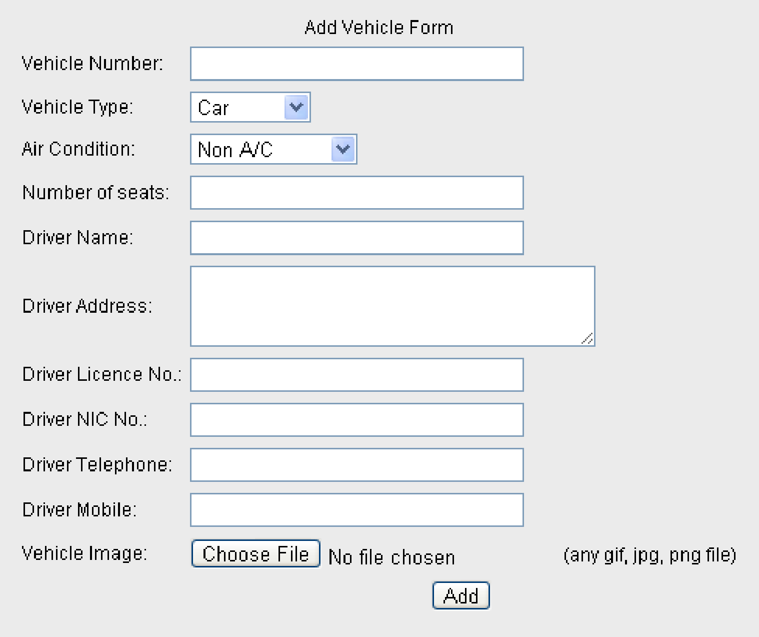
Address :

Telephone number :

Mobile number :

Figure D.2 Sample for add reservation form

Add vehicle form:



Add Vehicle Form

Vehicle Number:

Vehicle Type: Car

Air Condition: Non A/C

Number of seats:

Driver Name:

Driver Address:

Driver Licence No.:

Driver NIC No.:

Driver Telephone:

Driver Mobile:

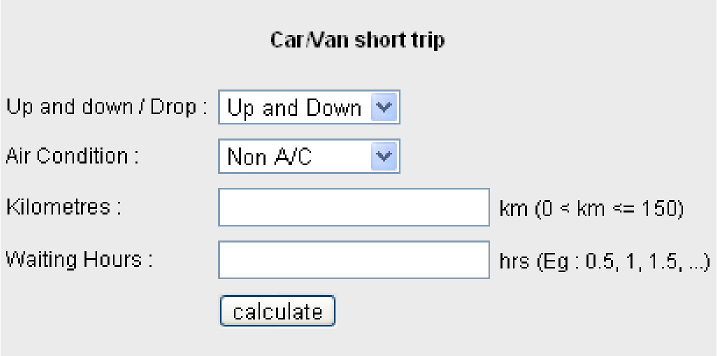
Vehicle Image: Choose File No file chosen (any gif, jpg, png file)

Add

Figure D.3 Add vehicle form

Chargers calculator form:

There are seven different chargers calculators for each reservation option. All are similar to each other as follows.



Car/Van short trip

Up and down / Drop : Up and Down

Air Condition : Non A/C

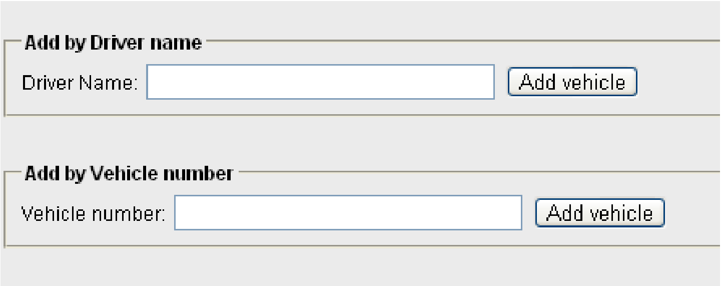
Kilometres : km (0 < km <= 150)

Waiting Hours : hrs (Eg : 0.5, 1, 1.5, ...)

calculate

Figure D.4 Sample charge calculator form

Add attendance form:



Add by Driver name

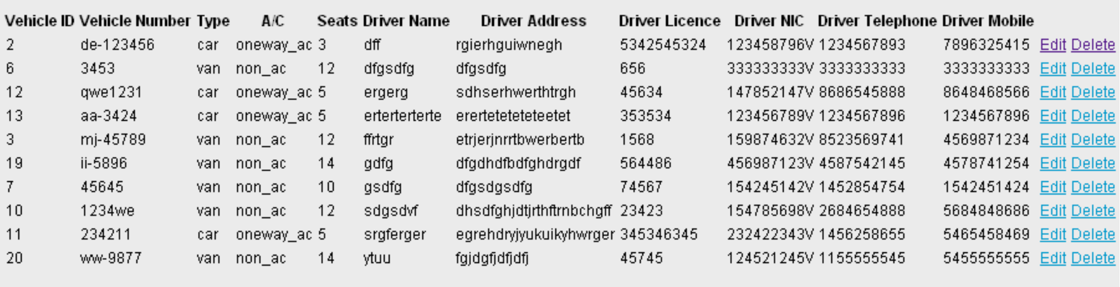
Driver Name: Add vehicle

Add by Vehicle number

Vehicle number: Add vehicle

Figure D.5 Add attendance form

For table data previews was used similar design. Following image shows the data preview design;

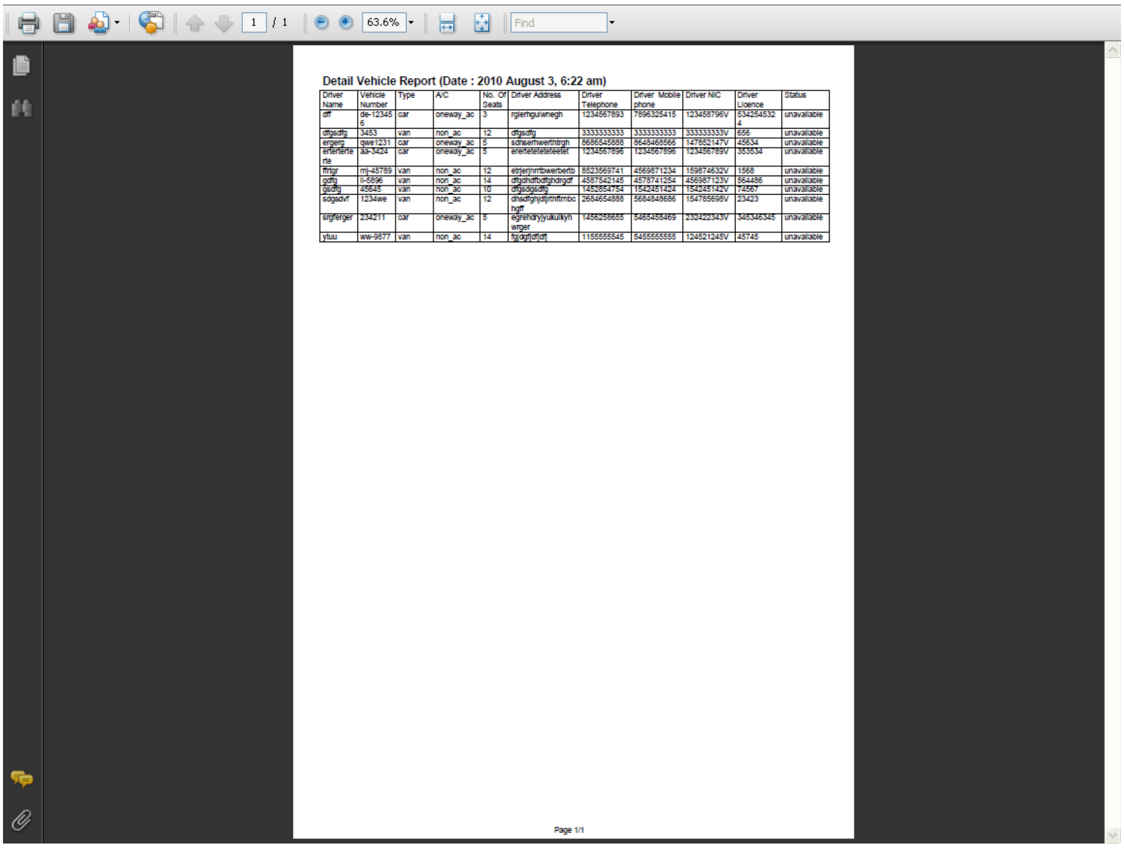


Vehicle ID	Vehicle Number	Type	A/C	Seats	Driver Name	Driver Address	Driver Licence	Driver NIC	Driver Telephone	Driver Mobile	
2	de-123456	car	oneway_ac	3	dff	rgierhguiwnegh	5342545324	123458796V	1234567893	7896325415	Edit Delete
6	3453	van	non_ac	12	dfgsdfg	dfgsdfg	656	333333333V	3333333333	3333333333	Edit Delete
12	qwe1231	car	oneway_ac	5	ergerg	sdhserhwerthtrgh	45634	147852147V	8686545888	8648468566	Edit Delete
13	aa-3424	car	oneway_ac	5	erterterterte	erertetetetetetet	353534	123456789V	1234567896	1234567896	Edit Delete
3	mj-45789	van	non_ac	12	ffrtgr	etrjerjnrrtbwerbertb	1568	159874632V	8523569741	4569871234	Edit Delete
19	ii-5896	van	non_ac	14	gdfg	dfgdhdfbdfghdrgdf	564486	456987123V	4587542145	4578741254	Edit Delete
7	45645	van	non_ac	10	gsdfg	dfgsdgsdfg	74567	154245142V	1452854754	1542451424	Edit Delete
10	1234we	van	non_ac	12	sdgsdvf	dhsdfghjdtjrthftrnbchgff	23423	154785698V	2684654888	5684848686	Edit Delete
11	234211	car	oneway_ac	5	srgferger	egrehdryjyukuikyhwrger	345346345	232422343V	1456258655	5465458469	Edit Delete
20	ww-9877	van	non_ac	14	ytuu	fgjdgfjdfjdfj	45745	124521245V	1155555545	5455555555	Edit Delete

Figure D.6 Sample data preview table

Appendix E - Management Reports

Vehicle detail report



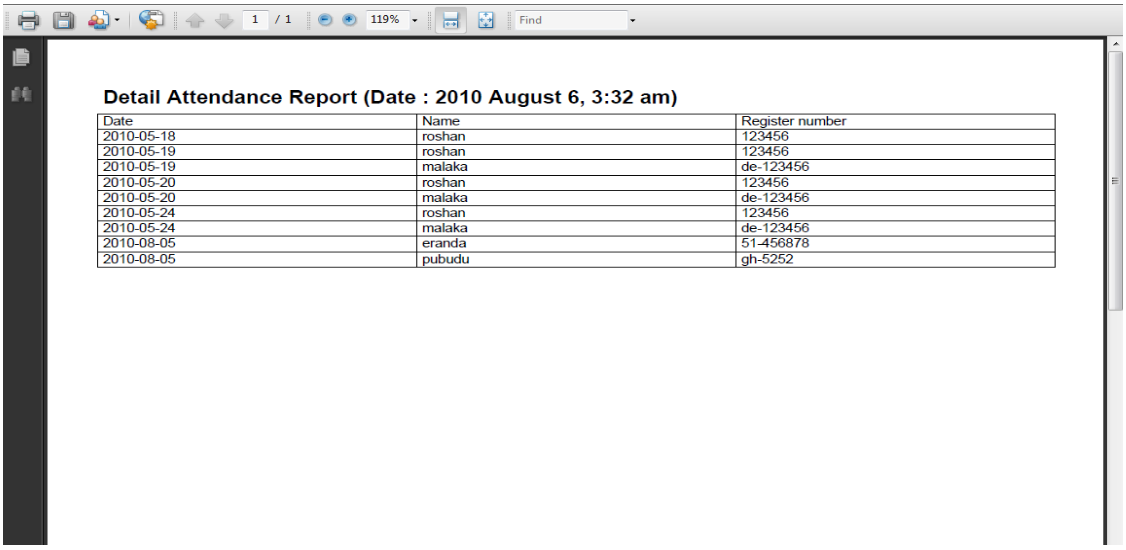
Detail Vehicle Report (Date : 2010 August 3, 6:22 am)

Driver Name	Vehicle Number	Type	A/C	No. Of Seats	Driver Address	Driver Telephone	Driver Mobile phone	Driver NIC	Driver Licence	Status
dff	de-123456	car	oneway_ac	3	rglerhgulwnegh	1234567893	7896325415	123458796V	5342545324	unavailable
dfgsdfg	3453	van	non_ac	12	dfgsdfg	3333333333	3333333333	333333333V	656	unavailable
ergerg	qwe1231	car	oneway_ac	5	sdhserhwerthtrgh	8686545888	8648468566	147852147V	45634	unavailable
erferferterte	aa-3424	car	oneway_ac	5	erertetetetteteetet	1234567896	1234567896	123456789V	353534	unavailable
ffrtgr	mj-45789	van	non_ac	12	etrjerjnrrtbwerbertb	8523569741	4569871234	159874632V	1568	unavailable
gdfg	ll-5896	van	non_ac	14	dfgdhdfbdfghdrgdf	4587542145	4578741254	456987123V	564486	unavailable
gsdfg	45645	van	non_ac	10	dfgsdgsdfg	1452854754	1542451424	154245142V	74567	unavailable
sdgsdvf	1234we	van	non_ac	12	dhsdfghjdtjrthftmbchgff	2684654888	5684848686	154785698V	23423	unavailable
srgferger	234211	car	oneway_ac	5	egrehdryjyukulkyhwrger	1456258655	5465458469	232422343V	345346345	unavailable
ytuu	ww-9877	van	non_ac	14	fgjdgfjdfjdfj	1155555545	5455555555	124521245V	45745	unavailable

Page 1/1

Figure E.1 Vehicle detail report

Attendance detail report



Detail Attendance Report (Date : 2010 August 6, 3:32 am)

Date	Name	Register number
2010-05-18	roshan	123456
2010-05-19	roshan	123456
2010-05-19	malaka	de-123456
2010-05-20	roshan	123456
2010-05-20	malaka	de-123456
2010-05-24	roshan	123456
2010-05-24	malaka	de-123456
2010-08-05	eranda	51-456878
2010-08-05	pubudu	gh-5252

Figure E.2 Attendance detail report

Total reservation comparison chart

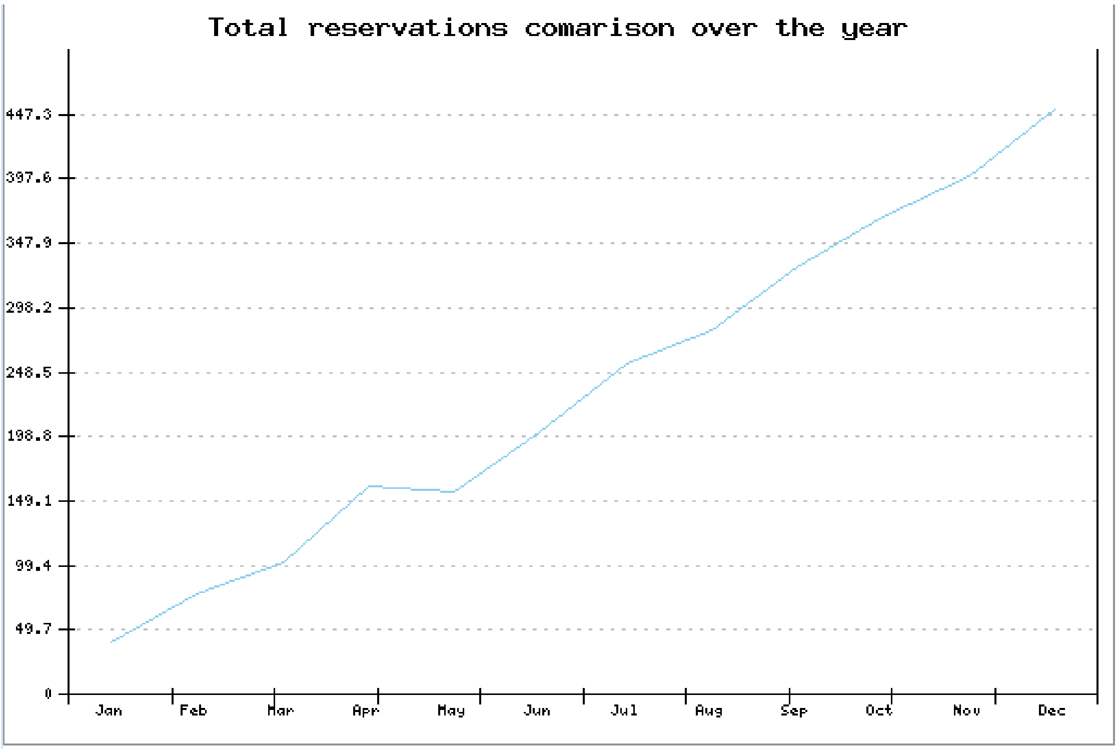


Figure E.3 Total reservation comparison chart

Reservation option comparison chart (over the time)

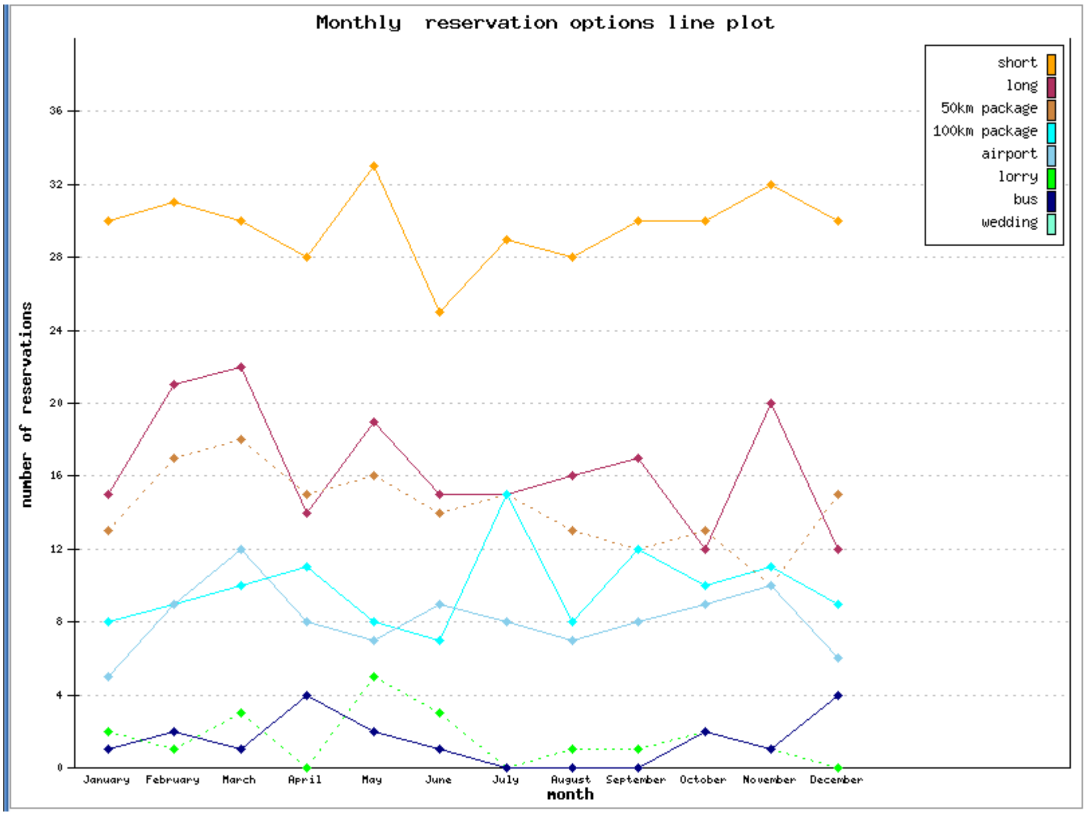


Figure E.4 Reservation option comparison chart (over the time)

Compares number of reservation between reservation options

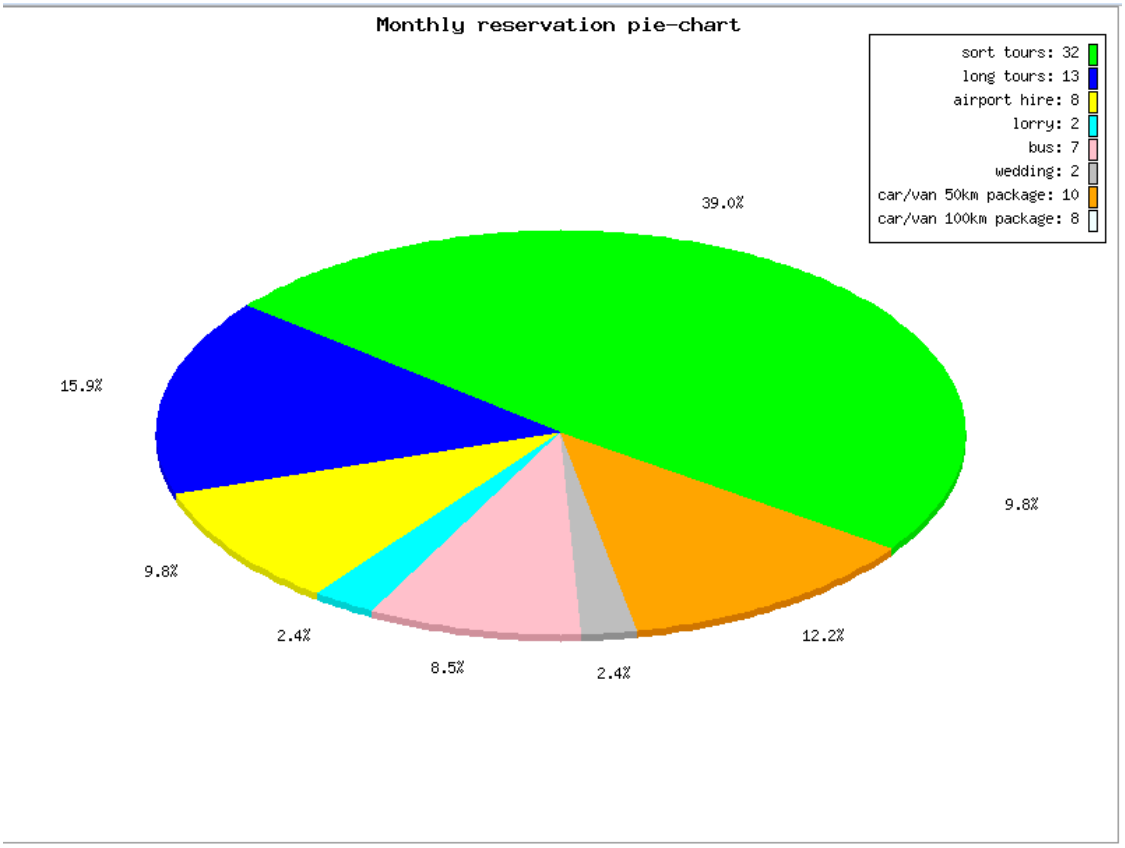


Figure E.5 Compares number of reservation between reservation options

UNION CABS
2 700800
Hiring Motor cars/ Vans/ Lorries
No. 05, Siddamulla, Piliyandala
Tel: 2 700800
Fax: 2 700907
Mobile: 0777537526
E-mail: unioncabs@sltnet.lk

5 August 2010

Project Coordinator,

University of Colombo School of Computing,

Dear Sir,

<u>Approving the Web based Vehicle Reservation Information Management System and Online Vehicle Vehicle Reservation System</u>

I'm glad to say that the above solution presented by Mr. Yasiru Gaminda Welandawe was accepted by the company, and he has successfully completed the solution which addresses all the requirements.

This letter was issued in response to a request made by him.

Thank You.

Yours faithfully,

……………………………………

Mr. Ravishan Gamage
Proprietor,
Union Cabs.