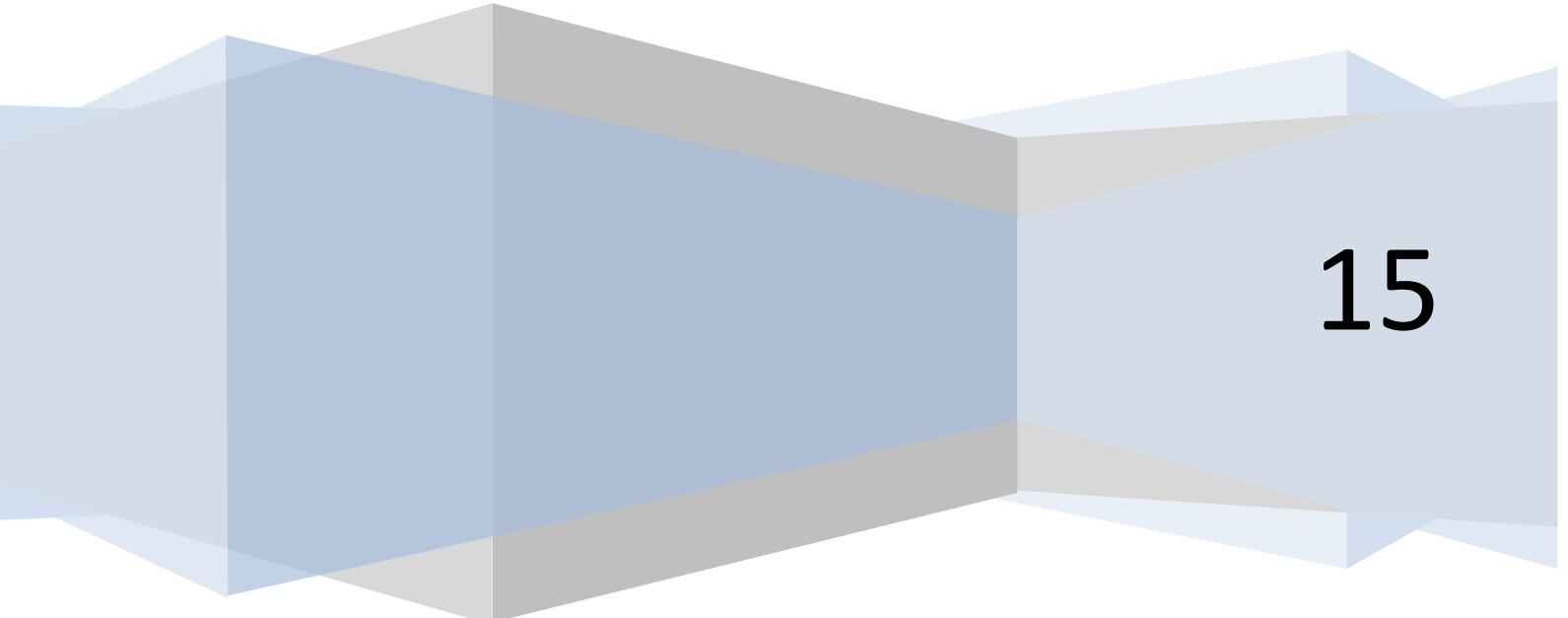
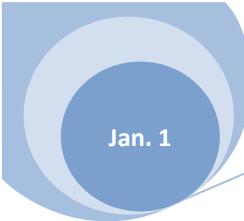


Travel Agency Management System

Priontu Chowdhury



15



Jan. 1

Travel Agency Management System

Travel Agency Management System

By Priorntu Chowdhury

Contents:

1. Definition, investigation and analysis: page

(i) Definition- nature of the problem -----	5
(ii) Investigation and analysis -----	14

2. Design:

(i) Nature of the solution-----	31
(ii) Intended benefits -----	79
(iii) Limits of the scope of the solution -----	80

3. Software development, testing and implementation:

(i) Development and Programming -----	83
(ii) Testing -----	190
(iii) Installation Instructions-----	340

4. Documentation:

(i) System maintenance documentation -----	345
(ii) User Documentation-----	465

5. Evaluation:

(i) Discussion of the degree of success in meeting the original objectives-----	499
(ii) Evaluation of the client's and user's response to the system-----	501
(iii) Desirable Extensions -----	503

(a)

**DEFINITION,
INVESTIGATION AND
- ANALYSIS**

(i) Definition – nature of the problem

❖ **Description of the organization**

Chittagong is a major port city with a lot of exciting attractions for the huge amount of tourists that come to visit every year. Being the major port city of Bangladesh, Chittagong's trade ships are vital to the economy of the country. Chittagong also holds one of the wonders of the world, Cox's Bazar. Therefore a lot of people visit Chittagong from many countries around the globe, and many people from Chittagong also travel around the world.

MAAS Travels and Tours was opened by a group of brothers at 2007. Their office is situated at R M Centre (5th floor), 53, Nasirabad/A, Chittagong. The MAAS travels and tours is a private retailer that provides travel and tourism related services. They also have a separate department devoted to making travel arrangements for business travelers. There are two main functions of the agency, the first being they offer fixed packages for travel around the globe, and the second is they provide vehicles for rentals for tourist or people who are in temporary need for transport.

❖ **How the -current system works**

The overall day functioning of the travel agency is overseen by the Chief Executive Officer (CEO) who makes sure the whole system is working properly. The chief executive officer takes care of both departments:

1. Holiday Packages
2. Vehicle rentals,

The main correspondent on behalf of the Travel agency management was the CEO, Mr. Sheikh, who took me through the daily procedure of the agency.

The customers have complete freedom of choosing between holiday packages, vehicle rentals or both. There are no restraints on behalf of the travel agency except the provision of fixed packages due to a cost plan determined by the agency for each package. The packages need to be booked in advance. Vehicles can be booked instantly or in advance and the booking is confirmed on cash payment. My project comprises of both sectors of the travel agency.

As I visited their office to examine how the activities are carried out and the problems associated with them, I divided the whole process into the following steps (top-down design) shown below. Most of the following details were found out throughout the analysis stage:

- There are several counters at the office. The customers, when they arrive at the counters, are asked if they are going to book the packages or vehicle rentals or both. According to their response they are provided with catalogs that contain all the options and facilities provided to the customers. The package catalogs contain all the package outlines and the vehicle reservation catalogs provide all the information about the vehicles provided by the travel agency. The customers are then given time to reflect on their options.
- After the customer is done thinking, he/she is asked for the option chosen.
- If the customer chooses a Package, the officer checks his records for the particular package to see if there are any vacancies. If any vacancies are present, the officer takes a form and fills the "Package information" section of the form and provides the form to the customer and asks them to fill the "Customer information" section of the form and provide the number of people for whom the Package is being booked. Then the customer calculates the net total and fills the rest of the form. The form is then added to a folder which contains all the Package Booking forms. Then a receipt is prepared for the customer and it is provided to the customer on instant cash payment. This receipt needs to be shown for access on the day of the journey. The receipt acts as an evidence of payment.
- After a booking, the officer is to add the record to the Customer list of Package bookings and assign a serial number to it. The serial number acts as vacancy check information i.e. if the serial number reaches the number of reservations made, it means there are no more vacancies left. The Customer list of Package Bookings is revised by the CEO periodically.
- If the customer chooses to book a vehicle, the officer first checks the records to see if the vehicle is available or if it is booked. If the vehicle is not available, the officer notifies the customer. If the vehicle is available, the officer fills up the "Vehicle information" section of the form and provides the form to the customer to fill up the "Customer information" section. Then the officer takes back the form and checks if the customer information provided is correct. The agency charges the customers on the basis of rate per kilometer. It also charges a late fine for every extra day the vehicle is kept by the customer, beyond the date specified.
- The officer prepares a "Token" using the information on the booking form. An advance cash payment is requested from the customer. The token is provided to the customer on payment. The token needs to be shown by the customer when receiving the vehicle. It acts as an evidence of an advance payment being made. The form is then added to a folder containing all the Vehicle Booking forms.
- After the booking the customer is to add the record of the booking made to the Customer list of Vehicle Bookings. This list is prepared at each counter. These lists are revised periodically by the CEO of MAAS Travels and Tours.
- On finishing a job, the customer needs to return the vehicle to the agency and collect a bill after making full payment for the renting the vehicle.
- All kinds of payments can only be made in cash and must be made on spot.
- When anyone comes to the agency seeking for jobs, they must consult at the counter at first. From here they are sent to the office of the Chief Executive Officer (CEO) where he

interviews the person seeking the job. If the person gets the job, they must report back to the counter to obtain an Employee entry form. After the form is filled up, the record of the employee is added to the Employee Details list, which contains the record of all the employees in a table and is revised by the CEO periodically.

- When Package expenses are to be added to the “Package expenses list” and to the “List of total Package Expenses”, they are done at the counters. The records are collected by the officers over a period of time and the “Package Expenses List” and “List of total Package Expenses” prepared by the officers. The lists are revised by the CEO periodically.
- When a new package is to be added or existing package information to be updated or a package is to be deleted, the information is sent to the counter by the CEO. The message is sent to all the counters at the same time on these occasions. When a package is to be added, the officer is supposed to add the package information to a list of Packages, containing all the information regarding the packages, including number of reservations, from where the officer compares the data. The catalogs are updated shortly after. The list is revised periodically by the CEO.
- When a new vehicle is bought by the agency, the vehicle information is sent to the counters. The officers are then supposed to add the vehicle information to the list of Vehicles. The vehicle information may be updated or deleted manually from the list as well. The List of Vehicles is revised by the CEO periodically.
- Fuel record data are collected periodically at the counters. A Fuel record report is made by the officers at each counter. The fuel record data from each counter are merged periodically and a final draft is sent to the CEO for revision.

❖ Origins and forms of data

	Field	Data Type	Origin of Data and Description
1.	Package type	String	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry
2.	Package name	String	Found in- 1.Package Entry 2.Package Expenses 3.Package Booking 4.Package Booking Updates
3.	Package ID	String: Only alphabetical and numerical data accepted	Found in- 1.Package Entry 2.Package Expenses 3.Package Booking 4.Package Booking Updates
4.	Cost per person	Currency	Found in- 1.Package Entry 2.Package Booking 3.Package Booking Updates

Travel Agency Management System

5.	Number of people	Integer	Found in- 1.Package Booking 2.Package Booking Updates
6.	Net total	Currency	Found in- 1.Package Booking 2.Package Booking Update 3.Vehicle Billing
7.	Itinerary	String	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry
8.	Package Details	String	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry
9.	Accommodation type	String	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry
10.	Journey Date	Date/Time	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry 4.Vehicle Billing 5.Vehicle Booking 6.Vehicle Booking Update
11.	Arrival time	Date/Time	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry
12.	Starting time	Date/Time	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry
13.	Duration	Integer	Found in- 1.Package Booking 2.Package Booking Update 3.Package Entry

Travel Agency Management System

14.	Booking ID	String: Only alphabetical and numerical data accepted.	Found in- 1.Package Booking 2.Package Booking Update 3.Vehicle Billing 4.Vehicle Booking 5.Vehicle Booking Update
15.	Customer Name	String: Only alphabetical data accepted	Found in- 1.Package Booking 2.Package Booking Update 3.Vehicle Billing 4.Vehicle Booking 5.Vehicle Booking Update
16.	Address	String	Found in- 1.Package Booking 2.Package Booking Update 3.Vehicle Billing 4.Vehicle Booking 5.Vehicle Booking Update 6.Employee Entry
17.	Contact number	String: Only numerical data accepted	Found in- 1.Package Booking 2.Package Booking Update 3.Vehicle Billing 4.Vehicle Booking 5.Vehicle Booking Update 6.Employee Entry
18.	Date	Date/Time	Found in- All forms
19.	Date booking made	Date/Time	Found in- 1.Package Booking Update 2.Vehicle Billing 3.Vehicle Booking Update
20.	Amount	Currency	Found in- 1.Package Expenses 2.Vehicle Servicing
21.	User type	String	Found in- 1.User Creation 2.User Log-in

Travel Agency Management System

22.	User ID	String: Only alphabetical and numerical data accepted.	Found in- 1.User Creation 2.User Log-in
23.	Password	String: Only alphabetical and numerical data accepted.	Found in- 1.User Creation 2.User Log-in
24.	Confirm Password	String: Only alphabetical and numerical data accepted.	Found in- 1.User Creation 2.User Log-in
25.	Vehicle type	String: Only alphabetical data accepted.	Found in- 1.Vehicle Entry 2.Vehicle Update 3.Vehicle Booking 4.Vehicle Booking Update 5.Vehicle Servicing 6.Vehicle Billing
26.	Make	String: Only alphabetical data accepted.	Found in- 1.Vehicle Entry 2.Vehicle Update 3.Vehicle Booking 4.Vehicle Booking Update 5.Vehicle Billing
27.	Model	String	Found in- 1.Vehicle Entry 2.Vehicle Update 3.Vehicle Booking 4.Vehicle Booking Update 5.Vehicle Billing
28.	Vehicle ID	String: Only alphabetical and numerical data accepted.	Found in- 1.Vehicle Entry 2.Vehicle Update 3.Vehicle Booking 4.Vehicle Booking Update 5.Vehicle Servicing 6.Vehicle Billing

Travel Agency Management System

29.	Registration Number	String: Only alphabetical, numerical data and “-” accepted.	Found in- 1.Vehicle Entry 2.Vehicle Update 3.Vehicle Booking 4.Vehicle Booking Update 5.Vehicle Billing
30.	Driver name	String: Only alphabetical data accepted.	Found in- 1.Vehicle Booking 2.Vehicle Booking Update 3.Vehicle Billing
31.	Number of seats	Integer	Found in- 1.Vehicle Entry 2.Vehicle Update 3.Vehicle Booking 4.Vehicle Booking Update 5.Vehicle Billing 6.Vehicle Servicing
32.	Starting kilometer-	Long	Found in- 1.Vehicle Booking 2.Vehicle Booking Update 3.Vehicle Billing
33.	Ending kilometer	Long	Found in- 1.Vehicle Billing
34.	Kilometers travelled	Integer	Found in- 1.Vehicle Billing
35.	Rate per kilometer	Currency	Found in- 1.Vehicle Booking 2.Vehicle Booking Update 3.Vehicle Billing
36.	Cost of travel	Currency	Found in- 1.Vehicle Billing
37.	Booking date	Date/Time	Found in- 1.Vehicle Billing 2.Vehicle Booking Update

38.	Drop-off date	Date/Time	Found in- 1.Vehicle Booking 2.Vehicle Booking Update 3.Vehicle Billing
39.	Billing date	Date/Time	Found in- 1.Vehicle Billing
40.	Late fine	Currency	Found in- 1.Vehicle Billing
41.	Booking charge	Currency	Found in- 1.Vehicle Billing
42.	Driver charge	Currency	Found in- 1.Vehicle Billing
43.	Additional expenditures	Currency	Found in- 1.Vehicle Billing
44.	Gross total	Currency	Found in- 1.Vehicle Billing
45.	Discount	Currency	Found in- 1.Vehicle Billing
46.	Advance paid	Currency	Found in- 1.Vehicle Booking 2.Vehicle Booking Update 3.Vehicle Billing
47.	Servicing details	String	Found in- 1.Vehicle Servicing

48.	Employee ID	String: Only alphabetical and numerical data accepted.	Found in- 1.Employee Entry
49.	Name of Employee	String: Only alphabetical and numerical data accepted.	Found in- 1.Employee Entry
50.	Designation	String: Only alphabetical data accepted.	Found in- 1.Employee Entry
51.	Basic salary	Currency	Found in- 1.Employee Entry
52.	Fuel type	String	Found in- 1.Fuel Record
53.	Quantity	Single	Found in- 1.Fuel Record
54.	Cost	Currency	Found in- 1.Fuel Record

➤ There are 5 documents used by the system. They are-

i. Holiday Packages-

1. List of Packages.
2. Customer list Package Booking
3. Package Expenses list
4. List of Total Package Expenses
5. Receipt

ii. Vehicle rentals-

1. List of Vehicles
2. Customer list of Vehicle Booking
3. Fuel Record
4. Bill
5. Token

These are the full description of the organization and methods that are currently used by the organization, with a complete description of the origin of the data to be used by the organization, with a complete description of the origin of data to be used and some indication that data takes.

(ii) Investigation and Analysis:

In order to get a detailed look of the current system, I decided to interview the people that are involved with the current system, give the employees questionnaires to fill in and observe current practices to help identify problem areas and examine how the current system works.

❖ Investigation Methods and Plan Investigation

- **Questionnaire:** I prepared a common questionnaire for the Users of the system who would be servicing the customers directly. They would have to answer a fixed set of questions, many of which were to get to know the current system better and to know what improvement the users would like to the current system. Confidentiality is allowed in hopes of honest response. The users will also be allowed the option of interview, as per their wish.
- **Interview:** The end users will then be interviewed along with the executive officer in order to obtain further details, and to seek their approach to the present system. The interview with the Executive Officer provided me with an insight of the use of the data produced by the system and how they are processed further for decision making. The Executive officer was interviewed informally at first which gave me an insight on the problems that the system holds. It is expected that the conversation with the end users would allow me to understand the lacking the current system holds and to know how the process could be made more efficient. I expect the interviews to succeed as it would allow me to change my questions according to the responses of the interviewee or as suitable for the atmosphere. The interviewee can show me any given faults first hand and if necessary I can change my attitude as required.
- **Collection of samples:** samples of data of various stages of the process will be taken and displayed in the following pages. This would give me an idea of how data is actually stored and confirm the fields required to store a particular record to me,
- **Observation:** I also plan on sitting with the end users while they work at the counters and observe their method of operating. I hope my presence will not cause any abnormality to their work. This will allow me to collect data without engaging the users of the system. If they do want to show me the problems in the system while using it then I shall record what they present if possible.

Finally I will write a report on the current system which will be checked later by Mr. Sheikh to seek for any mistakes. After the system is overviewed and corrected, I will finally start designing the new

software. First I will design manually and show it to Mr. sheikh at every stage to avoid misunderstandings later.

In the next pages I have given samples of interviews and questionnaires. I have also given some of data representation of the current system after I have observed the whole system. The samples of the interviews that are given were first recorded on a tape recorder and extracts of those are typed into the computer.

■ *Interview with the Executive Officer of Travel Agency, Mr. Sheikh*

Me- Good morning sir, I would appreciate it if you could tell me about the services provided by the Travel Agency.

Mr. Sheikh – The Travel Agency provides 2 types of services. They are – Holiday Packages and Vehicle Rentals. We provide fixed packages for customers complete with accommodation according to the best interests of the customer at low budgets. We also provide vehicle rentals for potential customers who may be visiting Bangladesh from other countries and are in need of temporary transport.

Me – Does the travel agency provide bookings in advance?

Mr. Sheikh – Yes. We provide bookings in advance for potential customers.

Me – how are the bookings managed?

Mr. Sheikh – We keep records of all the customers who have booked the packages in a list that is updated every day. In case of vehicle rentals, we keep records of the customers in a list and also a record of all the vehicles that are available to us. These are also updated every day.

Me – Does the agency have any intention to expand soon? I mean, does the agency plan on increasing the number of packages offered or increasing the number of vehicles soon?

Mr. Sheikh – No. The agency does not have any intentions of such sort.

Me- Could you tell me about the day to day functions of the travel agency and how the overall agency is managed?

Mr. Sheikh – there are 4 types of counters. They are – primary counter, secondary counter, tertiary counter and hotlines counter. The customer collects the catalogs and primary information from the primary counter. They can seek for any further details on the services here. Then the customer is directed to the secondary counter to obtain the forms and obtain any further information on registration. Any information required on the filling of the forms can be found here. After the forms are filled, the customer pays the bill at the tertiary counter. In case the customers cannot make it to the office they can use the hotlines counters to make early bookings and pay their bills through credit cards or balance transfers.

Me - How many individuals are involved in the functioning of the system and what do they do?

Mr. Sheikh - For the types primary and secondary, there are 2 counters each. There are 4 hotlines counters as more people make bookings by phones rather than coming directly to the agency. There are 3 tertiary counters to prevent crowds at a particular counter. One individual

is required to be working at each of these counters. So there are a total of 9 individuals working at the agency at a time. The employee at the primary counter provides the catalogs containing the options for the customer and provides information on them.

Me – How is the handing over of the vehicle managed?

Mr. Sheikh – after payment, the customers receive a token. This token serves as their identification. They are to take this token from the counter and show it to the in charge of vehicle management in order for him to give access and show them to their vehicles. If the customers make registration through hotlines, the token can be mailed to them or they can collect it any time from the office tertiary counter.

Me – can the information a customer already provided be changed?

Mr. Sheikh – it becomes a little hard for us to manage any changes that may be required to the already provided information. But we try our best to make amendments. If it is absolutely necessary, a new form is provided to the customer and the old one is rejected.

Me – how necessary does it become for the agency to protect the information provided by the customer?

Mr. Sheikh – well, the agency maintains a confidentiality agreement with all its customers so the information is not allowed to be leaked at any cost. It becomes required as at present, all the information is stored in papers and kept in folders and the information can be required any time. But we hope to achieve better security through the new system.

ME – So who do you think should have access to these records?

Mr. Sheikh – I would have to say that it would be preferable if only the CEO and I, being the executive officer, only had access to the information.

Me – How is the billing managed when it comes to Packages or Vehicle Billing?

Mr. Sheikh – when it comes to packages, a receipt is provided to the customers after full payment is done. In case of vehicle rentals, an advance payment needs to be made on booking which is recorded in the booking form. Then, a billing form is used at the event of full payment which contains all the billing information as well as the information from the booking form. You may ask for further details from any of the officers at the counters.

Me – what are your thoughts on the software? What would you want it to be able to do?

Mr. Sheikh – the software is expected to automate all the manual operations of the job. It needs to take over the whole system and improve the consistency and overall quality of the documents. The software may also be required to print booking receipts, bills and any other daily or optional reports. The ability to add and cancel packages or vehicles would be very helpful as they might require frequent or occasional change. There must also be options to edit the information already provided on a package, vehicle or booking information provided by a customer. There should be options to delete records in case a customer wants to cancel his booking.

Me – Do you have any particular requirement or idea about the design of the forms?

Mr. Sheikh – No, there is no particular requirement at the moment.

Me – How are the current booking forms, bills and receipts made?

Mr. Sheikh – They Agency uses pre-printed forms, bills and receipts. The forms are provided to the customers so that they fill them up themselves. In case of bills and receipts, the officers fill them up themselves and obtain any customer information from the forms.

Me – how are the employee information managed?

Mr. Sheikh – we keep a separate section which contains all the employee information. When they are required we can obtain their information from this section.

Me – How is the servicing data of the vehicles managed?

Mr. Sheikh – there is no particular way for us to manage the vehicle servicing information. We have to contact the servicing centre when any information on the status or servicing of the vehicle is required. This makes it a little hard for us to make decisions. We hope the new system will provide ways for us to focus on this aspect as well.

Me – Is there anything else you would like to add to the new system?

Mr. Sheikh – There is no particular requirement at the moment. I will contact you if I think of something in the future.

Me- Okay sir, I will get back to you if I need any further clarification on anything. Thank you for your precious time.

Mr. Sheikh – Yes, sure. I am available anytime you need me.

Me – One last thing sir. May I have the permission to observe your current manual processing system in detail tomorrow?

Mr. Sheikh – Yes. You are welcome to do that.

■ *Interview with a Reception Officer*

Me – Good Morning. Thank you for giving me your precious time to take this interview. I would like to ask you some questions regarding the Travel Agency.

Officer – yes. Sure. Please proceed.

Me – Please describe in detail the procedure of direct Package Booking. Please elaborate till the payment procedure.

Officer – At first the customer must obtain information on the packages provided by the agency from the primary counter. The Package Booking is done by entering the customer information in a form that the customers themselves must fill up. The form is collected from the secondary counter. The booking is confirmed by making the full payment of the package at the tertiary counter. After payment a receipt is provided to the customer.

Me – Okay, now can you please describe the procedure of direct vehicle booking? Please elaborate till the payment procedure in this case as well.

Officer – The procedure for vehicle booking is almost the same. The information on the options provided by the agency is obtained from the primary counter. The forms are obtained from the secondary counter. There is just one change when it comes to the payment. Instead of full payment, the customer is allowed to make an advance payment before and obtain a token. Then the customer can use this token as proof of the advance payment while receiving the vehicle. After the job is done, the customer has to make the full payment. After full payment, a billing form is filled by the officers at the tertiary counters. Then a bill is provided to the customer and a copy of the bill is kept by the agency.

Me – The information you have provided me with is on direct registration. Can you provide me with any information on hotline registration?

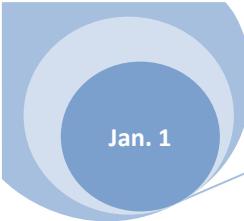
Officer – The form of registration is a bit different when it comes to registration through hotlines. The customer has to make a call at the hotline counters. After that the officers at the counters have to explain the options and features of the packages or vehicles to the customers, as per their response. Then the officer has to take a form and accordingly ask the questions required to fill the forms. After the forms are filled the customer is asked how they are going to pay the bills. After the payment, booking is confirmed for a particular customer and their forms are transferred to the tertiary counter and a token is prepared for them. The tokens are sent to the customers through mail or kept at the counters for the customers to pick them up later, as they wish.

Me – What are the documents used in manual process? Could you describe them?
Officer –

Me – can you tell me some of the problems you face while using the current system?
Officer – well, to start with, it is very hard to manage the documents. For example, there are many applications every day so it becomes exhaustive to move the file containers from one place to another. When a particular customer's form is required, it becomes hard to search for it in the huge pile of forms. After the job is done or a booking is cancelled, it becomes hard to search for the form. All these flaws also make the current system very time consuming. It also becomes impossible to make corrections to information already provided if a customer makes mistakes or if any change occurs to customer's data and he wants to update it. The process of checking availability of vehicles is pretty tedious as well.

Me – Would you want the new system to be able to do?
Officer – it would be helpful if the new system focused on the problems I already mentioned. Other than these we are pretty much satisfied with the system.

Me – Thank you for your cooperation. It would be really helpful if you could provide me with some sample documents so that I could get to know the fields better.
Officer – You are most welcome. I will fetch them as fast as possible.



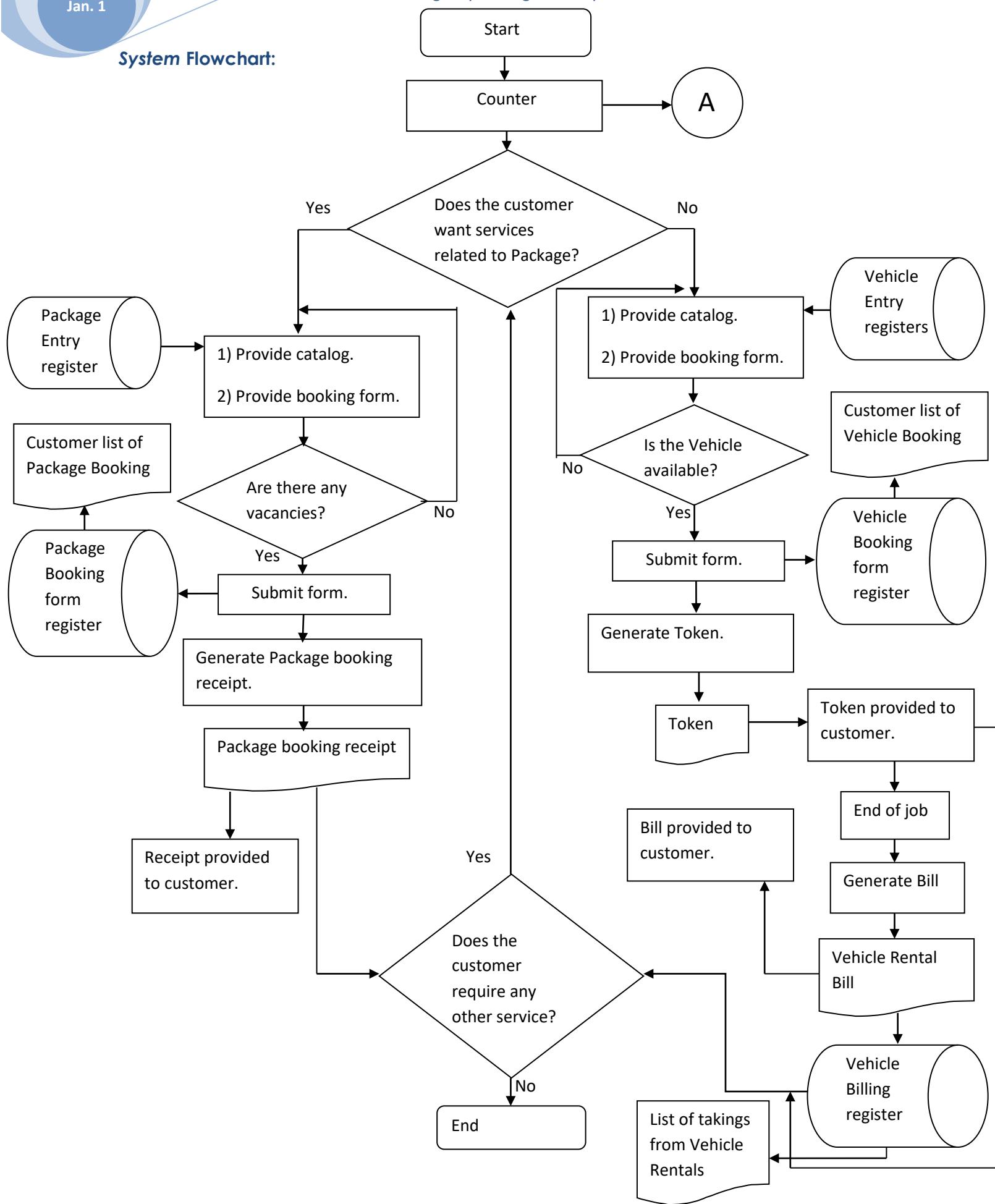
Jan. 1

Travel Agency Management System

Jan. 1

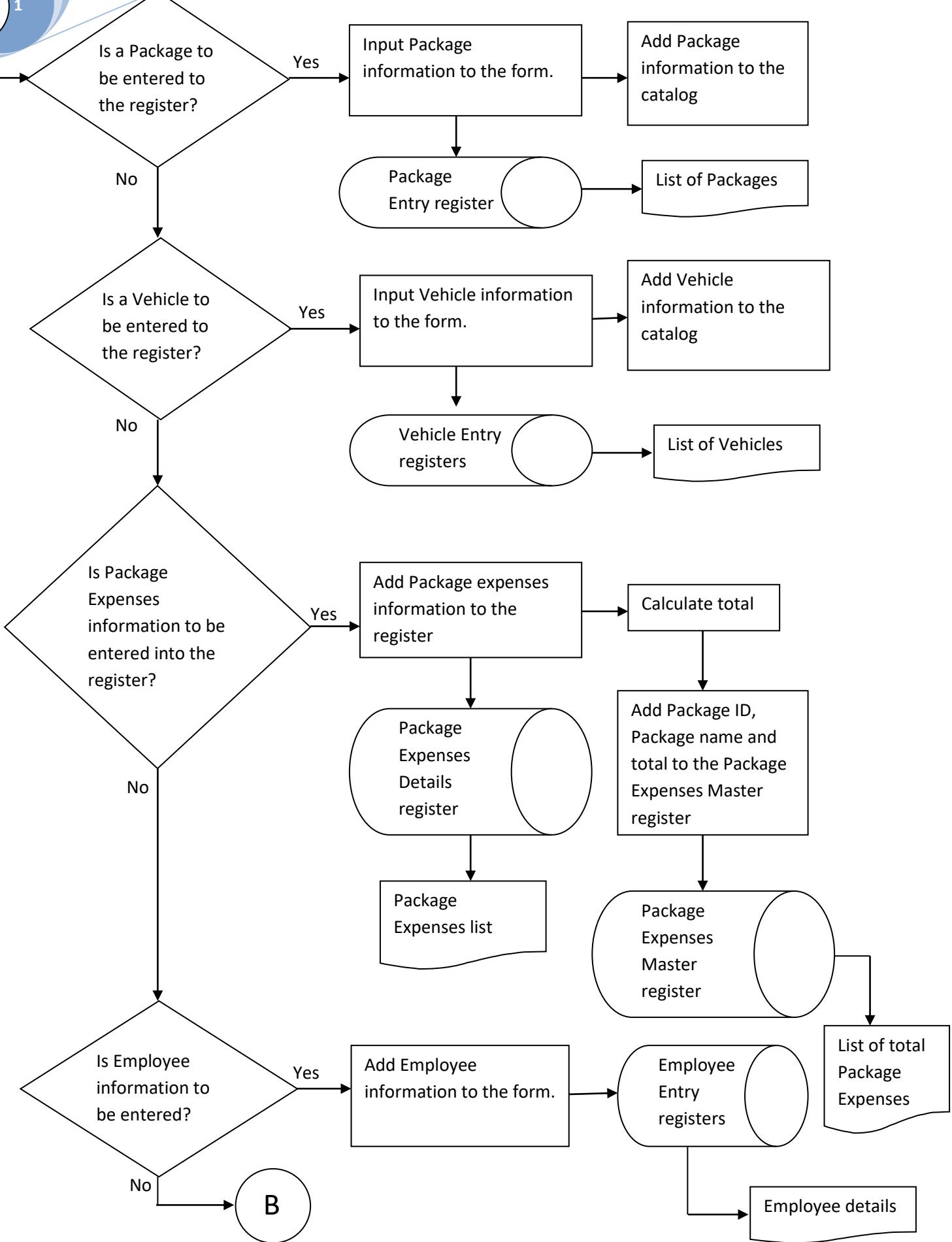
System Flowchart:

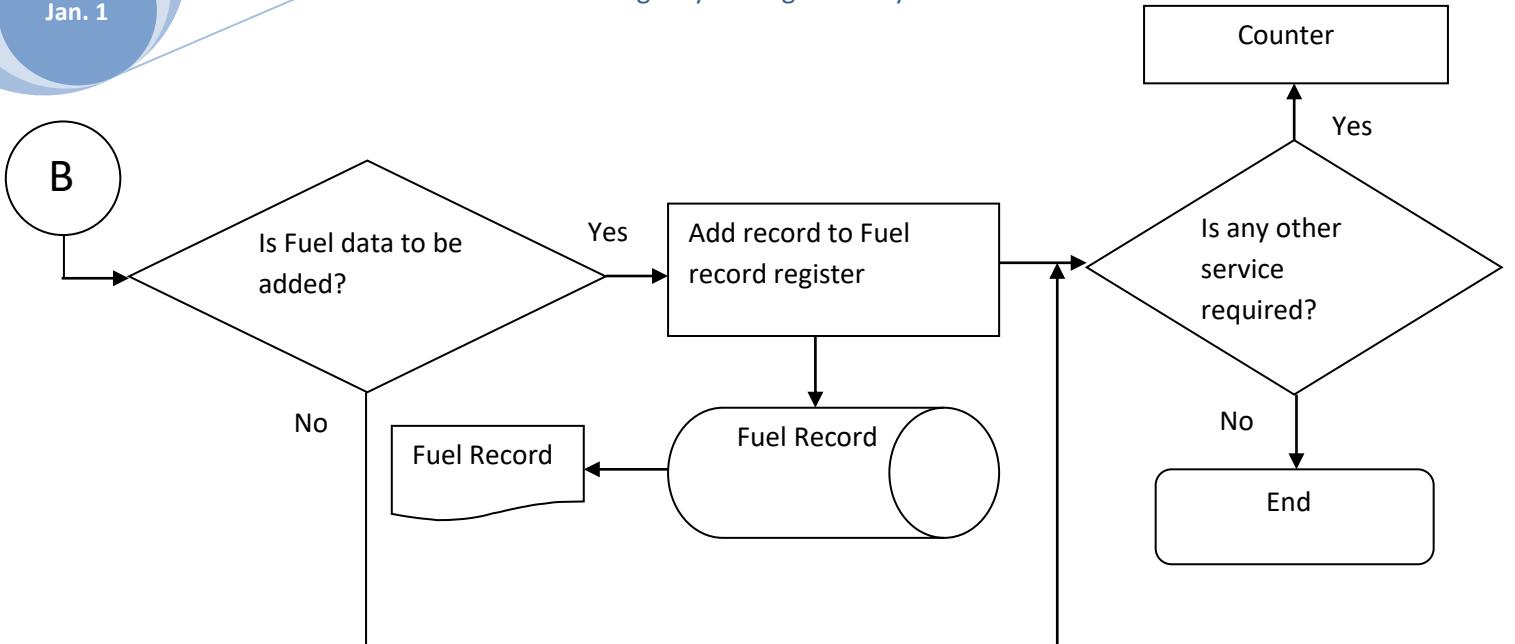
Travel Agency Management System



1

Travel Agency Management System





Requirement specification

❖ Problems in the current system

- 1) The manual system is very difficult to follow for the customer and it is also time consuming.
- 2) It is labor intensive as it requires continuous monitoring to ensure each transaction is accounted for. The time used for the constant monitoring could be used for more productive activities.
- 3) It becomes boring for the staff to follow the manual system which decreases slows them down and decreases job efficiency.
- 4) Whenever a customer wants to make a vehicle booking, an officer has to go through all the vehicle availability records which is time consuming.
- 5) Errors in calculation are highly probable. There is also the possibility of a vehicle being booked twice, which can cause problems. There may also be errors in the information provided by the customer or the customer may make errors while filling up the form.
- 6) The customer files are stored in folders. It is time consuming to search through the folders when searching for a file. There is also a high chance of losing documents of the customer. The pile of folders also consumes a lot of valuable space.
- 7) Any changes made to the options, such as changes in price, change in destination when it comes to package, addition or deletion of packages or vehicles etc. cannot be implemented immediately in the catalogs and forms.

- 8) It becomes hard to keep track of employee information and it becomes hard and time consuming to search for the information of a particular employee in the middle of a pile of files.
- 9) It becomes hard to keep track of package expenses, which may be required to make judgments about the Packages section of the travel agency.
- 10) It becomes hard to keep track of vehicle servicing details, which may be required to make decisions about the future about Vehicle rentals section of the Agency.
- 11) There is a lack of security of guest and employee information as they are maintained in folders and may be stolen, harming employee privacy.
- 12) Dealing with so many documents is tiresome for the staff. This makes them more prone to making mistakes or doing their jobs in a tedious manner.
- 13) The large amounts of documents are hard to handle, search and store. The process of handling of documents is time consuming and hard so this can cause disturbances when it comes to job satisfaction.
- 14) There is no way to backup data in the current system as all the information are on paper and backing up the data would require carbon copying the papers, which is expensive, or hand copying the information, which is labor-intensive, time consuming and expensive as well. In case of any hazard (such as a fire), therefore, there is no way of retrieving the data that may be lost due to single copy.
- 15) The old system is not cost-effective due to the large amount of paper or other materials that is required to operate the system.
- 16) In the modern world, all systems are becoming computerized. Therefore, any business organization that doesn't use a computerized system is at a disadvantage when it comes to competition with other businesses. The computerized system allows the work that may be started at the office to be finished at home by use of data transfer devices such as pendrives. It also allows backing up of large amount of data through portable hard drives. The system prevents human errors and a computer doesnot forget to do anything (such as backing up data or updating the availability of a vehicle). Therefore, a manual system doesnot allow a business to keep up the pace with the modern world, which the computerized system provides.

❖ User Requirements:

After close observation of the system and a long conversation with Mr. Sheikh on the system, I narrowed down to a summary of user requirements:

- 1) The new system should be faster than the manual system used at present. It must allow data, transactions to be processed and bills to be issued faster and related information (such as availability of vehicles) to be updated automatically.
- 2) The system should provide better security for customer and employee information security and privacy protection. The employee and customer information should be stored separately to allow them to be easily distinguishable. The people who are allowed to access the system should be provided with a unique name and password.
- 3) The system must allow for easy and faster formation of new user accounts. Each user would be recognized by the system by their User ID.
- 4) It should be easy to add, edit and delete information about packages or vehicles. Entries of packages, vehicles or employee records should be able to be made easily.
- 5) The system should provide an easy method to add new bookings frequently. It should also allow customer information to be easily searched. The system should allow the information to be updated easily. If a customer cancels a booking, the system should allow the customer information to be deleted easily and related information (such as availability of vehicles or number of vacancies for a package) to be updated automatically.
- 6) Unique Booking IDs should be provided to each customer to help distinguish between them. Validation checks are made to make sure no User ID is used twice.
- 7) Booking IDs, Package IDs and Vehicle IDs should be generated automatically and also validation checks provided to make sure no ID is used twice by any chance. This is done in order to protect the security of information present in the database.
- 8) Search buttons should be provided in any update or billing forms so that all the information can be found at the click of a button by just providing the Booking ID. The booking ID would be provided and the search button would be pressed in order to automatically load all the necessary information from the database and fill the information on the form automatically. If a Booking ID is not present in the database by any chance, error messages should be provided to make sure the user is notified. Where this system is not used (such as employee entry form, Vehicle update form and Package entry form) there will be user interactive lists to make sure the user can navigate through options to

choose find the required record. The fields on the form will be filled on making a selection from the list.

- 9) Calculations should be done automatically after all the information is entered in the forms.
- 10) Availability of vehicles and number of vacancies of packages should be updated automatically after every booking and they should be checked before every booking to prevent rebooking of any particular vehicle or limit of reservation of a package to be crossed. If a particular vehicle is booked, it should be removed from the options provided in the booking form.
- 11) Vehicle rent bill should be generated at the end of a job. All the booking information should be loaded automatically from booking information and vehicle information databases at the provision of Booking ID.
- 12) The new system should provide a way to keep track of package expenses.
- 13) The new system must provide a way to keep track of vehicle servicing information.
- 14) The system should provide a method to keep track of the fuel records.
- 15) The system should be easy to use and user friendly. It should provide help and have an option for backing up data.

❖ **Hardware requirements:**

Hardware list	Purpose
Intel core i3 processor with speed 2.0 GHz	<ul style="list-style-type: none"> • To run all the software required and the operating system. • To allow the system to function smoothly.
1 GB RAM	<ul style="list-style-type: none"> • To load the system files that need to be processed properly. • Acts as temporary storage of data.
160 GB hard disk	<ul style="list-style-type: none"> • To keep installment files of the system. • To keep the database. • To keep the backup files.
15/19" inch LCD or LED Monitor	<ul style="list-style-type: none"> • To obtain visual display of forms. • To obtain a visual display of the documents such as receipts or bills.
Printer(Laser, inkjet or dot matrix)	<ul style="list-style-type: none"> • To print bills or receipts. • To print other documents when required.
Combo drive	<ul style="list-style-type: none"> • To write the backup copy database into a CD • To access files from CD or DVD
UPS	<ul style="list-style-type: none"> • To prevent loss data if power failure occurs.
Removable flash drive (Pen drive, 32GB)	<ul style="list-style-type: none"> • To keep back-up of the files for further reference, in case the computer is not working.
Mouse and keyboard	<ul style="list-style-type: none"> • Keyboard is required to enter data into the system using forms and mouse is required to navigate the whole system; i.e. select menus, draw controls etc.

❖ **Software requirements:**

1. **Windows 7 or 8** – to run the software
2. **Ms Access** – this is a RDBMS used to create and maintain a database easily and efficiently.
3. **Visual basic** – it is a well-known programming language. It is used to develop the interfaces of the software and to run the software.
4. **Crystal report** – this is a report making software where any output report is designed very quickly.
5. **Adobe reader** – this will be required because I plan to write a help manual for the user in a portable document format.

Alternative solutions:

There objectives could be pursued by using a number of methods:

- Database management software such as Microsoft Access can be used to make a solution. It can be used to store all the records and can also create different forms for entering the customer names, details etc. during transaction. The records in the database can be used easily to produce the receipts, bills etc. but these forms cannot be customized as the user wants. Although few mistyped errors can be avoided, this is very limited.
- Microsoft excel could also be used as a solution to the existing problem. It can be used in the package booking and vehicle booking and billing sections. The customer names, booking IDs etc. can be stored in the registers. Besides, it can be used to calculate total amount while producing bill, receipts etc. However, features offered by Microsoft Excel are not enough to solve this problem. It cannot store records in an organized way because the data is stored in cells. It also cannot reduce chances of mistyped error.
- Computerized solution (Custom – Made): A program can be made by combining visual basic and a database program (e.g. Ms Access); the program will consist of forms appearing on screen for data to be input and saved in the database; a search engine can be made to extract necessary information from the database regarding particular items, also reports can be generated. Besides visual basic allows setting all rules that needs to be followed when data is entered. It allows the user to create forms as they like. It can be linked with database software to store all records easily.so it can add, alter, delete or access the data from the database easily. This method of solution allows me to meet all objectives set. It eliminates almost all the shortcomings of the current system being used. Though this solution will generate more costly in the short term than the previous two methods, it is bound to be most cost efficient and cheaper in the long term. Therefore, in my opinion this solution is the best option available.

Letter from the Administrative Head of MAAS Travels and Tours:

MAAS Travels and Tours
852, Green Line Road,
GEC, Chittagong



March 10, 2015

Priontu Chowdhury

53, Nasirabad I/A, Chittagong, Bangladesh

Dear Priontu Chowdhury,

We are writing this to inform you of our decisions regarding the kind of solution we want to use at MAAS Travels and Tours. After close examination of the alternative solutions to the problem, we have agreed that a customized solution is what we want; after all, the other solutions do not quite fulfill all our requirements and objectives. The finalized decision is to convert to the tailor-made system that you intend to design.

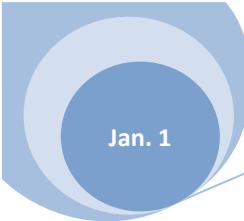
Thank you for your kind co-operation.

MAAS Travels and Tours encourages you and wishes you luck with the design of the solution.

Warm regards,

A handwritten signature in black ink, appearing to read "Abdullah Sheikh".

Mr. Abdullah Sheikh
Administrative Head
MAAS Travels and Tours



Jan. 1

Travel Agency Management System

(b)

System Design

(i) The Nature of the solution

❖ *Objectives of the system*

1. System should be password protected and allow access to the user only when correct User type, User ID and Password are provided.
2. The system should allow the user to change his/her password as required, only on provision of correct User type and User ID.
3. Validation checks to make sure incorrect data cannot be entered into the database and key fields (such as Booking ID, Employee ID, Package ID etc.) cannot be changed.
4. The system must allow new User accounts to be added, edited and deleted easily.
5. The system should allow Packages to be added, edited and deleted easily.
6. The system should allow Vehicle details to be added, edited and deleted easily.
7. The system should allow Employee records to be added, edited and deleted easily.
8. The fields on the booking forms must be organized and must prompt the user to ask necessary questions to the customer in order to fill the form on a step-by-step basis.
9. Packages and vehicles can be booked easily.
10. The system must provide a way to change booking information when required.
11. The system must provide a way to cancel a booking if required.
12. All arithmetical calculations should be done automatically in the forms to reduce mistakes.
13. The entry and booking forms must generate IDs automatically to prevent overwrite. ID checks should be provided for all kinds of IDs in all forms so that the same ID cannot be used twice.
14. A list of the names or packages should be present in certain forms (such as Package Entry, Vehicle Entry, Vehicle Update, Vehicle Servicing and Employee Entry etc.) which should be updated after every booking. This allows the system administrator to keep the options provided by the agency in supervision. Also, the he can choose directly from the list when he/she wants to edit or delete a particular record. All the information on the form appears immediately on selection.
15. Easy to use search system that only requires the user to provide the Booking ID only when it comes to searching for bookings. The system can be used when the customer wants to edit or delete any booking information or just wants to check the information.
16. User interactive forms that show options according to customer choice and react to customer's choices to maintain security.
17. If a vehicle is already booked, it is automatically removed from booking the forms. It automatically reappears when job is done or the booking is deleted. Therefore the system automatically keeps track of the availability of vehicles. The vehicles are made unavailable on booking, and made available again on cancellation of a booking, when a customer changes his choice of vehicle (where one vehicle is made available while another one is made unavailable) or after the job is done on billing. The availability options of vehicles are updated after every transaction.
18. The system should automatically check for the number of vacancies available for a package before every booking. The customer should be informed immediately if there are no vacancies for a particular package. The vacancy check information should be updated automatically after every booking transaction made (such as a new booking or cancellation of a booking or update of a booking).
19. A form for Vehicle Billing that will automatically load all required information from Vehicle Booking database on the customer and the services provided to him/her on

- provision of their Booking ID. The rest of the information needs to be provided manually and then all the calculations should be done automatically on bill generation.
- 20. A Package Expenses Entry form needs to be made to keep track of the expenses on the package. The package expenses details are stored in two different tables, one containing individual expenses details and another containing total expense on a particular package.
 - 21. A Vehicle servicing form needs to be made to keep track of information on servicing and status of vehicle. The form will servicing information for a particular vehicle on selection from the Vehicle ID list and any new information will be added to the list and to the database on addition of the record.
 - 22. The system contains a “Fuel Records” form that stores data to a Fuel record table to provide a way to store fuel records data.
 - 23. Presence checks to make sure all the information are provided before information is being saved, searched, updated or deleted on all firms.
 - 24. Character type checks to make sure non-numerical data cannot be input in fields such as Cost per person, Net total and Contact number and so on.
 - 25. Length checks to make sure there are 11 digits provided for contact number.
 - 26. A “Refresh” button present in all forms to allow the user to clear the forms immediately as they wish. This allows users to start anew.
 - 27. Generate receipt for Package booking.
 - 28. Generate token for Vehicle booking.
 - 29. Generate bill for Vehicle Billing.
 - 30. Generate reports.
 - 31. Provide User manual and option to back-up.

Letter from the Administrative Head of MAAS Travels and Tours:

MAAS Travels and Tours
852, Green Line Road,
GEC, Chittagong



March 16, 2015

Priontu Chowdhury

53, Nasirabad I/A, Chittagong, Bangladesh

Dear Priontu Chowdhury,

This is to inform you that we have read and agreed to the objectives of the new system that you have listed. The objectives tackle most of the problems in the current system. As of yet, we do not feel the need to include anything else in your list.

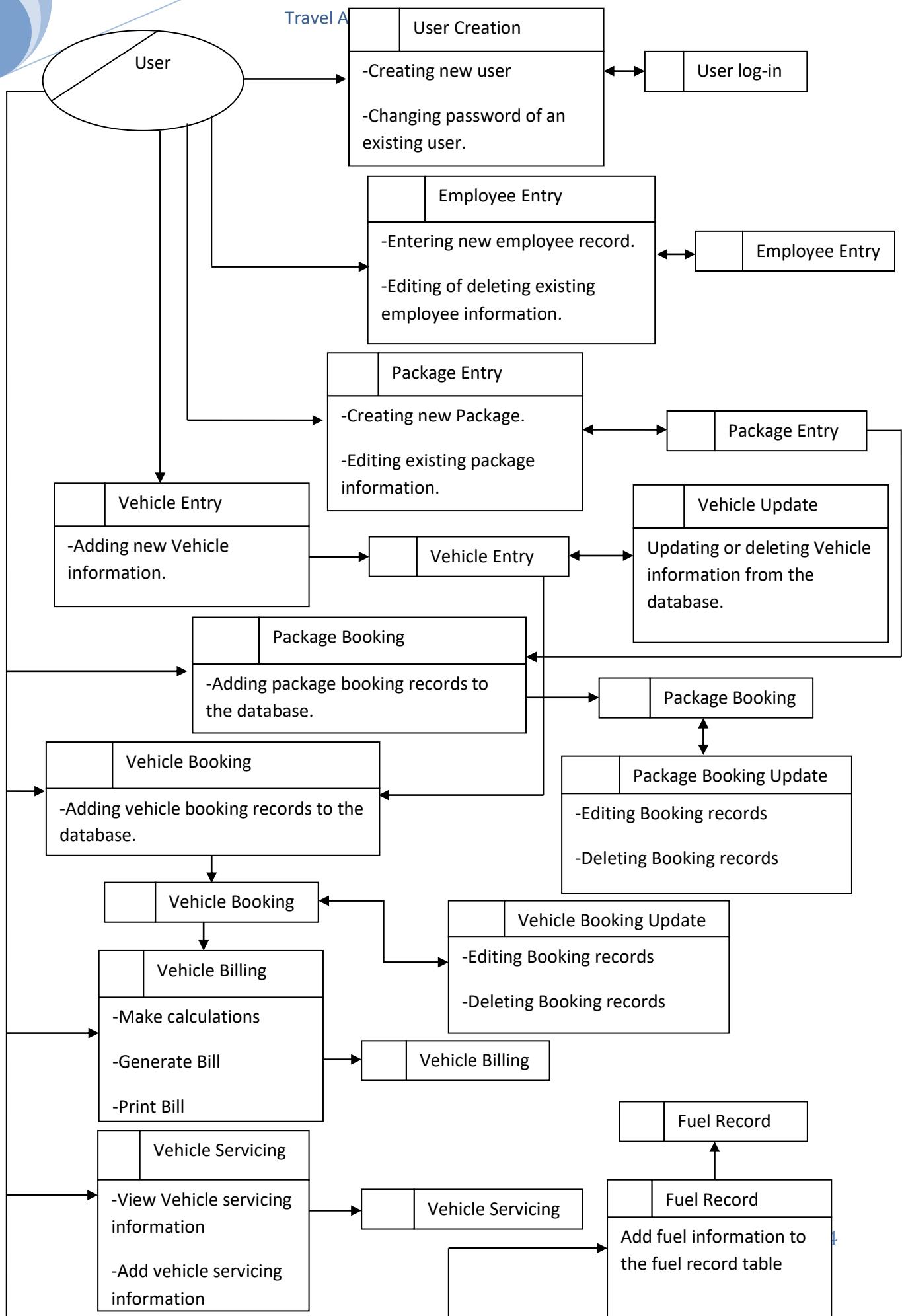
We encourage you to design the new system based on the proposed list of objectives.

MAAS Travels and Tours thanks you for your co-operation, and looks forward to the progress of the design of the new system.

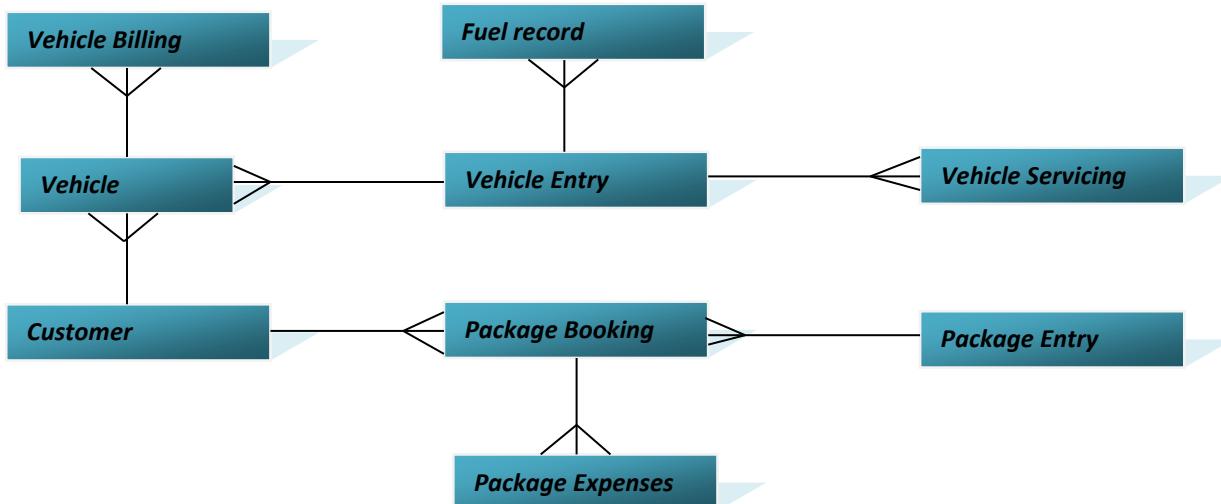
Warm regards,


Mr. Abdullah Sheikh
Administrative Head
MAAS Travels and Tours

Jan. 1



❖ *Entity Relationship Diagram*



❖ *Proposed file, record and data structures:*
There are 11 tables in the proposed system.

- User login
- Vehicle entry
- Employee entry
- Fuel record
- Package entry
- Package booking
- Vehicle booking
- Package expenses master
- Package expenses details
- Vehicle Billing
- Vehicle Servicing

1. User_login

The table stores the details of the users of the system. It is connected to the “User Log-in” and “User Creation” form.

Field name	Data type	Size	Note	Example	Validation
User_type	Text	10	User type of user	Administrator/ General	Presence check
User_ID	Text	10	A ID for identification of user, unique for each user.	JohnyB2112	Presence check
Password	Text	20	The password of the user.		Presence check

2. Vehicle_entry

Field name	Data type	Size	Note	Example	Validation
Vehicle_type	Text	10	Type of the vehicle.	SUV/Sedan/Convertible/minivan	Presence check
VehicleID	text	6	An ID for identification of vehicle, unique to each vehicle.	Ad2150	Presence check
RegistrationNo	Text	7	Legal registration number of vehicle.	22-2122	Presence check
Make	Text	10	The name of the company that made the vehicle.	toyota/mercedes/Bentley/chevrolet	Presence check
Model	Text	6	The model of the vehicle.	Vista/arnage/aveo	Presence check
Nseats	Number	4	Number of seats in the vehicle.	5	Presence check,character type check
Rate_per_km	Number	8	Amount charged per kilometer travelled.	\$125.99	Presence check,character type check
Date_of_entry	Date/Time	10	Date on which the entry was made.	10/03/2015	Presence check
IsBooked?	Yes/No	1	A check to see if a vehicle is available for booking.	Yes or No, where No means vehicle is available.	Presence check

This table stores the information about the vehicles of the agency. It is connected to the "Vehicle Entry", "Vehicle Update", "Vehicle Booking" and "Vehicle Booking Update" form. It receives data from "Vehicle entry", sends data to "Vehicle Booking" and "vehicle Booking Update" form and both data receives form and sends data to "Vehicle Update" form.

3.Employee_entry

Field name	Data type	Size	Note	Example	Validation
EmployeeID	Text	6	ID of an employee working for the agency, unique to each employee.	Bessy1212	Presence check
EName	Text	20	Name of the employee.	Bessy Graham	Presence check, character type check
Address	Text	30	Address of employee	Road 4, house 8, khulshi, Chittagong	Presence check
Contact	Text	16	Contact number of employee.	+8801819637911/+917204255875	Presence check, character type check

Designation	Text	10	Category of classification of an employee. Role or position of employee within agency	General manager/Chief executive officer/chief of staff/ executive officer	Presence check
Basic salary	Currency	8	Salary of an employee.	Salary of an employee.	Presence check
Date_of_entry	Date/Time	10	The date on which the entry was made.	10/03/2015	Presence check

This table stores information related to the employees. When anyone from the staff or any senior officer wants to check or alter employee information, they can select the employee ID from the list of Employee IDs in the “Employee Entry” form. This table sends and receives data to and from “Employee Entry” form.

4.Fuel_record

Field name	Data type	Size	Note	Example	Validation
VehicleID	Text	6	Unique ID to identify a vehicle.	Ad2133	Presence check
Date_of_enty	Text	10	Date on which entry was made.	10/03/2015	Presence check
FuelType	Text	10	Type of the fuel.	Octane/diesel	Presence check
FuelQuantity/litres	Number	8	Quantity of fuel.	5	Presence check
FuelCost	Currency	8	Cost of fuel.	\$125.45	Presence check

This table stores fuel records of the vehicle. It receives data from the “Fuel record” form.

5. Package_entry

Field name	Data type	Size	Note	Example	Validation
Package_type	Text	20	Type of the package.	Special package/deluxe package/best on budget	Presence check
Package_name	Text	20	Name of the package.	Beach travel/ Asia tour	Presence check, character type check
Package_details	Text	70	Information on the package.		Presence check
Cost_per_person	Currency	8	Amount charged per person.	\$215.55	Presence check
Date_of_entry	Date/Time	10	Date on which an entry was made.	10/03/2015	Presence check
PackageID	Text	6	An ID for identification of a package, unique to each package.	Qw1152	Presence check
Itinerary	Text	70	Complete travel schedule.		Presence check
Journey_date	Date/Time	10	Date on which the start of the journey is due.	21/08/2015	Presence check
Arrival_time	Date/Time	10	Time of arrival of the customers.	10:00:00 AM	Presence check
Starting_time	Date/Time	10	The time at which the journey is going to start.	10:30:00 AM	Presence check
Duration/days	Number	4	The number of days the journey is going to occur.	5	Presence check
Accommodation_type	Text	10	The type of accommodation the package plan provides.	Deluxe/luxury/low budget	Presence check
NumberReserved	Number	4	The number of bookings made of a particular package.	25	Presence check
NumberOfReservations	Number	4	The number of bookings than can be made for a particular package.	30	Presence check

This table stores information and all the details of all the packages provided by the Agency. This table used when package information need to be altered or new packages need to be added to the options. this table sends data to and receives data from "Package Entry" form. It also sends data to the "Package Booking" form and "Package Booking Update" form as required.

6. Package_Booking

Field name	Data type	Size	Note	Example	Validation
PackageBookingID	Text	6	An ID to distinguish between bookings, unique to each customer.	PG9485	Presence check
Customer_name	Text	20	The name of the customer.	Brett Snow	Presence check, character type check
Customer_address	Text	30	Address of the customer.	Road 6, house 8,Nasirabad housing society, Chittagong	Presence check
Contact	Text	16	The contact number of the customer.	+8801819637911	Presence check, character type check
Package_name	Text	20	The name of the package.	Beach travel/Asia tour	Presence check, character type check
Cost_per_person	Currency	8	Amount charged per person for the package.	\$124.99	Presence check
Number_of_people	Number	4	The number of people for which a booking is being made.	3	Presence check
Net_total	Currency	8	Net amount that a customer needs to make.	\$499.96	Presence check
Date_of_booking	Date/Time	20	The date on which the booking is made.	10/03/2015	Presence check
Package_type	Text	20	The type of the package.	Special package/deluxe package/best on budget	Presence check
Accommodation_type	Text	10	The type of accommodation provided by the package plan.	Deluxe/Luxury/low Budget	Presence check
Journey_date	Date/Time	10	The date on which the beginning of the journey is due.	15/06/2015	Presence check
Arrival_time	Date/Time	10	The time of arrival of the customer.	10:00:00 AM	Presence check
Starting_time	Date/Time	10	The time at which the journey will start.	10:30:00 AM	Presence check
Duration/days	Number	4	The number of days for which the tour will carry on	5	Presence check
Itinerary	Text	70	Complete schedule of the journey		Presence check
Package_details	Text	70	Detailed information about the package.		Presence check
PackageID	Text	6	An ID for identification of package, unique to each.	PG3215	Presence check

This table stores all kinds of package booking information. It receives data from "Package Booking" form and sends and receives data to and from "Package Booking Update" form.

7. Vehicle_booking

Field name	Data type	Size	Note	Example	Validation
VehicleBookingID	Text	6	Booking ID unique to each customer, for differentiation.	789754	Presence check
VehicleID	Text	6	A ID number for differentiation of vehicles, unique to each vehicle.	VC2323	Presence check
Customer_name	Text	20	The name of the customer.	Gary P. Gale	Presence check, character type check
Customer_address	Text	30	Address of the customer.	Road 4, house 5, Khulshi, Chittagong	Presence check
Contact	Text	16	Contact number of the customer.	+8801819637911	Presence check, character type check
Vehicle_type	Text	10	Type of the vehicle.	Sedan/SUV/Convertible/minivan	Presence check
Journey_date	Date/Time	10	The date on which the journey was due.	12/06/2015	Presence check
Driver_name	Text	20	Name of driver.	Peter Gatsby	Presence check, character type check
Date_of_booking	Date/Time	10	Date on which booking was made.	13/05/2015	Presence check
Registration_number	Text	6	Legal registration number of the vehicle.	22-3215	Presence check
Drop_off_date	Date/Time	10	The day on which the vehicle is supposed to be returned according to the deal with the customer.	06/07/2015	Presence check
Make	Text	10	The name of the company that made the vehicle.	Toyota/Mercedes/Bentley	Presence check
Model	Text	6	The model of the vehicle.	Premio/Vista/Arnage/Prius	Presence check
Rate_per_kilometer	Currency	8	The amount charged per kilometer travelled.	\$125.99	Presence check
Number_of_seats	Number	4	Number of seats in the vehicle.	4	Presence check
Advance_paid	Currency	8	The amount that needs to be paid at booking.	\$1400.00	Presence check, character type check
Starting_km	Number	4	The value of kilometer travelled before the vehicle is turned in to a new customer.	00431556	Presence check

This table stores all the information on the Booking of vehicles. It receives data from "Vehicle Booking" form. It sends data to "Vehicle Billing" form. It also sends data to and receives data from "Vehicle Booking Update".

8. Package_expenses_master

Field name	Data type	Size	Note	Example	Validation
PackageID	Text	6	Unique ID that serves as identification for each package.	PG2313	Presence check
Package_name	Text	20	The name of the package.	Beach travels/Asia tour	Presence check, character type check
Total	Currency	8	The total numerical amount spent on package expenses.	\$1400.85	Presence check

This table stores overall information on the total expenses on packages. It receives data from “Package Expenses Entry” form.

9. package_expenses_details

Field name	Data type	Size	Note	Example	Validation
PackageID	Text	6	An ID that serves as identification for each package, unique to each package.	PG8945	Presence check
Expenses	Text	20	The names of the expenses.	Travel food costs	Presence check
Amount	currency	8	The amount charged for a particular expense.	\$555.86	Presence check

This table stores all individual expenses and the information on the package. It receives data from “Package Expenses Entry” form.

10. Vehicle Billing

Field name	Data type	Size	Note	Example	Validation
BookingID	Text	6	An identification number for a booking, unique for each customer.	320305	Presence check, character check
Customer_name	Text	20	The name of the customer	Lu Kyang	Presence check
Customer_address	Text	50	The address of the customer.	45 th , avenue street, Chittagong	Presence check
VehicleID	Text	6	The identification number for a Vehicle, unique for each Vehicle.	Ad4646	Presence check, Character check
Kilometer_travelled	Number	4	The number of kilometers travelled during the journey with the vehicle.	10	Presence check
Rate_per_kilometer	Currency	8	The amount of money charged for each kilometer travelled.	\$50	Presence check
Cost_of_travel	Currency	8	The total cost of the travel.	\$500	Presence check
Journey_date	Date/Time	10	The date on which the journey is due.	10/10/2015	Presence check
Drop_off_date	Date/Time	10	The date on which the vehicle is supposed to be dropped off.	20/10/2015	Presence check
Booking_charge	Currency	8	The amount charged for making a booking.	\$300	Presence check
Driver_charge	Currency	8	The amount charged for hiring a driver.	\$150	Presence check
Additional_expenditures	Currency	8	Any additional expenditures on the vehicle booking by the customer.	\$30	Presence check
Gross_total	Currency	8	The gross total on the vehicle.	\$980	Presence check
Discount	Currency	8	Any discount provided to the customer.	\$40	Presence check
Advance_paid	Currency	8	The advance paid by the customer.	\$200	Presence check
Net_total	Currency	8	The net total for the booking.	\$780	Presence check
Billing_date	Date/Time	10	The date on which the billing is being done.	20/10/2015	Presence check

This table contains all the billing information of any booked vehicle. It receives information from "Vehicle Booking" form.

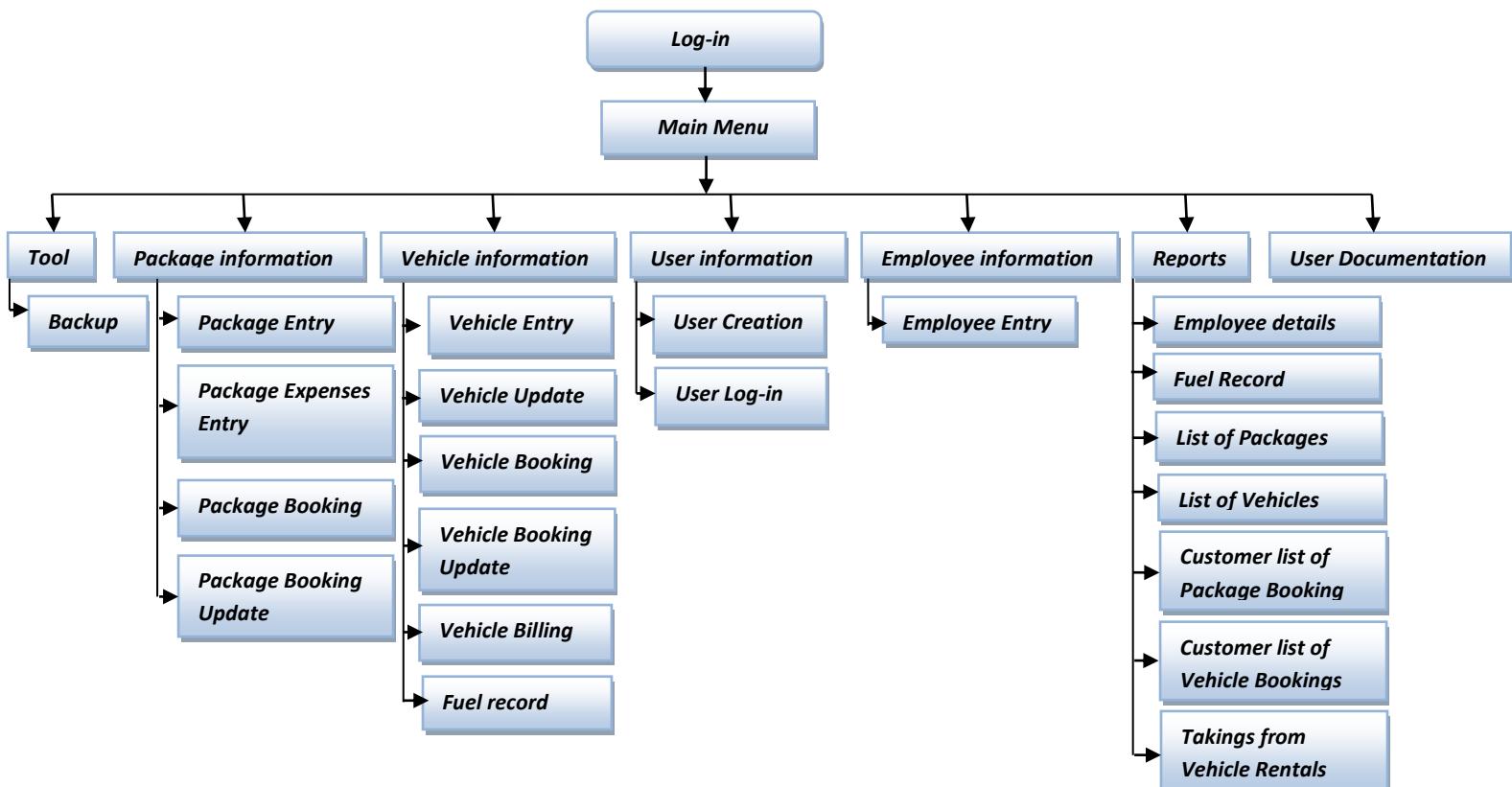
11. Vehicle servicing

Field name	Data type	Size	Note	Example	Validation
Vehicle_type	Text	10	The type of the vehicle being serviced.	SUV	Presence check
VehicleID	Text	6	And identification for the recognition of the vehicle.	Ad2656	Presence check, character check
Servicing_details	Text	70	The information on the servicing of the vehicle.		Presence check
Amount	Currency	8	The amount paid in servicing of the vehicle.	\$1500	Presence check
Date	Date/time	10	The date on which the servicing transaction was made.	10/2/2013	Presence check

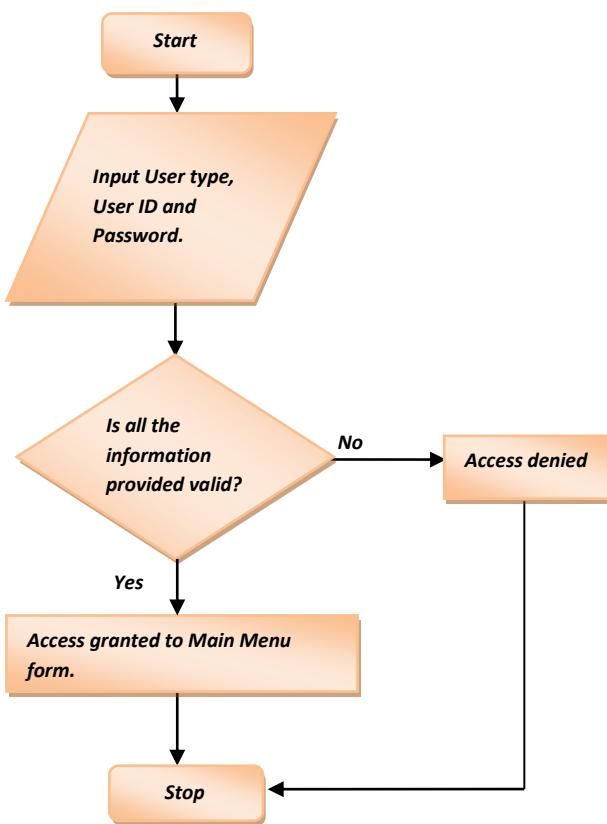
This table contains information on the servicing details of the agency. It receives and sends information to the “Vehicle Servicing” form.

❖ Forms:

- There are 14 forms. The form names are listed below –
 1. Log-in form
 2. Main (Navigation) form
 3. User Creation form
 4. Employee Entry form
 5. Package Entry
 6. Package Booking
 7. Package Booking Update
 8. Package Expenses Entry
 9. Vehicle Entry
 10. Vehicle Update
 11. Vehicle Booking
 12. Vehicle Booking Update
 13. Vehicle Billing
 14. Vehicle servicing
 15. Fuel Record
 16. Backup



1. Login form:



The input User type, User ID and password will be verified with the predefined data provided while making the User account, stored in the User log-in table. Error message will be shown if the information provided is not found in the database or if invalid information is provided. If the information provided is valid then the User will get access to the Main Menu. Only the CEO and the Executive officers will have “Administrator” accounts with special rights. The staff will obtain the “General” accounts.

User Log-in	
User Log-in	
Date	<input type="text"/>
User type	<input type="text"/> <input checked="" type="checkbox"/>
User ID	<input type="text"/>
Password	<input type="text"/>
Log-in	Cancel

2. Main Menu form:

Main Menu						
Tools	Package Information	Vehicle Information	User information	Employee information	Reports	User Documentation
Backup	Package Entry	Vehicle Entry	User Creation	Employee Entry	Employee details	
Help	Package Expenses entry	Vehicle Update	User log-in		Fuel record	
	Package Booking	Vehicle Booking			List of Packages	
	Package Booking Update	Vehicle Booking Update			List of Vehicles	
		Vehicle Billing			Customer list of Package Booking	
		Vehicle Servicing			Customer list of Vehicle Booking	
					Takings from Vehicle Rentals	

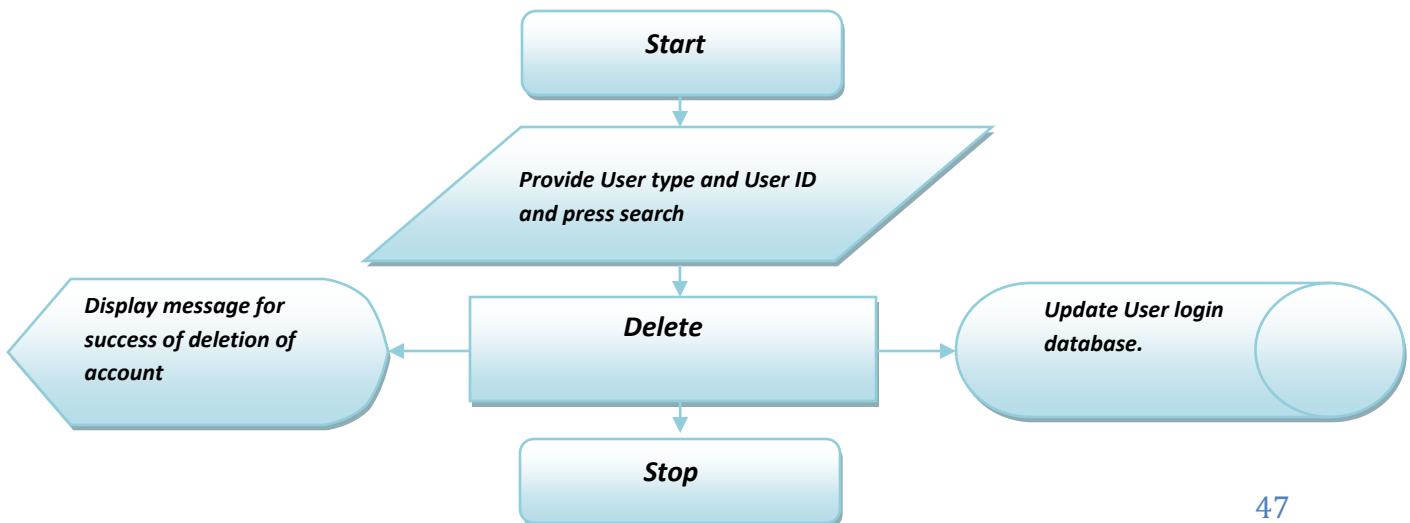
This form contains 6 options. They are Files, Package information, Vehicle information, User data, Fuel data and Employee information. Files contains: Backup and Help. Package Information contains: Package Entry, Package Expenses, Package Booking and Package Booking Update. Vehicle Information contains: Vehicle Entry, Vehicle Update, Vehicle Booking, Vehicle Booking Update, Vehicle Billing and Vehicle Servicing. User data contains: User Creation and User log-in. Fuel data contains Fuel records. Employee information contains: Employee information.

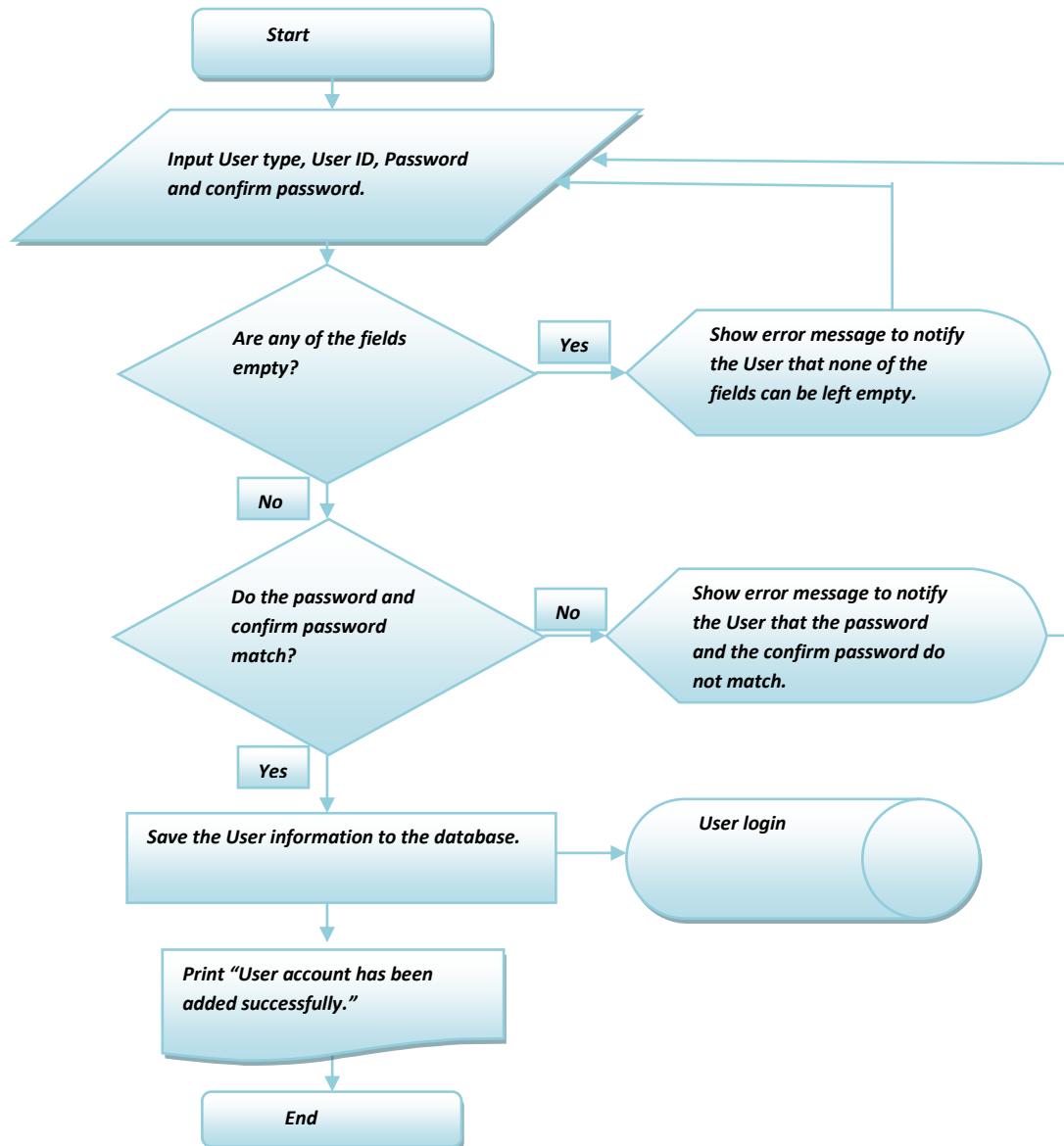
3. User Creation:

User Account Creation	
User Creation	
<i>User type</i>	<input type="text"/> <input checked="" type="checkbox"/>
<i>User ID</i>	<input type="text"/>
<i>Password</i>	<input type="text"/>
<i>Confirm Password</i>	<input type="text"/>
Date	<input type="text"/>
<input type="button" value="Create Account"/> <input type="button" value="Search"/> <input type="button" value="Update"/> <input type="button" value="Refresh"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>	

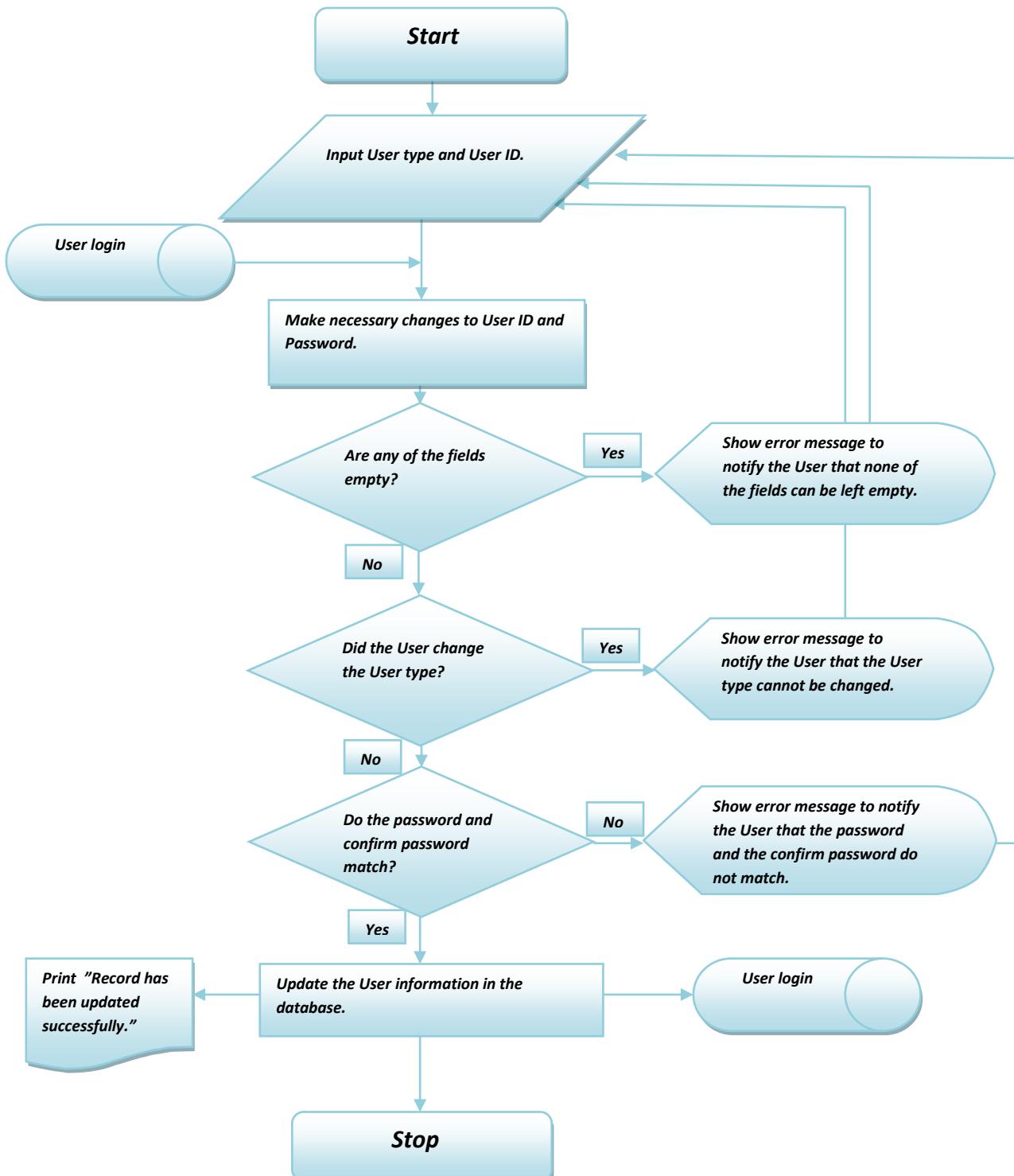
This form is used to enter new user data into the User login database. It is used in case the agency has employed new staff, in case any of the staff members want to change their User data or if any of the staff is fired. A “Confirm Password” box is used to ensure the User provides the correct desired password for the creation of the account. Refresh button allows the User to start anew. Delete button allows records to be deleted easily from the database.

1. Deletion of existing User:



2. Adding a new User account:

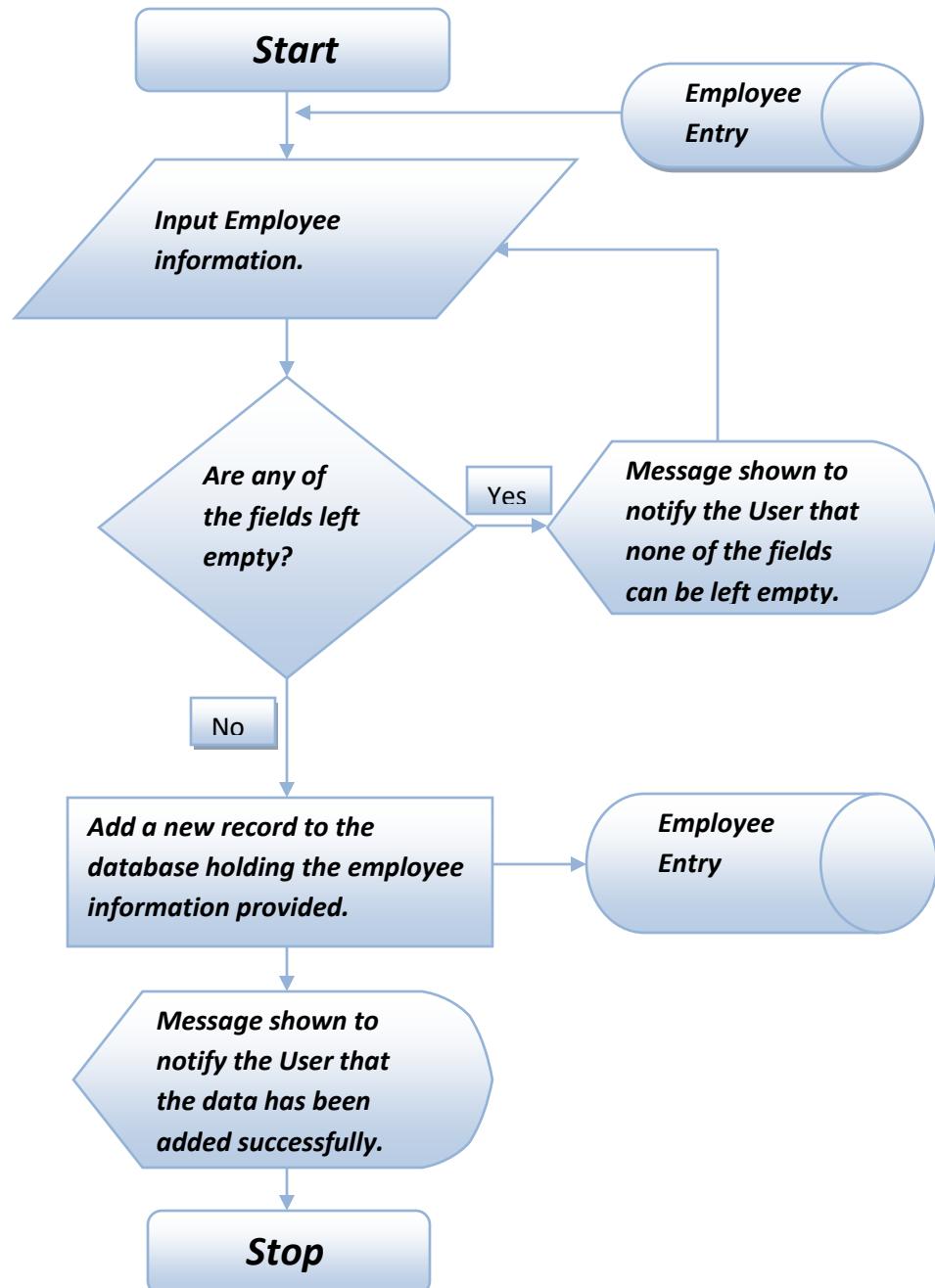
3. Updating User Information:



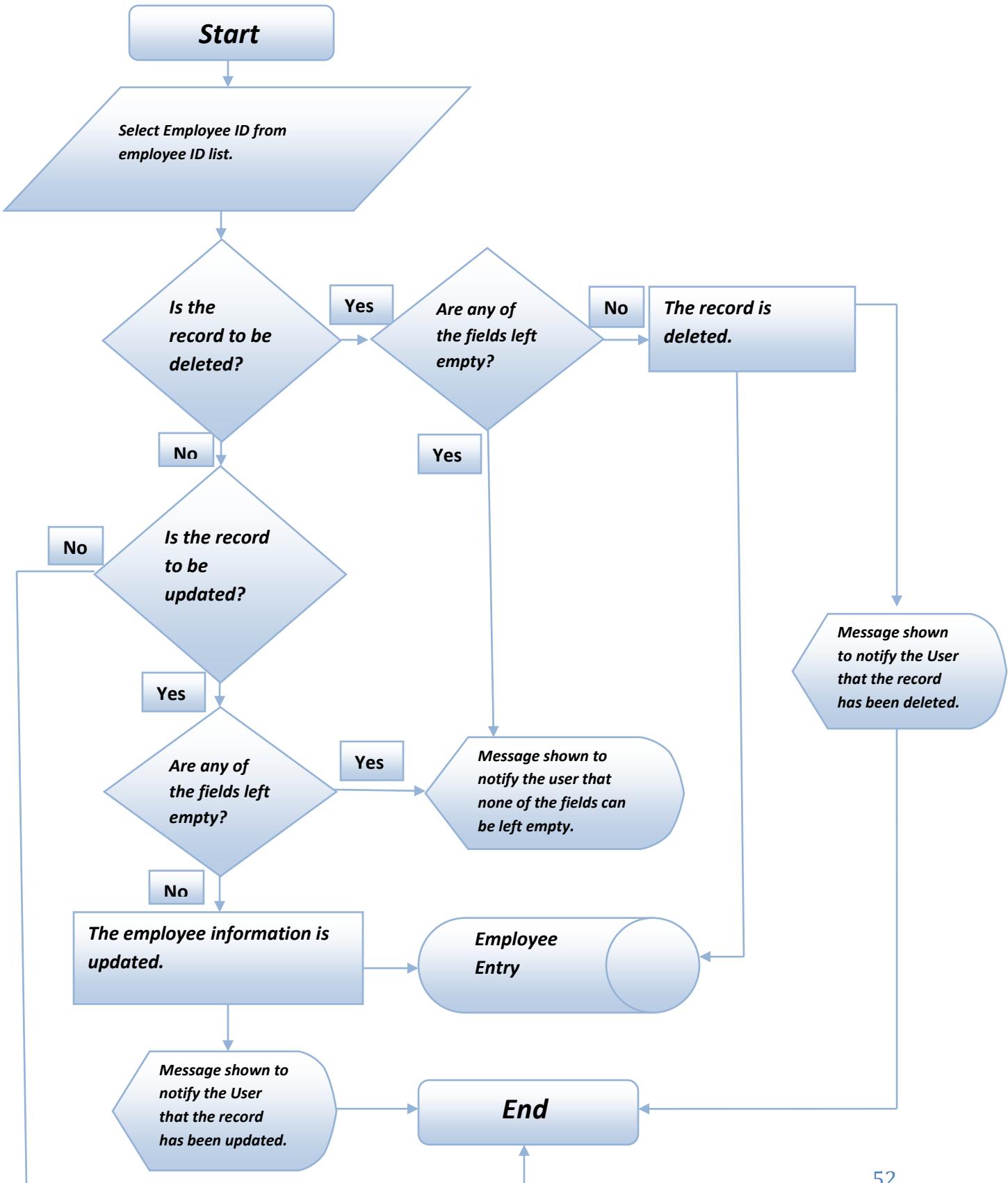
4. Employee Entry form:

Employee Entry		
EMPLOYEE ENTRY		
Date <input type="text"/>		
Employee information <i>Employee ID</i> <input type="text"/> <i>Name of Employee</i> <input type="text"/> <i>Address</i> <input type="text"/> <i>Contact number</i> <input type="text"/> <i>Designation</i> <input type="text"/> <i>Basic salary</i> <input type="text"/>	Update <input type="text"/> <i>Employee ID list</i> <input type="text"/> <input type="button" value="Update"/> <input type="button" value="Delete"/>	
<input type="button" value="Make Entry"/>	<input type="button" value="Refresh"/>	<input type="button" value="Cancel"/>

This form is used to enter the information of the employee into the database. This form is used when a new employee joins, or when an existing employee wants to update his her personal information, or when an employee is fired and the existing information of the employee is to be deleted from the database. A list of Employee IDs is present on the Update section of the form. The employee information is loaded into the form from the database at the click of an Employee ID. After that the Update button can be used to update the employee information or the Delete button could be used to delete the record.

1. Adding Employee Information to the database:

2. Updating or deleting employee information:



5. Package Entry:

Package Entry

Package Entry

Date

Package information

Package type

Name of Package

Package ID

Cost per person

Accommodation type

Journey Date

Arrival time

Starting time

Duration Days

Update

Package type

Package Name list

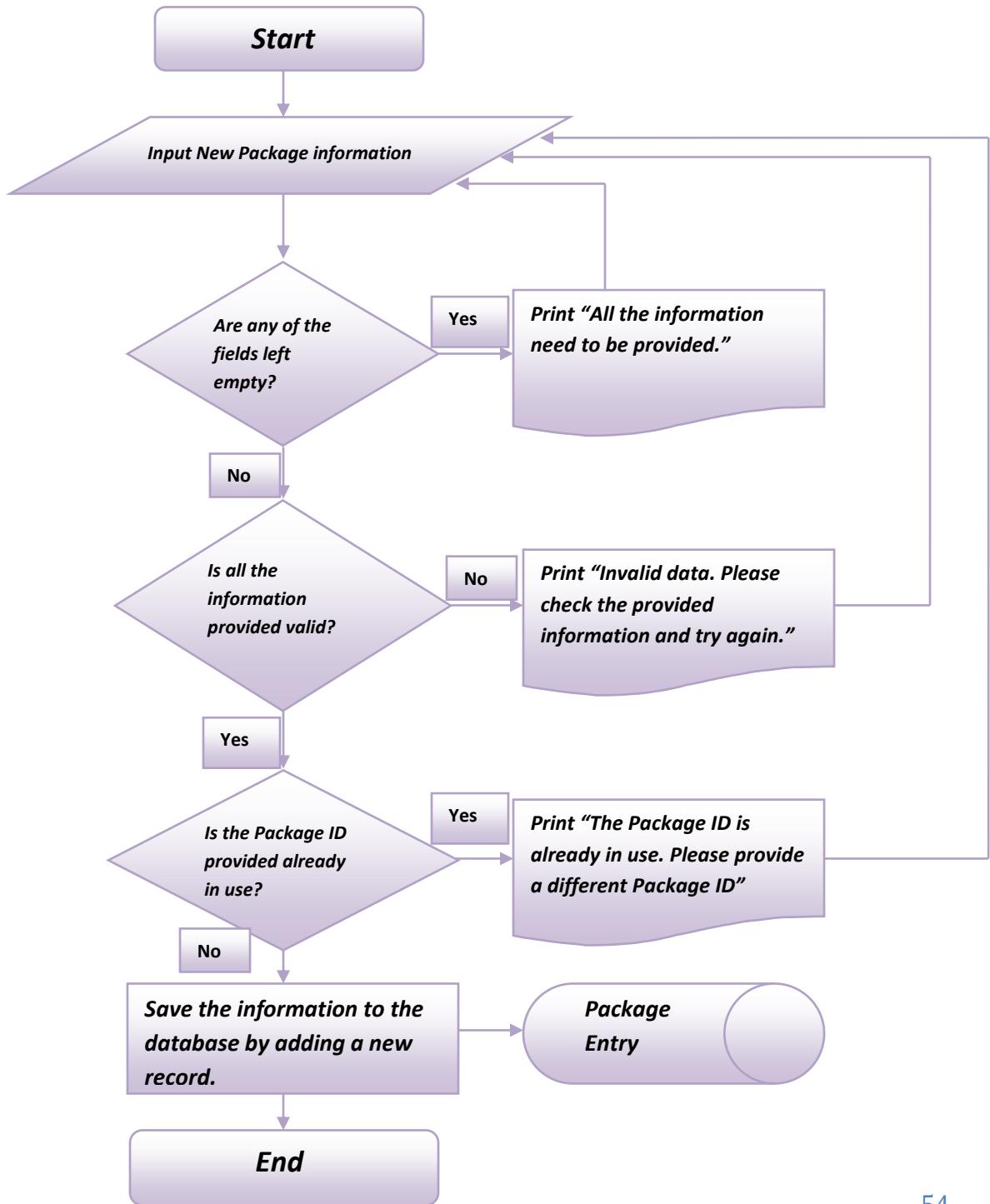
Number of reservations available?

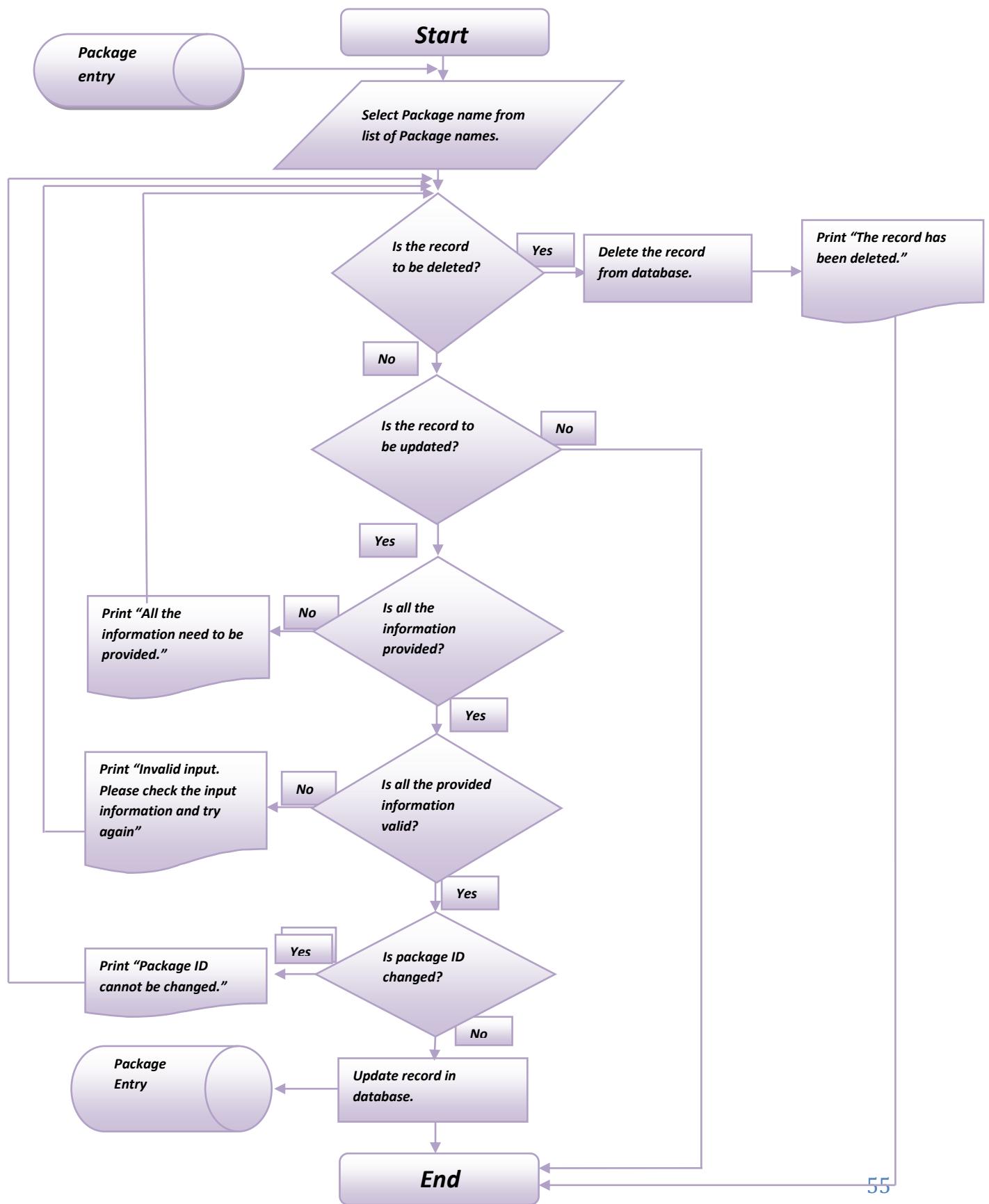
Itinerary

Package details

This form can be used to add new packages to the Package Entry table of the database. It is particularly used to add new packages to the database. It can also be used to update the data of the existing packages or delete any of the existing packages. The form contains an update section which contains the list of names of all the packages that are provided by the agency. The desired package can be selected from the list in case any alterations need to be made or the information needs to be reviewed. The package information is updated automatically at the selection of the package.

1. Adding new Packages to the database:



2.Package update or delete:

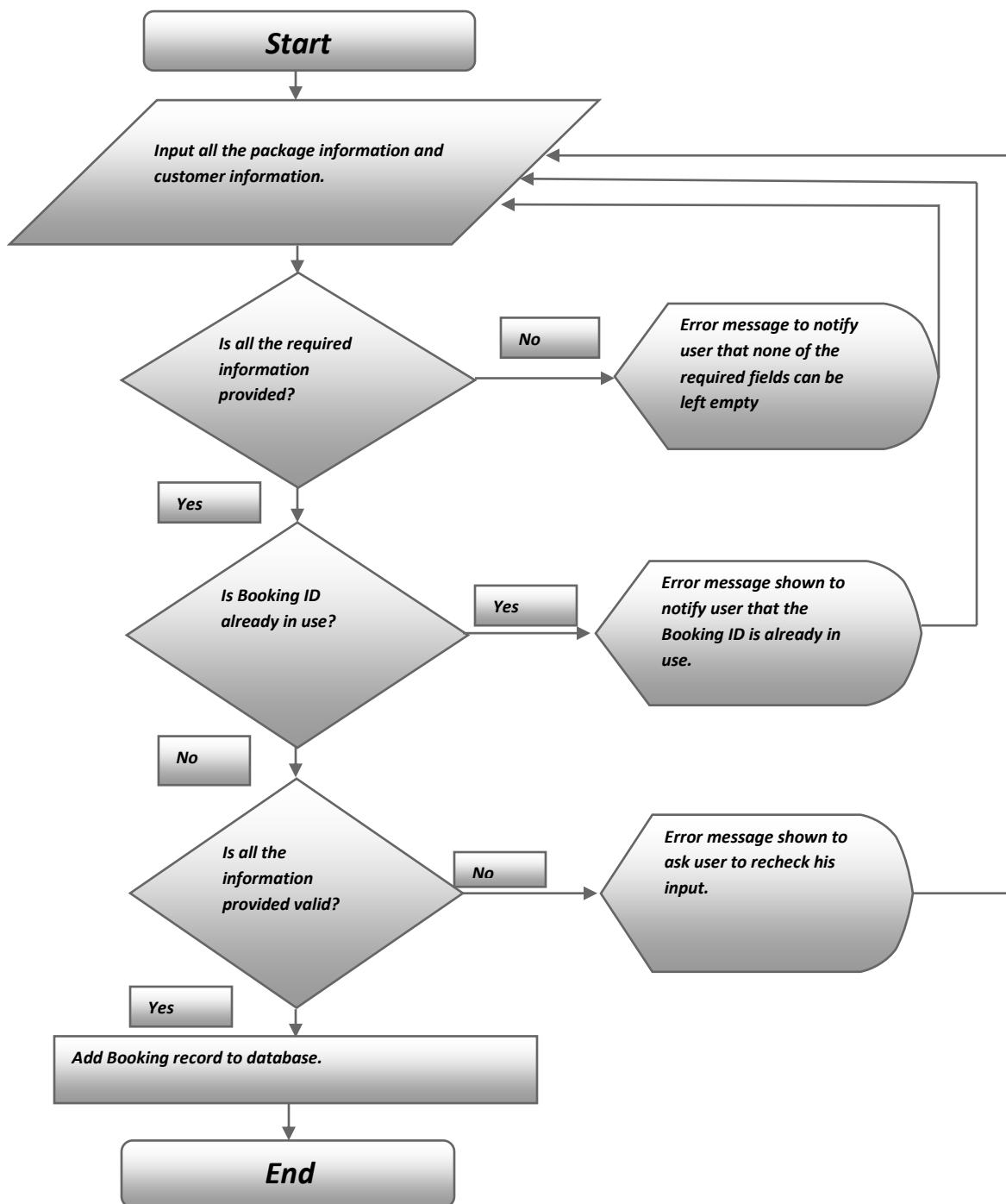
6. Package Booking form:

Package Booking

Date **Package Booking****Package Specifications****Package type****Package name****Package ID****Cost per person****Number of people****Net total****Information on Package****Itinerary****Package Details****Accommodation type****Journey date****Arrival time****Starting time****Duration** Days**Number of vacancies****Customer Details****Booking ID****Customer name****Customer address****Contact number****Confirm****Refresh****Cancel**

This form can be used to add Bookings to the database. The fields are separated for Package specifications, information on Package and customer information to reduce confusion. The list for Package names is loaded according to the choosing of Package type of the user. When a package name is chosen from the list the Information on Package is automatically updated, along with other Package specifications details such as cost per person, package ID etc. All these information is loaded from Package Entry table of the database. The provided information is checked for presence of all the valuable information, validity and if Booking ID used by the User is already in use to prevent errors. After all the validation checks are done, the information is saved to the Package Booking table of the database.

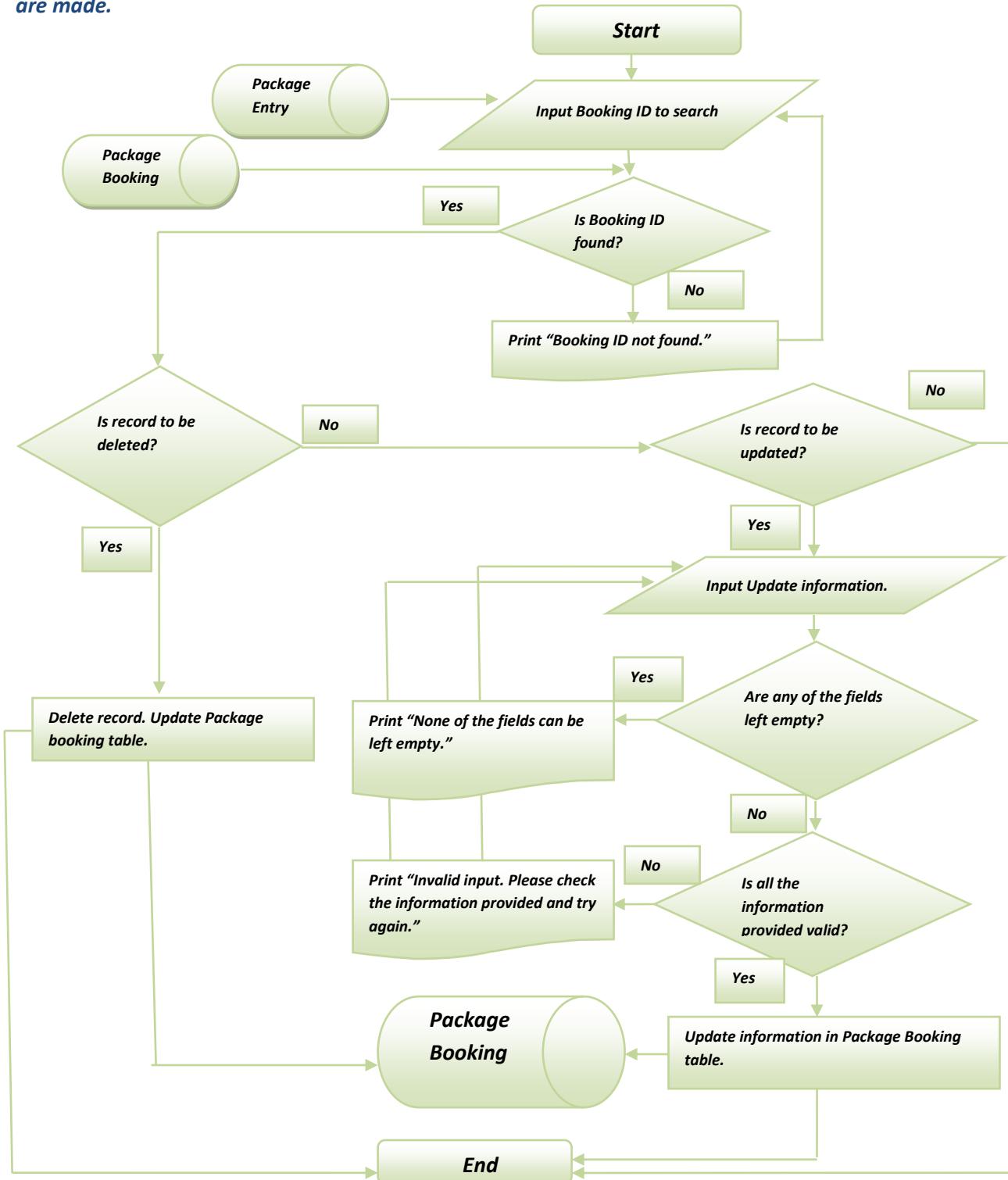
Adding information to the form:



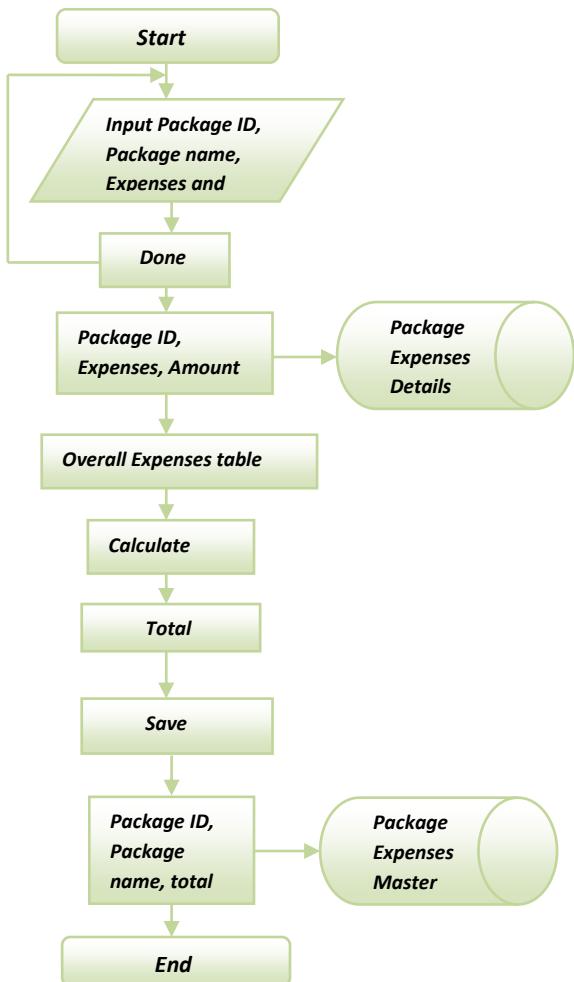
7. Package Booking Update:

Package Booking Update		
<h3>Package Booking Update</h3>		
Date <input type="text"/>		
Customer Details <i>Booking ID</i> <input type="text"/> <i>Customer name</i> <input type="text"/> <i>Customer address</i> <input type="text"/> <i>Contact number</i> <input type="text"/>	Information on Package <i>Itinerary</i> <input type="text"/> <i>Package Details</i> <input type="text"/>	
Package Specifications <i>Package type</i> <input type="text"/> <input type="button" value="▼"/> <i>Package name</i> <input type="text"/> <input type="button" value="▼"/> <i>Package ID</i> <input type="text"/> <i>Cost per person</i> <input type="text"/> <i>Number of people</i> <input type="text"/> <i>Net total</i> <input type="text"/>	 <i>Accommodation type</i> <input type="text"/> <i>Journey date</i> <input type="text"/> <i>Arrival time</i> <input type="text"/> <i>Starting time</i> <input type="text"/> <i>Number of vacancies</i> <input type="text"/> Days <input type="text"/>	Actions <input type="button" value="Search"/> <input type="button" value="Update"/> <input type="button" value="Refresh"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>

This form receives data from Package Entry table and Package Booking table of the database. The information stored in the Package Booking database can be altered here. Information of the database can be edited or deleted here. The database information is automatically updated after any changes are made.



8. Package Expenses Entry:



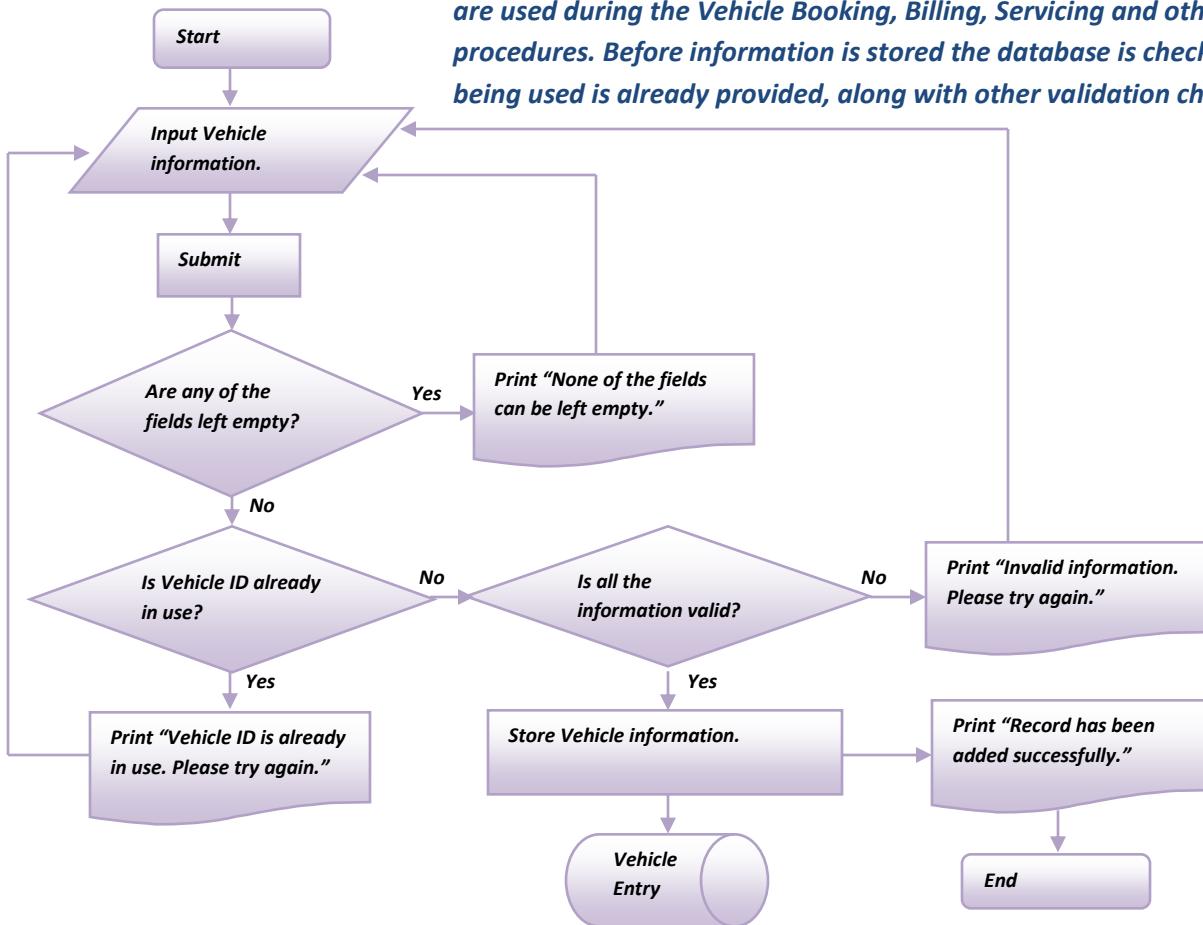
This form keeps track of all the expenditure due to Packages by the agency. It receives package information from the Package Entry table and saves data to two different tables: the Package Master table and the Package details table.

Package Expenses Entry																				
Package Expenses Entry																				
Date <input type="text"/>																				
Package Expenses Information																				
Package ID	<input type="text"/>																			
Package Name	<input type="text"/>																			
Expenses	<input type="text"/> <input checked="" type="checkbox"/>																			
Amount	<input type="text"/>																			
Done																				
<table border="1"> <thead> <tr> <th>Package ID</th> <th>Expenses</th> <th>Amount</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </tbody> </table> <div style="display: flex; justify-content: space-between;"> Total <input type="text"/> <div style="border: 1px solid black; padding: 5px;"> <input type="button" value="Calculate"/> <input type="button" value="Refresh"/> <input type="button" value="Save"/> <input type="button" value="Cancel"/> </div> </div>			Package ID	Expenses	Amount															
Package ID	Expenses	Amount																		

9. Vehicle Entry form:

Vehicle Entry	
Vehicle Entry	
Date	<input type="text"/>
Vehicle Type	<input type="text"/> <input checked="" type="checkbox"/>
Vehicle ID	<input type="text"/>
Make	Registration number <input type="text"/>
Model <input type="text"/>	
Number of seats <input type="text"/>	Rate per kilometer <input type="text"/>
<input type="button" value="Submit"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>	

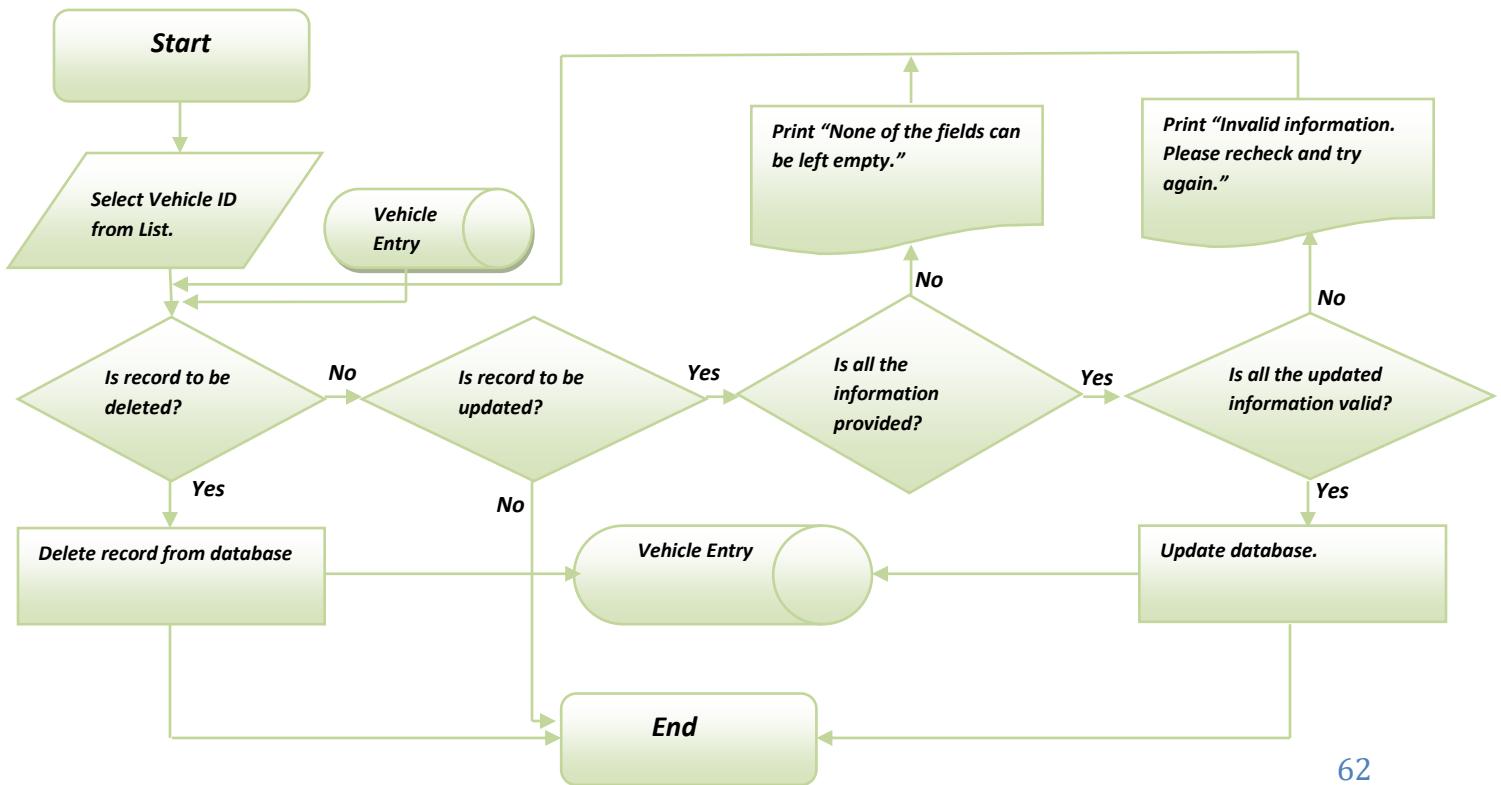
This form is connected to the Vehicle Entry table of the database. It allows the system to store the information of a new vehicle of the agency to the database. These information are used during the Vehicle Booking, Billing, Servicing and other Vehicle related procedures. Before information is stored the database is checked to ensure the Vehicle ID being used is already provided, along with other validation checks and presence checks.



10. Vehicle Update:

Vehicle Update											
<h3 style="text-align: center;">Vehicle Update</h3> <div style="text-align: right; margin-bottom: 5px;"> Date <input type="text"/> </div> <table border="0" style="width: 100%;"> <tr> <td style="width: 30%; vertical-align: top;"> Vehicle ID <input type="text"/> </td> <td style="width: 70%; vertical-align: top;"> Vehicle Information <table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> Vehicle type <input type="text"/> </td> <td style="width: 50%; vertical-align: top;"> Number of seats <input type="text"/> </td> </tr> <tr> <td style="vertical-align: top;"> Registration number <input type="text"/> </td> <td style="vertical-align: top;"> Rate <input type="text"/> </td> </tr> <tr> <td style="vertical-align: top;"> Make <input type="text"/> </td> <td style="vertical-align: top;"> Model <input type="text"/> </td> </tr> </table> </td> </tr> <tr> <td colspan="2" style="text-align: center; padding-top: 10px;"> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/> </td> </tr> </table>		Vehicle ID <input type="text"/>	Vehicle Information <table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> Vehicle type <input type="text"/> </td> <td style="width: 50%; vertical-align: top;"> Number of seats <input type="text"/> </td> </tr> <tr> <td style="vertical-align: top;"> Registration number <input type="text"/> </td> <td style="vertical-align: top;"> Rate <input type="text"/> </td> </tr> <tr> <td style="vertical-align: top;"> Make <input type="text"/> </td> <td style="vertical-align: top;"> Model <input type="text"/> </td> </tr> </table>	Vehicle type <input type="text"/>	Number of seats <input type="text"/>	Registration number <input type="text"/>	Rate <input type="text"/>	Make <input type="text"/>	Model <input type="text"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>	
Vehicle ID <input type="text"/>	Vehicle Information <table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> Vehicle type <input type="text"/> </td> <td style="width: 50%; vertical-align: top;"> Number of seats <input type="text"/> </td> </tr> <tr> <td style="vertical-align: top;"> Registration number <input type="text"/> </td> <td style="vertical-align: top;"> Rate <input type="text"/> </td> </tr> <tr> <td style="vertical-align: top;"> Make <input type="text"/> </td> <td style="vertical-align: top;"> Model <input type="text"/> </td> </tr> </table>	Vehicle type <input type="text"/>	Number of seats <input type="text"/>	Registration number <input type="text"/>	Rate <input type="text"/>	Make <input type="text"/>	Model <input type="text"/>				
Vehicle type <input type="text"/>	Number of seats <input type="text"/>										
Registration number <input type="text"/>	Rate <input type="text"/>										
Make <input type="text"/>	Model <input type="text"/>										
<input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>											

The Vehicle information can be altered using this form. It receives data and save to Vehicle Entry form. A list of Vehicle IDs exists on the form from which when an ID is selected the information is automatically loaded onto the form. The data can be edited and the information in database updated on click of Update button, or the selected record can be deleted from the database table by pressing delete.



11. Vehicle Booking:

Vehicle Booking

Vehicle Booking

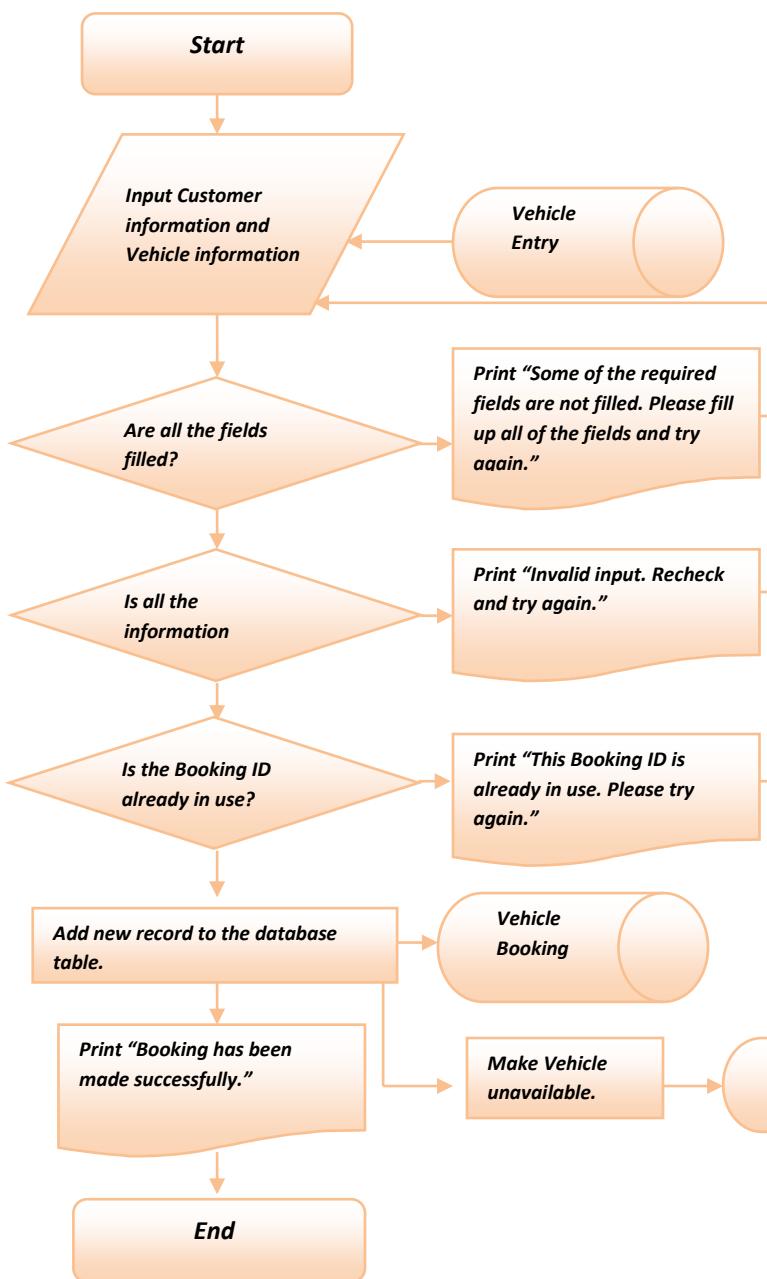
Date

Customer Information

<i>Booking ID</i>	<input type="text"/>	<i>Address</i>	<input type="text"/>
<i>Customer Name</i>	<input type="text"/>		
<i>Contact number</i>	<input type="text"/>		
<i>Journey Date</i>	<input type="text"/> <input checked="" type="button" value="▼"/>	<i>Drop-off Date</i>	<input type="text"/> <input checked="" type="button" value="▼"/>

Vehicle Information

<i>Vehicle type</i>	<input type="text"/>	<i>Registration number</i>	<input type="text"/>
<i>Make</i>	<input type="text"/>	<i>Model</i>	<input type="text"/>
<i>Vehicle ID</i>	<input type="text"/>		
<i>Rate per kilometer</i>	<input type="text"/>	<i>Advance paid</i>	<input type="text"/>
<i>Number of seats</i>	<input type="text"/>	<i>Driver name</i>	<input type="text"/>
<i>Starting kilometer</i>	<input type="text"/>		



This form is used to make bookings of vehicles. It contains two sections: Customer information and Vehicle information, both of which have to be filled completely in order to make a booking. The form receives data from Vehicle Entry table of the database, which is needed to show the customer the options the agency provides. The User is provided with different choices based on his/her previous choice. Once an option is chosen the required information automatically appears. The information is loaded on the form according to the choice of the User. Some data such as all the Customer information, Advance paid and Starting kilometer etc. need to be provided by the customer. The provided information is checked for validity and presence of all necessary information before storage. After filling the form, the confirm button is used to store the information into the database, completing the Booking procedure.

12. Vehicle Booking Update:

Vehicle Booking Update

Vehicle Booking Update

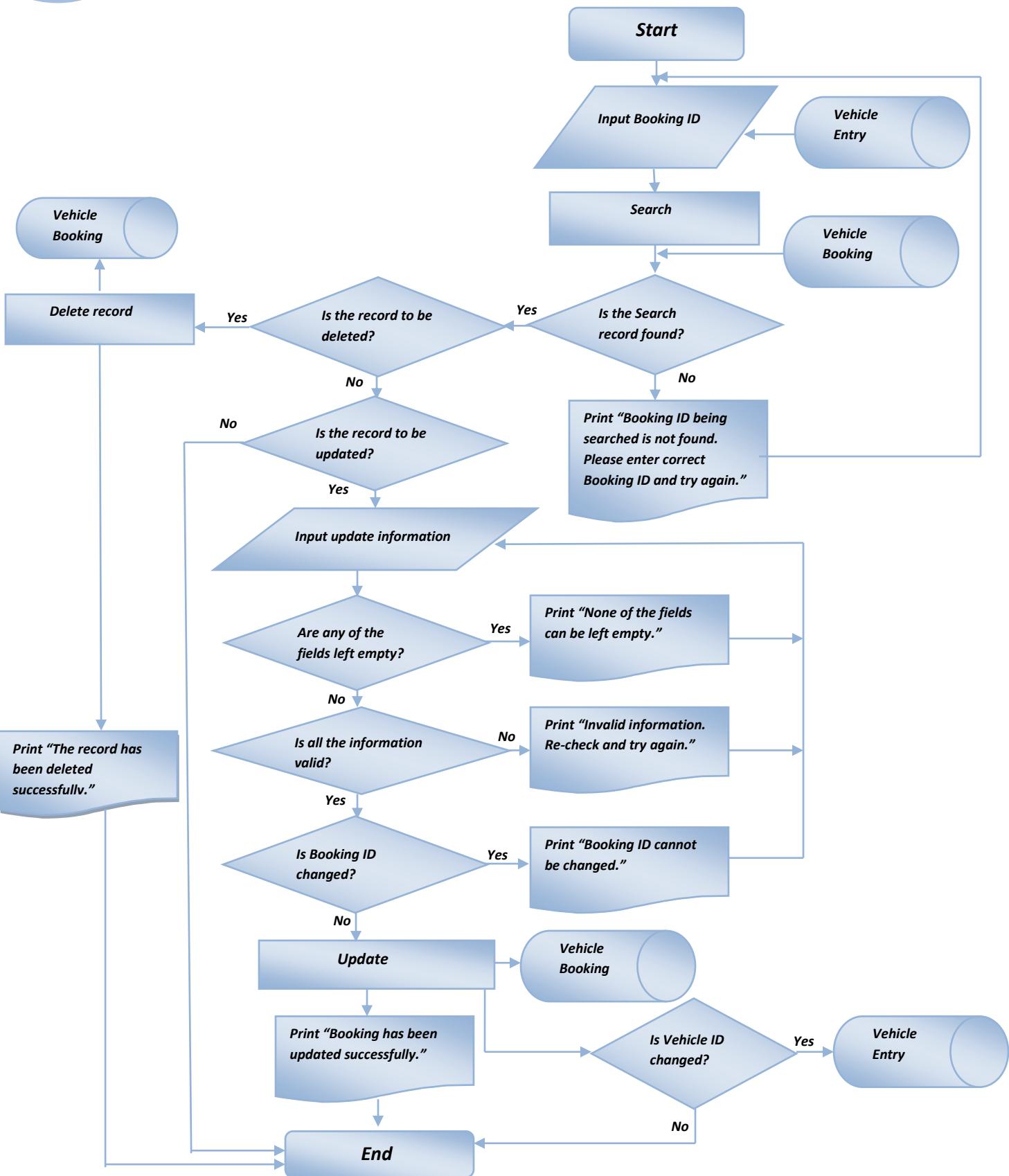
Date

Customer Information Booking ID <input type="text"/> Customer name <input type="text"/> Address <input type="text"/> Contact number <input type="text"/> Journey date <input type="text"/> <input type="button" value="V"/> Drop-off date <input type="text"/> <input type="button" value="V"/> Booking date <input type="text"/>	Vehicle Information Vehicle type <input type="text"/> <input type="button" value="V"/> Make <input type="text"/> <input type="button" value="V"/> Model <input type="text"/> <input type="button" value="V"/> Vehicle ID <input type="text"/> <input type="button" value="V"/> Registration number <input type="text"/> Number of seats <input type="text"/> Rate per kilometer <input type="text"/> Advance paid <input type="text"/> Driver name <input type="text"/> Starting kilometer <input type="text"/>	<input type="button" value="Search"/> <input type="button" value="Delete"/> <input type="button" value="Update"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>
---	--	--

This form shows Vehicle Booking information, loaded from Vehicle Booking table. The Vehicle Booking information can be edited or deleted as the User wishes to change something. This form receives data from Vehicle Entry table and Vehicle Booking table of the database. The Booking ID needs to be entered and the search button pressed to load the Booking information of a particular customer to the form. Error message is shown if the Booking ID is not found in the database. After any alterations the data in Vehicle Booking table is automatically updated at the press of the Update button, or the record deleted at pressing delete button. Vehicle Entry information is updated for availability of vehicle after every transaction.

Jan. 1

Travel Agency Management System



13. Vehicle Billing:

Vehicle Billing

Vehicle BillingDate **Customer Information**

Booking ID	<input type="text"/>
Customer Name	<input type="text"/>
Contact number	<input type="text"/>
Address	<input type="text"/>

Search**Cancel****Billing Information****Cost of travel**

Starting kilometer	<input type="text"/>
Ending kilometer	<input type="text"/>
Kilometer travelled	<input type="text"/>
Rate per kilometer	<input type="text"/>
Cost of travel	<input type="text"/>

Billing

Booking charge	<input type="text"/>
Driver charge	<input type="text"/>
Cost of travel	<input type="text"/>
Late fine (If any)	<input type="text"/>
Additional expenditures	<input type="text"/>
Gross total	<input type="text"/>

Vehicle Information

Vehicle type	<input type="text"/>
Make	<input type="text"/>
Model	<input type="text"/>
Vehicle ID	<input type="text"/>
Registration number	<input type="text"/>
Driver name	<input type="text"/>
Number of seats	<input type="text"/>

Dates

Booking date	<input type="text"/> <input type="button" value="▼"/>
Journey date	<input type="text"/> <input type="button" value="▼"/>
Drop off date	<input type="text"/> <input type="button" value="▼"/>
Billing date	<input type="text"/> <input type="button" value="▼"/>
Late fine (If any)	<input type="text"/>

(Less)

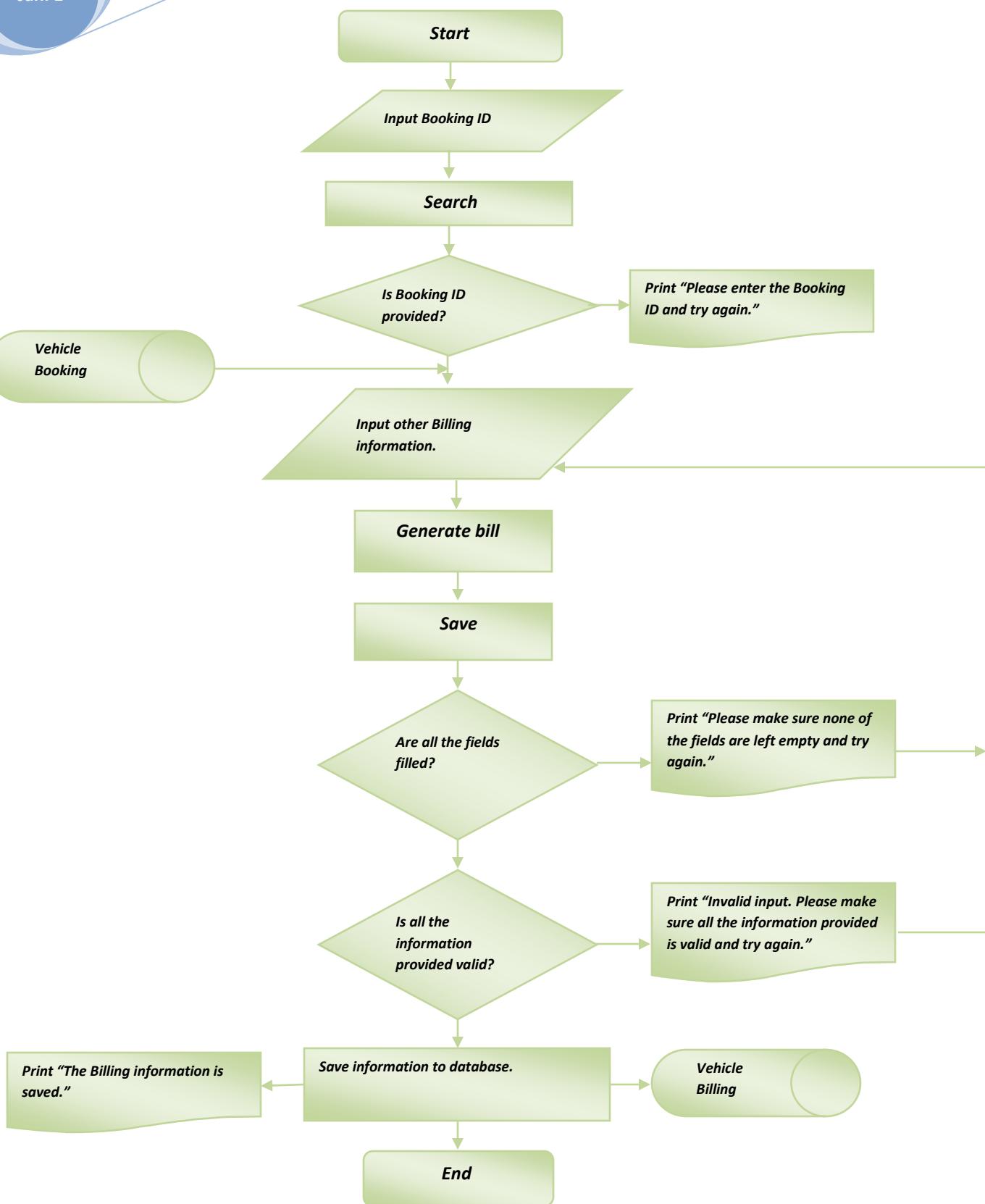
Discount (If any)	<input type="text"/>
Advance paid	<input type="text"/>

Net total**Generate Bill** **Save** **Cancel**

Billing for the Vehicle rental is done with this form. The Booking ID is typed in and the search button is clicked which loads rest of the information from the database, if the Booking ID is found. The rest of the necessary information is then provided by the User and the "Generate bill" button is used to carry out the rest of the calculations by the system. Then the information is saved to the "Vehicle Billing" database table by pressing the save button. "Print" button is used to print the bill.

Jan. 1

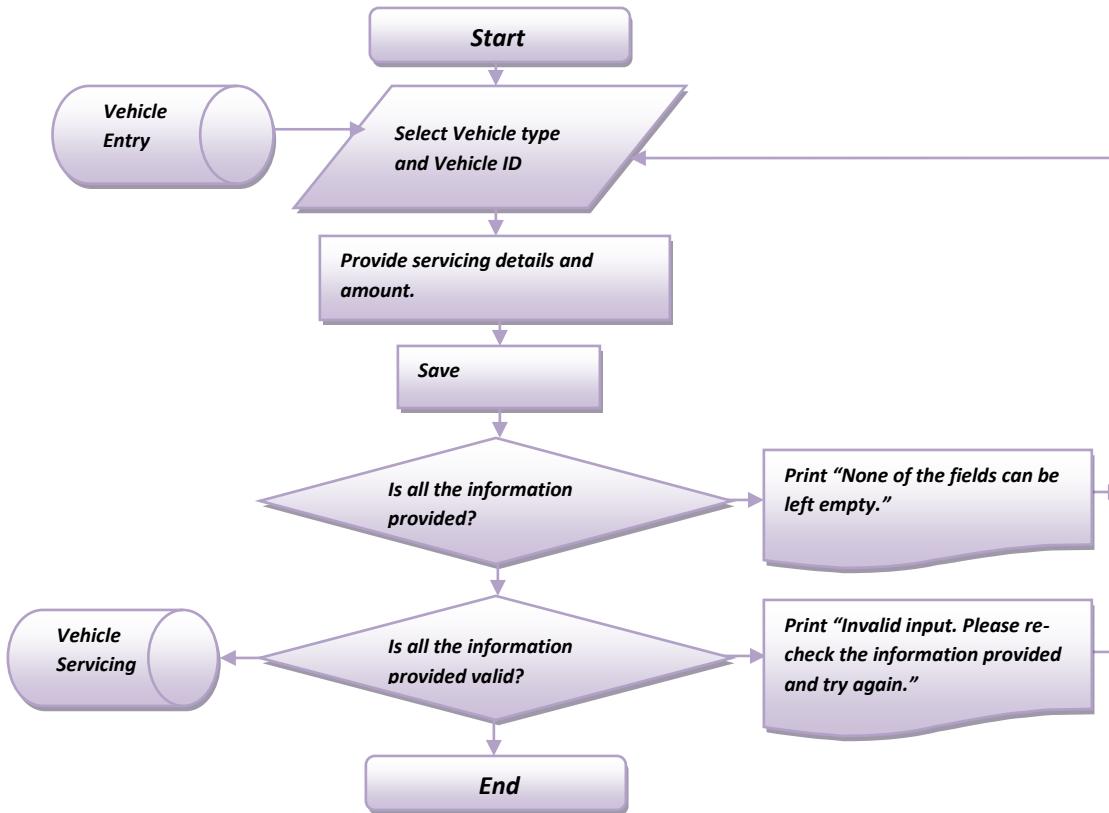
Travel Agency Management System



14. Vehicle Servicing:

Vehicle Servicing

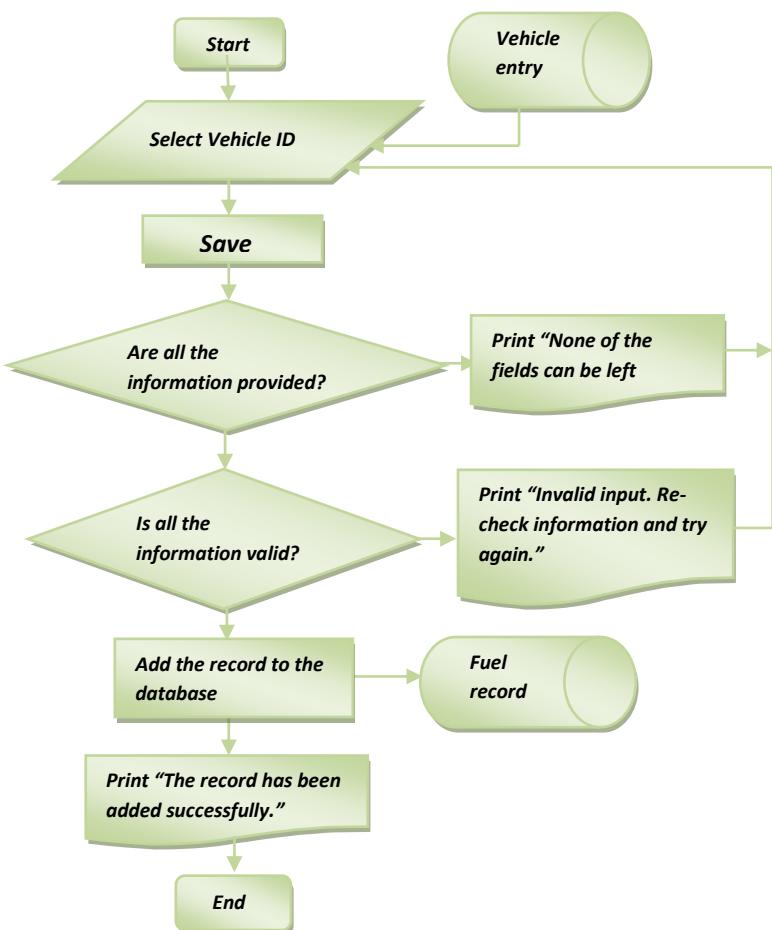
Vehicle type <input type="text"/>	Date <input type="text"/>												
Vehicle ID <input style="width: 100%; height: 150px; margin-bottom: 5px;" type="text"/>													
Servicing details <input type="text"/> Amount <input type="text"/> <div style="display: flex; justify-content: space-around; width: 100%;"> <input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/> </div>													
Servicing history preview <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Date</th> <th>Servicing details</th> <th>Amount</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </tbody> </table>		Date	Servicing details	Amount									
Date	Servicing details	Amount											



This form views all the servicing information of a vehicle. It is also used to input new servicing information into the database. It receives vehicle information from the Vehicle Entry table and stores servicing information to the Vehicle Servicing table. The Vehicle ID list is updated on selection of a Vehicle type and the servicing information is automatically loaded on selection of a Vehicle ID. Any new transaction is automatically added to the end of "Servicing history preview" table.

15. Fuel Record:

Fuel Record	
Fuel Record	
<p>Vehicle ID <input type="text"/> <input type="button" value="▼"/></p> <p>Fuel type <input type="text"/> <input type="button" value="▼"/></p>	<p>Date <input type="text"/></p> <p>Quantity <input type="text"/></p> <p>Cost <input type="text"/></p>
<input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>	

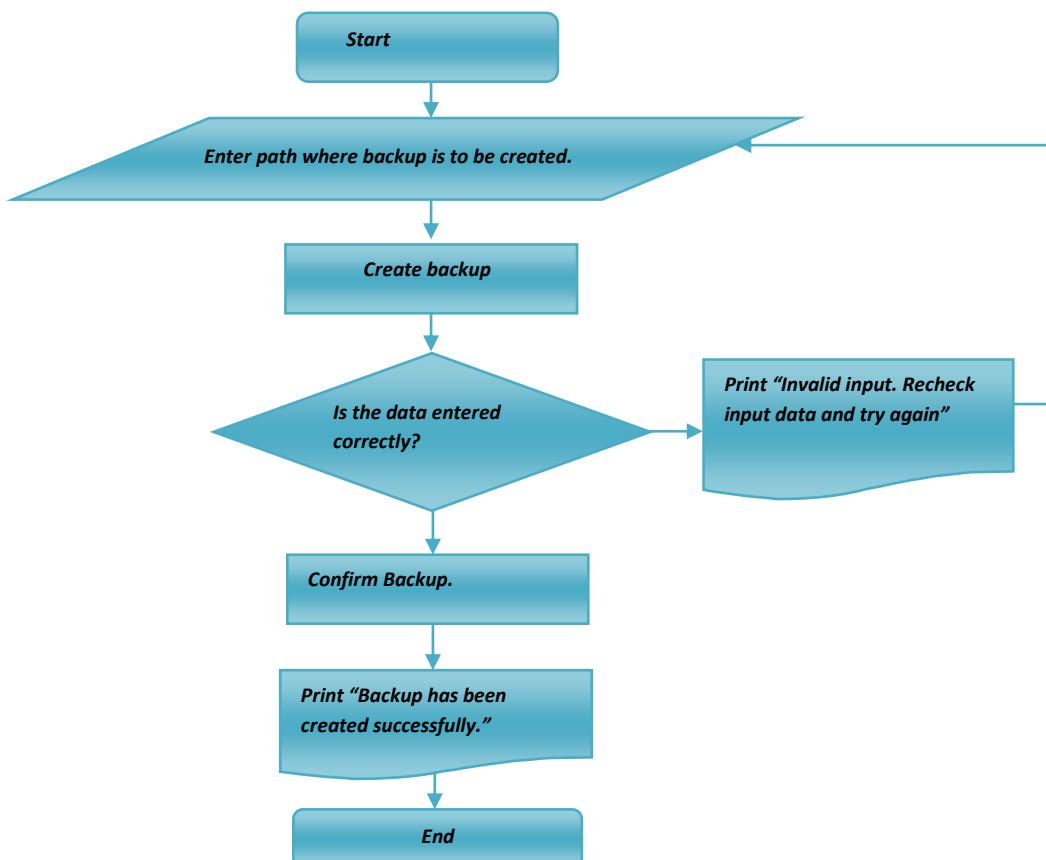


This form is used to save the necessary information involving fuel to the database. A particular Vehicle ID is selected and the other information is input. The form receives the Vehicle IDs from the Vehicle Entry table and the information is saved to the Fuel Record table by clicking the save button.

Back-up form:

This form is used to backup data.

Backup	
Choose Directory for target file	Choose directory for Backup file
<input type="text"/>	<input type="text"/>
<input type="button" value="Backup files"/>	<input type="button" value="Cancel"/>



◆ Reports:

- *Employee Details report:*

**Logo**Date

Employee Details

Employee ID	Employee Name	Address	Contact number	Designation	Basic salary	Date of employment

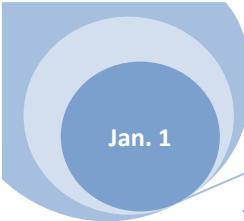
- *Fuel Record report:*

**Logo**

Fuel Record

Date

Vehicle ID	Fuel type	Quantity	Cost	Date of entry



Jan. 1

Travel Agency Management System

➤ *List of Packages:*



Logo

List of Packages

Package type	Package name	Package ID	Itinerary	Package details	Cost per person	Accommodation type	Journey date	Duration	Date of entry

➤ *List of Vehicles:*



Logo

List of Vehicles

Vehicle type	Vehicle ID	Registration number	Make	Model	Number of seats	Rate per kilometer	Date of entry

Jan. 1

Travel Agency Management System

➤ *Customer list for Package Booking:*



Logo

Customer List for Package Booking

Package ID

Booking ID	Customer name	Address	Contact number	Number of people	Net total	Date of Booking
------------	---------------	---------	----------------	------------------	-----------	-----------------

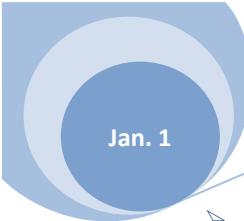
➤ *Customer list of Vehicle Booking:*



Logo

Customer List for Vehicle Booking

Booking ID	Customer name	Address	Contact number	Vehicle ID	Advance paid	Date of Booking	Journey date	Drop off date	Driver name
------------	---------------	---------	----------------	------------	--------------	-----------------	--------------	---------------	-------------



Jan. 1

Travel Agency Management System

- *List of takings from Vehicle rentals:*

Logo

List of Takings from Vehicle Rentals

Booking ID

Customer name

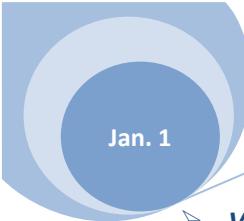
Vehicle ID

Net total

➤ *Lists of Package Expenses:*

Package Expenses list		
Package ID	Expenses	Amount

List of Total Package Expenses		
Package ID	Expenses	Amount



Jan. 1

Travel Agency Management System

➤ **Vehicle Servicing Details list:**



Vehicle Servicing Details List

Vehicle type	Vehicle ID	Servicing Details	Amount	Date of Service
--------------	------------	-------------------	--------	-----------------

➤ **Package Booking Receipt:**

Logo	<u>Package Booking Receipt</u>
	Date <input type="text"/>
Booking ID	<input type="text"/>
Date of Booking	<input type="text"/>
Customer name	<input type="text"/>
Package ID	<input type="text"/>
Cost per person	<input type="text"/>
Number of people	<input type="text"/>
Net total	<input type="text"/>

➤ ***Vehicle Rental Token:***

	<u>Vehicle Rental Token</u>
Date of Booking	
Booking ID	
Customer name Vehicle ID Rate per Kilometer Starting Kilometer Journey date Drop-off date Driver name	
Advance paid	

➤ ***Vehicle Rental bill:***

	<u>Vehicle Rental Bill</u>	Date of Billing
Booking ID Customer name Journey date Drop-off date Kilometers travelled Rate per kilometer		Booking charge Driver charge Cost of travel Late fine Additional Expenditures Gross total (Less) Discount Advance paid Net total

Validation checks:

Validation checks are used to reduce the amount of inappropriateness of records and cut off human errors including mistyped errors since most of the data will be input through the keyboard. It will also minimize calculation mistakes made by the users and to reduce the number of errors and inappropriateness of record. A variety of validation checks are used in the system. Examples include: character type check to check the correct type of characters are used (e.g. - numbers cannot be entered as name), presence check (the field cannot be kept empty), and length check (e.g. - phone number must be 11 digits) etc.

(ii) Intended Benefits

- *The new system will be easier to use as it tends the customers to provide the certain necessary information that the agency requires that is fixed by the system. All the necessary information is also generated automatically onto the forms. The system also generates reports such as list of packages, list of vehicles, customer list of package bookings, customer list of vehicle bookings, list of takings from vehicle rentals, list of package expenses, list of total package expenses and list of vehicle servicing automatically. It also produces Package Booking Receipt, Token and Vehicle Rental bill automatically.*
- *The new computerized system is faster than the current system so there will be fewer queues, if any, and the customer can be done with the bookings and billings instantly.*
- *There will be fewer chances of errors as there will be validation checks to make sure all the information is provided correctly, minimizing the errors. Chances of miscalculations will also be minimized. The data in some of the forms will also be loaded automatically, reducing user intervention and further reducing the certainty of errors.*
- *Less amount of paper is required for the new system as the forms do not to be printed anymore and the receipts and bills do not need to be kept (as the information will be stored automatically to the database), creating fewer piles. This will also reduce paper and printing costs.*
- *Entry and updates of new records is easier.*
- *There are fewer chances of any vehicle being booked more than once.*
- *It is easier to keep expenses details of packages and servicing details of vehicles.*

(iii) Limits of the scope of the solution

Table Name	Size of fields in each record in the table/bytes	Estimated number of records in each month/bytes	Size of database in each month/bytes	Estimated size in a year/bytes	Maximum size of database per year/Megabytes	Maximum size of record with overhead/Megabytes
User log-in	30	10	300	3600	$3600/(1024*1024) =0.0034$	0.00374
Vehicle entry	62	2	(This barely changes in a month)	124	$124/(1024*1024) =0.0001$	0.00011
Employee entry	104	10	1040	12480	$12480/(1024*1024) =0.0119$	0.01309
Fuel record	42	100	4200	50400	$50400/(1024*1024) =0.0481$	0.05291
Package entry	256	10	2560	30720	$30720/(1024*1024) =0.0293$	0.03223
Package Booking	342	600	205200	2462400	$2462400/(1024*1024) =2.3483$	2.58313
Vehicle Booking	184	60	11040	132480	$132480/(1024*1024) =0.1263$	0.13893
Package Master	34	100	3400	40800	$40800/(1024*1024) =0.0389$	0.04279
Package details	34	1000	34000	408000	$408000/(1024*1024) =0.3891$	0.42801
Vehicle billing	188	50	9400	112800	$112800/(1024*1024) =0.1076$	0.11836
Vehicle servicing	104	500	52000	624000	$624000/(1024*1024) =0.5951$	0.65461

Maximum size of the files (per year) =

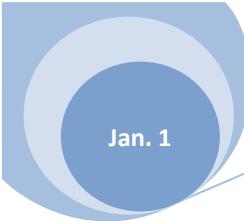
$0.00374+0.00011+0.01309+0.05291+0.03223+2.58313+0.13893+0.04279+0.42801+0.11836+0.65461$

= 4.06791 MB

There are few factors which have to be considered before new system is implemented. The new system is completely different from the existing system. It requires purchase of new devices like printers, UPS, monitor and other parts of computer system including the software which has to be installed. So initially a significant amount needs to be spent by the Agency. Since the Agency uses its own power generator during power supply in case of power failure so this would not be a problem. On the other hand, the processing was previously done manually so the Users may need little time and training to get used to with the new system.

Adjustment:

The new system may require some basic skills on how to operate a computer system and input data using keyboard and mouse. Severe use of computer system causes health problems which need to be taken into account and the new system may need some getting used to both physically and mentally as it requires much less movement and long hours of staring at a screen. Therefore proper health posture should be maintained while working hours. There should be 20" – 29" distance maintained from the screen to prevent eye strain, headache or nausea. The computer should be approximately 15 degrees tilted to prevent pain in the neck and the back. The place where computer is stored should be sufficient enough to avoid wrong postures to be taken forcefully due to limited space. Anti RSI keyboard should be used at proper heights to prevent wrist tissue injuries. For proper health maintenance, periodical breaks should be taken from work to prevent health issues and injuries. Chairs with proper adjustments should be used to avoid back pain. Care should be taken when using electrical appliances to prevent shock and liquid substances should be kept away from work area at all times to prevent hazardous shocks or short circuit. Proper distance should also be kept from plugs and sockets.



Jan. 1

(c)

Software Development,
Programming, Testing
and Implementation

(I)Development and (II) Programming

I created a database file named “travel agency database” and also created the tables that are shown below in design view in MS Access.

➤ Tables

▪ User login table

The screenshot shows the Microsoft Access 2007 interface with the 'User_login' table open in Design view. The table structure is as follows:

Field Name	Data Type	Description
User_type	Text	
User_ID	Text	
Password	Text	

The 'User_type' field is currently selected. The 'Field Properties' pane at the bottom displays the following properties for this field:

- General tab: Field Size is set to 255.
- Lookup tab: Not applicable for this field type.
- Other properties: Required is set to No, Allow Zero Length is Yes, Indexed is No, Unicode Compression is No, IME Mode is No Control, and IME Sentence Mode is None.

▪ Employee Entry table

The screenshot shows the Microsoft Access 2007 interface with the 'Employee_entry' table open in Design view. The table structure is as follows:

Field Name	Data Type	Description
EmployeeID	Text	
EName	Text	
Address	Text	
Contact	Text	
Designation	Text	
Basic_salary	Currency	
Date_of_entry	Date/Time	

The 'EmployeeID' field is currently selected. The 'Field Properties' pane at the bottom displays the following properties for this field:

- General tab: Field Size is set to 255.
- Lookup tab: Not applicable for this field type.
- Other properties: Required is set to No, Allow Zero Length is Yes, Indexed is Yes (No Duplicates), Unicode Compression is No, IME Mode is No Control, and IME Sentence Mode is None.

Jan. 1

Travel Agency Management System

■ Package Entry table

The screenshot shows the Microsoft Access application window with the 'Design' tab selected in the ribbon. The 'Table Tools' ribbon tab is also visible. A security warning message is displayed at the top left. The 'Navigation Pane' on the left lists the table 'Package_entry'. The main area displays the table structure with the following fields:

Field Name	Data Type	Description
Package_type	Text	
Package_name	Text	
Package_details	Text	
Cost_per_person	Currency	
Date_of_entry	Date/Time	
PackageID	Text	
Itinerary	Text	
Journey_date	Date/Time	
Arrival_time	Date/Time	
Starting_time	Date/Time	
Duration/days	Number	

The 'Field Properties' pane on the right shows the properties for the 'Duration/days' field, specifically setting the 'Format' to 'Long Integer'. A note in the pane states: 'Alignment of text in control'.

■ Package Booking table

The screenshot shows the Microsoft Access application window with the 'Design' tab selected in the ribbon. The 'Table Tools' ribbon tab is also visible. A security warning message is displayed at the top left. The 'Navigation Pane' on the left lists the table 'Package_booking'. The main area displays the table structure with the following fields:

Field Name	Data Type	Description
PackageBookingID	Text	
Customer_name	Text	
Customer_address	Text	
Contact	Text	
Package_name	Text	
Cost_per_person	Currency	
Number_of_people	Number	
Net_total	Currency	
Date_of_booking	Date/Time	
Package_type	Text	
Accommodation_type	Text	

The 'Field Properties' pane on the right shows the properties for the 'PackageBookingID' field, specifically setting the 'Format' to '255'. A note in the pane states: 'A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.'

■ Package expenses details table

The screenshot shows the Microsoft Access 2007 interface in Design view. The table 'Package_expenses_details' is open. The 'PackageID' field is selected, and its properties are displayed in the 'Field Properties' pane. The properties include:

Property	Value
Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A note in the pane states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names."

■ Package expenses master table

The screenshot shows the Microsoft Access 2007 interface in Design view. The table 'Package_expenses_master' is open. The 'PackageID' field is selected, and its properties are displayed in the 'Field Properties' pane. The properties include:

Property	Value
Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A note in the pane states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names."

■ Vehicle Entry table

Vehicle_entry

Field Name	Data Type	Description
Vehicle_type	Text	
VehicleID	Text	
RegistrationNo	Text	
Make	Text	
Model	Text	
NSeats	Number	
Rate_per_km	Number	
Date_of_entry	Date/Time	
IsBooked?	Yes/No	

Field Properties

General

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

■ Vehicle Booking table

Vehicle_booking

Field Name	Data Type	Description
VehicleBookingID	Text	
VehicleID	Text	
Customer_name	Text	
Customer_address	Text	
Contact	Text	
Vehicle_type	Text	
Journey_date	Date/Time	
Driver_name	Text	
Date_of_booking	Date/Time	
Registration_number	Text	
Drop_off_date	Date/Time	

Field Properties

General

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

■ ***Vehicle Billing table***

Field Name	Data Type	Description
BookingID	Text	
Customer_name	Text	
Customer_address	Text	
VehicleID	Text	
Kilometer_travelled	Number	
Rate_per_kilometer	Currency	
Cost_of_travel	Currency	
Journey_date	Date/Time	
Drop_off_date	Date/Time	
Booking_charge	Currency	
Driver_charge	Currency	

■ ***Vehicle Servicing table***

Field Name	Data Type	Description
Vehicle_type	Text	
VehicleID	Text	
Servicing_details	Text	
Amount	Currency	
Date	Date/Time	

- ***Fuel record table***

The screenshot shows the Microsoft Access 2007 interface in Design view for the 'Fuel_record' table. The table structure is as follows:

Field Name	Data Type	Description
VehicleID	Text	
Date_of_entry	Text	
FuelType	Text	
FuelQuantity/litres	Number	
FuelCost	Currency	

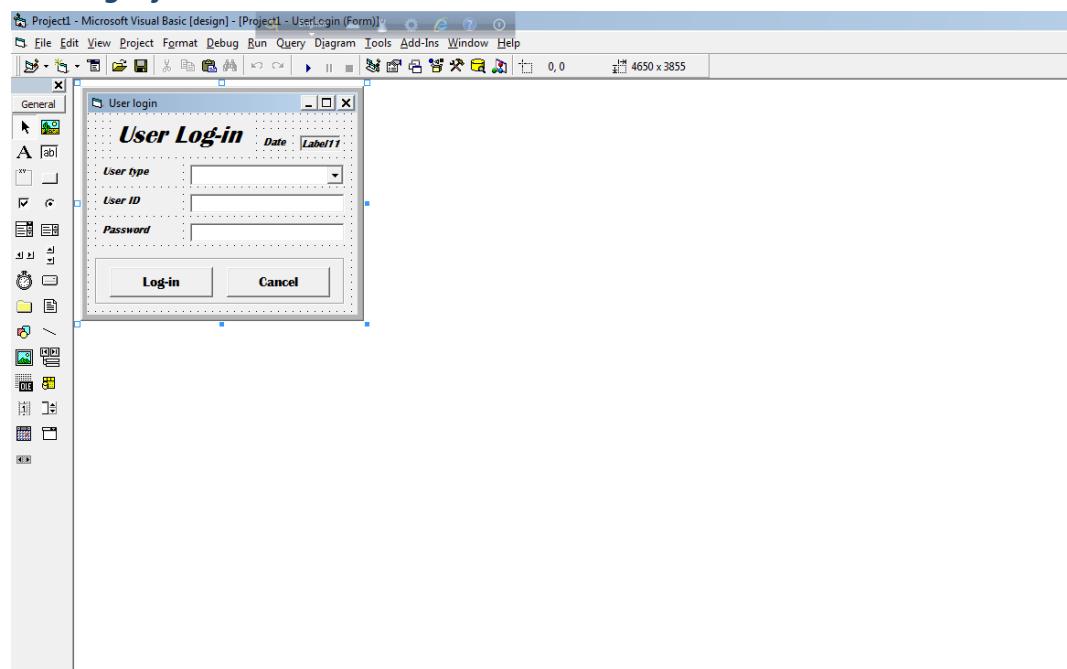
The 'VehicleID' field is selected, and its properties are displayed in the 'Field Properties' pane on the right. The 'General' tab is selected, showing the following settings:

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A note in the pane states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names."

- **Forms** – All these forms are created by me using VB editor.

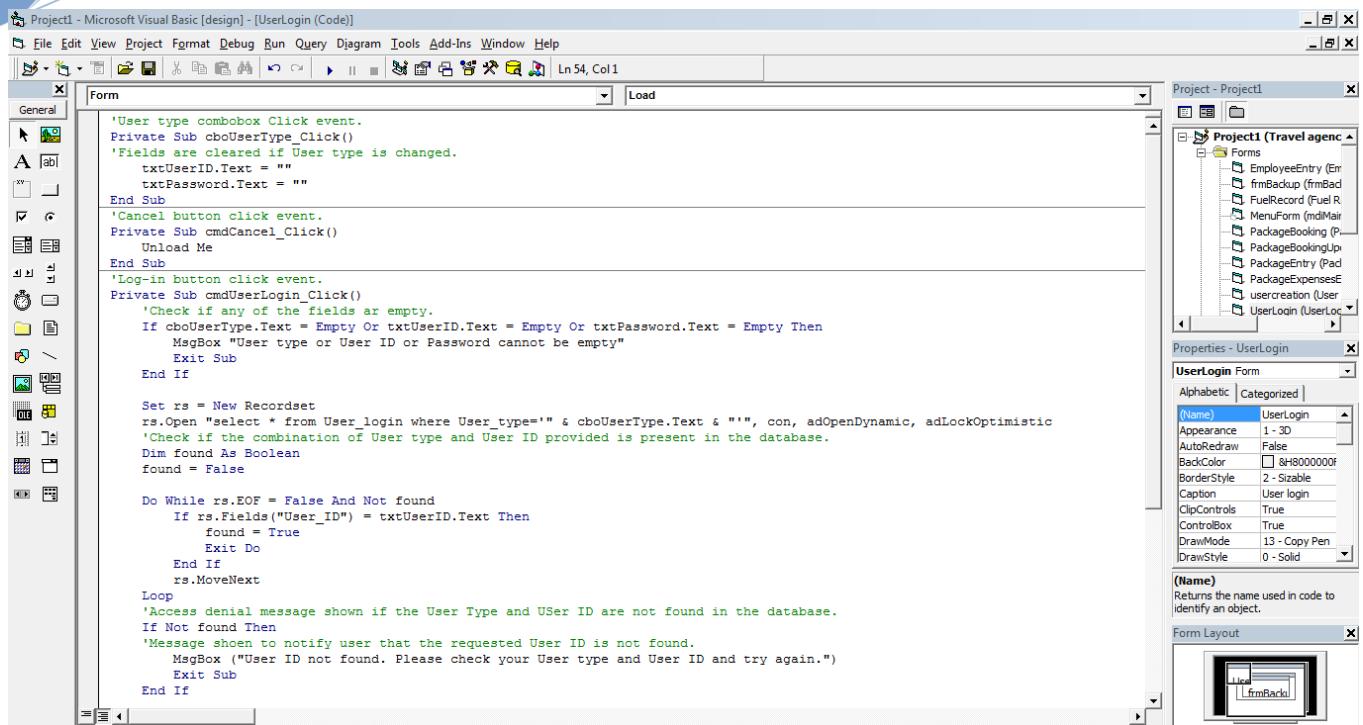
1. User Login form



The screenshot shows the actual layout of the login form.



Travel Agency Management System



```
'User type combobox Click event.
Private Sub cboUserType_Click()
'Fields are cleared if User type is changed.
    txtUserID.Text = ""
    txtPassword.Text = ""
End Sub

'Cancel button click event.
Private Sub cmdCancel_Click()
    Unload Me
End Sub

'Log-in button click event.
Private Sub cmdUserLogin_Click()
    'Check if any of the fields ar empty.
    If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Then
        MsgBox "User type or User ID or Password cannot be empty"
        Exit Sub
    End If

    Set rs = New Recordset
    rs.Open "select * from User_login where User_type=''" & cboUserType.Text & "", con, adOpenDynamic, adLockOptimistic
    'Check if the combination of User type and User ID provided is present in the database.
    Dim found As Boolean
    found = False

    Do While rs.EOF = False And Not found
        If rs.Fields("User_ID") = txtUserID.Text Then
            found = True
        End If
        rs.MoveNext
    Loop
    'Access denial message shown if the User Type and User ID are not found in the database.
    If Not found Then
        'Message shoen to notify user that the requested User ID is not found.
        MsgBox ("User ID not found. Please check your User type and User ID and try again.")
        Exit Sub
    End If
End Sub
```

```
If rs.Fields("User_ID") = txtUserID.Text Then
    found = True
    Exit Do
End If
rs.MoveNext
Loop
'Access denial message shown if the User Type and USer ID are not found in the database.
If Not found Then
    'Message shoen to notify user that the requested User ID is not found.
    MsgBox ("User ID not found. Please check your User type and User ID and try again.")
    Exit Sub
End If
'Check for validation of password if combination of User type and User ID are accepted.
If txtPassword.Text = rs.Fields("Password") Then
    MsgBox "Login successful."
Else
    'Access denial message shown if password is not accepted.
    MsgBox "Invalid password. Please try again."
End If

MenuForm.Show
End Sub

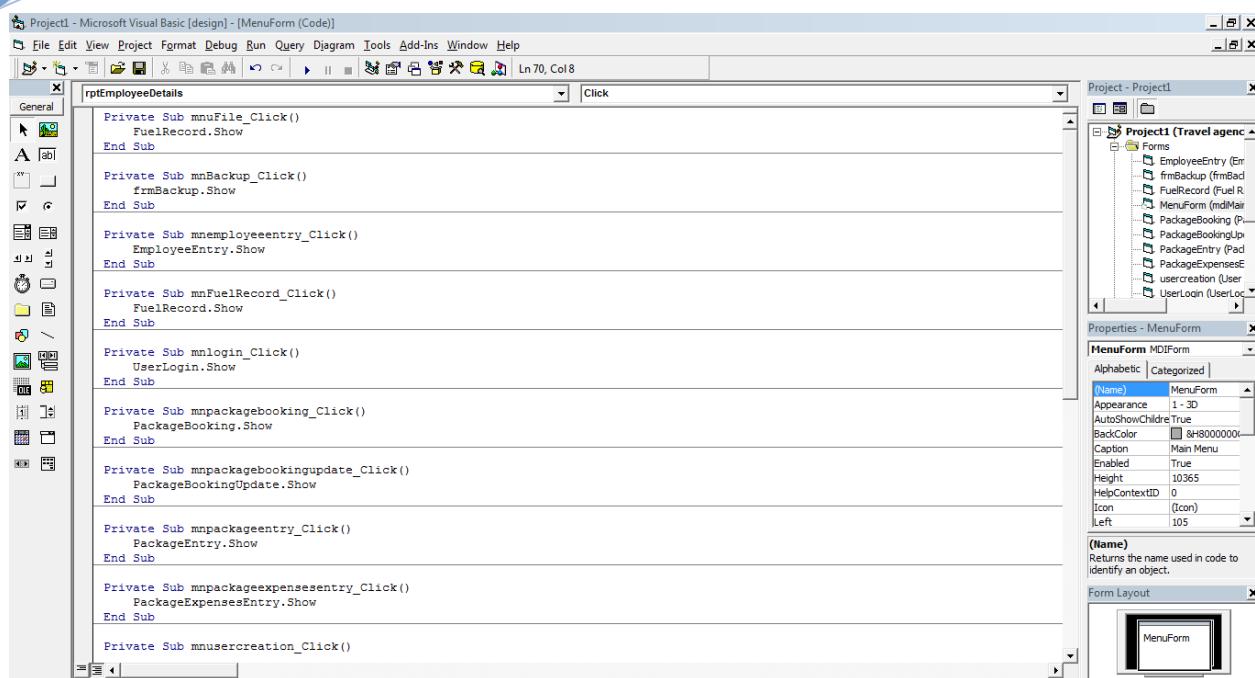
'Form load event.
Private Sub Form_Load()
    'Calling the function to connect the database.
    Call dblink
    lblDate.Caption = Date
End Sub
```

2. Main Menu



The screenshot shows the actual layout of the Main Menu form.

Travel Agency Management System



```

Private Sub mnuFile_Click()
    FuelRecord.Show
End Sub

Private Sub mnBackup_Click()
    frmBackup.Show
End Sub

Private Sub mnemployeeentry_Click()
    EmployeeEntry.Show
End Sub

Private Sub mnFuelRecord_Click()
    FuelRecord.Show
End Sub

Private Sub mnlogin_Click()
    UserLogin.Show
End Sub

Private Sub mnpackagebooking_Click()
    PackageBooking.Show
End Sub

Private Sub mnpackagebookingupdate_Click()
    PackageBookingUpdate.Show

```

```
End Sub
```

```
Private Sub mnpackageentry_Click()  
    PackageEntry.Show  
End Sub
```

```
Private Sub mnpackageexpensesentry_Click()  
    PackageExpensesEntry.Show  
End Sub
```

```
Private Sub mnusercreation_Click()  
    usercreation.Show  
End Sub
```

```
Private Sub mnvehiclebilling_Click()  
    VehicleBilling.Show  
End Sub
```

```
Private Sub mnvehiclebooking_Click()  
    VehicleBooking.Show  
End Sub
```

```
Private Sub mnvehiclebookingupdate_Click()  
    VehicleBookingUpdate.Show  
End Sub
```

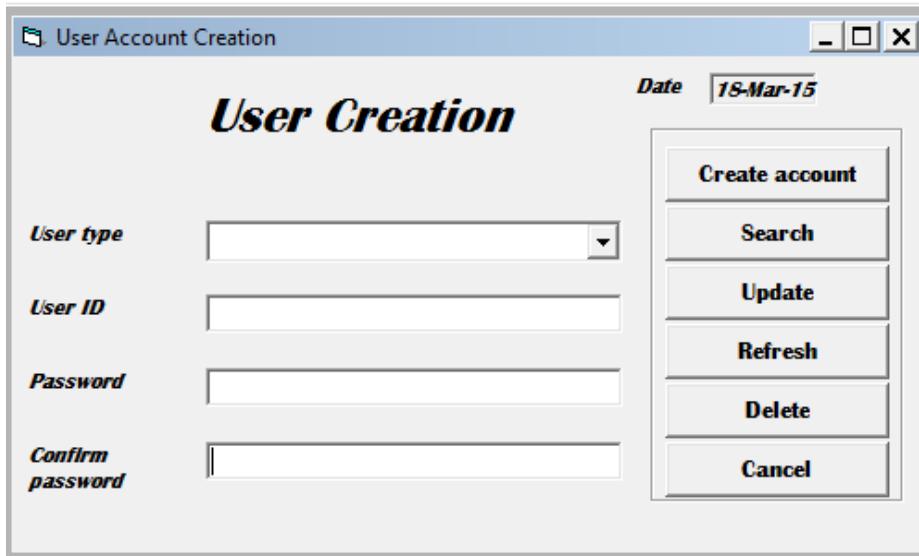
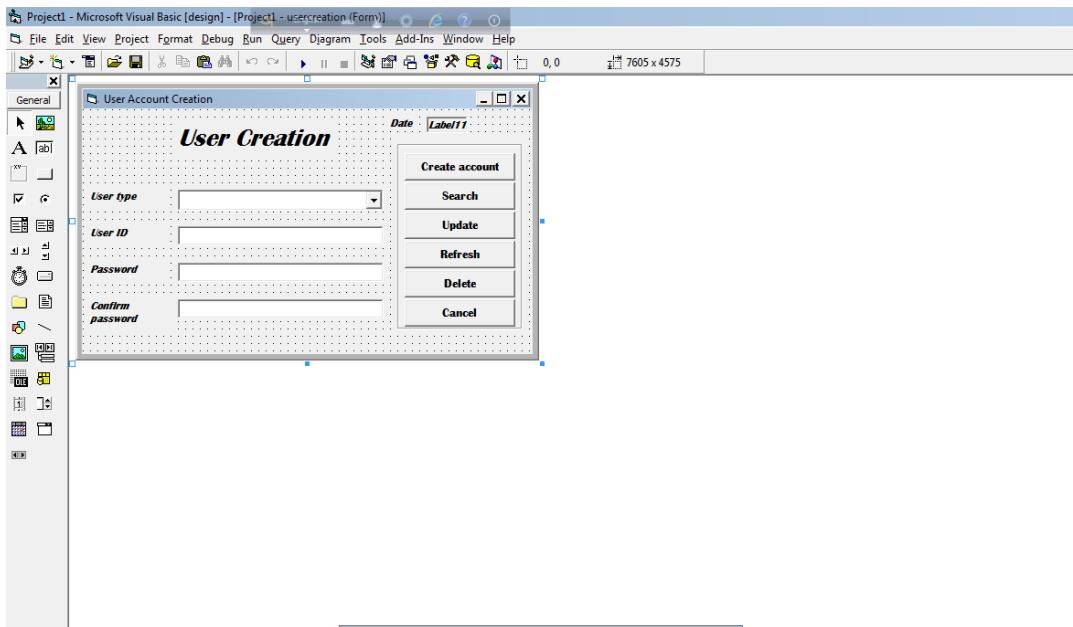
```
Private Sub mnvehicleentry_Click()  
    VehicleEntry.Show  
End Sub
```

```
Private Sub mnvehicleservicing_Click()  
    VehicleServicing.Show  
End Sub
```

```
Private Sub mnvehicleupdate_Click()  
    VehicleUpdate.Show  
End Sub
```

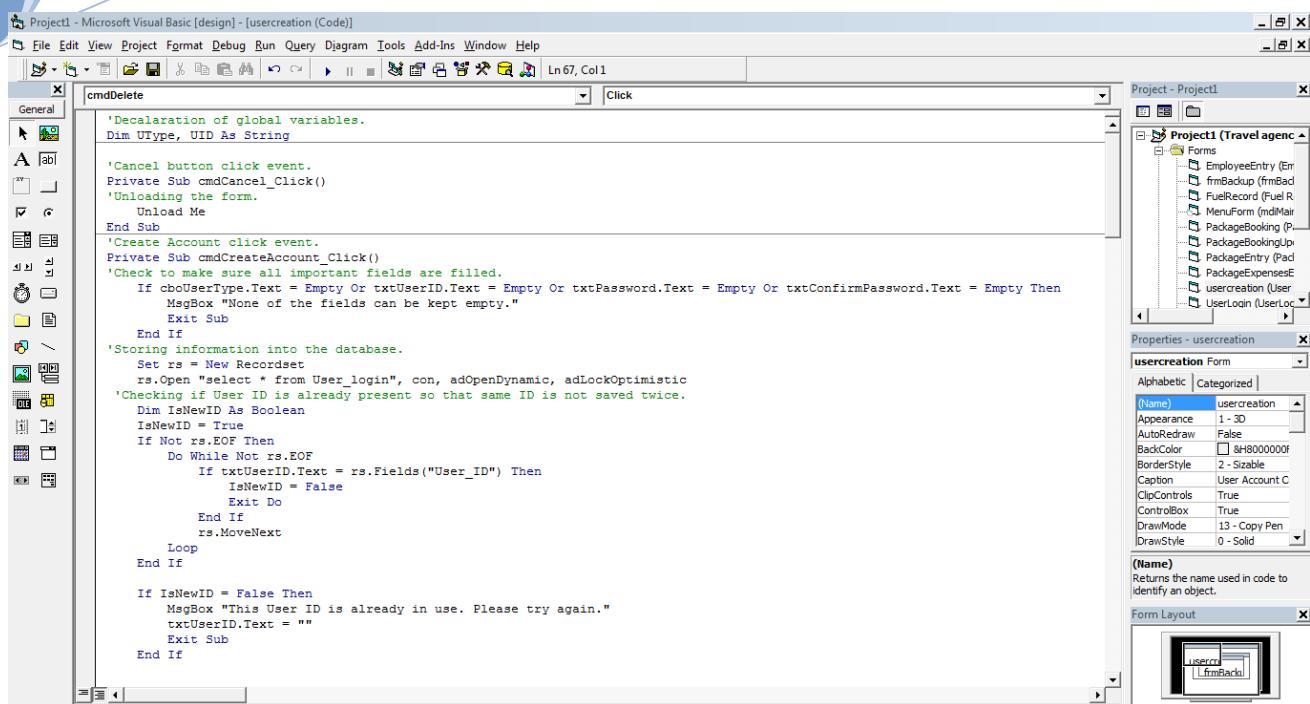
```
Private Sub rptEmployeeDetails_Click()  
    rpt1.ReportFileName = App.Path & "\Reports\Employee Details.rpt"  
    rpt1.Action = 2  
End Sub
```

3. User Creation



The screenshot shows the actual layout of the User Creation form.

Travel Agency Management System



'Declaration of global variables.

Dim UType, UID As String

'Cancel button click event.

Private Sub cmdCancel_Click()

'Unloading the form.

Unload Me

End Sub

'Create Account click event.

Private Sub cmdCreateAccount_Click()

'Check to make sure all important fields are filled.

If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Or
txtConfirmPassword.Text = Empty Then

MsgBox "None of the fields can be kept empty."

Exit Sub

End If

'Storing information into the database.

Set rs = New Recordset

rs.Open "select * from User_login", con, adOpenDynamic, adLockOptimistic

'Checking if User ID is already present so that same ID is not saved twice.

Dim IsNewID As Boolean

IsNewID = True

If Not rs.EOF Then

Do While Not rs.EOF

If txtUserID.Text = rs.Fields("User_ID") Then

IsNewID = False

Exit Do

```
End If
rs.MoveNext
Loop
End If

If IsNewID = False Then
    MsgBox "This User ID is already in use. Please try again."
    txtUserID.Text = ""
    Exit Sub
End If

rs.AddNew
'Check made to see if password and confirm password match if User type and User ID are Valid.
If txtPassword.Text = txtConfirmPassword.Text Then
    rs.Fields("Password") = txtPassword.Text
Else
    MsgBox ("The password and the confirm password do not match. Please re-enter them and try
again")
    txtPassword.Text = ""
    txtConfirmPassword.Text = ""
    Exit Sub
End If

rs.Fields("User_type") = cboUserType.Text
rs.Fields("User_ID") = txtUserID.Text

rs.Update

MsgBox ("User account has been created successfully.")
'The fields are emptied after the information is stored.
cboUserType.Text = ""
txtUserID.Text = ""
txtPassword.Text = ""
txtConfirmPassword = ""

End Sub

'Delete button click event.
Private Sub cmdDelete_Click()
'Check to make sure all important fields are filled.
If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Or
txtConfirmPassword.Text = Empty Then
    MsgBox "None of the fields can be kept empty."
    Exit Sub
End If
```

```
'The selected record is deleted.  
rs.Delete  
MsgBox "The record has been deleted."  
'The fields on the form are cleared after deletion.  
cboUserType.Text = ""  
txtUserID.Text = ""  
txtPassword.Text = ""  
txtConfirmPassword = ""  
End Sub  
'Refresh button click event.  
Private Sub cmdRefresh_Click()  
'The fields are cleared.  
cboUserType.Text = ""  
txtUserID.Text = ""  
txtPassword.Text = ""  
txtConfirmPassword = ""  
End Sub  
'Search button click event.  
Private Sub cmdSearch_Click()  
'Fields are checked to see if any of the fields are empty.  
If cboUserType.Text = Empty Or txtUserID.Text = Empty Then  
    MsgBox "Please provide User type and User ID both."  
    Exit Sub  
End If  
'Information is loaded from the database.  
Set rs = New Recordset  
rs.Open "select * from User_login where User_type ='" & cboUserType.Text & "'", con,  
adOpenDynamic, adLockOptimistic  
  
Dim found As Boolean  
found = False  
'The database is checked to see if the User ID is present.  
Do While rs.EOF = False And Not found  
    If rs.Fields("User_ID") = txtUserID.Text Then  
        found = True  
        Exit Do  
    End If  
    rs.MoveNext  
Loop  
'Message if database is not found.  
If Not found Then  
    MsgBox ("User ID not found. Please check your User type and User ID and try again.")  
    Exit Sub  
End If  
  
txtPassword.Text = rs.Fields("Password")  
txtConfirmPassword.Text = txtPassword.Text
```

```
UType = cboUserType.Text
UID = txtUserID.Text

End Sub
Private Sub cmdUpdate_Click()
'The fields are checked to see if they are all filled.
If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Or
txtConfirmPassword.Text = Empty Then
    MsgBox "None of the fields can be kept empty."
    Exit Sub
End If
'Validation to ensure User type is not changed.
If cboUserType.Text <> UType Then
    MsgBox "User type cannot be changed."
    cboUserType.Text = UType
    Exit Sub
End If
'Validation to check if User ID is already present if User plans on changing the User ID.
If txtUserID <> UID Then
    If Not rs.EOF Then
        Dim IsNewID As Boolean
        IsNewID = True
        Do While Not rs.EOF
            If txtUserID.Text = rs.Fields("User_ID") Then
                IsNewID = False
                Exit Do
            End If
            rs.MoveNext
        Loop
    End If
    If IsNewID = False Then
        MsgBox "This User ID is already in use. Please try again."
        txtUserID.Text = ""
        Exit Sub
    End If
End If

'User information in the database is updated.
rs.MoveFirst

'Validation to make sure password and confirm password match.
If txtPassword.Text = txtConfirmPassword.Text Then
    rs.Fields("Password") = txtPassword.Text
```

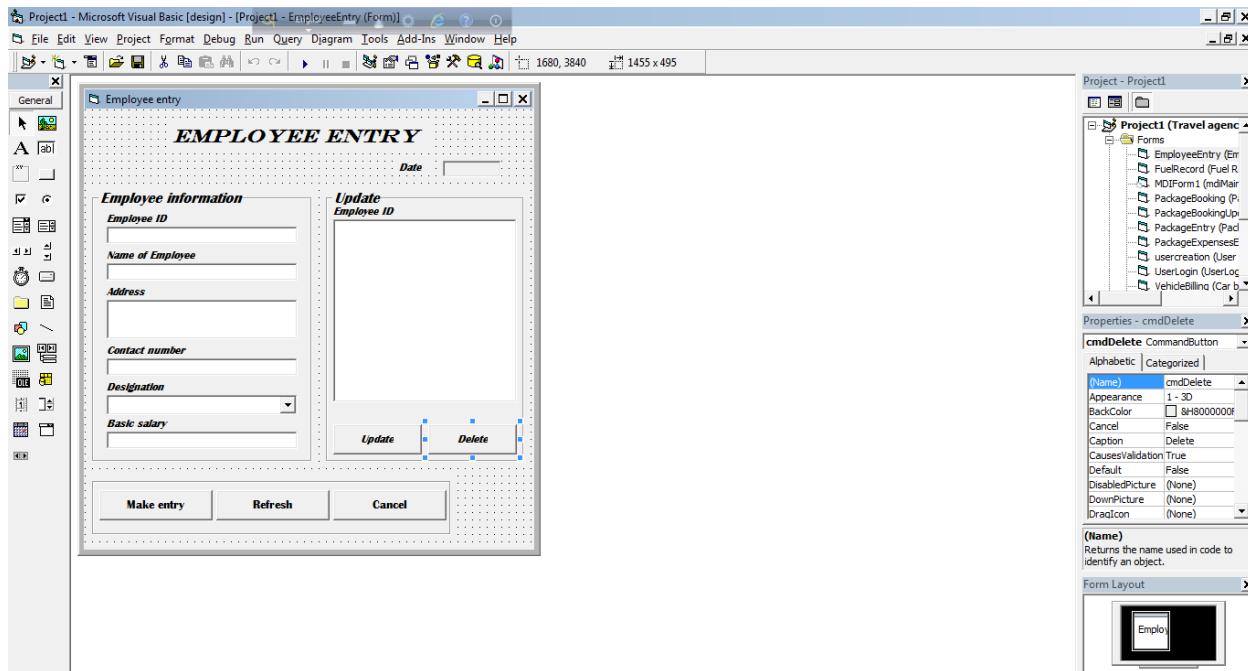
```
Else
'Message shown if password and confirm password don't match
    MsgBox ("The password and the confirm password do not match. Please check them and try again")
    Exit Sub
End If

rs.Fields("User_ID") = txtUserID.Text

rs.Update
'Message shown to notify the User that the Update was successful.
    MsgBox ("Record has been Updated successfully.")
'The fields are cleared after the Update is successful.
cboUserType.Text = ""
txtUserID.Text = ""
txtPassword.Text = ""
txtConfirmPassword = ""

End Sub
'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
    Call dblink
    lblDate.Caption = Date
End Sub
```

4. Employee entry



This screenshot shows the actual running interface of the 'Employee entry' form. The title bar reads 'Employee entry'. The main title is 'EMPLOYEE ENTRY'. A date field shows 'Date 22-03-2015'. The left panel is labeled 'Employee information' and contains fields for Employee ID, Name of Employee, Address, Contact number, Designation, and Basic salary. The right panel is labeled 'Update Employee ID' and lists the same seven values as the design view. At the bottom are 'Update' and 'Delete' buttons, and at the very bottom are 'Make entry', 'Refresh', and 'Cancel' buttons.

The screenshot shows the actual layout of the Employee Entry form.

The screenshot shows the Microsoft Visual Basic IDE interface. The title bar reads "Project1 - Microsoft Visual Basic [design] - [EmployeeEntry (Code)]". The menu bar includes File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-Ins, Window, Help. The toolbar has various icons for file operations like Open, Save, Print, and a search icon. The code editor window is titled "cmdDelete" and contains the following VB code:

```

Dim EID As String

'Cancel button click event
Private Sub cmdCancel_Click()
'The form is unloaded.
Unload Me
End Sub

'Delete button click event.
Private Sub cmdDelete_Click()
| If txtEmployeeID.Text = "" Or txtEmployeeName.Text = "" Or txtEmployeeAddress.Text = "" Or txtContactNumber.Text = ""
| Message shown to notify the user that none of the fields can be left empty.
| MsgBox "Select the record to be deleted."
| Exit Sub

End If
'The selected record is deleted.
rs.Delete
'The Employee ID list is updated.
lstEmployeeID.RemoveItem (lstEmployeeID.ListIndex)
lstEmployeeID.Refresh
'Message shown to notify the user that the record is deleted from the database.
MsgBox "The Record has been deleted."
'The fields are cleared after deletion of a record.
txtEmployeeID.Text = ""
txtEmployeeName.Text = ""
txtEmployeeAddress.Text = ""
txtContactNumber.Text = ""
.cboDesig.Text = ""
.txtBasicSalary.Text = ""
lstEmployeeID.Clear
Call Form_Load
End Sub

'Make Entry button click event.
Private Sub cmdMakeEntry_Click()

'The fields are checked to make sure none of them are empty.
If txtEmployeeID.Text = "" Or txtEmployeeName.Text = "" Or txtEmployeeAddress.Text = "" Or txtContactNumber.Text = "" Then
| Message shown to notify the user that none of the fields can be left empty.
| MsgBox "Select the record to be deleted."
| Exit Sub
End If

```

The properties window on the right shows the following settings for the "EmployeeEntry Form":

- (Name) EmployeeEntry
- Appearance: 1 - 3D
- AutoRedraw: False
- BackColor: #FFFFFF
- BorderStyle: 2 - Sizable
- Caption: Employee entry
- ClipControls: True
- ControlBox: True
- DrawMode: 13 - Copy Pen
- DrawStyle: 0 - Solid

The form layout shows a single form named "EmployeeEntry".

Dim EID As String

'Cancel button click event

Private Sub cmdCancel_Click()

'The form is unloaded.

Unload Me

End Sub

'Delete button click event.

Private Sub cmdDelete_Click()

If txtEmployeeID.Text = "" Or txtEmployeeName.Text = "" Or txtEmployeeAddress.Text = "" Or txtContactNumber.Text = "" Or cboDesig.Text = "" Or txtBasicSalary.Text = "" Then

'Message shown to notify the user that none of the fields can be left empty.

MsgBox "Select the record to be deleted."

Exit Sub

End If

'The selected record is deleted.

rs.Delete

'The Employee ID list is updated.

lstEmployeeID.RemoveItem (lstEmployeeID.ListIndex)

lstEmployeeID.Refresh

'Message shown to notify the user that the record is deleted from the database.

MsgBox "The Record has been deleted."

'The fields are cleared after deletion of a record.

txtEmployeeID.Text = ""

txtEmployeeName.Text = ""

txtEmployeeAddress.Text = ""

txtContactNumber.Text = ""

```
cboDesig.Text = ""
txtBasicSalary.Text = ""
lstEmployeeID.Clear
Call Form_Load
End Sub

'Make Entry button click event.
Private Sub cmdMakeEntry_Click()

'The fields are checked to make sure none of them are empty.
If txtEmployeeID.Text = Empty Or txtEmployeeName.Text = Empty Or txtEmployeeAddress.Text =
Empty Or txtContactNumber.Text = Empty Or cboDesig.Text = Empty Or txtBasicSalary.Text = Empty
Then
'Message shown to notify the user that none of the fields can be left empty.
    MsgBox "None of the fields can be left empty."
    Exit Sub
End If

'Length check to make sure contact number is of 11 digits.
If Len(txtContactNumber.Text) <> 11 Then
    MsgBox "Number of digits used for contact number is invalid."
    Exit Sub
End If

'The data of the new employee are added to the Employee Entry database.
Set rs = New Recordset
rs.Open "select * from Employee_entry", con, adOpenDynamic, adLockOptimistic
'Validation check to make sure same Employee is not used twice.
If Not rs.EOF Then
    Dim IsNewID As Boolean
    IsNewID = True
    Do While Not rs.EOF
        If txtEmployeeID.Text = rs.Fields("EmployeeID") Then
            IsNewID = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID = False Then
    MsgBox "This ID is already in use. Please try again."
    txtEmployeeID.Text = ""
    Exit Sub
End If

rs.AddNew
rs.Fields("EmployeeID") = txtEmployeeID.Text
```

```

rs.Fields("EName") = txtEmployeeName.Text
rs.Fields("Address") = txtEmployeeAddress.Text
rs.Fields("Contact") = txtContactNumber.Text
rs.Fields("Designation") = cboDesig.Text
rs.Fields("Basic_salary") = txtBasicSalary.Text
rs.Fields("Date_of_entry") = lblDate.Caption
rs.Update

```

'Message shown to notify the user that the record has been added successfully.

```
MsgBox "Record has been added successfully."
```

'The fields are cleared after the information is added to the database.

```

txtEmployeeID.Text = ""
txtEmployeeName.Text = ""
txtEmployeeAddress.Text = ""
txtContactNumber.Text = ""
cboDesig.Text = ""
txtBasicSalary.Text = ""

```

```

lstEmployeeID.Clear
lstEmployeeID.Refresh

```

```
Call Form_Load
```

```
End Sub
```

'Refresh button click event.

```
Private Sub cmdRefresh_Click()
```

'The fields are cleared at the click of the button.

```

lstEmployeeID.Clear
lstEmployeeID.Refresh
txtEmployeeID.Text = ""
txtEmployeeName.Text = ""
txtEmployeeAddress.Text = ""
txtContactNumber.Text = ""
cboDesig.Text = ""
txtBasicSalary.Text = ""

```

```
Call Form_Load
```

```
End Sub
```

'Update button click event.

```
Private Sub cmdUpdate_Click()
```

'Validation check to make sure none of the fields are kept empty.

```

If txtEmployeeID.Text = "" Or txtEmployeeName.Text = "" Or txtEmployeeAddress.Text = "" Or
txtContactNumber.Text = "" Or cboDesig.Text = "" Or txtBasicSalary.Text = "" Then
    MsgBox "Check if all the required information are filled and then try again."
    Exit Sub
End If

```

'Ensuring the Employee ID is not changed as it serves as the basis of identification of the employee.

```
If txtEmployeeID.Text <> EID Then
    MsgBox "Employee ID cannot be changed."
    txtEmployeeID.Text = EID
    Exit Sub
End If
```

'Updating Employee Entry database information for a particular Employee record.

```
rs.Fields("EName") = txtEmployeeName.Text
rs.Fields("Address") = txtEmployeeAddress.Text
rs.Fields("Contact") = txtContactNumber.Text
rs.Fields("Designation") = cboDesig.Text
rs.Fields("Basic_salary") = txtBasicSalary.Text
rs.Update
```

'Message shown to notify the user that the update was successful.

```
MsgBox "The record has been updated."
```

'The fields on the form are emptied to ready the form for next transaction.

```
IstEmployeeID.Clear
IstEmployeeID.Refresh
txtEmployeeID.Text = ""
txtEmployeeName.Text = ""
txtEmployeeAddress.Text = ""
txtContactNumber.Text = ""
cboDesig.Text = ""
txtBasicSalary.Text = ""
```

```
Call Form_Load
```

End Sub

Private Sub Form_Load()

'Calling the function to connect to the database.

```
Call dblink
```

'Updating the date.

```
lblDate.Caption = Format(Now, "dd/mm/yyyy")
```

'The Employee IDs present in the Employee database are loaded onto the Employee ID list.

```
Set rs = New Recordset
rs.Open "select * from Employee_entry", con, adOpenDynamic, adLockOptimistic
Do While Not rs.EOF
    IstEmployeeID.AddItem (rs.Fields("EmployeeID"))
    rs.MoveNext
Loop
```

```
rs.MoveLast
txtEmployeeID.Text = CInt(rs!EmployeeID) + 1
```

```
End Sub
```

```
'Employee ID list click event.
```

```
Private Sub lstEmployeeID_Click()
```

```
    Set rs = New Recordset
```

```
'Employee information loaded from database on selection of Employee ID from Employee list.
```

```
    rs.Open "select * from Employee_entry where EmployeeID ='" & lstEmployeeID.Text & "'", con,  
adOpenDynamic, adLockOptimistic
```

```
    txtEmployeeID.Text = rs.Fields("EmployeeID")
```

```
    txtEmployeeName.Text = rs.Fields("EName")
```

```
    txtEmployeeAddress.Text = rs.Fields("Address")
```

```
    txtContactNumber.Text = rs.Fields("Contact")
```

```
    cboDesig.Text = rs.Fields("Designation")
```

```
    txtBasicSalary.Text = rs.Fields("Basic_salary")
```

```
    EID = txtEmployeeID.Text
```

```
End Sub
```

```
Private Sub txtContactNumber_KeyPress(KeyAscii As Integer)
```

```
'Validating that the new vat on picnicspot entered is numeric
```

```
    If KeyAscii = 8 Then Exit Sub
```

```
    If Not IsNumeric(Chr(KeyAscii)) Then
```

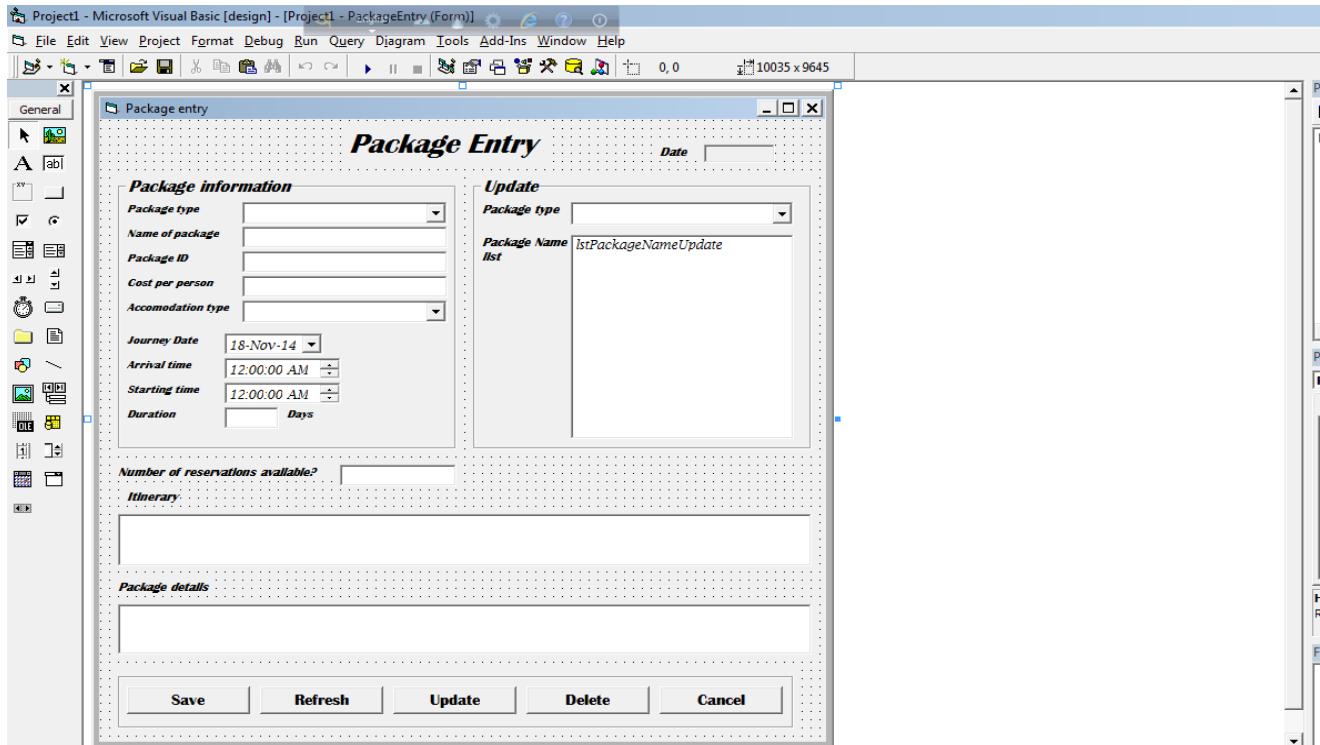
```
        MsgBox "Please input numbers only"
```

```
    Exit Sub
```

```
    End If
```

```
End Sub
```

5. Package Entry



Package entry

Package Entry

Date **29-03-2015**

Package information

Package type:

Name of package:

Package ID:

Cost per person:

Accommodation type:

Journey Date:

Arrival time:

Starting time:

Duration: Days

Number of reservations available?

Itinerary

Package details

Update

Package type:

Package Name list:

- 1stPackageNameUpdate

Save Refresh Update Delete Cancel

The screenshot shows the actual layout of the Package Entry form.

Travel Agency Management System

The screenshot shows the Microsoft Visual Basic IDE interface. The main window displays the code for the 'PackageEntry' form. The code handles various events such as package type selection, loading package names, cancel button click, and delete button click. The Project Explorer on the right shows the project structure under 'Project1 (Travel agency)'. The Properties window shows settings for the 'PackageEntry' form, and the Form Layout window shows a diagram of the form.

```

Dim OldPID As String
'Click event Package type combobox of the update section of the form.
Private Sub cboPackageTypeUpdate_Click()
'The Package name list cleared if package type is changed.
    lstPackageNameUpdate.Clear
    Set rs = New Recordset
    'Opening the Package Entry database.
    rs.Open "select * from Package_entry where Package_type ='" & cboPackageTypeUpdate.Text & "'", con, adOpenDynamic, adLockOptimistic
    'Loading the Package names of the particular type selected into the Package name list.
    Do While Not rs.EOF
        lstPackageNameUpdate.AddItem (rs.Fields("Package_name"))
        rs.MoveNext
    Loop
End Sub
'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
    Unload Me
End Sub
'Delete button click event
Private Sub cmdDelete_Click()
    Set rs = New Recordset
    'Validation check to make sure none of the important fields are left empty
    If cboPackageType.Text = Empty Or txtPackageName.Text = Empty Or txtPackageDetails.Text = Empty
    'Message shown if any empty fields are fields are found.
        MsgBox "All the information need to be provided."
        Exit Sub
    End If
    'Deletion of selected record with the particular Package name from the Package Entry database.
    rs.Open "delete * from Package_entry where Package_name ='" & lstPackageNameUpdate.Text & "' and Package_type ='" & cboPackageType.Text & "'"
    lstPackageNameUpdate.Clear
    lstPackageNameUpdate.Refresh
    cboPackageTypeUpdate.Clear
    cboPackageTypeUpdate.Refresh
    cboPackageTypeUpdate.Text = ""
End Sub

```

Dim OldPID As String

'Click event Package type combobox of the update section of the form.

Private Sub cboPackageTypeUpdate_Click()

'The Package name list cleared if package type is changed.

 lstPackageNameUpdate.Clear

 Set rs = New Recordset

'Opening the Package Entry database.

 rs.Open "select * from Package_entry where Package_type ='" & cboPackageTypeUpdate.Text & "'",
con, adOpenDynamic, adLockOptimistic

'Loading the Package names of the particular type selected into the Package name list.

 Do While Not rs.EOF

 lstPackageNameUpdate.AddItem (rs.Fields("Package_name"))

 rs.MoveNext

 Loop

End Sub

'Cancel button click event.

Private Sub cmdCancel_Click()

'Unloading the form.

 Unload Me

End Sub

'Delete button click event

Private Sub cmdDelete_Click()

 Set rs = New Recordset

 'Validation check to make sure none of the important fields are left empty

 If cboPackageType.Text = Empty Or txtPackageName.Text = Empty Or txtPackageDetails.Text = Empty

 Or txtCostPerPerson.Text = Empty Or cboAccommodation = Empty Or dtpJourneyDate.Value = Date Or

 dtpStartingTime = Default Or dtpArrivalTime.Value = Default Or txtDuration.Text = Empty Or

 txtItinerary.Text = Empty Then

 'Message shown if any empty fields are fields are found.

```
    MsgBox "All the information need to be provided."
    Exit Sub
End If

'Deletion of selected record with the particular Package name from the Package Entry database.
rs.Open "delete * from Package_entry where Package_name=""" & lstPackageNameUpdate.Text & """
and Package_type=""" & cboPackageTypeUpdate.Text & """", con, adOpenDynamic, adLockOptimistic

lstPackageNameUpdate.Clear
lstPackageNameUpdate.Refresh
cboPackageTypeUpdate.Clear
cboPackageTypeUpdate.Refresh

cboPackageTypeUpdate.Text = """
cboPackageType.Text = """
txtPackageName.Text = """
txtCostPerPerson.Text = """
cboAccommodation.Text = """
dtpJourneyDate.Value = Default
dtpStartingTime.Value = Default
dtpArrivalTime.Value = Default
txtDuration.Text = """
txtItinerary.Text = """
txtPackageDetails.Text = """
txtPackageID.Text = """
txtNRes.Text = """
lstPackageNameUpdate.Clear
lstPackageNameUpdate.Refresh
MsgBox "Record has been deleted successfully."

lstPackageNameUpdate.Clear
lstPackageNameUpdate.Refresh

Call Form_Load
End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
'Emptying all the fields and refreshing the forms if the user requires it.
cboPackageTypeUpdate.Text = """
cboPackageType.Text = """
txtPackageName.Text = """
txtCostPerPerson.Text = """
cboAccommodation.Text = """
dtpJourneyDate.Value = Default
dtpStartingTime.Value = Default
dtpArrivalTime.Value = Default
txtDuration.Text = """
txtItinerary.Text = """

```

```
txtPackageDetails.Text = ""
txtPackageID.Text = ""
txtNRes.Text = ""
lstPackageNameUpdate.Clear
lstPackageNameUpdate.Refresh

Call Form_Load

End Sub

'Save button click event.
Private Sub cmdSave_Click()
'Validation check to make sure none of the important fields are left empty
If cboPackageType.Text = Empty Or txtPackageName.Text = Empty Or txtPackageDetails.Text = Empty
Or txtCostPerPerson.Text = Empty Or cboAccommodation = Empty Or dtpJourneyDate.Value = Date Or
dtpStartingTime = Default Or dtpArrivalTime.Value = Default Or txtDuration.Text = Empty Or
txtItinerary.Text = Empty Then
'Message shown if any empty fields are found.
MsgBox "All the information need to be provided."
Exit Sub
End If

'Validation check to make sure same Package ID is not used twice.
If Not rs.EOF Then
Dim IsOldID As Boolean
IsOldID = False
Do While Not rs.EOF
If txtPackageID.Text = rs.Fields("PackageID") Then
IsOldID = True
Exit Do
End If
rs.MoveNext
Loop
End If

If IsOldID = True Then
MsgBox "This Package ID is already in use. Please provide a differnt Package ID."
txtPackageID.Text = ""
Exit Sub
End If

'Storing Package information into the database.
Set rs = New Recordset
rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic
rs.AddNew
rs.Fields("Package_type") = cboPackageType.Text
rs.Fields("Package_name") = txtPackageName.Text
rs.Fields("Accommodation_type") = cboAccommodation.Text
rs.Fields("Journey_date") = dtpJourneyDate.Value
rs.Fields("Starting_time") = dtpStartingTime.Value
```

```

rs.Fields("Duration/days") = txtDuration.Text
rs.Fields("Arrival_time") = dtpArrivalTime.Value
rs.Fields("Cost_per_person") = txtCostPerPerson.Text
rs.Fields("Date_of_entry") = lblDate.Caption
rs.Fields("Itinerary") = txtItinerary.Text
rs.Fields("Package_details") = txtPackageDetails.Text
rs.Fields("PackageID") = txtPackageID.Text
rs.Fields("NumberOfReservations") = txtNRes.Text

rs.Update
'Message shown to notify the user that the package information have been added successfully.
MsgBox "Record has been added successfully"
'Emptying the fields after the information is added to the database.
cboPackageTypeUpdate.Text = ""
cboPackageType.Text = ""
txtPackageName.Text = ""
txtCostPerPerson.Text = ""
cboAccommodation.Text = ""
dtpJourneyDate.Value = Default
dtpStartingTime.Value = Default
dtpArrivalTime.Value = Default
txtDuration.Text = ""
txtItinerary.Text = ""
txtPackageDetails.Text = ""
txtPackageID.Text = ""
txtNRes.Text = ""
lstPackageNameUpdate.Clear
lstPackageNameUpdate.Refresh

rs.MoveLast
txtPackageID.Text = CInt(rs!PackageID) + 1

'Reloading the names on the Package Type combobox of the update section to make sure that the new
names appear immediately.
Set rs = New Recordset
rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic

If rs.EOF = False Then
  Do While rs.EOF = False
    Dim i As Integer
    Dim IsNewType As Boolean
    IsNewType = True
    For i = 0 To cboPackageTypeUpdate.ListCount - 1
      If cboPackageTypeUpdate.List(i) = rs.Fields("Package_type") Then
        IsNewType = False
      End If
    Next
  End If
End If

```

```

If IsNewType = True Then
    cboPackageTypeUpdate.AddItem (rs.Fields("Package_type"))
End If
rs.MoveNext
Loop
End If

End Sub

'Update button click event.
Private Sub cmdUpdate_Click()
'Validation check to make sure none of the fields are left empty.
    If cboPackageType.Text = Empty Or txtPackageName.Text = Empty Or txtPackageDetails.Text = Empty
        Or txtCostPerPerson.Text = Empty Or cboAccommodation = Empty Or dtpJourneyDate.Value = Default
        Or dtpStartingTime = Default Or dtpArrivalTime.Value = Default Or txtDuration.Text = Empty Or
        txtItinerary.Text = Empty Then
        MsgBox "All the information need to be provided."
        Exit Sub
    End If
'Ensuring Package ID is not changed.
If OldPID <> txtPackageID.Text Then
    MsgBox "PackageID cannot be changed."
    txtPackageID.Text = OldPID
    Exit Sub
End If

'Updating information on the database.
    rs.Fields("Package_type") = cboPackageType.Text
    rs.Fields("Package_name") = txtPackageName.Text
    rs.Fields("Accommodation_type") = cboAccommodation.Text
    rs.Fields("Journey_date") = dtpJourneyDate.Value
    rs.Fields("Starting_time") = dtpStartingTime.Value
    rs.Fields("Duration/days") = txtDuration.Text
    rs.Fields("Arrival_time") = dtpArrivalTime.Value
    rs.Fields("Cost_per_person") = txtCostPerPerson.Text
    rs.Fields("Date_of_entry") = IblDate.Caption
    rs.Fields("Itinerary") = txtItinerary.Text
    rs.Fields("Package_details") = txtPackageDetails.Text
    rs.Fields("PackageID") = txtPackageID.Text
    rs.Fields("NumberOfReservations") = txtNRes.Text
    rs.Fields("NumberReserved") = 0
    rs.Update
'Message shown to notify user that the information is updated.
    MsgBox "Package information has been updated successfully."
'The fields are updated after the database is updated and the form is refreshed.
    cboPackageTypeUpdate.Text = ""
    cboPackageType.Text = ""

```

```
txtPackageName.Text = ""  
txtCostPerPerson.Text = ""  
cboAccommodation.Text = ""  
dtpJourneyDate.Value = Default  
dtpStartingTime.Value = Default  
dtpArrivalTime.Value = Default  
txtDuration.Text = ""  
txtItinerary.Text = ""  
txtPackageDetails.Text = ""  
txtPackageID.Text = ""  
txtNRes.Text = ""  
lstPackageNameUpdate.Clear  
lstPackageNameUpdate.Refresh
```

Call Form_Load

End Sub

'Form load event.

```
Private Sub Form_Load()
```

'Calling function to connect to database.

```
    Call dblink
```

'Showing current date.

```
    lblDate.Caption = Format(Now, "dd/mm/yyyy")
```

'Current date assigned to Journey date datepicker.

```
    dtpJourneyDate.Value = Date
```

'Loading Package types into the Package type combobox of the update section.

```
    Set rs = New Recordset
```

```
    rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic
```

If rs.EOF = False Then

```
    Do While rs.EOF = False
```

```
        Dim i As Integer
```

```
        Dim IsNewType As Boolean
```

```
        IsNewType = True
```

```
        For i = 0 To cboPackageTypeUpdate.ListCount - 1
```

```
            If cboPackageTypeUpdate.List(i) = rs.Fields("Package_type") Then
```

```
                IsNewType = False
```

```
            End If
```

```
        Next
```

```
        If IsNewType = True Then
```

```
            cboPackageTypeUpdate.AddItem (rs.Fields("Package_type"))
```

```
        End If
```

```
        rs.MoveNext
```

```
    Loop
```

```
End If
```

'Updating Package ID textbox.

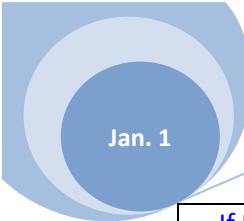
```

rs.MoveLast
txtPackageID.Text = CInt(rs!PackageID) + 1

End Sub
'Click event of the list of Package names.
Private Sub lstPackageNameUpdate_Click()
'The package information is loaded onto the form at the selection of a Package name.
    Set rs = New Recordset
    rs.Open "select * from Package_entry where Package_name ='" & lstPackageNameUpdate.Text & "'",
con, adOpenDynamic, adLockOptimistic
    cboPackageType.Text = rs.Fields("Package_type")
    txtPackageName.Text = rs.Fields("Package_name")
    txtItinerary.Text = rs.Fields("Itinerary")
    txtPackageDetails.Text = rs.Fields("Package_details")
    cboAccommodation.Text = rs.Fields("Accommodation_type")
    dtpStartingTime.Value = rs.Fields("Starting_time")
    txtDuration.Text = rs.Fields("Duration/days")
    dtpArrivalTime.Value = rs.Fields("Arrival_time")
    dtpJourneyDate.Value = rs.Fields("Journey_date")
    txtCostPerPerson.Text = rs.Fields("Cost_per_person")
    txtPackageID.Text = rs.Fields("PackageID")
    txtNRes.Text = rs.Fields("NumberOfReservations")
    rs.Update

    OldPID = txtPackageID.Text
End Sub
'Cost per person keypress event.
Private Sub txtCostPerPerson_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
    If KeyAscii = 8 Then Exit Sub
    If Not IsNumeric(Chr(KeyAscii)) Then
        MsgBox "Please input numbers only."
        Exit Sub
    End If
End Sub
'Duration keypress event.
Private Sub txtDuration_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
    If KeyAscii = 8 Then Exit Sub
    If Not IsNumeric(Chr(KeyAscii)) Then
        MsgBox "Please input numbers only."
        Exit Sub
    End If
End Sub
'Number of reservations keypress event.
Private Sub txtNRes_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.

```

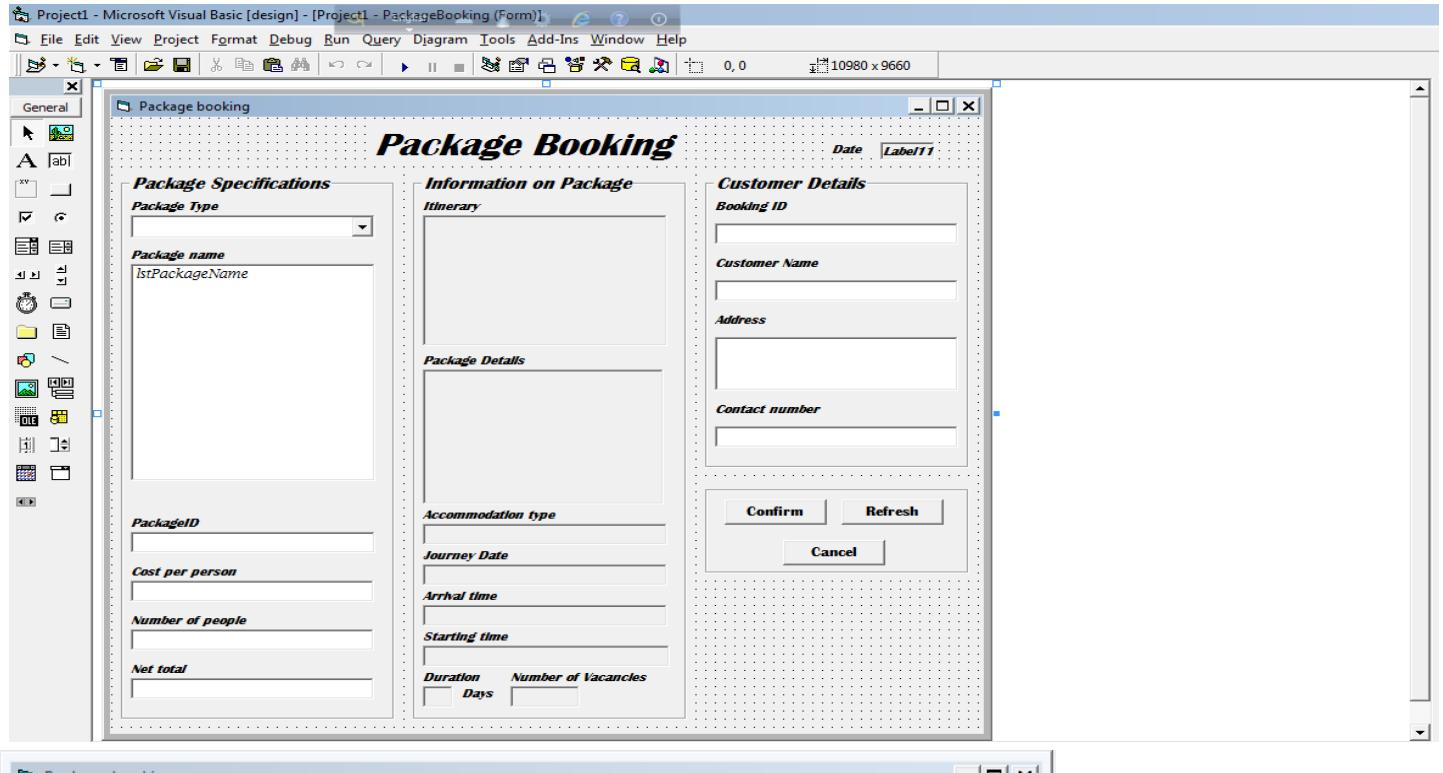


Jan. 1

Travel Agency Management System

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
End Sub
```

6. Package Booking



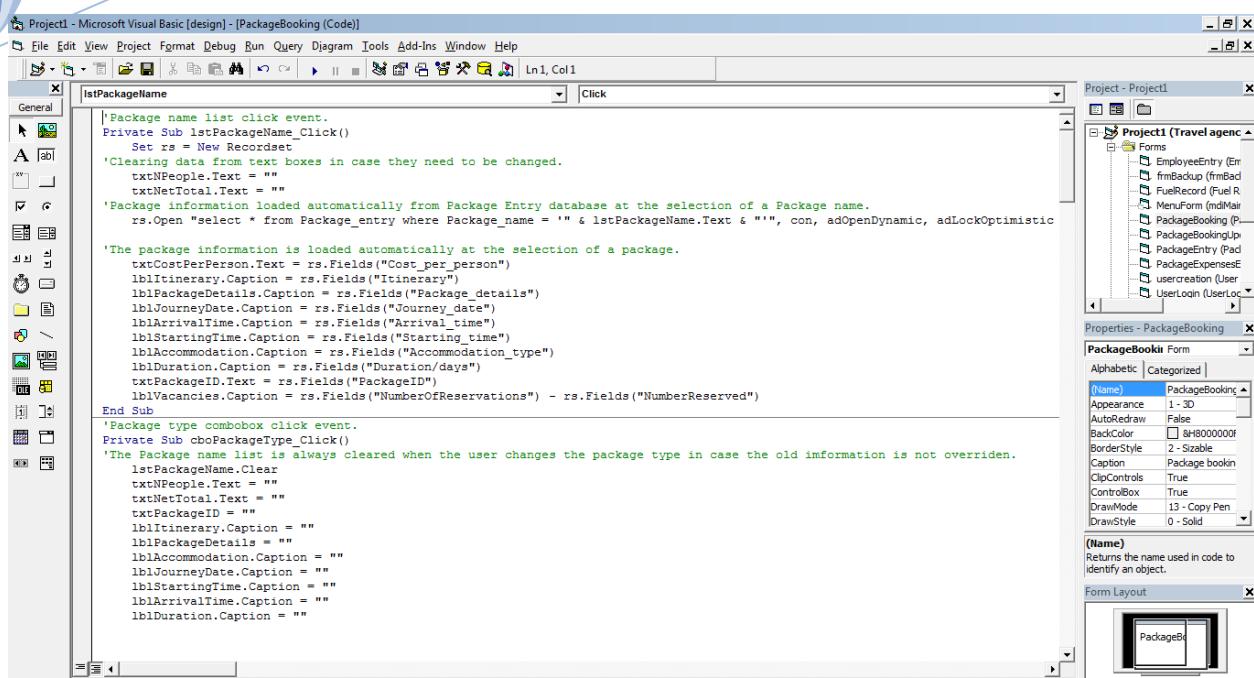
Package Booking

Date **22-03-2015**

Package Specifications		Information on Package		Customer Details	
Package Type <input type="button" value="..."/>		Itinerary Package Details Accommodation type <input type="text"/> Journey Date <input type="text"/> Arrival time <input type="text"/> Starting time <input type="text"/> Duration Number of Vacancies <input type="text"/> Days <input type="text"/>		Booking ID <input type="text"/> Customer Name <input type="text"/> Address <input type="text"/> Contact number <input type="text"/> <input type="button" value="Confirm"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>	
Package name	<input type="list"/>	Accommodation type	<input type="text"/>	Booking ID	<input type="text"/>
PackageID	<input type="text"/>	Journey Date	<input type="text"/>	Customer Name	<input type="text"/>
Cost per person	<input type="text"/>	Arrival time	<input type="text"/>	Address	<input type="text"/>
Number of people	<input type="text"/>	Starting time	<input type="text"/>	Contact number	<input type="text"/>
Net total	<input type="text"/>	Duration	<input type="text"/>	Confirm	<input type="button" value="Confirm"/>

The screenshot shows the actual layout of the Package Booking form.

Travel Agency Management System



```
'Package name list click event.
Private Sub lstPackageName_Click()
    Set rs = New Recordset
    'Clearing data from text boxes in case they need to be changed.
    txtNPeople.Text = ""
    txtNetTotal.Text = ""
    'Package information loaded automatically from Package Entry database at the selection of a Package name.
    rs.Open "select * from Package_entry where Package_name = '" & lstPackageName.Text & "'", con, adOpenDynamic, adLockOptimistic

    'The package information is loaded automatically at the selection of a package.
    txtCostPerPerson.Text = rs.Fields("Cost_per_person")
    lblItinerary.Caption = rs.Fields("Itinerary")
    lblPackageDetails.Caption = rs.Fields("Package_details")
    lblJourneyDate.Caption = rs.Fields("Journey_date")
    lblArrivalTime.Caption = rs.Fields("Arrival_time")
    lblStartingTime.Caption = rs.Fields("Starting_time")
    lblAccommodation.Caption = rs.Fields("Accommodation_type")
    lblDuration.Caption = rs.Fields("Duration/days")
    txtPackageID.Text = rs.Fields("PackageID")
    lblVacancies.Caption = rs.Fields("NumberOfReservations") - rs.Fields("NumberReserved")
End Sub

'Package type combobox click event.
Private Sub cboPackageType_Click()
    'The Package name list is always cleared when the user changes the package type in case the old imformation is not overriden.
    lstPackageName.Clear
    txtNPeople.Text = ""
    txtNetTotal.Text = ""
    txtPackageID = ""
    lblItinerary.Caption = ""
    lblPackageDetails = ""
    lblAccommodation.Caption = ""
    lblJourneyDate.Caption = ""
    lblStartingTime.Caption = ""
    lblArrivalTime.Caption = ""
    lblDuration.Caption = ""
End Sub
```

```
IstPackageName.Clear
txtNPeople.Text = ""
txtNetTotal.Text = ""
txtPackageID = ""
lblItinerary.Caption = ""
lblPackageDetails = ""
lblAccommodation.Caption = ""
lblJourneyDate.Caption = ""
lblStartingTime.Caption = ""
lblArrivalTime.Caption = ""
lblDuration.Caption = ""
```

'Package names are loaded onto the Package names list on the selection of a particular package type.
Only packages of this particular type are selected and loaded onto the list.

'Opening the Package Entry database.

```
Set rs = New Recordset
```

'Opening the Package Entry database.

```
rs.Open "select * from Package_entry where Package_type ='" & cboPackageType.Text & "'", con,
adOpenDynamic, adLockOptimistic
```

'Loading the Package names of the particular type selected into the Package name list.

```
Do While Not rs.EOF
```

```
IstPackageName.AddItem (rs.Fields("Package_name"))
```

```
rs.MoveNext
```

```
Loop
```

End Sub

'Confirm button click event.

```
Private Sub cmdConfirm_Click()
```

'Validation check to make sure all the important fields are filled.

```
If cboPackageType.Text = "" Or txtPackageID.Text = "" Or txtCostPerPerson.Text = "" Or txtNPeople =
"" Or txtBookingID.Text = "" Or txtCustomerName.Text = "" Or txtAddress.Text = "" Or
txtContactNumber.Text = "" Then
```

'Message to notify the user that all of the information need to be provided.

```
MsgBox "All the Package information and customer information need to provided in order to
proceed with the booking."
```

```
Exit Sub
```

```
End If
```

'Length check to make sure contact number is of 11 digits.

```
If Len(txtContactNumber.Text) <> 11 Then
```

```
MsgBox "Number of digits used for contact number is invalid."
```

```
Exit Sub
```

```
End If
```

```
Set rs = New Recordset
```

'Updating reservation information in the Package Entry database.

```

rs.Open "select * from Package_entry where Package_name ='" & lstPackageName.Text & "'", con,
adOpenDynamic, adLockOptimistic
p = rs.Fields("NumberReserved")
p = p + txtNPeople.Text
If rs.Fields("NumberOfReservations") > p Then
    rs.Fields("NumberReserved") = p
    rs.Update
Else
    'Validation check to check if there are any vacancies.
    'Message to notify the user that there are no vacancies that are available.
    MsgBox "Sorry, there are no vacancies for this package. All the slots have been booked. Please try another package."
    Exit Sub
End If
rs.Close
'Opening the Package booking database.
rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic
'Validation check to make sure that the same ID cannot be used twice.
Dim IsNewID As Boolean
IsNewID = True
If Not rs.EOF Then
    Do While Not rs.EOF
        If txtBookingID.Text = rs.Fields("PackageBookingID") Then
            IsNewID = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID = False Then
    'Message to notify that the user if booking ID is already in use.
    MsgBox "This Booking ID is already in use. Please try again."
    txtBookingID.Text = ""
    Exit Sub
End If

'Storage of Booking information into the database.
rs.AddNew
rs.Fields("PackageBookingID") = txtBookingID.Text
rs.Fields("Customer_name") = txtCustomerName.Text
rs.Fields("Customer_address") = txtAddress.Text
rs.Fields("Package_type") = cboPackageType.Text
rs.Fields("Journey_Date") = lblJourneyDate.Caption
rs.Fields("Contact") = txtContactNumber.Text
rs.Fields("Cost_per_person") = txtCostPerPerson.Text
rs.Fields("Package_name") = lstPackageName.Text

```

```

rs.Fields("Number_of_people") = txtNPeople.Text
rs.Fields("Net_total") = txtNetTotal.Text
rs.Fields("Date_of_booking") = lblDate.Caption
rs.Fields("PackageID") = txtPackageID.Text
rs.Fields("Accommodation_type") = lblAccommodation.Caption
rs.Fields("Journey_date") = lblJourneyDate.Caption
rs.Fields("Starting_time") = lblStartingTime.Caption
rs.Fields("Arrival_time") = lblArrivalTime.Caption
rs.Fields("Duration/days") = lblDuration.Caption
rs.Fields("Itinerary") = lblItinerary.Caption
rs.Fields("Package_details") = lblPackageDetails.Caption
rs.Update
'Message shown to notify the user that storage was sucessful.
MsgBox "Booking has been made successfully."
rs.Close

```

'The fields are emptied after the storage of a record.

```

txtBookingID.Text = ""
txtCustomerName.Text = ""
txtAddress.Text = ""
cboPackageType.Text = ""
txtContactNumber.Text = ""
txtCostPerPerson.Text = ""
txtNPeople.Text = ""
txtNetTotal.Text = ""
txtPackageID = ""
lblItinerary.Caption = ""
lblPackageDetails = ""
lblAccommodation.Caption = ""
lblJourneyDate.Caption = ""
lblStartingTime.Caption = ""
lblArrivalTime.Caption = ""
lblDuration.Caption = ""
lstPackageName.Clear
lstPackageName.Refresh

```

Call Form_Load

End Sub

'Cancel button click event.

Private Sub cmdCancel_Click()

'Unloading the form.

Unload Me

End Sub

'Refresh button click event.

Private Sub cmdRefresh_Click()

'The fields are emptied and the form is refreshed to allow the user to start anew.

```

txtBookingID.Text = ""

```

```

txtCustomerName.Text = ""
txtAddress.Text = ""
cboPackageType.Text = ""
txtContactNumber.Text = ""
txtCostPerPerson.Text = ""
txtNPeople.Text = ""
txtNetTotal.Text = ""
txtPackageID = ""
lblItinerary.Caption = ""
lblPackageDetails = ""
lblAccommodation.Caption = ""
lblJourneyDate.Caption = ""
lblStartingTime.Caption = ""
lblArrivalTime.Caption = ""
lblDuration.Caption = ""
lstPackageName.Clear
lstPackageName.Refresh

Call Form_Load
End Sub
'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
Call dblink
'Showing current date on the form.
lblDate.Caption = Format(Now, "dd/mm/yyyy")
Set rs = New Recordset
rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic
rs.MoveLast
txtBookingID.Text = CInt(rs!PackageBookingID) + 1
rs.Close

Set rs = New Recordset
rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic
'Loading Package Types from Package Entry Database.
'A check is made to make sure that each type is not loaded twice.
If rs.EOF = False Then
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboPackageType.ListCount - 1
            If cboPackageType.List(i) = rs.Fields("Package_type") Then
                IsNewType = False
            End If
        Next
        If IsNewType = True Then

```

```
cboPackageType.AddItem (rs.Fields("Package_type"))
End If
rs.MoveNext
Loop
End If

End Sub
```

```
'Contact Number keypress event.
Private Sub txtContactNumber_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```

```
'Cost per person keypress event.
Private Sub txtCostPerPerson_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```

```
'Net total keypress event.
Private Sub txtNetTotal_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```

```
'Number of people keypress event.
Private Sub txtNPeople_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```

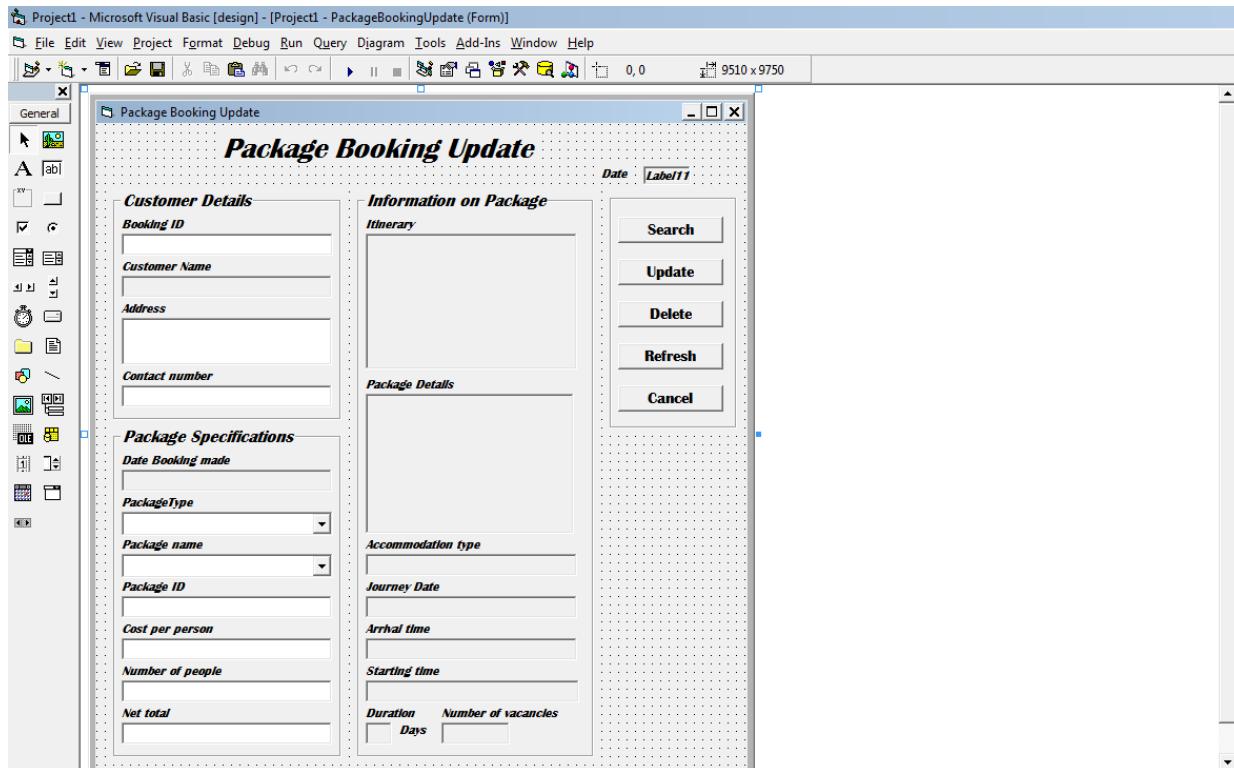
```
'Number of people keyup event.
Private Sub txtNPeople_KeyUp(KeyCode As Integer, Shift As Integer)
'The bar moves to booking ID textbox when enter is pressed after the number of people textbox is filled.
```

Jan. 1

Travel Agency Management System

```
If KeyCode = vbKeyReturn Then  
    txtBookingID.SetFocus  
End If  
  
End Sub  
  
'Number of people lostfocus event.  
Private Sub txtNPeople_LostFocus()  
'Calculation of net total by multiplying Number of people with Cost per person.  
    txtNetTotal.Text = txtNPeople.Text * txtCostPerPerson.Text  
End Sub
```

7. Package Booking Update



Package Booking Update

Customer Details

Information on Package

Date **29-03-2015**

Package Specifications

Itinerary

Package Details

Accommodation type

Journey Date

Arrival time

Starting time

Duration **Number of vacancies**

Days

Search

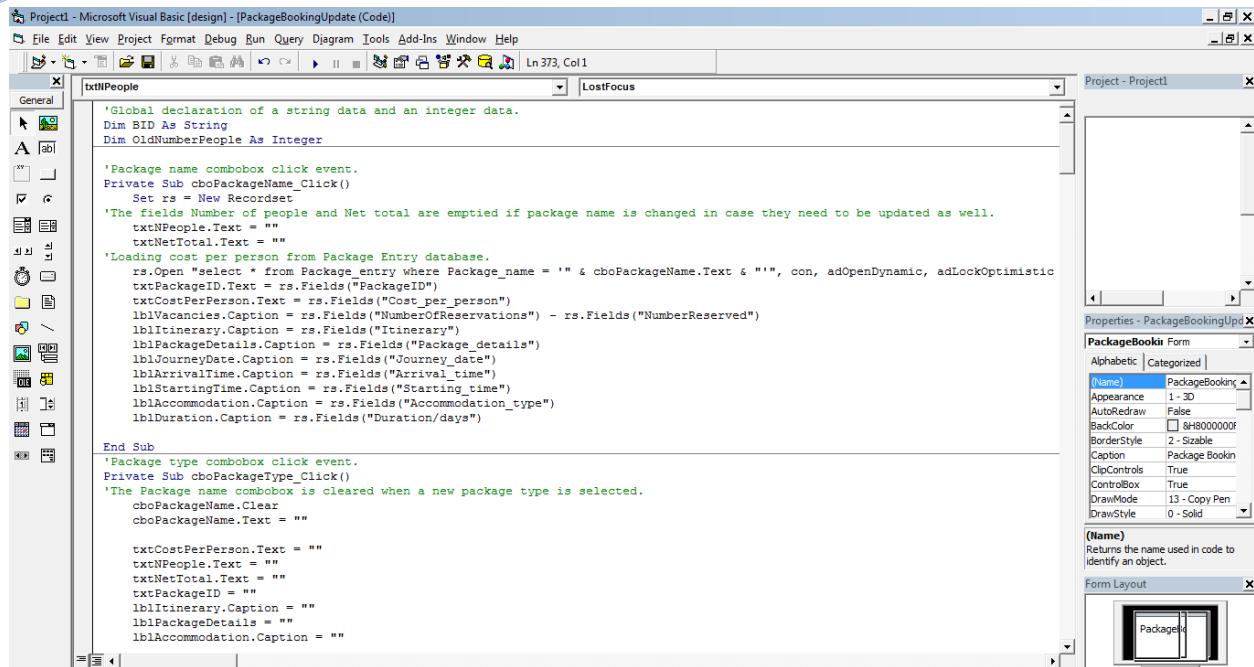
Update

Delete

Refresh

Cancel

The screenshot shows the actual layout of the Package Booking Update form.



'Global declaration of a string data and an integer data.

Dim BID As String

Dim OldNumberPeople As Integer

'Package name combobox click event.

Private Sub cboPackageName_Click()

 Set rs = New Recordset

'The fields Number of people and Net total are emptied if package name is changed in case they need to be updated as well.

 txtNPeople.Text = ""

 txtNetTotal.Text = ""

'Loading cost per person from Package Entry database.

 rs.Open "select * from Package_entry where Package_name = '" & cboPackageName.Text & "'", con, adOpenDynamic, adLockOptimistic

 txtPackageID.Text = rs.Fields("PackageID")

 txtCostPerPerson.Text = rs.Fields("Cost_per_person")

 lblVacancies.Caption = rs.Fields("NumberOfReservations") - rs.Fields("NumberReserved")

 lblItinerary.Caption = rs.Fields("Itinerary")

 lblPackageDetails.Caption = rs.Fields("Package_details")

 lblJourneyDate.Caption = rs.Fields("Journey_date")

 lblArrivalTime.Caption = rs.Fields("Arrival_time")

 lblStartingTime.Caption = rs.Fields("Starting_time")

 lblAccommodation.Caption = rs.Fields("Accommodation_type")

 lblDuration.Caption = rs.Fields("Duration/days")

'Package type combobox click event.

```
Private Sub cboPackageType_Click()
```

'The Package name combobox is cleared when a new package type is selected.

```
cboPackageName.Clear  
cboPackageName.Text = ""
```

```
txtCostPerPerson.Text = ""
```

```
txtNPeople.Text = ""
```

```
txtNetTotal.Text = ""
```

```
txtPackageID = ""
```

```
lblItinerary.Caption = ""
```

```
lblPackageDetails = ""
```

```
lblAccommodation.Caption = ""
```

```
lblJourneyDate.Caption = ""
```

```
lblStartingTime.Caption = ""
```

```
lblArrivalTime.Caption = ""
```

```
lblDuration.Caption = ""
```

```
lblItinerary.Caption = ""
```

```
lblPackageDetails.Caption = ""
```

'The Package names under the selected Package type are loaded.

```
Set rs = New Recordset
```

```
rs.Open "select * from Package_entry where Package_type=''" & cboPackageType.Text & "'", con,  
adOpenDynamic, adLockOptimistic
```

```
Do While Not rs.EOF
```

```
    cboPackageName.AddItem (rs.Fields("Package_name"))
```

```
    rs.MoveNext
```

```
Loop
```

```
End Sub
```

'Cancel button click event.

```
Private Sub cmdCancel_Click()
```

'Unloading the form.

```
Unload Me
```

```
End Sub
```

'Delete button click event.

```
Private Sub cmdDelete_Click()
```

'Validation check to make sure none of the fields are left empty.

```
If txtBookingID.Text = Empty Or lblName.Caption = Empty Or txtAddress.Text = Empty Or  
cboPackageType.Text = Empty Or txtContactNumber.Text = Empty Or txtCostPerPerson.Text = Empty Or  
cboPackageName.Text = Empty Or txtNPeople.Text = Empty Or txtNetTotal = Empty Or  
txtPackageID.Text = Empty Then
```

'Message shown to notify the user that none of the fields can be left empty.

```
    MsgBox "None of the fields can be left empty."
```

```
    Exit Sub
```

```
End If
```

'The selected record is deleted.

```
    rs.Delete
```

'Message shown to notify user that the record has been deleted successfully.

```
MsgBox "The Booking has been deleted."
rs.Close
'The reservation data is updated in Package Entry database after deletion of a booking.
rs.Open "select * from Package_entry where Package_name ='" & cboPackageName.Text & "'", con,
adOpenDynamic, adLockOptimistic
Dim NReserve As Integer
NReserve = rs.Fields("NumberReserved")
NReserve = NReserve - txtNPeople.Text
rs.Fields("NumberReserved") = NReserve
rs.Update
'The fields are emptied after deletion of the record.
cboPackageName.Text = ""
txtBookingID.Text = ""
lblName.Caption = ""
txtAddress.Text = ""
cboPackageType.Text = ""
txtContactNumber.Text = ""
txtCostPerPerson.Text = ""
txtNPeople.Text = ""
txtNetTotal.Text = ""
txtPackageID = ""
lblItinerary.Caption = ""
lblPackageDetails = ""
lblAccommodation.Caption = ""
lblJourneyDate.Caption = ""
lblStartingTime.Caption = ""
lblArrivalTime.Caption = ""
lblDuration.Caption = ""
lblBookingDate.Caption = ""

Call Form_Load

End Sub
'Refresh button click event.
Private Sub cmdRefresh_Click()
'The fields are cleared and the form is refreshed on the click of a button.
cboPackageName.Text = ""
txtBookingID.Text = ""
lblName.Caption = ""
txtAddress.Text = ""
cboPackageType.Text = ""
txtContactNumber.Text = ""
txtCostPerPerson.Text = ""
txtNPeople.Text = ""
txtNetTotal.Text = ""
txtPackageID = ""
lblItinerary.Caption = ""
```

```

lblPackageDetails = ""
lblAccommodation.Caption = ""
lblJourneyDate.Caption = ""
lblStartingTime.Caption = ""
lblArrivalTime.Caption = ""
lblDuration.Caption = ""
lblItinerary.Caption = ""
lblPackageDetails.Caption = ""

cboPackageType.Clear
cboPackageName.Clear

Call Form_Load

End Sub

'Search button click event.
Private Sub cmdSearch_Click()
'Making sure Booking ID is provided.
If txtBookingID.Text = "" Then
    MsgBox "Please provide Booking ID and try again."
    Exit Sub
End If

'The Package booking database is searched for a paticular Booking ID.
Set rs = New Recordset
rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic
Dim found As Boolean
found = False
rs.MoveFirst
Do While Not rs.EOF And Not found
    If rs.Fields("PackageBookingID") = txtBookingID.Text Then
        found = True
    End If
    rs.MoveNext
Loop

If found = False Then
'Message shown to notify User if Booking ID not found.
    MsgBox "There is no record for this Booking ID."
    Exit Sub
End If
rs.Close

'If booking ID found, the information the record containing this Booking ID holds is loaded.
    rs.Open "select * from Package_booking where PackageBookingID ='" & txtBookingID.Text & "'", con,
adOpenDynamic, adLockOptimistic
    cboPackageType.Text = rs.Fields("Package_type")

```

```

txtPackageID.Text = rs.Fields("PackageID")
lblName.Caption = rs.Fields("Customer_name")
txtAddress.Text = rs.Fields("Customer_address")
lblJourneyDate.Caption = rs.Fields("Journey_date")
txtContactNumber.Text = rs.Fields("Contact")
txtCostPerPerson.Text = rs.Fields("Cost_per_person")
cboPackageName.Text = rs.Fields("Package_name")
txtNPeople.Text = rs.Fields("Number_of_people")
txtNetTotal.Text = rs.Fields("Net_total")
lblBookingDate.Caption = rs.Fields("Date_of_booking")
lblItinerary.Caption = rs.Fields("Itinerary")
lblPackageDetails.Caption = rs.Fields("Package_details")
lblAccommodation.Caption = rs.Fields("Accommodation_type")
lblJourneyDate.Caption = rs.Fields("Journey_date")
lblArrivalTime.Caption = rs.Fields("Arrival_time")
lblStartingTime.Caption = rs.Fields("Starting_time")
lblDuration.Caption = rs.Fields("Duration/days")
lblItinerary.Caption = rs.Fields("Itinerary")
lblPackageDetails.Caption = rs.Fields("Package_details")
rs.Update

```

'Values are assigned to the variables BID and OldNumberPeople.

```

BID = txtBookingID.Text
OldNumberPeople = txtNPeople.Text
End Sub

```

'Update button click event.

```
Private Sub cmdUpdate_Click()
```

'Making sure a record is opened to cause alterations.

```

If txtBookingID.Text = "" Then
    MsgBox "Please provide Booking ID, press search and try again."
    Exit Sub
End If

```

'Validation check to make sure Booking ID is not changed.

```

If txtBookingID.Text <> BID Then
    MsgBox "Booking ID cannot be changed."
    txtBookingID = BID
    Exit Sub
End If

```

'Presence check to make sure all the required fields are filled by the customer.

```

If cboPackageType.Text = "" Or cboPackageName.Text = "" Or txtPackageID.Text = "" Or
txtCostPerPerson.Text = "" Or txtNPeople = "" Or txtAddress.Text = "" Or txtContactNumber.Text = ""
Then

```

'Message to notify the user that all of the information need to be provided.

```
MsgBox "All the Package information and customer information need to provided in order to"
```

proceed with the booking."

```
    Exit Sub  
End If
```

'Length check to make sure contact number is of 11 digits.

```
If Len(txtContactNumber.Text) <> 11 Then  
    MsgBox "Number of digits used for contact number is invalid."  
    Exit Sub  
End If
```

```
Set rs = New Recordset
```

'Validation check to ensure the updated reservations doesnot exceed the reservation limit.

```
    rs.Open "select * from Package_entry where Package_name ='" & cboPackageName.Text & "'", con,  
adOpenDynamic, adLockOptimistic
```

```
    Dim NewNumberReserved As Integer  
    NewNumberReserved = rs.Fields("NumberReserved")
```

```
NewNumberReserved = NewNumberReserved - OldNumberPeople + txtNPeople.Text
```

```
If NewNumberReserved > rs.Fields("NumberOfReservations") Then  
    MsgBox "There are not enough vacancies to make this reservation."  
    Exit Sub  
End If
```

```
If NewNumberReserved < rs.Fields("NumberOfReservations") Then  
    rs.Fields("NumberReserved") = NewNumberReserved  
    rs.Update  
End If
```

```
rs.Close
```

'Validation check to ensure none of the fields are left empty.

```
If txtBookingID.Text = Empty Or lblName.Caption = Empty _  
Or txtAddress.Text = Empty Or cboPackageType.Text = Empty _  
Or txtContactNumber.Text = Empty Or txtCostPerPerson.Text = Empty _  
Or cboPackageName.Text = Empty Or txtNPeople.Text = Empty _  
Or txtNetTotal = Empty Or txtPackageID.Text = Empty Then
```

'Message shown to notify user that none of the fields can be kept empty.

```
    MsgBox "None of the fields can be left empty."  
    Exit Sub  
End If
```

'The fields are added to the database.

```
rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic  
rs.Fields("Customer_name") = lblName.Caption  
rs.Fields("Customer_address") = txtAddress.Text  
rs.Fields("Package_type") = cboPackageType.Text  
rs.Fields("Journey_Date") = lblJourneyDate.Caption  
rs.Fields("Contact") = txtContactNumber.Text
```

```

rs.Fields("Cost_per_person") = txtCostPerPerson.Text
rs.Fields("Package_name") = cboPackageName.Text
rs.Fields("Number_of_people") = txtNPeople.Text
rs.Fields("Net_total") = txtNetTotal.Text
rs.Fields("Date_of_booking") = lblDate.Caption
rs.Fields("Accommodation_type") = lblAccommodation.Caption
rs.Fields("Journey_date") = lblJourneyDate.Caption
rs.Fields("Starting_time") = lblStartingTime.Caption
rs.Fields("Arrival_time") = lblArrivalTime.Caption
rs.Fields("Duration/days") = lblDuration.Caption
rs.Update

```

MsgBox "Booking information has been Updated successfully."

'The fields on the form are emptied.

```

cboPackageName.Text = ""
txtBookingID.Text = ""
lblName.Caption = ""
txtAddress.Text = ""
cboPackageType.Text = ""
txtContactNumber.Text = ""
txtCostPerPerson.Text = ""
txtNPeople.Text = ""
txtNetTotal.Text = ""
txtPackageID = ""
lblItinerary.Caption = ""
lblPackageDetails = ""
lblAccommodation.Caption = ""
lblJourneyDate.Caption = ""
lblStartingTime.Caption = ""
lblArrivalTime.Caption = ""
lblDuration.Caption = ""
lblItinerary.Caption = ""
lblPackageDetails.Caption = ""
lblBookingDate.Caption = ""

```

Call Form_Load

End Sub

'Form load event.

Private Sub Form_Load()

'Calling function to connect to database.

Call dblink

lblDate.Caption = Format(Now, "dd/mm/yyyy")

'Loading Package types from database.

Set rs = New Recordset

rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic

```
If rs.EOF = False Then
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboPackageType.ListCount - 1
            If cboPackageType.List(i) = rs.Fields("Package_type") Then
                IsNewType = False
            End If
        Next
        If IsNewType = True Then
            cboPackageType.AddItem (rs.Fields("Package_type"))
        End If
        rs.MoveNext
    Loop
End If

End Sub
```

'Booking ID keyup event.

```
Private Sub txtBookingID_KeyUp(KeyCode As Integer, Shift As Integer)
'Shifts to Search button when "Enter" pressed.
If KeyCode = vbKeyReturn Then
    cmdSearch.SetFocus
End If
End Sub
```

'Contact number keypress event.

```
Private Sub txtContactNumber_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```

'Cost per person keypress event.

```
Private Sub txtCostPerPerson_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```

'Net total keypress event.

```
Private Sub txtNetTotal_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

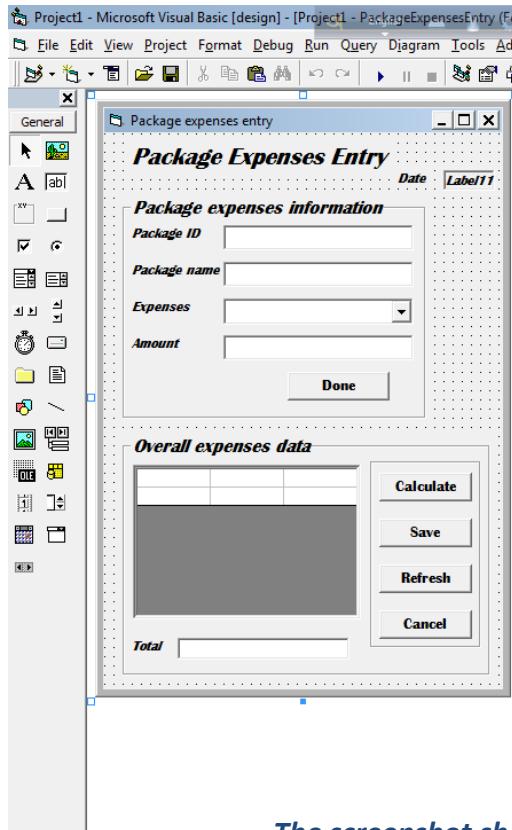
'Number of people keypress event.
Private Sub txtNPeople_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub

'Number of people Keyup event.
Private Sub txtNPeople_KeyUp(KeyCode As Integer, Shift As Integer)
'Character type check for numerical input.
If KeyCode = vbKeyReturn Then
    cmdUpdate.SetFocus
End If

End Sub

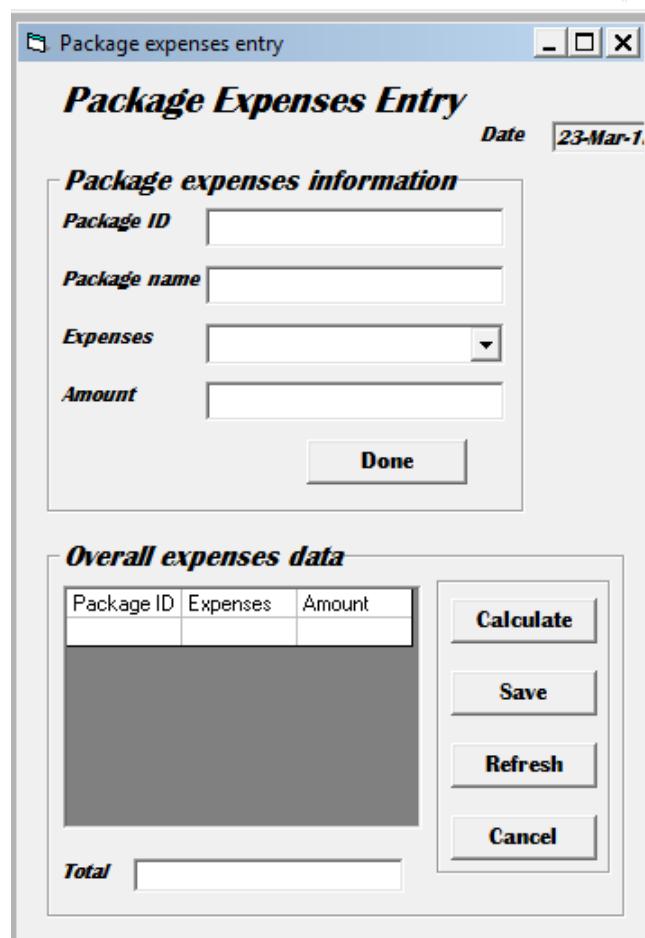
'Number of people Lostfocus event.
Private Sub txtNPeople_LostFocus()
'Calculation of Net total.
txtNetTotal.Text = txtNPeople.Text * txtCostPerPerson.Text
End Sub
```

8. Package Expenses Entry

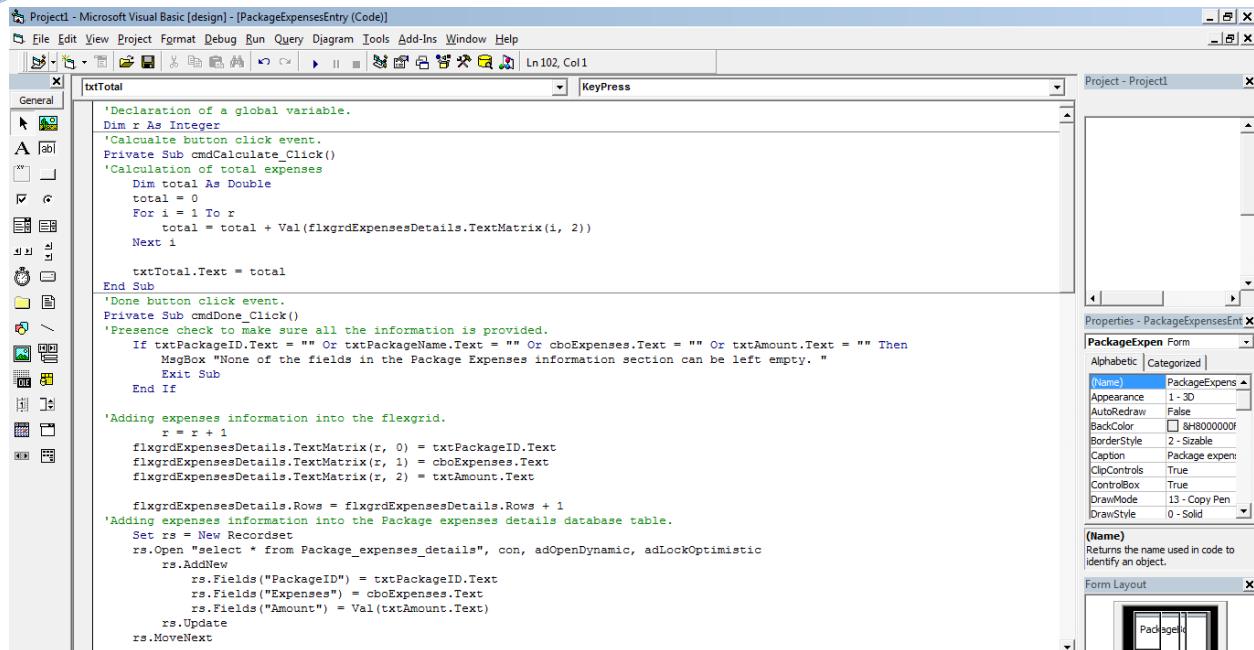


The screenshot shows the Microsoft Visual Basic design mode interface. A window titled "Project1 - Microsoft Visual Basic [design] - [Project1 - PackageExpensesEntry (Form)]" is open. The form itself is titled "Package expenses entry". It contains two main sections: "Package expenses information" and "Overall expenses data". The "Package expenses information" section includes fields for "Package ID", "Package name", "Expenses" (a dropdown menu), and "Amount". Below these is a "Done" button. The "Overall expenses data" section features a grid table with columns "Package ID", "Expenses", and "Amount", a "Calculate" button, a "Save" button, a "Refresh" button, and a "Cancel" button. A "Total" input field is also present. The "Date" field in the top right of the form is set to "Label11".

The screenshot shows the actual layout of the Package Expenses Entry form.



The screenshot shows the "Package expenses entry" form running. The title bar displays "Package Expenses Entry" and the date "23-Mar-1". The "Package expenses information" section is identical to the design view. In the "Overall expenses data" section, the "Calculate", "Save", "Refresh", and "Cancel" buttons are visible on the right. The "Total" input field is present at the bottom. The "Date" field now displays the value "23-Mar-1".



'Declaration of a global variable.

Dim r As Integer

'Calcualte button click event.

Private Sub cmdCalculate_Click()

'Calculation of total expenses

 Dim total As Double

 total = 0

 For i = 1 To r

 total = total + Val(flxgrdExpensesDetails.TextMatrix(i, 2))

 Next i

 txtTotal.Text = total

End Sub

'Done button click event.

Private Sub cmdDone_Click()

'Presence check to make sure all the information is provided.

 If txtPackageID.Text = "" Or txtPackageName.Text = "" Or cboExpenses.Text = "" Or txtAmount.Text = "" Then

 MsgBox "None of the fields in the Package Expenses information section can be left empty."

 Exit Sub

 End If

'Adding expenses information into the flexgrid.

 r = r + 1

 flxgrdExpensesDetails.TextMatrix(r, 0) = txtPackageID.Text

 flxgrdExpensesDetails.TextMatrix(r, 1) = cboExpenses.Text

 flxgrdExpensesDetails.TextMatrix(r, 2) = txtAmount.Text

```

flxgrdExpensesDetails.Rows = flxgrdExpensesDetails.Rows + 1
'Adding expenses information into the Package expenses details database table.
Set rs = New Recordset
rs.Open "select * from Package_expenses_details", con, adOpenDynamic, adLockOptimistic
rs.AddNew
rs.Fields("PackageID") = txtPackageID.Text
rs.Fields("Expenses") = cboExpenses.Text
rs.Fields("Amount") = Val(txtAmount.Text)
rs.Update
rs.MoveNext
'These fields are cleared in case they need to be changed.
cboExpenses.Text = ""
txtAmount.Text = ""
End Sub
'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
Unload Me
End Sub
'Refresh button click event.
Private Sub cmdRefresh_Click()
'All the fields are emptied and refreshed in case the user wants to start anew.
txtPackageID.Text = ""
txtPackageName.Text = ""
cboExpenses.Text = ""
flxgrdExpensesDetails.Clear
flxgrdExpensesDetails.Rows = 2
Call Form_Load
txtTotal.Text = ""
End Sub
'Save button click event.
Private Sub cmdSave_Click()
'Storing final cost and expenses information in the Package expenses master database.
Set rs = New Recordset
rs.Open "select * from Package_expenses_master", con, adOpenDynamic, adLockOptimistic
rs.AddNew
rs.Fields("PackageID") = txtPackageID.Text
rs.Fields("Package_name") = txtPackageName.Text
rs.Fields("Total") = txtTotal.Text
rs.Update
MsgBox "The record has been saved."
End Sub
'Form load event.
Private Sub Form_Load()
'Calling the function to connect the form to the database.
Call dblink
Showing the current date on the form.

```

```
lblDate.Caption = Date  
'Assigning fields to the flexgrid columns.  
flxgrdExpensesDetails.TextMatrix(0, 0) = "Package ID"  
flxgrdExpensesDetails.TextMatrix(0, 1) = "Expenses"  
flxgrdExpensesDetails.TextMatrix(0, 2) = "Amount"  
r = 0  
End Sub
```

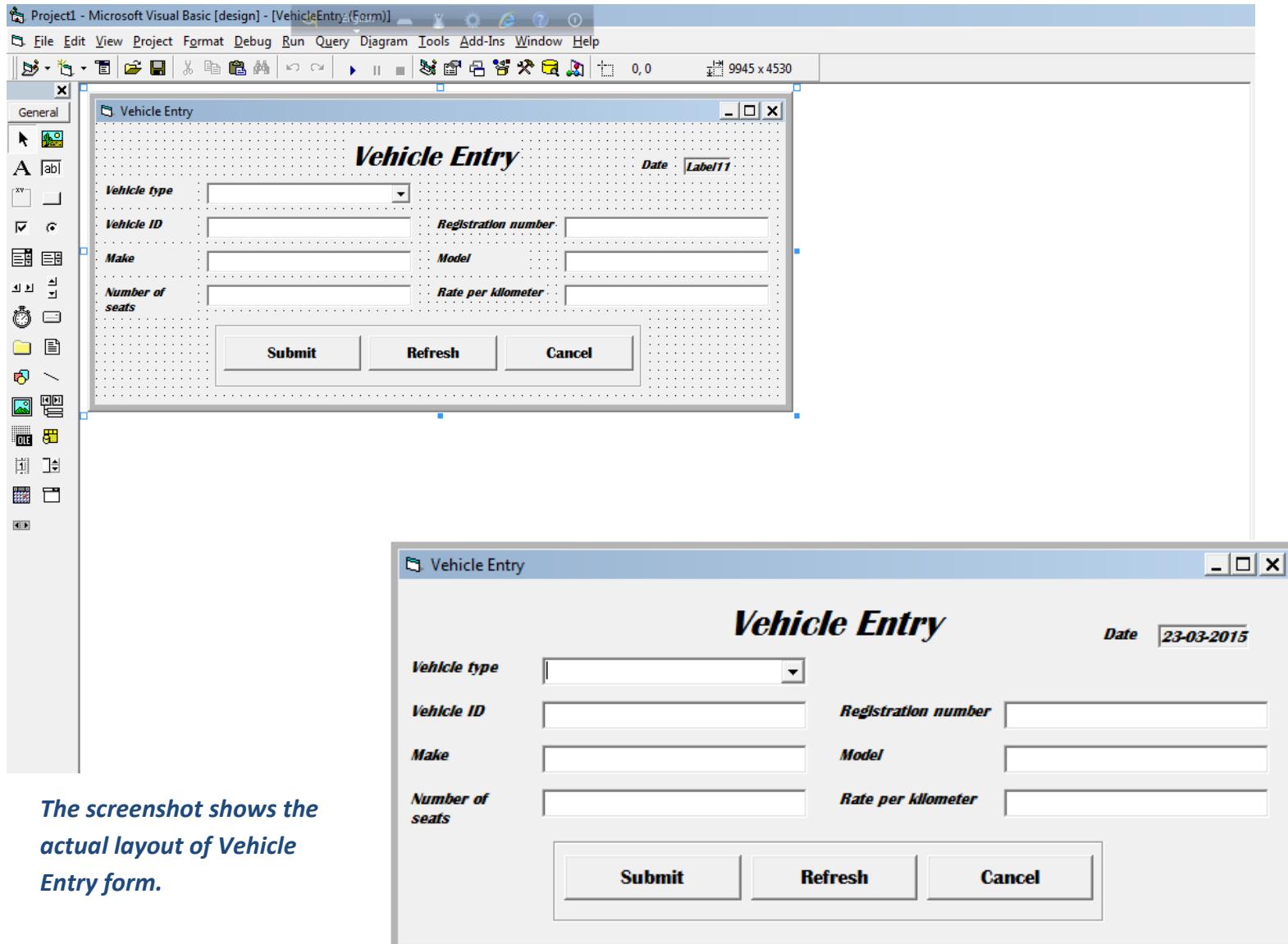
```
'Amount keypress event.  
Private Sub txtAmount_KeyPress(KeyAscii As Integer)  
'Character type check to make sure only numerical data is entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If
```

```
End Sub
```

```
'Amount textbox keyup event.  
Private Sub txtAmount_KeyUp(KeyCode As Integer, Shift As Integer)  
'Directing the cursor to the Done button after information is inserted in the amount textbox.  
If KeyCode = vbKeyReturn Then  
    cmdDone.SetFocus  
End If  
End Sub
```

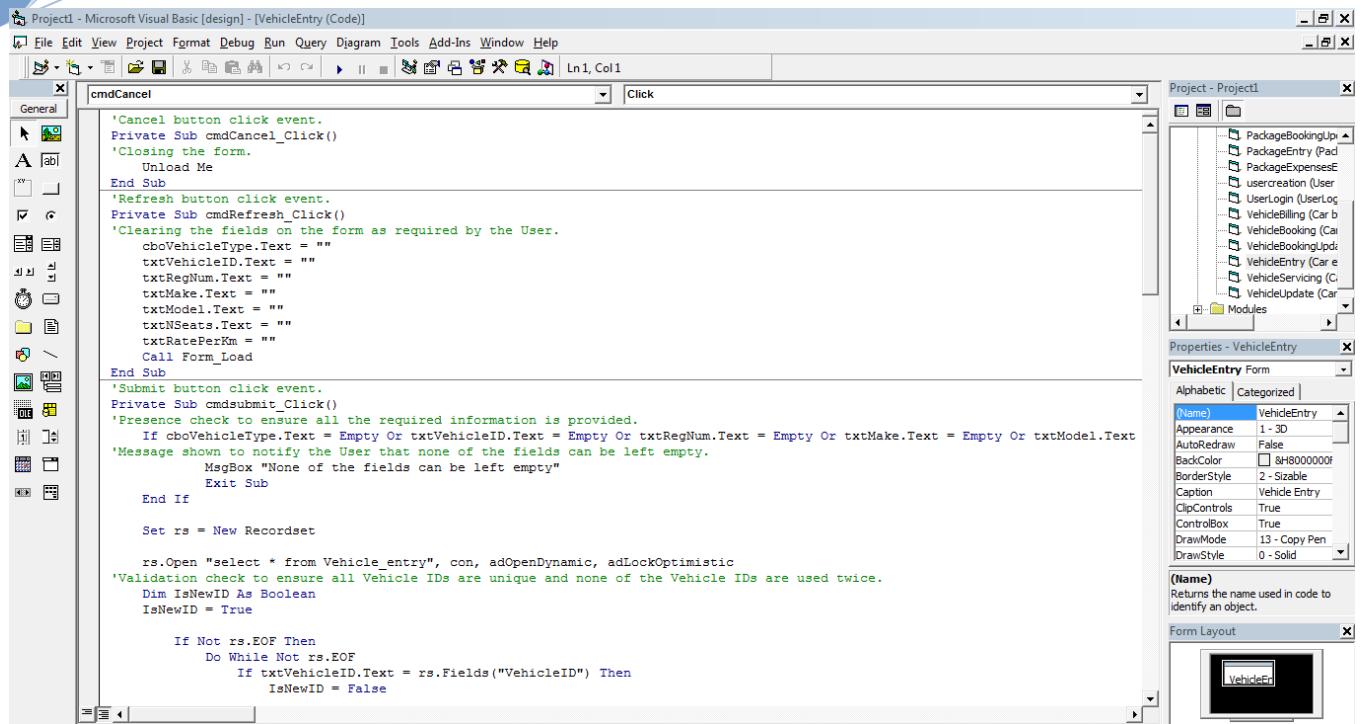
```
'Total keypress event.  
Private Sub txtTotal_KeyPress(KeyAscii As Integer)  
'Character type check to make sure only numerical data is entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
End Sub
```

9. Vehicle Entry



The screenshot shows the actual layout of Vehicle Entry form.

Travel Agency Management System



```

'Cancel button click event.
Private Sub cmdCancel_Click()
    'Closing the form.
    Unload Me
End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
    'Clearing the fields on the form as required by the User.
    cboVehicleType.Text = ""
    txtVehicleID.Text = ""
    txtRegNum.Text = ""
    txtMake.Text = ""
    txtModel.Text = ""
    txtNSeats.Text = ""
    txtRatePerKm = ""
    Call Form_Load
End Sub

'Submit button click event.
Private Sub cmdsubmit_Click()
    'Presence check to ensure all the required information is provided.
    If cboVehicleType.Text = Empty Or txtVehicleID.Text = Empty Or txtRegNum.Text = Empty Or
    txtMake.Text = Empty Or txtModel.Text = Empty Or txtNSeats.Text = Empty Or txtRatePerKm.Text =
    Empty Then
        MsgBox "None of the fields can be left empty"
    End If
End Sub

```

```
Exit Sub
End If

Set rs = New Recordset

rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Validation check to ensure all Vehicle IDs are unique and none of the Vehicle IDs are used twice.
Dim IsNewID As Boolean
IsNewID = True

If Not rs.EOF Then
    Do While Not rs.EOF
        If txtVehicleID.Text = rs.Fields("VehicleID") Then
            IsNewID = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID = False Then
    'Message shown if Vehicle ID is already in use.
    MsgBox "This Vehicle ID is already in use. Please try again."
    txtVehicleID.Text = ""
    Exit Sub
End If
rs.MoveFirst
Dim IsNewID2 As Boolean
IsNewID2 = True

If Not rs.EOF Then

    Do While Not rs.EOF
        If txtRegNum.Text = rs.Fields("RegistrationNo") Then
            IsNewID2 = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID2 = False Then
    MsgBox "This Registration number is already in use. Please try again."
    txtRegNum.Text = ""
    Exit Sub
End If
```

```
'Adding new Vehicle information into the database.
```

```
rs.AddNew
rs.Fields("Vehicle_type") = cboVehicleType.Text
rs.Fields("VehicleID") = txtVehicleID.Text
rs.Fields("RegistrationNo") = txtRegNum.Text
rs.Fields("Make") = txtMake.Text
rs.Fields("Model") = txtModel.Text
rs.Fields("NSeats") = txtNSeats.Text
rs.Fields("Rate_per_km") = txtRatePerKm.Text
rs.Fields("Date_of_entry") = lblDate.Caption
rs.Update
```

```
'Message shown to notify the User that new vehicle information is added successfully.
```

```
MsgBox "Vehicle information has been added successfully"
```

```
'Clearing the fields to ready the form for new transaction.
```

```
cboVehicleType.Text = ""
txtVehicleID.Text = ""
txtRegNum.Text = ""
txtMake.Text = ""
txtModel.Text = ""
txtNSeats.Text = ""
txtRatePerKm = ""
```

```
Call Form_Load
```

```
End Sub
```

```
'Form load event.
```

```
Private Sub Form_Load()
```

```
'Calling function to connect to database.
```

```
Call dblink
```

```
'Showing the current date on the form.
```

```
lblDate.Caption = Format(Now, "dd/mm/yyyy")
```

```
Set rs = New Recordset
```

```
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
```

```
rs.MoveLast
```

```
txtVehicleID.Text = CInt(rs!VehicleID) + 1
```

```
End Sub
```

```
'Number of seats Keypress event.
```

```
Private Sub txtNSeats_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure only numerical data is entered.
```

```
If KeyAscii = 8 Then Exit Sub
```

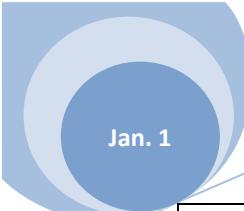
```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

```
End If
```

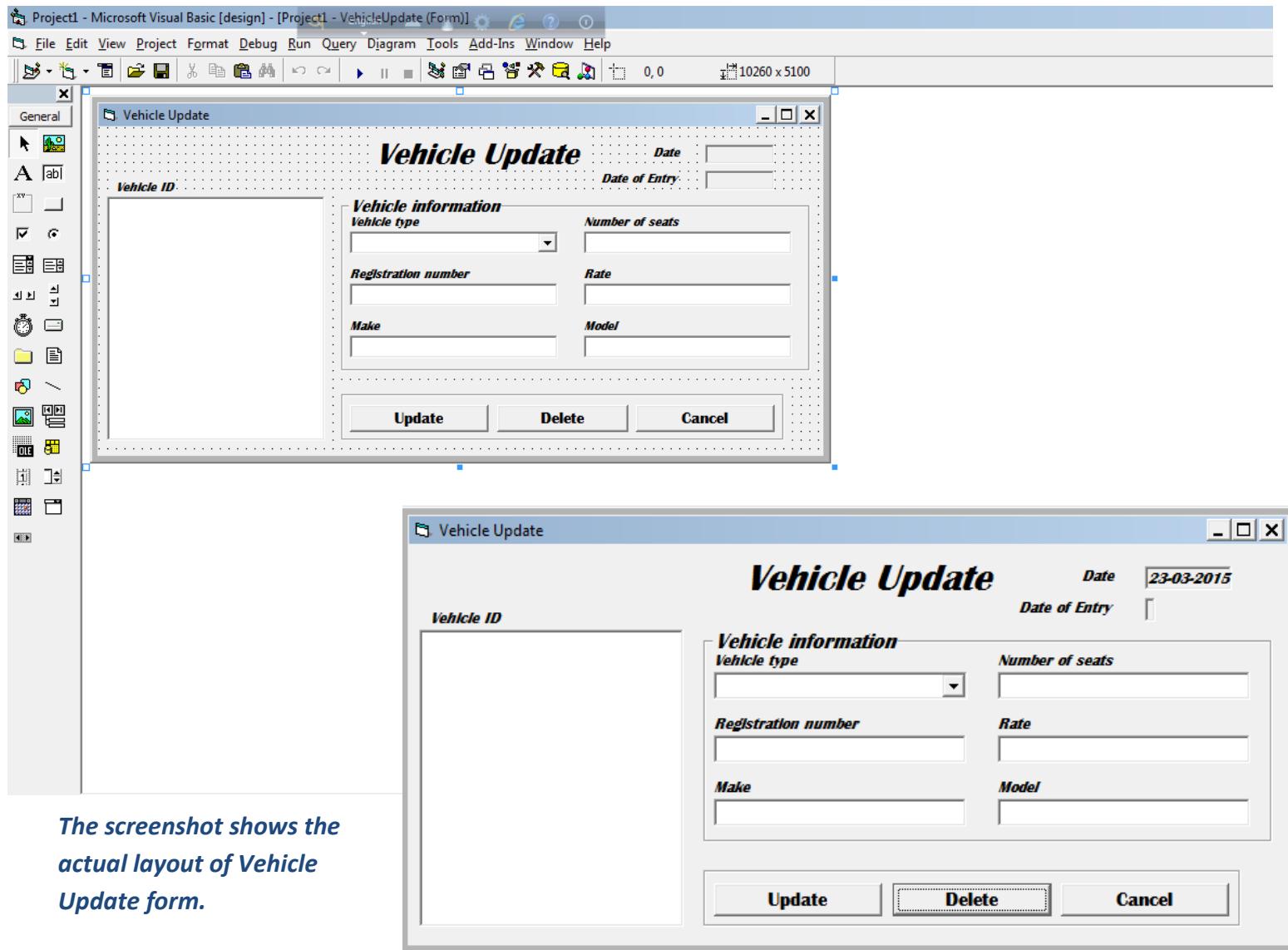
```
End Sub
```



Jan. 1

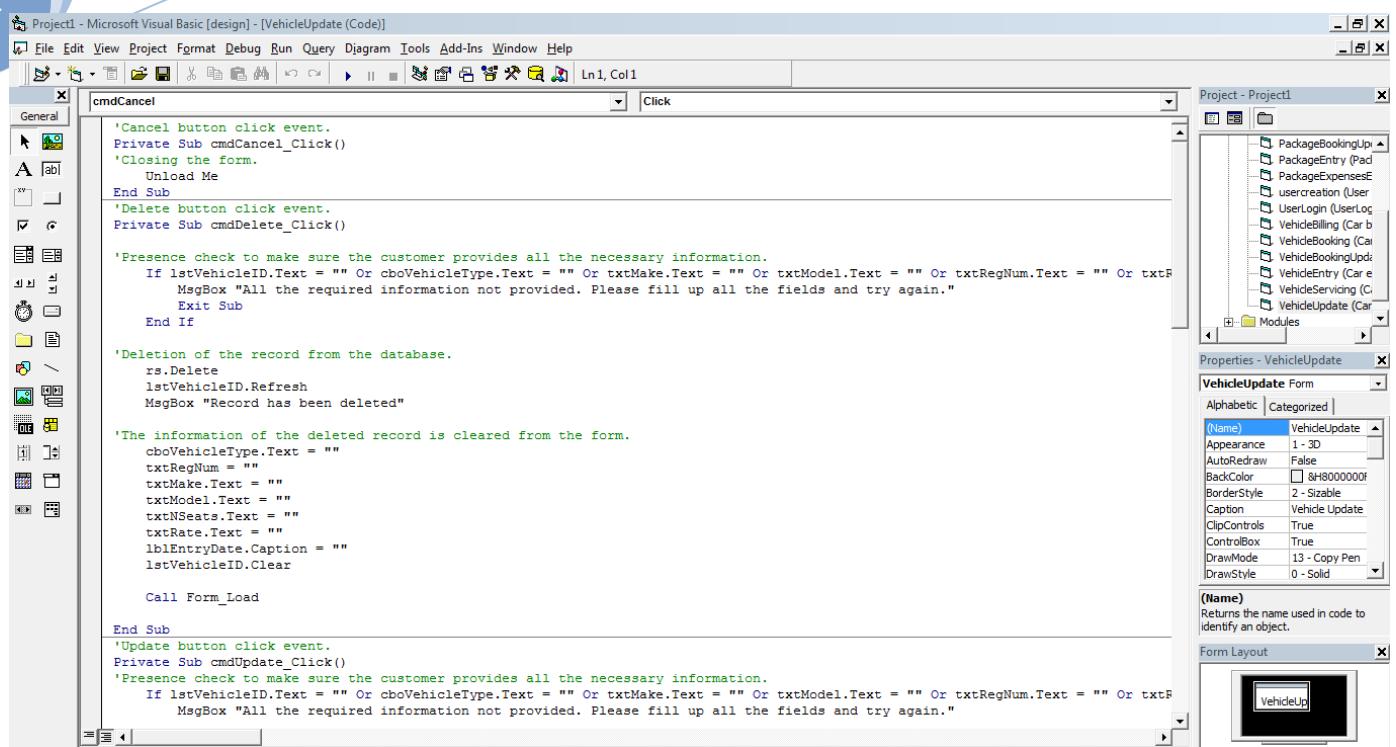
Travel Agency Management System

```
'Rate per kilometer Keypress event.  
Private Sub txtRatePerKm_KeyPress(KeyAscii As Integer)  
'Character type check to make sure only numerical data is entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
End Sub
```

10. Vehicle Update

The screenshot shows the actual layout of Vehicle Update form.

Travel Agency Management System



```
'Cancel button click event.
Private Sub cmdCancel_Click()
'Closing the form.
Unload Me
End Sub

'Delete button click event.
Private Sub cmdDelete_Click()

'Presence check to make sure the customer provides all the necessary information.
If lstVehicleID.Text = "" Or cboVehicleType.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or
txtRegNum.Text = "" Or txtRate.Text = "" Or txtNSeats.Text = "" Then
    MsgBox "All the required information not provided. Please fill up all the fields and try again."
    Exit Sub
End If

'Deletion of the record from the database.
rs.Delete
lstVehicleID.Refresh
MsgBox "Record has been deleted"

'The information of the deleted record is cleared from the form.
cboVehicleType.Text = ""
txtRegNum = ""
txtMake.Text = ""
txtModel.Text = ""
txtNSeats.Text = ""
```

```

txtRate.Text = ""
lblEntryDate.Caption = ""
lstVehicleID.Clear

Call Form_Load

End Sub

'Update button click event.
Private Sub cmdUpdate_Click()
'Presence check to make sure the customer provides all the necessary information.
If lstVehicleID.Text = "" Or cboVehicleType.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or
txtRegNum.Text = "" Or txtRate.Text = "" Or txtNSeats.Text = "" Then
    MsgBox "All the required information not provided. Please fill up all the fields and try again."
    Exit Sub
End If

'The Vehicle information in the database are updated.
rs.Fields("Vehicle_type") = cboVehicleType.Text
rs.Fields("RegistrationNo") = txtRegNum.Text
rs.Fields("Make") = txtMake.Text
rs.Fields("Model") = txtModel.Text
rs.Fields("NSeats") = txtNSeats.Text
rs.Fields("Rate_per_km") = txtRate.Text
rs.Update
'Message shown to notify the User that the Update was successful.
MsgBox "Vehicle information has been updated"
'Clearing the fields in the form to ready form for the next job.
cboVehicleType.Text = ""
txtRegNum = ""
txtMake.Text = ""
txtModel.Text = ""
txtNSeats.Text = ""
txtRate.Text = ""
lblEntryDate.Caption = ""
lstVehicleID.Clear

Call Form_Load
End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
Call dblink
'Showing the current date on the form.
lblDate.Caption = Format(Now, "dd/mm/yyyy")
'Loading the Vehicle ID information into the list of Vehicle IDs from the database.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic

```

```
Do While Not rs.EOF
    lstVehicleID.AddItem (rs.Fields("VehicleID"))
    rs.MoveNext
Loop

End Sub

'Vehicle ID list click event.
Private Sub lstVehicleID_Click()
    'Loading information for the particular Vehicle ID selected into the form from the database.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where VehicleID = '" & lstVehicleID.Text & "'", con,
adOpenDynamic, adLockOptimistic

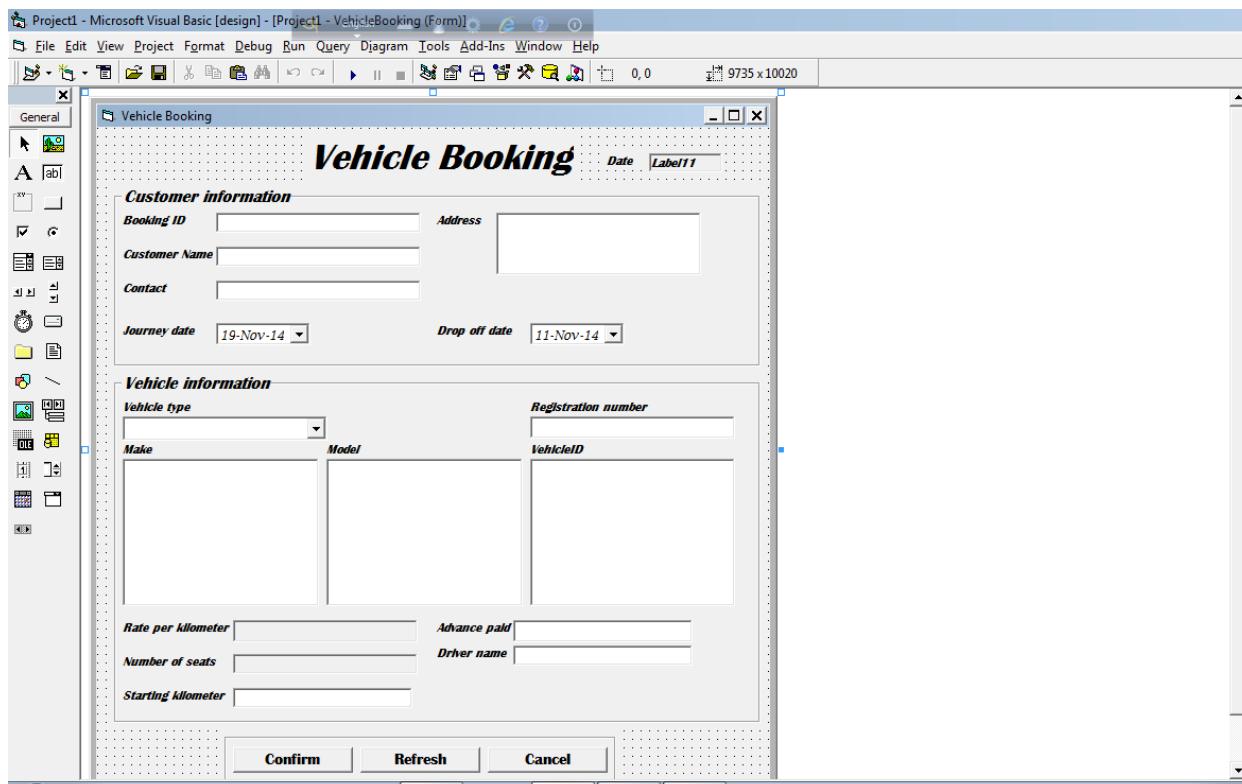
    cboVehicleType.Text = rs.Fields("Vehicle_type")
    txtRegNum.Text = rs.Fields("RegistrationNo")
    txtMake.Text = rs.Fields("Make")
    txtModel.Text = rs.Fields("Model")
    txtNSeats.Text = rs.Fields("NSeats")
    txtRate.Text = rs.Fields("Rate_per_km")
    lblEntryDate.Caption = rs.Fields("Date_of_entry")
    rs.Update
End Sub

'Number of seats Keypress event.
Private Sub txtNSeats_KeyPress(KeyAscii As Integer)
    'Character type check to make sure only numerical data is input.
    If KeyAscii = 8 Then Exit Sub
    If Not IsNumeric(Chr(KeyAscii)) Then
        MsgBox "Please input numbers only."
        Exit Sub
    End If

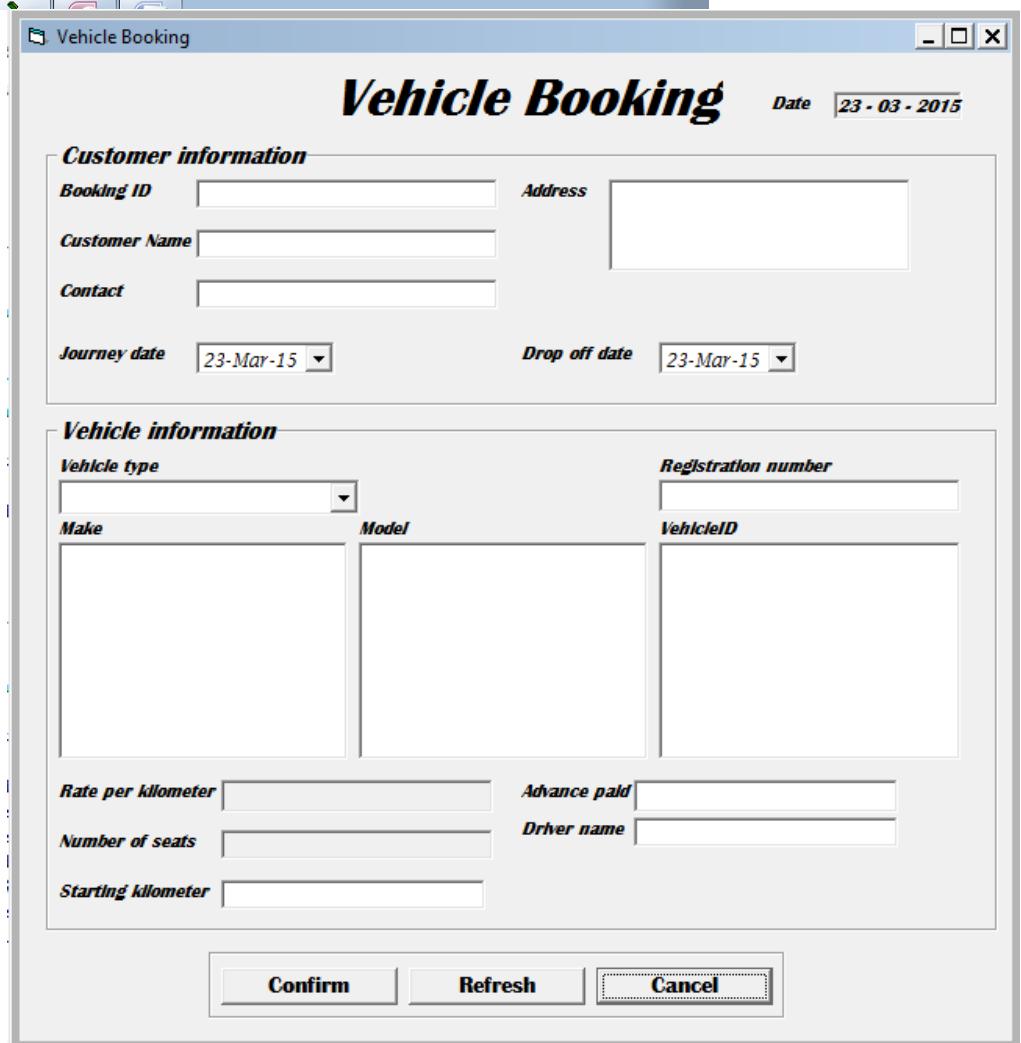
End Sub

'Rate per kilometer keypress event.
Private Sub txtRate_KeyPress(KeyAscii As Integer)
    'Character type check to make sure only numerical data is input.
    If KeyAscii = 8 Then Exit Sub
    If Not IsNumeric(Chr(KeyAscii)) Then
        MsgBox "Please input numbers only."
        Exit Sub
    End If

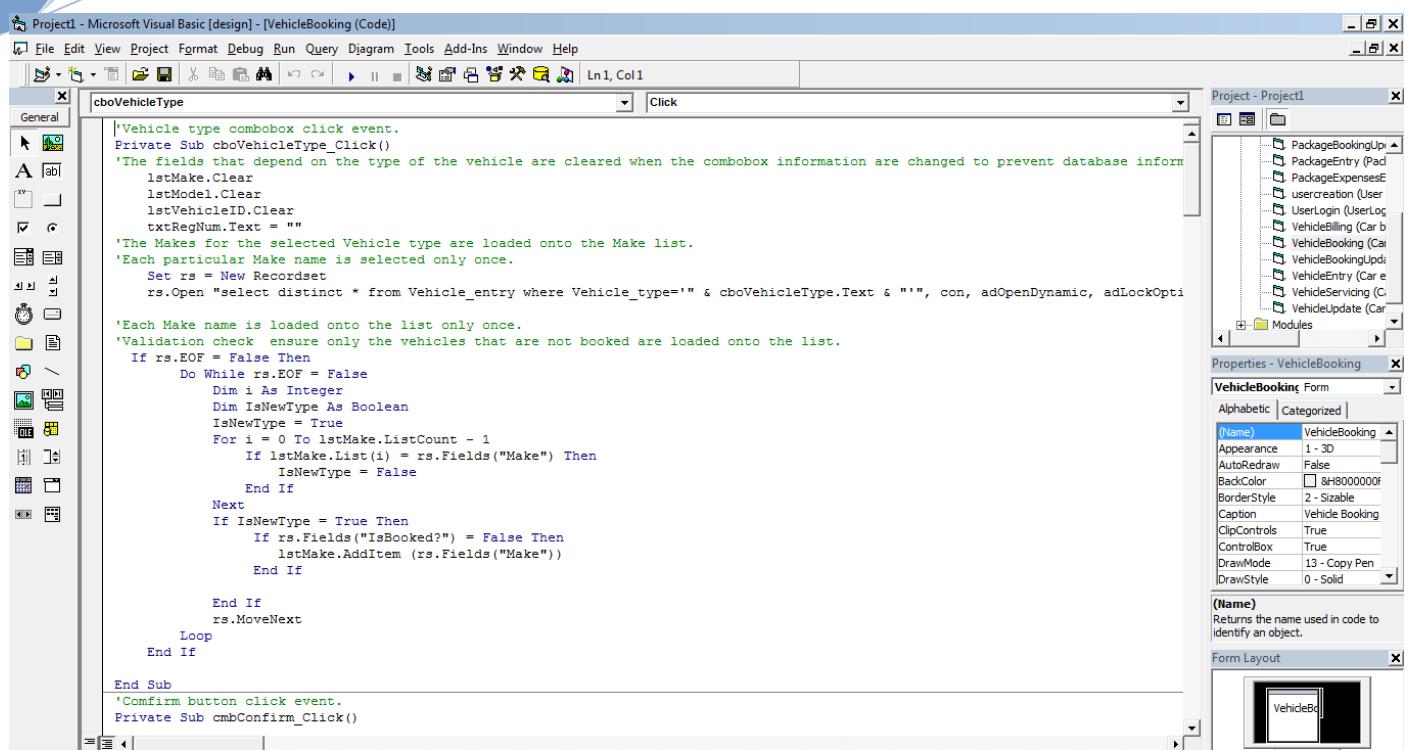
End Sub
```

11. Vehicle Booking

The screenshot shows the actual layout of Vehicle Booking form.



Travel Agency Management System



'Vehicle type combobox click event.

Private Sub cboVehicleType_Click()

'The fields that depend on the type of the vehicle are cleared when the combobox information are changed to prevent database information storage override.

IstMake.Clear

IstModel.Clear

IstVehicleID.Clear

txtRegNum.Text = ""

'The Makes for the selected Vehicle type are loaded onto the Make list.

'Each particular Make name is selected only once.

Set rs = New Recordset

rs.Open "select distinct * from Vehicle_entry where Vehicle_type= '" & cboVehicleType.Text & "'", con, adOpenDynamic, adLockOptimistic

'Each Make name is loaded onto the list only once.

'Validation check ensure only the vehicles that are not booked are loaded onto the list.

If rs.EOF = False Then

Do While rs.EOF = False

Dim i As Integer

Dim IsNewType As Boolean

IsNewType = True

For i = 0 To IstMake.ListCount - 1

If IstMake.List(i) = rs.Fields("Make") Then

IsNewType = False

End If

```

Next
If IsNewType = True Then
    If rs.Fields("IsBooked?") = False Then
        lstMake.AddItem (rs.Fields("Make"))
    End If

End If
rs.MoveNext
Loop
End If

End Sub
'Comfirm button click event.
Private Sub cmbConfirm_Click()
'Presence check to make sure that none of the fields are left empty.
    If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = "" Or
    cboVehicleType.Text = "" Or txtRegNum.Text = "" Or txtAdvancePaid.Text = "" Or txtDriverName.Text =
    "" Or txtStartingKm.Text = "" Then
        MsgBox "Some of the required fields are not filled. Please fill up all of the fields and try again."
        Exit Sub
    End If

'Length check to make sure contact number is of 11 digits.
    If Len(txtContact.Text) <> 11 Then
        MsgBox "Number of digits used for contact number is invalid."
        Exit Sub
    End If

Set rs = New Recordset

    rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic
'Validation check to make sure the same Booking ID is not used twice.
    If Not rs.EOF Then
        Dim IsNewID As Boolean
        IsNewID = True
        Do While Not rs.EOF
            If txtBookingID.Text = rs.Fields("VehicleBookingID") Then
                IsNewID = False
                Exit Do
            End If
            rs.MoveNext
        Loop
    End If
'Message shown to notify the user that the Booking ID is in use if the same User ID is found in database.
    If IsNewID = False Then
        MsgBox "This Booking ID is already in use. Please try again."
        txtBookingID.Text = ""
    End If

```

```

    Exit Sub
End If
'Storage of Vehicle Booking information into the database.
rs.AddNew
    rs.Fields("VehicleBookingID") = txtBookingID.Text
    rs.Fields("Customer_name") = txtName.Text
    rs.Fields("Customer_address") = txtAddress.Text
    rs.Fields("Contact") = txtContact.Text
    rs.Fields("Journey_date") = dtpJourneyDate.Value
    rs.Fields("Drop_off_date") = dtpDropOffDate.Value
    rs.Fields("Vehicle_type") = cboVehicleType.Text
    rs.Fields("Registration_number") = txtRegNum.Text
    rs.Fields("Make") = lstMake.Text
    rs.Fields("Model") = lstModel.Text
    rs.Fields("VehicleID") = lstVehicleID.Text
    rs.Fields("Rate_per_kilometer") = lblRatePerKm.Caption
    rs.Fields("Number_of_seats") = lblNSeats.Caption
    rs.Fields("Advance_paid") = txtAdvancePaid.Text
    rs.Fields("Driver_name") = txtDriverName.Text
    rs.Fields("Date_of_booking") = lblDate.Caption
    rs.Fields("Starting_Km") = txtStartingKm.Text
rs.Update

'Message shown notify the User that Booking was successful.
MsgBox "Booking has been made successfully."
rs.Close
'The Vehicle is made unavailable to further bookings.
rs.Open "select * from Vehicle_entry where VehicleID ='" & lstVehicleID.Text & "'", con,
adOpenDynamic, adLockOptimistic
    rs.Fields("IsBooked?") = True
rs.Update
'The fields are all cleared and the form is refreshed after the booking is made.
txtBookingID.Text = ""
txtName.Text = ""
txtContact.Text = ""
txtAddress.Text = ""
dtpJourneyDate.Value = Date
dtpDropOffDate.Value = Date
cboVehicleType.Text = ""
txtRegNum.Text = ""
lstMake.Clear
lstModel.Clear
lstVehicleID.Clear
lblRatePerKm.Caption = ""
lblNSeats.Caption = ""
txtAdvancePaid = ""
txtDriverName.Text = ""

```

```
txtStartingKm.Text = ""

Call Form_Load
End Sub

'Cancel button click event
Private Sub cmdCancel_Click()
'Unloading the form.
Unload Me
End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
'The fields are cleared at the click of the button as the User might require in case he/she needs to start anew.
txtBookingID.Text = ""
txtName.Text = ""
txtContact.Text = ""
txtAddress.Text = ""
dtpJourneyDate.Value = Date
dtpDropOffDate.Value = Date
cboVehicleType.Text = ""
txtRegNum.Text = ""
lstMake.Clear
lstModel.Clear
lstVehicleID.Clear
lblRatePerKm.Caption = ""
lblNSeats.Caption = ""
txtAdvancePaid = ""
txtDriverName.Text = ""
txtStartingKm.Text = ""

Call Form_Load

End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
Call dblink
Shows current date.
lblDate.Caption = Format(Now, "dd / mm / yyyy")
'The date pickers are assigned current dates as default.
dtpJourneyDate.Value = Date
dtpDropOffDate.Value = Date
'Loading the Vehicle type information onto the Vehicle type combobox from Vehicle entry database.

Set rs = New Recordset
rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic
rs.MoveLast
```

```

txtBookingID.Text = CInt(rs!VehicleBookingID) + 1
rs.Close
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Validity check to make sure each type is loaded only once.
If rs.EOF = False Then
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboVehicleType.ListCount - 1
            If cboVehicleType.List(i) = rs.Fields("Vehicle_type") Then
                IsNewType = False
            End If
        Next
        If IsNewType = True Then
            'Validity check to make sure any booked vehicles are not loaded.
            If rs.Fields("IsBooked") = False Then
                cboVehicleType.AddItem (rs.Fields("Vehicle_type"))
            End If
        End If
        rs.MoveNext
    Loop
End If

End Sub

```

```

'Make list click event.
Private Sub lstMake_Click()
    'The information dependent on the Make of the vehicle are cleared to make sure no override occurs
    'when storing data into the database.
    lstModel.Clear
    lstVehicleID.Clear
    txtRegNum.Text = ""
    'The particular Models are loaded for the selected Make.
    Set rs = New Recordset
    rs.Open "select distinct * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "' and
    Make ='" & lstMake.Text & "'", con, adOpenDynamic, adLockOptimistic
    'Each Model name is loaded onto the list only once.
    'Validation check to make sure only the vehicles that are not booked are loaded onto the list.
    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True

```

```

For i = 0 To lstModel.ListCount - 1
    If lstModel.List(i) = rs.Fields("Model") Then
        IsNewType = False
    End If
Next
If IsNewType = True Then
    If rs.Fields("IsBooked?") = False Then
        lstModel.AddItem (rs.Fields("Model"))
    End If

End If
rs.MoveNext
Loop
End If

End Sub
'Model list click event.
Private Sub lstModel_Click()
'The information that are dependant on the Make and Model are refreshed if Model is changed, to
prevent database storage override.
    lstVehicleID.Clear
    txtRegNum.Text = ""
'The Vehicle IDs of the Model of Vehicles of the particular Make is loaded.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = "" & cboVehicleType & "" and Make = "" &
lstMake.Text & "" and Model = "" & lstModel.Text & "", con, adOpenDynamic, adLockOptimistic
    rs.MoveFirst

'Each Make name is loaded onto the list only once.
'Validation check ensure only the vehicles that are not booked are loaded onto the list.
    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True
            For i = 0 To lstVehicleID.ListCount - 1
                If lstVehicleID.List(i) = rs.Fields("VehicleID") Then
                    IsNewType = False
                End If
            Next
            If IsNewType = True Then
                If rs.Fields("IsBooked?") = False Then
                    lstVehicleID.AddItem (rs.Fields("VehicleID"))
                End If

            End If
        rs.MoveNext
    End If

```

```
Loop  
End If  
End Sub
```

'Vehicle ID list click event.

```
Private Sub IstVehicleID_Click()
```

'Further required information for the selected Vehicle is loaded.

```
    Set rs = New Recordset  
    rs.Open "select * from Vehicle_entry where VehicleID ='" & IstVehicleID.Text & "'", con,  
adOpenDynamic, adLockOptimistic  
    lblRatePerKm.Caption = rs.Fields("Rate_per_km")  
    lblNSeats.Caption = rs.Fields("NSeats")  
    txtRegNum.Text = rs.Fields("RegistrationNo")  
End Sub
```

'Advance paid keypress event.

```
Private Sub txtAdvancePaid_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure only numerical data is entered.

```
    If KeyAscii = 8 Then Exit Sub  
    If Not IsNumeric(Chr(KeyAscii)) Then  
        MsgBox "Please input numbers only."  
        Exit Sub  
    End If  
End Sub
```

'Contact number keypress event.

```
Private Sub txtContact_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure only numerical data is entered.

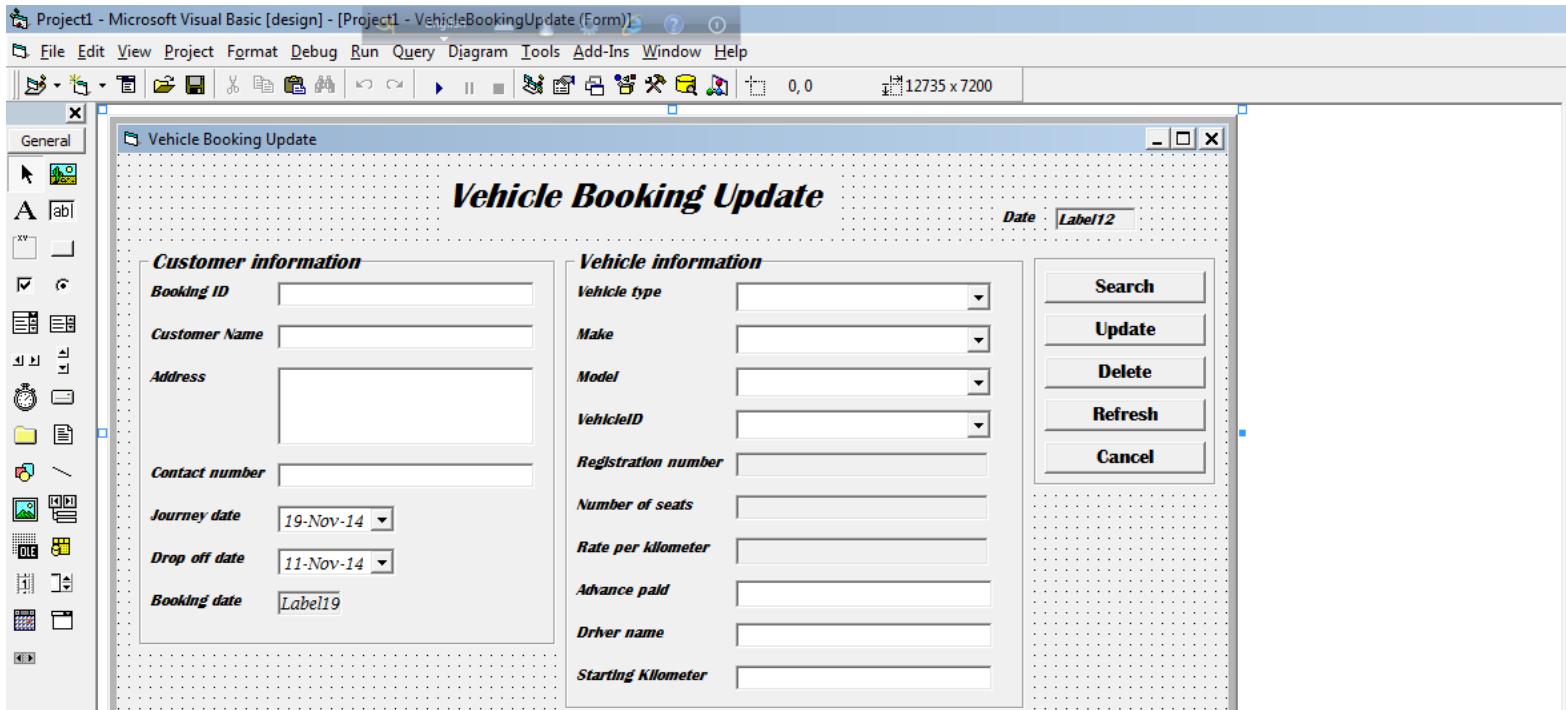
```
    If KeyAscii = 8 Then Exit Sub  
    If Not IsNumeric(Chr(KeyAscii)) Then  
        MsgBox "Please input numbers only."  
        Exit Sub  
    End If  
End Sub
```

'Starting kilometer keypress event.

```
Private Sub txtStartingKm_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure only numerical data is entered.

```
    If KeyAscii = 8 Then Exit Sub  
    If Not IsNumeric(Chr(KeyAscii)) Then  
        MsgBox "Please input numbers only."  
        Exit Sub  
    End If  
End Sub
```

12. Vehicle Booking Update

The screenshot shows the actual layout of Vehicle Booking Update form.

Customer information		Vehicle information		Date
Booking ID	<input type="text"/>	Vehicle type	<input type="text"/>	23-03-2015
Customer Name	<input type="text"/>	Make	<input type="text"/>	
Address	<input type="text"/>	Model	<input type="text"/>	
Contact number	<input type="text"/>	VehicleID	<input type="text"/>	
Journey date	23-Mar-15	Registration number	<input type="text"/>	
Drop off date	23-Mar-15	Number of seats	<input type="text"/>	
Booking date	Label19	Rate per kilometer	<input type="text"/>	
		Advance paid	<input type="text"/>	
		Driver name	<input type="text"/>	
		Starting Kilometer	<input type="text"/>	
				Search
				Update
				Delete
				Refresh
				Cancel

Travel Agency Management System

```

Project1 - Microsoft Visual Basic [design] - [VehicleBookingUpdate (Code)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln1, Col1
(General) (Declarations)
'Declaring global variables.
Dim BID As String
Dim OldVID As String
'Make combobox click event.
Private Sub cboMake_Click()
'Clearing the fields that are dependant on Make of the vehicle to prevent errors while booking, if Make of the Vehicle is changed.
cboModel.Clear
cboVehicleID.Clear
lblRegNum.Caption = ""
cboModel.Text = ""
cboVehicleID.Text = ""
'Loading the Models available for the selected Make onto the list combobox from the database.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "' and Make ='" & cboMake.Text & "'", con,
'Ensuring that every available Model is loaded only once.
Do While rs.EOF = False
    Dim i As Integer
    Dim IsNewType As Boolean
    IsNewType = True
    For i = 0 To cboModel.ListCount - 1
        If cboModel.List(i) = rs.Fields("Model") Then
            IsNewType = False
            Exit For
        End If
    Next
    If IsNewType = True Then
        cboModel.AddItem (rs.Fields("Model"))
    End If
    rs.MoveNext
Loop
End Sub
'Model combobox click event.
Private Sub cboModel_Click()
'Clearing the fields that are dependant on the Model of the vehicle to prevent errors in booking information if Model chosen by the
cboVehicleID.Clear
lblRegNum.Caption = ""
cboVehicleID.Text = ""

```

'Declaring global variables.

Dim BID As String

Dim OldVID As String

'Make combobox click event.

Private Sub cboMake_Click()

'Clearing the fields that are dependant on Make of the vehicle to prevent errors while booking, if Make of the Vehicle is changed.

cboModel.Clear

cboVehicleID.Clear

lblRegNum.Caption = ""

cboModel.Text = ""

cboVehicleID.Text = ""

'Loading the Models available for the selected Make onto the list combobox from the database.

Set rs = New Recordset

rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "' and Make ='" & cboMake.Text & "'", con, adOpenDynamic, adLockOptimistic

'Ensuring that every available Model is loaded only once.

Do While rs.EOF = False

Dim i As Integer

Dim IsNewType As Boolean

IsNewType = True

For i = 0 To cboModel.ListCount - 1

If cboModel.List(i) = rs.Fields("Model") Then

IsNewType = False

Exit For

End If

```

Next
If IsNewType = True Then
    cboModel.AddItem (rs.Fields("Model"))
End If
rs.MoveNext
Loop
End Sub

'Model combobox click event.
Private Sub cboModel_Click()
'Clearing the fields that are dependant on the Model of the vehicle to prevent errors in booking
information if Model chosen by the user is changed.
    cboVehicleID.Clear
    lblRegNum.Caption = ""
    cboVehicleID.Text = ""
'Loading the Vehicle IDs for the Vehicle types of the selected Make and Model.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType & "' and Make = '" &
cboMake.Text & "' and Model = '" & cboModel.Text & "'", con, adOpenDynamic, adLockOptimistic
    rs.MoveFirst
    Do While Not rs.EOF
        cboVehicleID.AddItem (rs.Fields("VehicleID"))
        rs.MoveNext
    Loop
End Sub

'Vehicle ID combox click event.
Private Sub cboVehicleID_Click()
'Loading necessary booking information onto the form from the database for the selected Vehicle ID.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where VehicleID ='" & cboVehicleID.Text & "'", con,
adOpenDynamic, adLockOptimistic
    lblRatePerKm.Caption = rs.Fields("Rate_per_km")
    lblNSeats.Caption = rs.Fields("NSeats")
    lblRegNum.Caption = rs.Fields("RegistrationNo")
End Sub

'Vehicle type combobox click event.
Private Sub cboVehicleType_Click()
'Clearing the fields that are depemdamt on the Vehicle type to prevent errors in booking information if
the Vehicle type is changed.
    cboMake.Clear
    cboModel.Clear
    cboVehicleID.Clear
    lblRegNum.Caption = ""

    cboMake.Text = ""
    cboModel.Text = ""
    cboVehicleID.Text = ""
'The Makes that are available for the selected Vehicle type are loaded onto the Make combobox from

```

the database.

```

Set rs = New Recordset
rs.Open "select distinct * from Vehicle_entry where Vehicle_type ='" & cboVehicleType.Text & "'",
con, adOpenDynamic, adLockOptimistic
'Validation check to ensure that the vehicles that are booked are not loaded.
If rs.EOF = False Then
Do While rs.EOF = False
If rs.Fields("IsBooked?") = False Then
cboMake.AddItem (rs.Fields("Make"))
End If
rs.MoveNext
Loop
End If
'Message shown to notify the user that there are no Makes available for the chosen Vehicle type if
necessary.
If cboMake.ListCount = 0 Then
MsgBox "No cars of that type available"
cboVehicleType.Text = ""
Exit Sub
End If
End Sub

```

'Update button click event.

```

Private Sub cmbUpdate_Click()
'Validation check to make sure the User provides the Booking ID before update.
If txtBookingID.Text = Empty Then
MsgBox "Please enter the Booking ID."
Exit Sub
End If
'Presence check to make sure all the necessary information is provided.
If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = "" Or
cboVehicleType.Text = "" Or lblRegNum.Caption = "" Or cboMake.Text = "" Or cboModel.Text = "" Or
cboVehicleID.Text = "" Or lblRatePerKm.Caption = "" Or lblNSeats.Caption = "" Or txtAdvancePaid = "" Or
txtDriverName.Text = "" Then
MsgBox "None of the fields can be left empty."
Exit Sub
End If

```

'Length check to make sure contact number is of 11 digits.

```

If Len(txtContact.Text) <> 11 Then
MsgBox "Number of digits used for contact number is invalid."
Exit Sub
End If

```

'Checking if Booking ID is changed.

```

If txtBookingID.Text <> BID Then

```

'Message shown to notify the User that Booking ID cannot be changed.

```
MsgBox "Booking ID cannot be changed."
txtBookingID = BID
Exit Sub
End If
'Checking if Vehicle ID is changed, which means that the customer changed his choice of vehicle.
If cboVehicleID.Text <> OldVID Then
    Set rs = New Recordset
'Updating availability information if Vehicle ID is changed.
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
Do While Not rs.EOF
    If rs.Fields("VehicleID") = OldVID Then
        rs.Fields("IsBooked?") = False
    End If

    If rs.Fields("VehicleID") = cboVehicleID.Text Then
        rs.Fields("IsBooked?") = True
    End If

    rs.Update
    rs.MoveNext
Loop
rs.Close
End If

'Updating database information for the selected Booking ID
Set rs = New Recordset
rs.Open "select * from Vehicle_booking where VehicleBookingID = '" & txtBookingID.Text & "'", con,
adOpenDynamic, adLockOptimistic

rs.Fields("Customer_name") = txtName.Text
rs.Fields("Customer_address") = txtAddress.Text
rs.Fields("Contact") = txtContact.Text
rs.Fields("Journey_date") = dtpJourneyDate.Value
rs.Fields("Drop_off_date") = dtpDropOffDate.Value
rs.Fields("Vehicle_type") = cboVehicleType.Text
rs.Fields("Registration_number") = lblRegNum.Caption
rs.Fields("Make") = cboMake.Text
rs.Fields("Model") = cboModel.Text
rs.Fields("VehicleID") = cboVehicleID.Text
rs.Fields("Rate_per_kilometer") = lblRatePerKm.Caption
rs.Fields("Number_of_seats") = lblNSeats.Caption
rs.Fields("Advance_paid") = txtAdvancePaid.Text
rs.Fields("Driver_name") = txtDriverName.Text
rs.Fields("Date_of_booking") = lblDate.Caption
rs.Fields("Starting_Km") = txtStartingKm.Text
rs.Update
'Message shown to notify the User that the update was successful.
```

```

    MsgBox "Booking has been updated successfully."
'All the fields are cleared after the information is updated to ready form for next job.
    txtBookingID.Text = ""
    txtName.Text = ""
    txtContact.Text = ""
    txtAddress.Text = ""
    dtpJourneyDate.Value = Date
    dtpDropOffDate.Value = Date
    cboVehicleType.Text = ""
    lblRegNum.Caption = ""
    cboMake.Clear
    cboModel.Clear
    cboVehicleID.Clear
    lblRatePerKm.Caption = ""
    lblNSeats.Caption = ""
    txtAdvancePaid = ""
    txtStartingKm = ""
    txtDriverName.Text = ""

    Call Form_Load

End Sub
'Cancel button click event.
Private Sub cmdCancel_Click()
'Closing the form.
    Unload Me
End Sub
'Delete button click event.
Private Sub cmdDelete_Click()
'Presence check to make sure all the required information is provided.
    If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" And txtAddress.Text = "" Or
    cboVehicleType.Text = "" Or lblRegNum.Caption = "" Or cboMake.Text = "" Or cboModel.Text = "" Or
    cboVehicleID.Text = "" Or lblRatePerKm.Caption = "" Or lblNSeats.Caption = "" Or txtAdvancePaid = "" Or
    txtDriverName.Text = "" Then
        'Message shown to notify the User that all the information needs to be provided.
        MsgBox "None of the fields can be left empty."
        Exit Sub
    End If
'Deletion of the selected record from the database.
    rs.Delete
    MsgBox "Booking has been deleted."
    rs.Close
'Updating the availability information for the vehicle in the Vehicle Entry database.
    rs.Open " select * from Vehicle_entry where VehicleID = " & cboVehicleID.Text & " ", con,
    adOpenDynamic, adLockOptimistic
        rs.Fields("IsBooked?") = False
    rs.Update

```

```
'Clearing the unnecessary information on the form after deletion of the record.
```

```
txtBookingID.Text = ""  
txtName.Text = ""  
txtContact.Text = ""  
txtAddress.Text = ""  
dtpJourneyDate.Value = Date  
dtpDropOffDate.Value = Date  
cboVehicleType.Text = ""  
lblRegNum.Caption = ""  
cboMake.Clear  
cboModel.Clear  
cboVehicleID.Clear  
lblRatePerKm.Caption = ""  
lblINSeats.Caption = ""  
txtAdvancePaid = ""  
txtStartingKm = ""  
txtDriverName.Text = ""  
rs.Close
```

```
Call Form_Load
```

```
End Sub
```

```
'Refresh button click event.
```

```
Private Sub cmdRefresh_Click()
```

```
'Clearing all the fields on the form as required by the User in case the User wants to start anew.
```

```
txtBookingID.Text = ""  
txtName.Text = ""  
txtContact.Text = ""  
txtAddress.Text = ""  
dtpJourneyDate.Value = Date  
dtpDropOffDate.Value = Date  
cboVehicleType.Text = ""  
lblRegNum.Caption = ""  
cboMake.Clear  
cboModel.Clear  
cboVehicleID.Clear  
lblRatePerKm.Caption = ""  
lblINSeats.Caption = ""  
txtAdvancePaid = ""  
txtStartingKm.Text = ""  
txtDriverName.Text = ""
```

```
Call Form_Load
```

```
End Sub
```

```
'Search button click event.
```

```
Private Sub cmdSearch_Click()
```

```
'Presence check to ensure the User provides the Booking ID to be searched.  
If txtBookingID.Text = Empty Then  
'Message shown to ask for the Vehicle ID to be searched from the User.  
    MsgBox "Please enter the Booking ID."  
    Exit Sub  
End If  
  
'The Vehicle booking database is searched for a paticular Booking ID.  
Set rs = New Recordset  
rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic  
Dim found As Boolean  
found = False  
rs.MoveFirst  
Do While Not rs.EOF And Not found  
    If rs.Fields("VehicleBookingID") = txtBookingID.Text Then  
        found = True  
    End If  
    rs.MoveNext  
Loop  
  
If found = False Then  
'Message shown to notify User if Booking ID not found.  
    MsgBox "Booking ID being searched is not found. Please enter correct Booking ID and try again."  
    Exit Sub  
End If  
rs.Close  
'The Booking information for the particular Booking ID provided is loaded onto the form from the  
database.  
Set rs = New Recordset  
rs.Open "select * from Vehicle_booking where VehicleBookingID = '" & txtBookingID.Text & "'", con,  
adOpenDynamic, adLockOptimistic  
  
txtName.Text = rs.Fields("Customer_name")  
txtContact.Text = rs.Fields("Contact")  
txtAddress.Text = rs.Fields("Customer_address")  
dtpJourneyDate.Value = rs.Fields("Journey_date")  
dtpDropOffDate.Value = rs.Fields("Drop_off_date")  
lblBookingDate.Caption = rs.Fields("Date_of_booking")  
cboVehicleType.Text = rs.Fields("Vehicle_type")  
cboMake.Text = rs.Fields("Make")  
cboModel.Text = rs.Fields("Model")  
cboVehicleID.Text = rs.Fields("VehicleID")  
lblRegNum.Caption = rs.Fields("Registration_number")  
lblNSeats.Caption = rs.Fields("Number_of_seats")  
lblRatePerKm.Caption = rs.Fields("Rate_per_kilometer")  
txtAdvancePaid.Text = rs.Fields("Advance_paid")  
txtDriverName.Text = rs.Fields("Driver_name")
```

```
txtStartingKm.Text = rs.Fields("Starting_Km")

rs.Update

OldVID = cboVehicleID.Text
BID = txtBookingID.Text
End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
Call dblink
Showing the current date on the form.
lblDate.Caption = Format(Now, "dd/mm/yyyy")
dtpJourneyDate.Value = Date
dtpDropOffDate.Value = Date
'Loading Vehicle type information onto the Vehicle type combobox from the database.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Check made to make sure each type is loaded only once from the database.
If rs.EOF = False Then
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboVehicleType.ListCount - 1
            If cboVehicleType.List(i) = rs.Fields("Vehicle_type") Then
                IsNewType = False
            End If
        Next
        If IsNewType = True Then
            cboVehicleType.AddItem (rs.Fields("Vehicle_type"))
        End If
        rs.MoveNext
    Loop
End If
End Sub

'Advance paid keypress event.
Private Sub txtAdvancePaid_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub

Private Sub txtContact_KeyPress(KeyAscii As Integer)
```

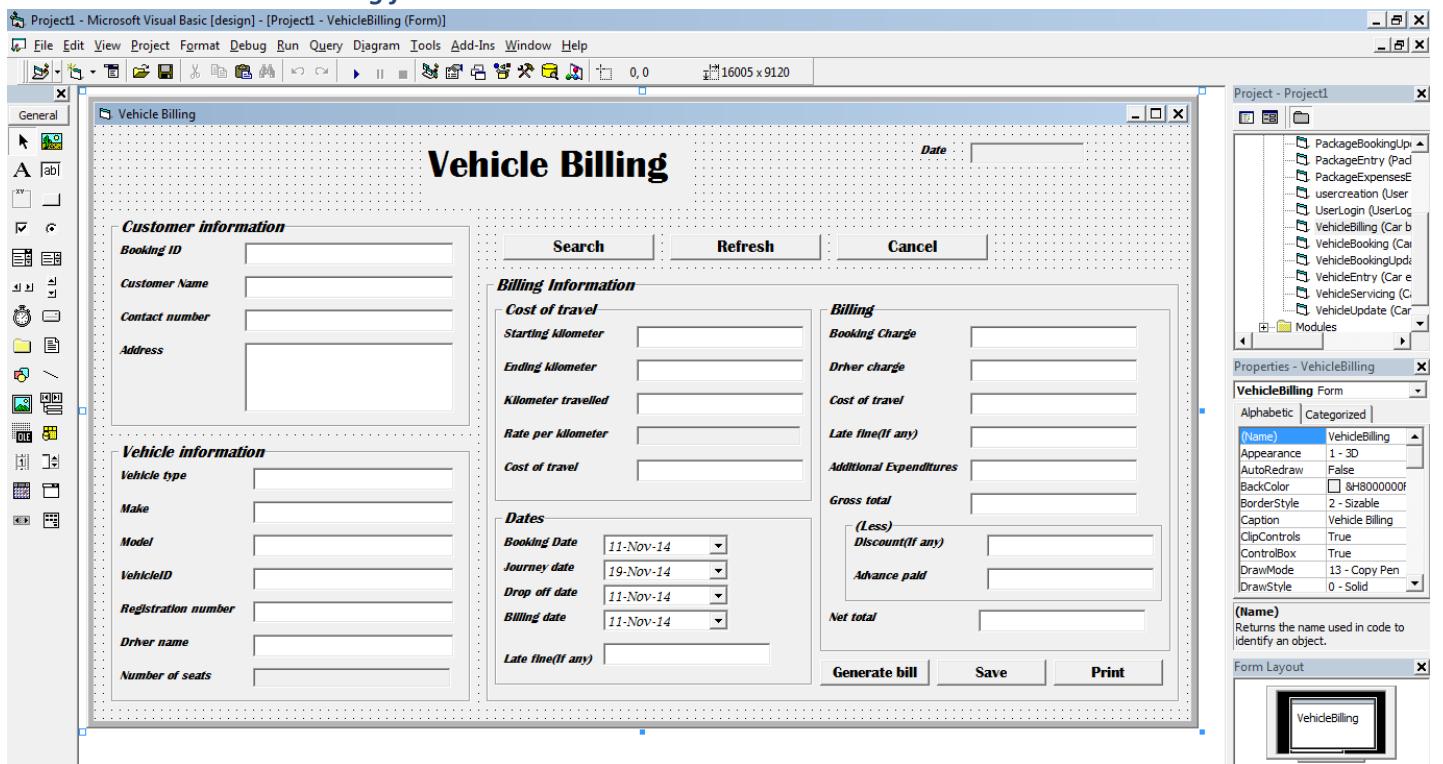
```
'Character type check to make sure only numerical data is entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
End Sub
```

```
Private Sub txtStartingKm_KeyPress(KeyAscii As Integer)
```

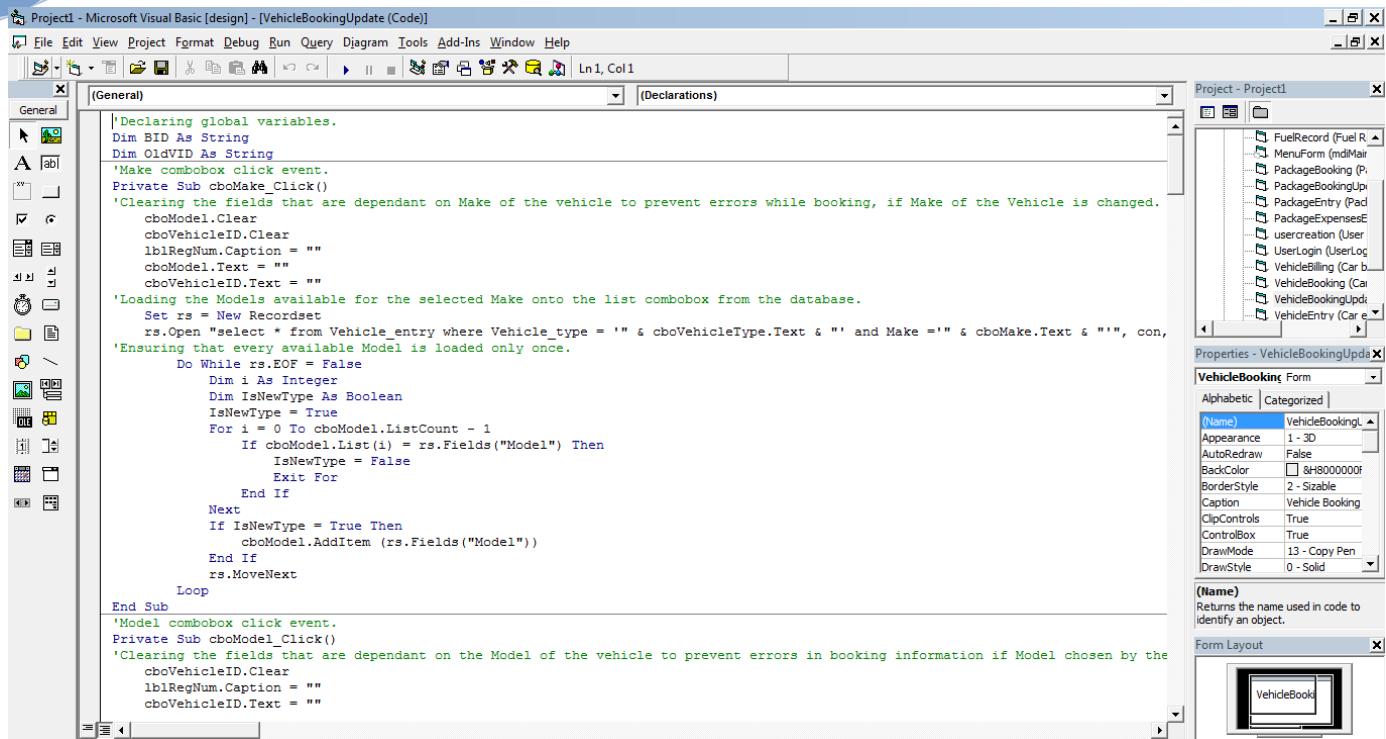
```
'Character type check to make sure only numerical data is entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
End Sub
```

13. Vehicle Billing form

The screenshot below is the actual layout of the Vehicle Billing Form.

The screenshot shows the actual 'Vehicle Billing' application window. The layout is identical to the design screen, featuring 'Customer information', 'Vehicle information', 'Billing Information', and 'Dates' sections with corresponding input fields and buttons. The window title is 'Vehicle Billing' and includes a 'Date' field set to '01-Apr-15'.



'Search button click event.

Private Sub cmbSearch_Click()

'Validation check to make sure Booking ID is provided.

If txtBookingID.Text = Empty Then

MsgBox "Please enter the Booking ID and try again."

Exit Sub

End If

'The Vehicle booking database is searched for a particular Booking ID.

Set rs = New Recordset

rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic

Dim found As Boolean

found = False

rs.MoveFirst

Do While Not rs.EOF And Not found

If rs.Fields("VehicleBookingID") = txtBookingID.Text Then

found = True

End If

rs.MoveNext

Loop

If found = False Then

'Message shown to notify User if Booking ID not found.

MsgBox "There is no record for this Booking ID."

Exit Sub

```

End If
rs.Close
'The Billing information is loaded from the Vehicle Booking database.
Set rs = New Recordset
rs.Open "select * from Vehicle_booking where VehicleBookingID=""" & txtBookingID.Text & """", con,
adOpenDynamic, adLockOptimistic
txtName.Text = rs.Fields("Customer_name")
txtContact.Text = rs.Fields("Contact")
txtAddress.Text = rs.Fields("Customer_address")
txtVehicleType.Text = rs.Fields("Vehicle_type")
txtMake.Text = rs.Fields("Make")
txtModel.Text = rs.Fields("Model")
txtRegNum.Text = rs.Fields("Registration_number")
txtVehicleID.Text = rs.Fields("VehicleID")
txtDriverName.Text = rs.Fields("Driver_name")
lblNSeats.Caption = rs.Fields("Number_of_seats")
txtStartingKm.Text = rs.Fields("Starting_km")
lblRatePerKm.Caption = rs.Fields("Rate_per_kilometer")
dtpBookingDate.Value = rs.Fields("Date_of_booking")
dtpJourneyDate.Value = rs.Fields("Journey_date")
dtpDropOffDate.Value = rs.Fields("Drop_off_date")
txtAdvancePaid.Text = rs.Fields("Advance_paid")

txtBookingID.Locked = True

'The difference between the dates is calculated.
Dim days As Integer
days = DateDiff("d", dtpDropOffDate.Value, dtpBillingDate.Value)
If dtpBillingDate.Value > dtpDropOffDate.Value Then
'Calculation of late fine.
    txtLateFine.Text = days * 50
Else
    txtLateFine.Text = 0
End If

txtLateFine2.Text = txtLateFine.Text
txtBookingCharge.Text = 2000
End Sub

'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
Unload Me
End Sub

'Generate Bill button click event.
Private Sub cmdGenerateBill_Click()
'Validation check to make sure none of the fields are left empty.
If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = Empty Or

```

```
txtVehicleType.Text = "" Or txtVehicleID.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or  
txtRegNum.Text = "" Or txtDriverName.Text = "" Or lblNSeats.Caption = "" Or txtStartingKm.Text = "" Or  
txtEndingKm.Text = "" Or txtKmTravelled.Text = "" Or txtCostOfTravel.Text = "" Or txtLateFine.Text = ""  
Or txtBookingCharge.Text = "" Or txtDriverCharge.Text = "" Or txtAdditionalExp.Text = "" Or  
txtDiscount.Text = "" Or txtAdvancePaid.Text = "" Then  
    MsgBox "Please make sure none of the fields are left empty and try again."  
    Exit Sub  
End If  
'Length check to make sure contact number is of 11 digits.  
If Len(txtContact.Text) <> 11 Then  
    MsgBox "Number of digits used for contact number is invalid."  
    Exit Sub  
End If  
  
If txtDiscount.Text = "" Then  
    txtDiscount.Text = 0  
End If  
  
txtGross.Text = Val(txtBookingCharge.Text) + Val(txtDriverCharge.Text) + Val(txtCostOfTravel2.Text) +  
Val(txtLateFine2.Text) + Val(txtAdditionalExp.Text)  
txtNet.Text = Val(txtGross.Text) - Val(txtDiscount.Text) - Val(txtAdvancePaid.Text)  
  
End Sub  
'Refresh button click event.  
Private Sub cmdRefresh_Click()  
'All the fields on the form are cleared on the click of the button as required by the User.  
txtBookingID.Text = ""  
txtName.Text = ""  
txtAddress.Text = ""  
txtContact.Text = ""  
txtVehicleType.Text = ""  
txtMake.Text = ""  
txtModel.Text = ""  
txtVehicleID = ""  
txtRegNum.Text = ""  
txtDriverName.Text = ""  
lblNSeats.Caption = ""  
txtStartingKm.Text = ""  
txtEndingKm.Text = ""  
txtKmTravelled.Text = ""  
lblRatePerKm.Caption = ""  
txtCostOfTravel.Text = ""  
dtpBookingDate.Value = Date  
dtpJourneyDate.Value = Date  
dtpDropOffDate.Value = Date  
dtpBillingDate.Value = Date  
txtLateFine.Text = ""
```

```

txtBookingCharge.Text = ""
txtDriverCharge.Text = ""
txtCostOfTravel2.Text = ""
txtLateFine2.Text = ""
txtAdditionalExp.Text = ""
txtGross.Text = ""
txtDiscount.Text = ""
txtAdvancePaid.Text = ""
txtNet.Text = ""

txtBookingID.Locked = False
End Sub

'Save button click event.
Private Sub cmdSave_Click()
'Validation check to make sure none of the fields are left empty.
If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = Empty Or
txtVehicleType.Text = "" Or txtVehicleID.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or
txtRegNum.Text = "" Or txtDriverName.Text = "" Or lblSeats.Caption = "" Or txtStartingKm.Text = "" Or
txtEndingKm.Text = "" Or txtKmTravelled.Text = "" Or txtCostOfTravel.Text = "" Or txtLateFine.Text = ""
Or txtBookingCharge.Text = "" Or txtDriverCharge.Text = "" Or txtAdditionalExp.Text = "" Or
txtGross.Text = "" Or txtDiscount.Text = "" Or txtAdvancePaid.Text = "" Or txtNet.Text = "" Then
'Message shown to notify the User that none of the fields can be left empty.
    MsgBox "Please make sure none of the fields are left empty and try again."
    Exit Sub
End If

'Length check to make sure contact number is of 11 digits.
If Len(txtContact.Text) <> 11 Then
    MsgBox "Number of digits used for contact number is invalid."
    Exit Sub
End If

'Storage of Billing information in the Vehicle Billing database.
Set rs = New Recordset
rs.Open "select * from Vehicle_billing", con, adOpenDynamic, adLockOptimistic
rs.AddNew
    rs.Fields("BookingID") = txtBookingID.Text
    rs.Fields("Customer_name") = txtName.Text
    rs.Fields("Customer_address") = txtAddress.Text
    rs.Fields("VehicleID") = txtVehicleID.Text
    rs.Fields("Kilometer_travelled") = txtKmTravelled.Text
    rs.Fields("Rate_per_kilometer") = lblRatePerKm.Caption
    rs.Fields("Cost_of_travel") = txtCostOfTravel.Text
    rs.Fields("Journey_date") = dtpJourneyDate.Value
    rs.Fields("Drop_off_date") = dtpDropOffDate.Value
    rs.Fields("Billing_date") = dtpBillingDate.Value
    rs.Fields("Booking_charge") = txtBookingCharge.Text

```

```

rs.Fields("Driver_charge") = txtDriverCharge.Text
rs.Fields("Cost_of_travel") = txtCostOfTravel.Text
rs.Fields("Late_fine") = txtLateFine.Text
rs.Fields("Additional_expenditures") = txtAdditionalExp.Text
rs.Fields("Gross_total") = txtGross.Text
rs.Fields("Discount") = txtDiscount.Text
rs.Fields("Advance_paid") = txtAdvancePaid.Text
rs.Fields("Net_total") = txtNet.Text
rs.Update

```

MsgBox "The Billing information is saved."

End Sub

'Form load click event.

Private Sub Form_Load()

'Calling the function to connect to database.

Call dblink

lblDate.Caption = Date

dtpBillingDate.Value = Date

End Sub

Private Sub txtAdditionalExp_KeyPress(KeyAscii As Integer)

'Character type check to make sure non-numerical data is not allowed to be entered.

If KeyAscii = 8 Then Exit Sub

If Not IsNumeric(Chr(KeyAscii)) Then

MsgBox "Please input numbers only."

Exit Sub

End If

End Sub

Private Sub txtBookingCharge_KeyPress(KeyAscii As Integer)

'Character type check to make sure non-numerical data is not allowed to be entered.

If KeyAscii = 8 Then Exit Sub

If Not IsNumeric(Chr(KeyAscii)) Then

MsgBox "Please input numbers only."

Exit Sub

End If

End Sub

Private Sub txtAdvancePaid_KeyPress(KeyAscii As Integer)

'Character type check to make sure non-numerical data is not allowed to be entered.

If KeyAscii = 8 Then Exit Sub

If Not IsNumeric(Chr(KeyAscii)) Then

MsgBox "Please input numbers only."

Exit Sub

```
End If  
End Sub
```

```
Private Sub txtContact_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtCostOfTravel_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtCostOfTravel2_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtDiscount_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtDriverCharge_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub

Private Sub txtEndingKm_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub

Private Sub txtGross_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub

Private Sub txtKmTravelled_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub

'Kilometers travelled textbox lostfocus event.
Private Sub txtKmTravelled_LostFocus()
'Calculation of total amount charged to the customer for the total distance travelled.
txtKmTravelled.Text = Val(txtEndingKm.Text) - Val(txtStartingKm.Text)
txtCostOfTravel.Text = Val(txtKmTravelled.Text) * Val(lblRatePerKm.Caption)
txtCostOfTravel2.Text = txtCostOfTravel.Text
End Sub

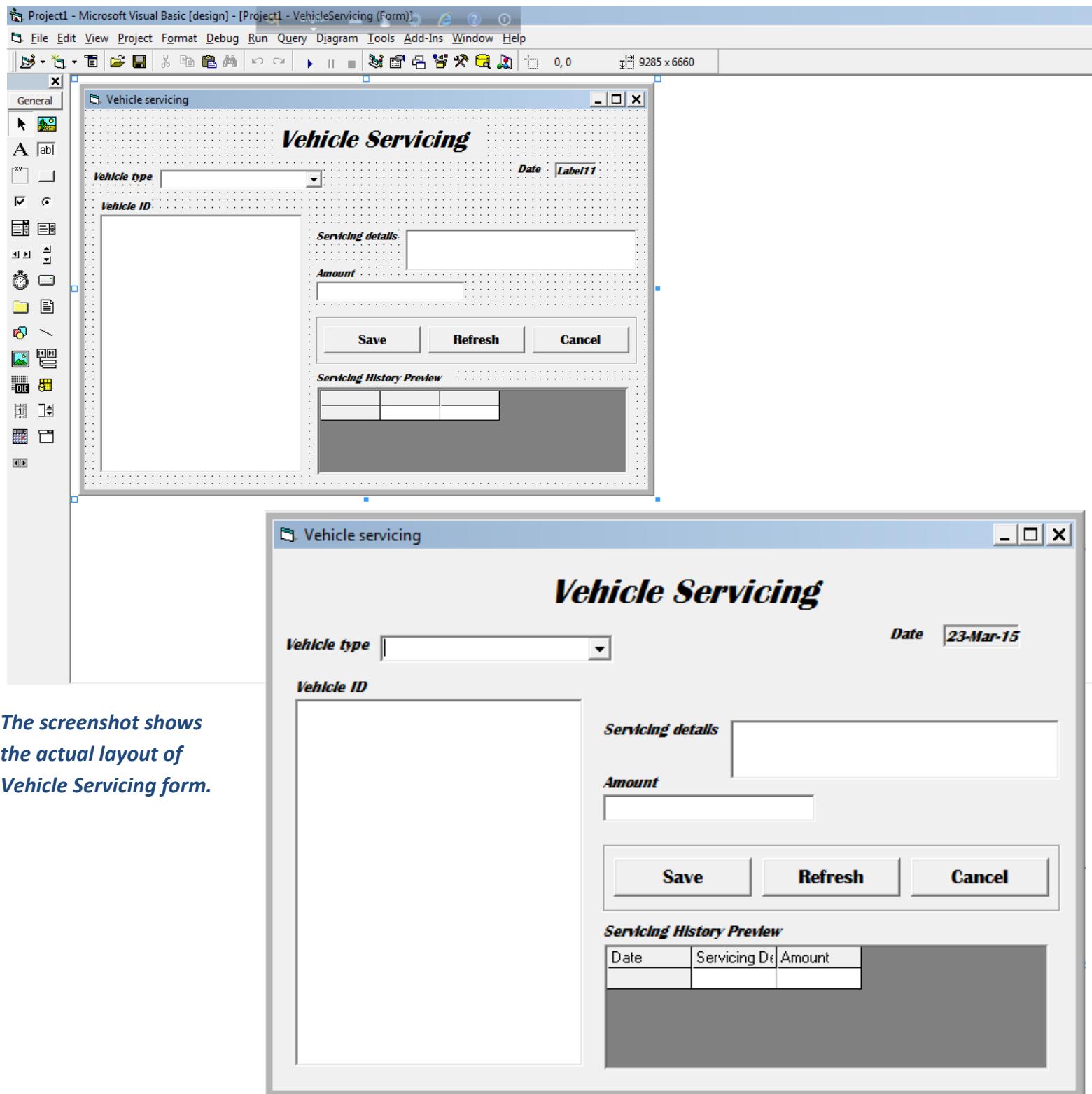
Private Sub txtLateFine_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

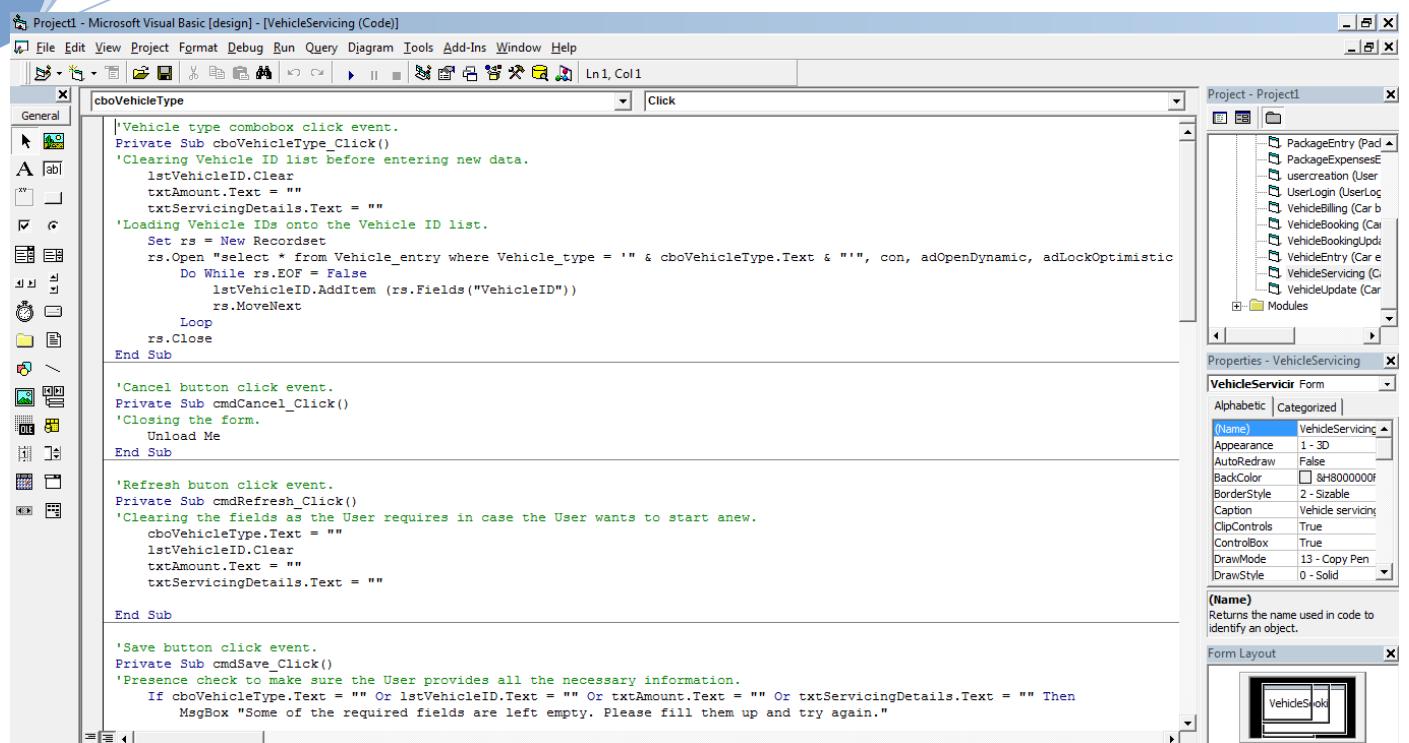
```
Private Sub txtLateFine2_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtNet_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
End Sub
```

```
Private Sub txtStartingKm_KeyPress(KeyAscii As Integer)  
'Character type check to make sure non-numerical data is not allowed to be entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

14. Vehicle Servicing

Travel Agency Management System



```

'Vehicle type combobox click event.
Private Sub cboVehicleType_Click()
'Clearing Vehicle ID list before entering new data.
    lstVehicleID.Clear
    txtAmount.Text = ""
    txtServicingDetails.Text = ""
'Loading Vehicle IDs onto the Vehicle ID list.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "'", con,
adOpenDynamic, adLockOptimistic
    Do While rs.EOF = False
        lstVehicleID.AddItem (rs.Fields("VehicleID"))
        rs.MoveNext
    Loop
    rs.Close
End Sub

```

```

'Cancel button click event.
Private Sub cmdCancel_Click()
'Closing the form.
    Unload Me
End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
'Clearing the fields as the User requires in case the User wants to start anew.
    cboVehicleType.Text = ""
    lstVehicleID.Clear

```

```
txtAmount.Text = ""  
txtServicingDetails.Text = ""  
  
End Sub  
  
'Save button click event.  
Private Sub cmdSave_Click()  
'Presence check to make sure the User provides all the necessary information.  
If cboVehicleType.Text = "" Or lstVehicleID.Text = "" Or txtAmount.Text = "" Or txtServicingDetails.Text = "" Then  
    MsgBox "Some of the required fields are left empty. Please fill them up and try again."  
    Exit Sub  
End If  
  
'Storing information into the database.  
Set rs = New Recordset  
rs.Open "select * from Vehicle_servicing", con, adOpenDynamic, adLockOptimistic  
rs.AddNew  
    rs.Fields("Vehicle_type") = cboVehicleType.Text  
    rs.Fields("VehicleID") = lstVehicleID.Text  
    rs.Fields("Servicing_details") = txtServicingDetails.Text  
    rs.Fields("Amount") = txtAmount.Text  
    rs.Fields("Date") = Date  
rs.Update  
  
'Clearing the fields to input new data.  
txtAmount.Text = ""  
txtServicingDetails.Text = ""  
flxgrdHistory.Rows = 2  
Call lstVehicleID_Click  
End Sub  
  
'Form load event.  
Private Sub Form_Load()  
'Calling funtion to connect to database.  
    Call dblink  
  
'Showing current date on the form.  
lblDate.Caption = Date  
  
'Loading the Vehicle type information onto the Vehicle type combobox from Vehicle entry database.  
Set rs = New Recordset  
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic  
  
'Validity check to make sure each type is loaded only once.  
If rs.EOF = False Then  
    Do While rs.EOF = False  
        Dim i As Integer  
        Dim IsNewType As Boolean  
        IsNewType = True  
        For i = 0 To cboVehicleType.ListCount - 1  
            If cboVehicleType.List(i) = rs.Fields("Vehicle_type") Then  
                IsNewType = False  
            End If  
    End If
```

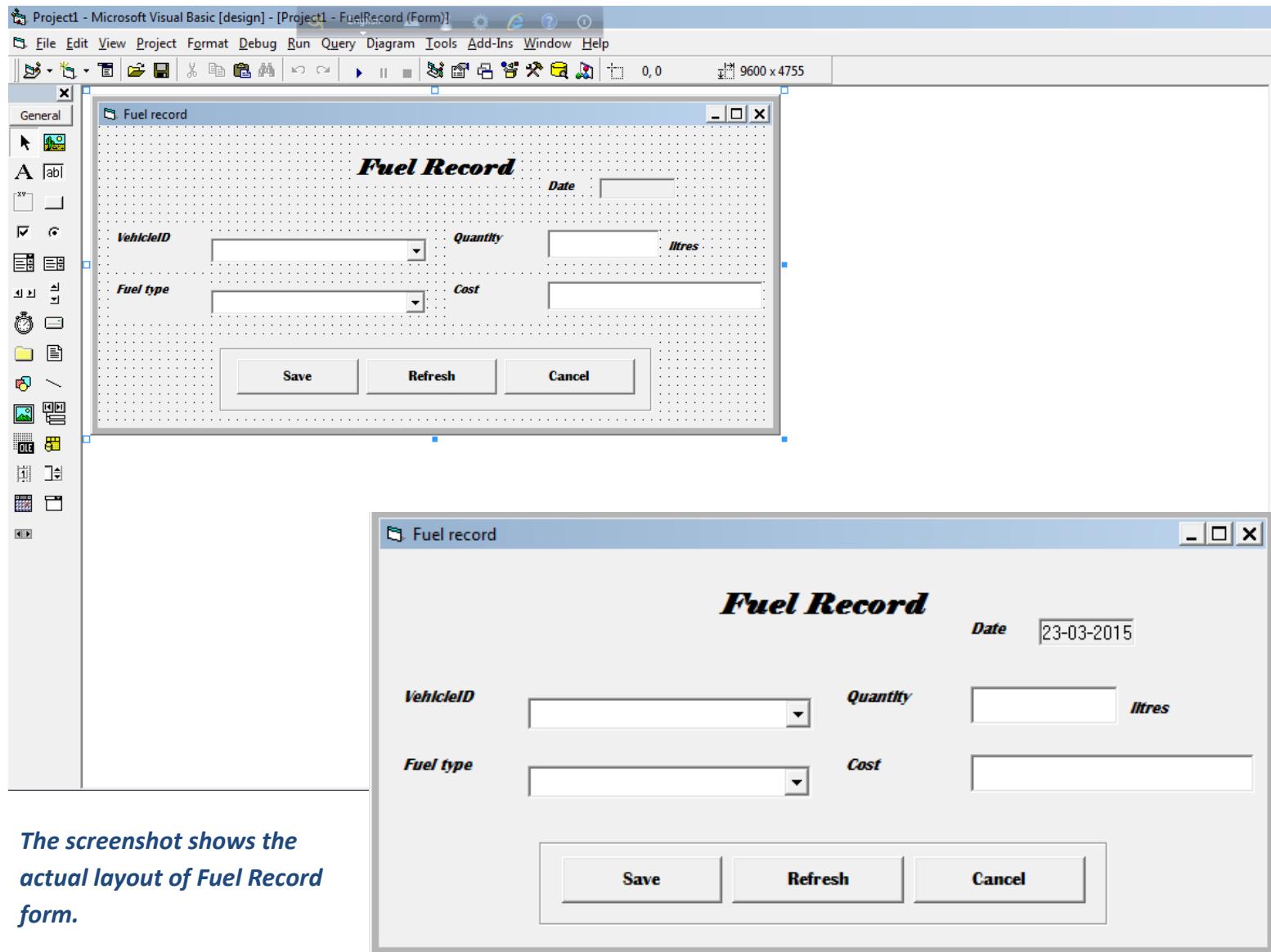
```
Next
If IsNewType = True Then
    cboVehicleType.AddItem (rs.Fields("Vehicle_type"))
End If
rs.MoveNext
Loop
End If
flxgrdHistory.TextMatrix(0, 0) = "Date"
flxgrdHistory.TextMatrix(0, 1) = "Servicing Details"
flxgrdHistory.TextMatrix(0, 2) = "Amount"
flxgrdHistory.ColWidth(1) = 3000
flxgrdHistory.RowHeight(1) = 500
End Sub

'Vehicle ID list click event.
Private Sub lstVehicleID_Click()
    Dim r As Integer
    r = 2
    txtAmount.Text = ""
    txtServicingDetails.Text = ""
    Set rs = New Recordset
    rs.Open "select * from Vehicle_servicing where VehicleID ='" & lstVehicleID.Text & "'", con,
adOpenDynamic, adLockOptimistic
    While Not rs.EOF
        r = r + 1

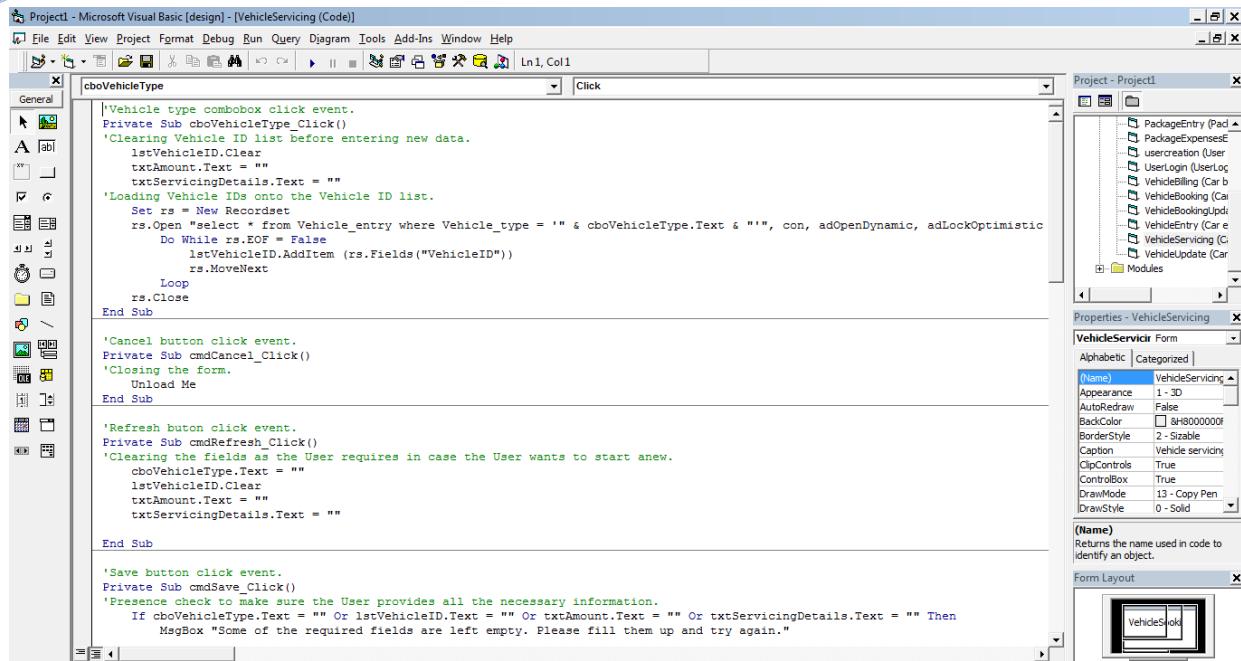
        flxgrdHistory.TextMatrix(r, 0) = rs!Date
        flxgrdHistory.TextMatrix(r, 1) = rs!Servicing_details
        flxgrdHistory.TextMatrix(r, 2) = rs!Amount

        rs.MoveNext
        flxgrdHistory.Rows = flxgrdHistory.Rows + 1
    Wend
End Sub

'Amount keypress event.
Private Sub txtAmount_KeyPress(KeyAscii As Integer)
    'Character type check to make sure only numerical data is input.
    If KeyAscii = 8 Then Exit Sub
    If Not IsNumeric(Chr(KeyAscii)) Then
        MsgBox "Please input numbers only."
        Exit Sub
    End If
End Sub
```

15. Fuel Record

Travel Agency Management System



```
'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
Unload Me
End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
'All the fields on the form are cleared.
cboVehicleID.Text = ""
cboFuelType.Text = ""
txtQuantity.Text = ""
txtCost.Text = ""

End Sub

'Save button click event.
Private Sub cmdSave_Click()
'Validation check to make sure that none of the important fields are kept empty.
If cboVehicleID.Text = Empty Or cboFuelType.Text = Empty Or txtQuantity.Text = Empty Or
txtCost.Text = Empty Then
    MsgBox "None of the fields can be left empty."
    Exit Sub
End If

'Storing information into the database.
Set rs = New Recordset
rs.Open "select * from Fuel_record", con, adOpenDynamic, adLockOptimistic
rs.AddNew
```

```

rs.Fields("VehicleID") = cboVehicleID.Text
rs.Fields("Date_of_entry") = lblDate.Caption
rs.Fields("FuelType") = cboFuelType.Text
rs.Fields("FuelQuantity/litres") = txtQuantity.Text
rs.Fields("FuelCost") = txtCost.Text
rs.Update
MsgBox "Record has been added successfully"
'The fields are cleared after the information are stored into the database.
cboVehicleID.Text = ""
cboFuelType.Text = ""
txtQuantity.Text = ""
txtCost.Text = ""

End Sub

```

```

'Form load click event.
Private Sub Form_Load()
'Calling function to connect to the database.
Call dblink
Shows current date.
lblDate.Caption = Format(Now, "dd/mm/yyyy")
'The vehicle types available on the Vehicle Entry database are loaded.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
Do While Not rs.EOF
    cboVehicleID.AddItem (rs.Fields("VehicleID"))
    rs.MoveNext
Loop
End Sub

```

```

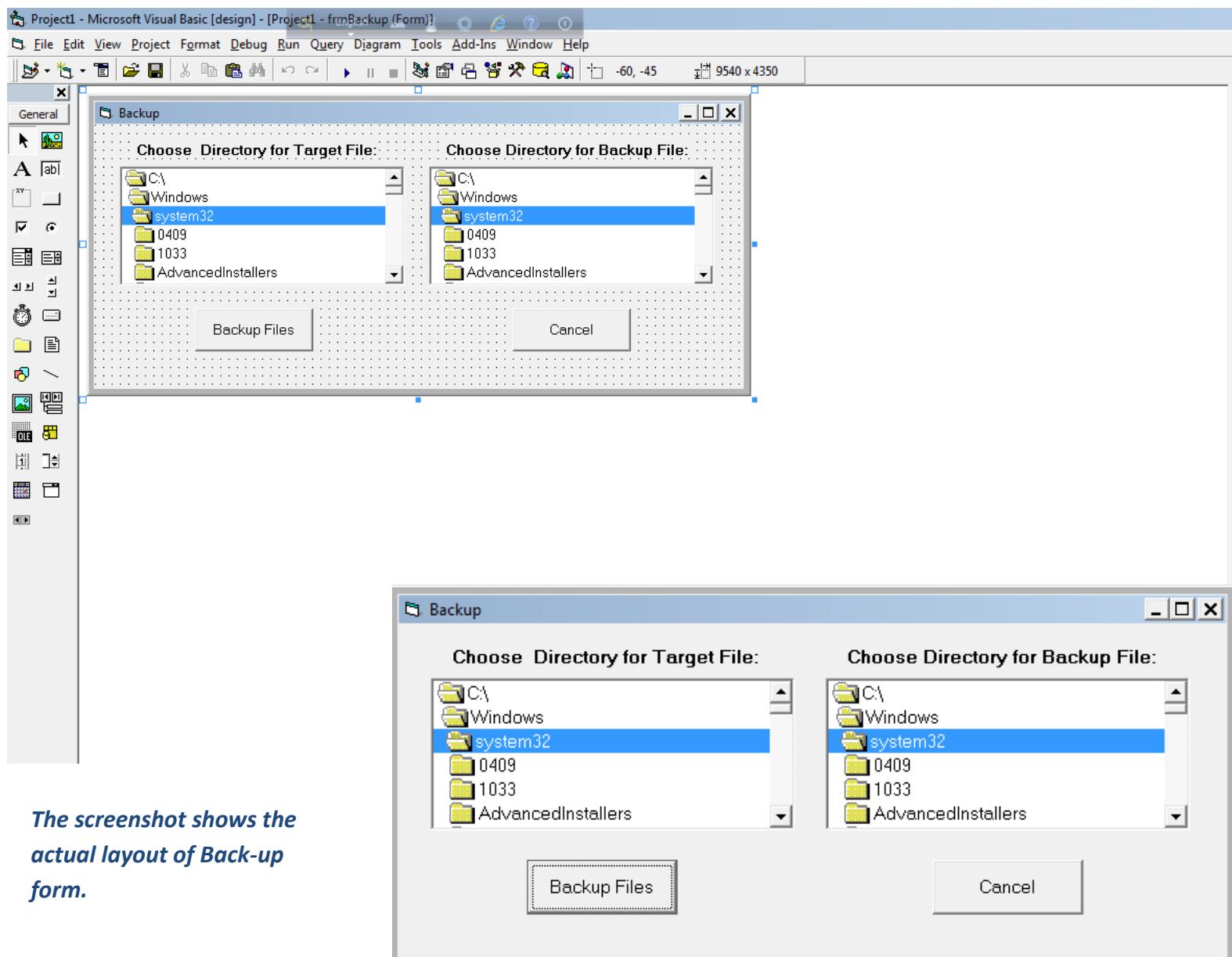
'Cost of fuel keypress event.
Private Sub txtCost_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub

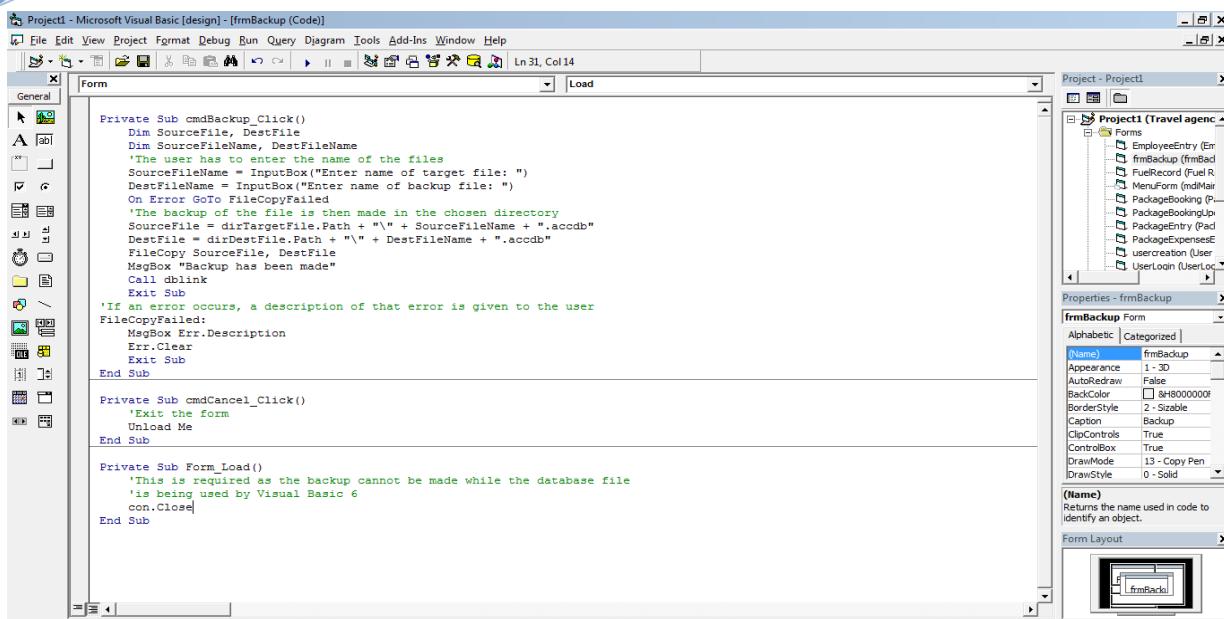
```

```

'Quantity of fuel keypress event.
Private Sub txtQuantity_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub

```

16. Backup form



```

Private Sub cmdBackup_Click()
    Dim SourceFile, DestFile
    Dim SourceFileName, DestFileName
    'The user has to enter the name of the files
    SourceFileName = InputBox("Enter name of target file: ")
    DestFileName = InputBox("Enter name of backup file: ")
    On Error GoTo FileCopyFailed
    'The backup of the file is then made in the chosen directory
    SourceFile = dirTargetFile.Path + "\" + SourceFileName + ".accdb"
    DestFile = dirDestFile.Path + "\" + DestFileName + ".accdb"
    FileCopy SourceFile, DestFile
    MsgBox "Backup has been made"
    Call dblink
    Exit Sub
    'If an error occurs, a description of that error is given to the user
    FileCopyFailed:
    MsgBox Err.Description
    Err.Clear
    Exit Sub
End Sub

Private Sub cmdCancel_Click()
    'Exit the form
    Unload Me
End Sub

Private Sub Form_Load()
    'This is required as the backup cannot be made while the database file
    'is being used by Visual Basic 6
    con.Close
End Sub

```

❖ Report Designs

Crystal report software design wizard is used to design all the reports

1. Employee details

The screenshot shows the Crystal Reports software interface for designing a report titled "Employee Details". The report structure is as follows:

- Report Header:** Contains the "MAAS Travels and Tours" logo.
- Page Header:** Contains the title "Employee Details".
- Details:** A table with columns: Employee ID, Employee Name, Address, Contact, Designation, Basic salary, and Date of entry.
- Report Footer:** Contains a "Page Number" field.
- Page Footer:** Contains a "Page Number" field.

2. Fuel Record

The screenshot shows the Crystal Reports software interface for designing a report titled "Report Title". The report structure is as follows:

- Page Header:** Contains the "MAAS Travels and Tours" logo.
- Details:** A table with columns: Vehicle ID, Fuel Type, Quantity/litres, Cost, and Date of entry.
- Report Footer:** Contains a "Page Number" field.
- Page Footer:** Contains a "Page Number" field.

3. List of Packages

The screenshot shows the Crystal Reports interface with the title bar "Crystal Reports - [List of Packages]". The menu bar includes File, Edit, View, Insert, Format, Database, Report, Analyzer, Window, and Help. The toolbar contains various icons for report creation and modification. The main window is divided into sections: Report Header, Page Header, Details, and Report Footer. The Report Header section contains a logo for "MAAS Travels and Tours" and a title "Report Title". The Details section displays a table with columns: Package type, Package name, Package ID, Package details, Accommodation type, Cost per person, and Journey date. The Report Footer section contains a page number placeholder "[Page Number]". The status bar at the bottom indicates "For Help, press F1" and "Records: 4".

4. List of Vehicles

The screenshot shows the Crystal Reports interface with a report titled "List of Vehicles". The report structure is as follows:

- Report Header:** Contains the "MAAS Travels and Tours" logo.
- Page Header:** Contains the placeholder text "Report Title".
- Table Section:** A table with the following columns:
 - Vehicle type
 - Vehicle ID
 - Registration number
 - Make
 - Model
 - Number of Seats
 - Rate per kilometer
 - Date of entry
- Details Section:** Displays the same table structure with specific field names:
 - Vehicle_type
 - VehicleID
 - RegistrationNo
 - Make
 - Model
 - NSeats
 - Rate_per_km
 - Date_of_entry
- Report Footer:** Contains a field labeled "Page Number".
- Page Footer:** Contains the placeholder text "Page Number".

Travel Agency Management System

5. Customer list of Package Booking

Customer List of Package Booking

Package ID	Booking ID	Customer name	Address	Contact number	Number of people	Net total	Date of booking
[PackageBookingID]	[Customer_name]	[Customer_address]	[Contact]	[Number_of_people]	[Net_total]	[Date_of_booking]	

6. Customer list of Vehicle Booking

Customer List of Vehicle Booking

Vehicle ID	Booking ID	Customer name	Address	Contact	Vehicle type	Advance paid	Date of booking	Journey date	Drop off date	Driver name
[VehicleBookingID]	[Customer_name]	[Customer_address]	[Contact]	[Vehicle_type]	[Advance_paid]	[Date_of_booking]	[Journey_date]	[Drop_off_date]	[Driver_name]	

7. List of Takings from Vehicle Rentals

Crystal Reports - [ListofTakingsFromVehicleRentals]

File Edit View Insert Format Database Report Analyzer Window Help

Design | Preview |

Report Header:

- Print Date
- 

Page Header:

Report Footer:

Page Number

Details:

Booking ID	Customer name	Vehicle ID	Net total
BookingID	Customer_name	VehicleID	Net_total

Report Footer:

Page Number

For Help, press F1

Records: 0

8. Package Expenses list

Crystal Reports - [Package Expenses List]

File Edit View Insert Format Database Report Analyzer Window Help

Design | Preview |

Report Header:

- 

Page Header:

Report Footer:

Page Number

Details:

PackageID	Expenses	Amount
PackageID	Expenses	Amount

Report Footer:

Page Number

Page Number

For Help, press F1

Records: 10

Jan. 1

Travel Agency Management System

9. List of Total Package Expenses

Crystal Reports - [ListOfTotalPackageExpenses]

File Edit View Insert Format Database Report Analyzer Window Help

04-Apr-15 2:46 1 of 1

Design Preview

PH

06-Apr-15

MAAS Travels and Tours

List of Total Package Expenses

Package ID	Package name	Total
1121	Summer Isles	\$22,650.00
1122	Wind of Brisbane	\$53,000.00

For Help, press F1

Records: 2 100%

10. Vehicle Servicing details

Crystal Reports - [Vehicle Servicing Details List]

File Edit View Insert Format Database Report Analyzer Window Help

04-Apr-15 2:46 1 of 1

Design Preview

Report Header

Page Header

MAAS Travels and Tours

Report Title

Vehicle type	Vehicle ID	Servicing details	Amount	Date of entry
Vehicle type	VehicleID	Servicing details	Amount	Date

Details

Report Footer

Page Footer

Page Number

For Help, press F1

Records: 4

Travel Agency Management System

11. Package Booking Receipt

Crystal Reports - [Package Booking Receipt]

File Edit View Insert Format Database Report Analyzer Window Help

Design | Preview |

Report Header

Page Header

Report Title



[Date] Print Date

Booking ID : PackageBookingID

Date of booking : Date_of_bg

Customer name : Customer_name	Customer name : Customer_name
Package ID : PackageID	Package ID : PackageID
Cost per person : per_person	Cost per person : per_person
Number of people : Number_of_ppl	Number of people : Number_of_ppl
Net total : Net_total	Net total : Net_total

Details

Report Footer

Page Footer

For Help, press F1

Page Number

Records: 3

12. Vehicle Rental Token

Crystal Reports - [Token]

File Edit View Insert Format Database Report Analyzer Window Help

Design | Preview |

Page Header

Vehicle Rental Token



Date of booking : Date_of_booking

BookingID : VehicleBookingID Print date : Print Date

Customer name : Customer_name	Customer name : Customer_name
Vehicle ID : VehicleID	Vehicle ID : VehicleID
Rate per kilometer : _kilometers	Rate per kilometer : _kilometers
Starting_km : Starting_km	Starting_km : Starting_km
Journey date : Journey	Journey date : Journey
Drop off date : Drop_of	Drop off date : Drop_of
Driver name : Driver_name	Driver name : Driver_name

Advance paid : Advance_paid

Details

Page Footer

For Help, press F1

Page Number

Records: 1

13. Vehicle Rental Bill

Crystal Reports - [VehicleRentalBill]

File Edit View Insert Format Database Report Analyzer Window Help

Design Preview |

Report Header

Page Header

Details

Print Footer

For Help, press F1

Records: 1

The report structure is as follows:

- Report Header:** Contains the MAAS Travels and Tours logo.
- Page Header:** Contains the title "Vehicle Rental Bill" and the date "Billing_date".
- Details:** Contains several data items grouped into two columns:
 - Left Column:** BookingID, Customer_name, Journey_date, Drop_off_date, Kilometers travelled, Rate per kilometer.
 - Right Column:** Booking_charge, Driver_charge, Cost of travel, Late_fine, Additional_expenditure, Gross_total, Less, Discount, Advance_paid, Net_total.

(iii) Testing

❖ Testing Strategy

Any system needs to be extensively tested to ensure that it works. This is a fairly obvious statement but in reality, comprehensive testing is impossible in all but the simplest of systems. It is simply not possible to test the results of every conceivable input or follow the path of every logical construction in the system. There are no developed programs are free from errors. As is a well known saying "Perfection is always a work in progress". A program, therefore, should be thoroughly tested before it is available for use. There are no possible ways to ensure complete error free program, since there are almost an infinite number of wrong combinations of events possible. This difficulty means that testing must be carefully planned. I have decided to use a test plan which covers maximum aspects (most features and screens) of the program. The different parts of the program will be exposed to abnormal, normal and even extreme data(where applicable). The expected values will then be compared with actual result to determine whether errors are present in a particular area. I have added some data to the database which will help me to carry out the testing thoroughly.

Test no.	Test details	Test Method and data to be entered in the respective boxes	Expected outcome	Objective fulfilled	Actual outcome	Evidence on page no.
1.	Testing whether the Main Menu form is loaded when correct User type, User ID and password is given in the 'Login' form.	User type: Administrator User ID: 1234 Password: 123 Click 'Login'	Confirmation message saying Login is successful. Main Menu form displayed.	1	Successful	
2.	Testing whether an error is generated when correct User ID and password but wrong user type given.	User type: General User ID: 1234 Password: 123 Click 'Login'	Error message displayed saying wrong User type or User ID provided.	1	Successful	
3.	Testing if error message is provided if correct User type and Password is provided but wrong User ID is provided.	User type: Administrator User ID: 123 Password: 123 Click 'Login'	Error message shown notifying the User that there is no such User ID in the database for this User type.	1	Successful	
4.	Testing if all the other fields are cleared when the User type is changed.	User type: Administrator User ID:1234 Password: 12	All the boxes are emptied.	16	Successful	

5.	Testing if error message is displayed if correct User type and password are given but wrong password is given.	User type: Administrator User ID: 123456 Password : 456 Click 'Login'	Error message shown to User notifying that the password provided is invalid and asks for correct password.	1	Successful	
6.	Testing if error is given if any of the User type box are left empty.	User type: , User ID: 1234, Password: 123	Error message shown notifying the User that none of the fields can be left empty.	23	Successful	
7.	Testing if error message is shown if User ID box is left empty.	User type: Administrator , User ID: , Password: 123	Error message shown notifying the User that none of the fields can be left empty.	23	Successful	
8.	Testing if error message is shown if Password is left empty	User type: Administrator , User ID:1234 , Password:	Error message shown notifying the User that none of the fields can be left empty.	23	Successful	
9.	Testing if User Creation form is displayed when selected from the main menu.	User type: Administrator , User ID:1234 , Password: 123 Select "User Creation" form from Main Menu.	User Creation form is displayed.		Successful	
10.	Testing if error message is shown if User type, User ID, Password or Confirm Password fields is left empty on pressing "Create Account" button.	User type: Staff , User ID: , Password:1321, Confirm Password:	Error message displayed notifying the User that none of the fields can be left empty.	23,26	Successful	
11.	Testing to check if error message shown if the User ID provided already exists in the User Login database table on pressing "Create account" button.	User type: Administrator , User ID:1234 , Password: 1321, Confirm password: 1321	Error message shown notifying the User that the User ID provided is already in use.	13	Successful	

12.	Testing to check if error message shown if Password does not match Confirm Password on pressing “Create account” button.	User type: Administrator , User ID:1111 , Password:1321, Confirm password: 1123	Error message shown to notify the User that the Password and Confirm Password do not match.	3	Successful	
13.	Testing to check if confirmation message is shown after the new record is entered, all the fields are cleared after, the new record is added to the database table and User login database table is updated.	User type: Administrator , User ID:1111 , Password:1321, Confirm password: 1321	Confirmation message shown notifying the User that the record is successfully added to the database.	4	Successful	
14.	Testing to check if error message shown if Search button is pressed without filling up User type and User ID.	User type: , User ID:	Error message asking for User type and User ID.	23		
15.	Testing to check if error message is shown if User ID for the particular User type is not found in the database.	User type: Administrator , User ID: 1321	Error message shown saying there is no User ID for the particular User type in the database.		Successful	
16.	Test to see if the required information appears if all the data is provided correctly on pressing search button.	User type: Administrator , User ID: 1111	The rest of the fields in the form are filled.	15,	Successful	
17.	Testing to check if error message shown if User type, User ID, password or confirm password not provided during update.	User type: Administrator , User ID: 1111, Password: , Confirm Password: 111	Error message shown saying none of the fields can be left empty.	23	Successful	
18.	Testing to check if error message is shown if password and confirm password don't match on pressing update button.	User type: Administrator , User ID: 1111, Password: 112 , Confirm Password: 111	Error message shown saying password and confirm password do not match.	3	Successful	

19.	Testing to check if error message is shown if User type is changed and update button pressed.	User type: Staff , User ID: 1111, Password: 111 , Confirm Password: 111	Error message shown saying User type cannot be changed.	3	Successful	
20.	Testing to check if error message shown if User ID is changed and the new User ID is already present in the database when update button is pressed.	User type: Administrator , User ID: 1234, Password: 111 , Confirm Password: 111	Error message shown saying User ID is already in use.	0	Successful	
21.	Testing to check if confirmation is shown when a record is updated, all the fields are emptied after and the information on the database record is changed in the database.	User type: Administrator , User ID: 1111, Password: 111 , Confirm Password: 111	Confirmation message shown saying the record has been updated.	4	Successful	
22.	Testing to check if all the fields are emptied on pressing the Refresh button.	User type: Administrator , User ID: 1111, Password: 111 , Confirm Password:	All the fields are emptied.	23,26	Successful	
23.	Testing to check if error message is shown if User type, User ID, password or confirm password box is left empty on pressing the Delete button.	User type: Administrator , User ID: , Password: 111 , Confirm Password:	Error message shown saying none of the fields can be left empty.	23	Successful	
24.	Testing to check if confirmation message shown on the deletion of a record, the fields in the form are emptied and the database is updated on the deletion of a record.	User type: Administrator , User ID: 1111 , Password: 111 , Confirm Password:	Confirmation message shown saying the record has been deleted.	4	Successful	
25.	Testing to check if form is closed on pressing cancel button.	User type: , User ID: , Password: , Confirm Password:	The form closes.		Successful	
26.	Testing to check if Employee Entry form appears on selection of Employee Entry form from the Main Menu, a new	Selecting Employee Entry form from Main Menu.	Employee Entry form is loaded and Employee ID list shows a number of	12,14	Successful	

	Employee ID is generated in the Employee ID box and the Employee ID list is loaded automatically.		Employee IDs.			
27.	Test to check if error message is shown if wrong character type, other than numbers is entered in the Contact number textbox.	Contact number: a	Error message shown asking for Numerical data.	16,24	Successful	
28.	Testing to check if error message is shown if any of the fields are left empty on pressing the Make Entry button.	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: ,Designation: Officer, Basic salary: \$10000	Error message shown saying none of the fields can be left empty.	23	Successful	
29.	Test to check if error message is shown if number of digits provided for Contact number is invalid.	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 017471685555, Designation: Officer, Basic salary: \$10000	Error message produced saying number of digits provided for contact number is invalid.	25	Successful	
30.	Testing to check if error message shown if Employee ID given is already provided present in the database and Employee ID box is automatically cleared, on pressing Make Entry button.	Employee ID: 111, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 01747168555, Designation: Officer, Basic salary: \$10000	Error message produced saying the employee ID provided is already present in the database.	13	Successful	

31.	Testing to make sure confirmation message is shown on addition of the new record to notify the User that the record is added successfully and the Employee ID list and the database are updated and the fields in the form are emptied automatically on pressing Make Entry button if all the information provided is valid. The Employee ID in the Employee ID textbox is also updated for next transaction.	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 01747168555, Designation: Officer, Basic salary: \$10000	Confirmation message shown to notify the User that the record has been added successfully. New Employee ID is added to the list. New record is added to the database. The Employee ID in the textbox is updated. The fields in the form are cleared.	7	Successful	
32.	Testing to check if all the fields are filled automatically on selection of an Employee ID from the Employee ID list	Select an Employee ID from the Employee ID list.	All the boxes are filled.	14,16		
33.	Testing to check if error message shown if Update button is clicked with any of the fields left empty.	Leaving basic salary empty and clicking Update button.	Error message saying none of the fields can be left empty	23		
34.	Testing if error message shown if Employee ID is changed on pressing Update button.	Employee ID: 118	Error message saying Employee ID cannot be changed.	3		
35.	Testing if confirmation message is provided on updating a record, the record is updated in Employee Entry database table, the fields on the form are emptied afterwards, Employee ID list is updated and a new Employee ID is generated in the Employee ID box on	Basic Salary: \$8000, Contact number: 01752433307	Confirmation message on Update of the record. The fields on the form are emptied. The Employee ID list is updated	7,14		

	pressing Update button, provided all valid information is provided and none of the fields are left empty.		with new Employee IDs. A new Employee ID is generated in the Employee ID box. The record in the database is updated.			
36.	Testing if confirmation is shown on deletion of a record, the fields on the form are emptied and the record is deleted from the database.	Select a record from Employee ID list and click Delete button.	Confirmation message for deletion of the record. Fields on the form are emptied. The record is deleted from the database.	7,14		
37.	Testing to check all the fields on the form are cleared on pressing the Refresh button and the Employee ID list is refreshed and Employee ID box is showing proper Employee ID for the next record.	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 01747168555, Designation: Officer, Basic salary: \$10000 Click Refresh button	All the fields are emptied.	26		
38.	Testing to check if form is closed on clicking Cancel button.	Cancel button is clicked	The form closes.		Successful	
Test #38 is repeated for all other forms and all of them proved successful.						
39.	Testing to check if Package Entry Form is loaded on selection from the Main	Package Entry selected from database.	Package Entry form appears.	12,14	Successful	

	Menu form, a new Package ID is generated for new transaction and the Package types are automatically loaded in the update section.		Package ID generated at Package ID textbox. Package types are loaded in the combo box.			
40.	Testing to check if error message shown if any of the fields are left empty on pressing Save button	All fields are filled except cost per person.	Error message shown saying none of the fields can be left empty.	23	Successful	
41.	Testing to check if error message shown if anything other than numerical data is provided for Cost per person or Duration.	Cost per person:	Error message shown saying only numerical data can be provided.	16,24	Successful	
42.	Testing to check if the Package ID provided is already in use. Error message shown if it is.	Package ID: 1121	Error message shown saying the Package ID provided is already in use.	0	Successful	
43.	Testing to check if confirmation message for addition of data to the database is shown, the fields are automatically cleared after message shown, record is added to database table, Package ID textbox is updated and Package type and Package name list is updated after addition of new record on pressing Save button, provided all the information entered is valid.	Package type: Multi-day Package name: River Run through Venice Package ID: 1124 Cost per person: 25000 Accommodation type: Normal Itinerary: 12/2/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/2/16 Sail through Venice 15/2/16 Day of	Confirmation message saying the data has been added successfully. The fields on the form are emptied. The new record is added to the database. Package type and Package name list is updated.	5,14	Successful	

		leisure Package details: 4 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options				
44.	Testing to check if all the fields are emptied on pressing Refresh button, Package ID box is updated and the Package type box is refreshed.	Click Refresh button.	All the fields are emptied.	26	Successful	
45.	Testing to check if Package name list in Update section is automatically loaded with names on selection of a Package type and all the other information in the form are automatically loaded on selection of a Package name from the list.	Selecting a Package type from combo box of Update section. Selecting a Package name from the names that appear on the list.	All the fields are automatically filled.	14,16	Successful	
46.	Testing to check if error message shown if Package ID is changed or any of the fields are empty on pressing the Update button.	Package ID is changed and Update button is pressed.	Error message shown saying the Package ID cannot be changed.	26	Successful	
47.	Testing to check if confirmation message is shown on Updating a record, all the fields are emptied afterwards, Package type combo box in the update section is updated, Package ID is generated for next transaction and record is updated in the database table on pressing the Update button, provided all the information entered is valid.	Cost per person: 20000	Confirmation message saying the record is updated. All the fields are emptied. Package types available are updated. Record is updated in the database.	5,14	Successful	

48.	Testing to check if error message shown if any of the fields are left empty on pressing the Delete button.	Cost per person is left empty Click Delete button.	Error message shown saying none of the fields can be left empty.	23	Successful	
49.	Testing to check if Deletion confirmation message is shown after deletion of a record, all the fields on the form are emptied automatically, Package ID generatd for next transaction and the database table and Package type and Package list combobox are updated on pressing the Delete button.	Select a Package type and Package name from the Update section. Click Delete button.	Confirmation message on deletion of record. All the fields are emptied. The Package name is removed from the list and Package type is updated.	5,14	Successful	
50.	Testing to check if Package Expenses Entry form opens on selection from Main Menu and names are automatically assigned to the columns of "Overall expenses data" section.	Click Package Expenses Entry from Main Menu.	Package Expenses Entry form is loaded.	20	Successful	
51.	Testing to check if error message is shown if non – numerical data is entered in the Amount section.	Amount: s	Error message shown saying only numerical data is accepted.	16,24	Successful	
52.	Testing to check if error message shown if any of the fields in the "Package Expenses information" section is left empty on pressing Done button.	"Amount" Field is left empty.	Error message shown saying none of the fields in the "Package Expenses information" can be left empty.	23	Successful	
53.	Testing to check if Package Expenses data is automatically added to the flexgrid in the "Overall expenses data" section of the form and to the Package Expenses Details database table on pressing Done button, provided all the information entered is	Package ID: 1122 Package name: Wind of Brisbane Expenses: Transportation, Amount: 5000 Expenses: Accommodation, Amount: 15000 Expenses: Entry	Data is added to the "Overall Expenses data" table in the form. Data is added to the Package Expenses Details database table.	20	Successful	

	valid.	fees, Amount: 3000 Expenses: Accommodation, Amount: 20000 Expenses: Transportation, Amount: 10000				
54.	Testing to check if all the amounts in the Overall Expenses data table are added, a total is calculated and the total value is shown at the Total textbox on pressing the calculate button	The data from the previous test is used.	A total amount appears on the total textbox.	11	Successful	
55.	Testing to check if a message is shown for confirmation that the data is saved and if Package ID, Package name and the total amount is added to the Package Expenses master table in the database on pressing Save button, provided all the information entered is valid.	The data from the previous test is used.	Confirmation message that the data is saved. New record is added to the database table.	20	Successful	
56.	Testing to check if all the fields in the form are emptied and the flexgrid table is refreshed on pressing the refresh button.	The Refresh button is clicked.	All the fields are emptied.	26	Successful	
57.	Testing to check if Package Booking form is loaded on selection from the Main Menu and the Package types are automatically loaded.	Selecting Package Booking form from Main Menu.	Package Booking form is loaded. Package types are loaded on the form.		Successful	
58.	Testing to check if the Package names list is loaded with names on selection of a Package type and all the required Package information is	Select a Package type. Select a Package name.	Package names appear on the list on selection of a Package type.	16	Successful	

	filled on the form on selection of a Package name.		Fields dependent on the Package names are filled.			
59.	Testing to check if the fields that depend on the Package name are automatically emptied if Package type is changed.	Select a Package type.	Fields dependent on Package name are emptied.	16	Successful	
60.	Testing to check if error message shown on provision of non-numerical data for fields such as Contact number, Number of people etc.	Number of people: p	Message shown asking to provide only numerical data.	16,24	Successful	
61.	Testing to check if error message shown if any of the fields are left empty on pressing Confirm button.	"Number of People" and "Net total" is left empty.	Error message shown saying none of the fields can be left empty.	23	Successful	
62.	Testing if Net total is automatically calculated on provision of Cost per person and Number of People.	Package type: Multi-day Package name: Sailing in Paris Number of People: 5	Net total is generated on provision of Cost per person and Net total	11		
63.	Testing to check if error message shown if number of digits of the contact number provided is invalid on pressing Confirm button.	Contact number: 0153452568888	Error message shown to notify that the number of digits for contact number is invalid.	25	Successful	
64.	Testing to check if error message shown if Number Reserved exceeds Number of Reservations limit on pressing Confirm button.	Number of people: 492	Error message saying there are no vacancies left for the particular Package.	18	Successful	
65.	Testing to check if confirmation message is provided on the addition of a new record to the Package Booking database table, the receipt is generated, all the fields on the form are emptied, the Booking ID and the database table are updated on pressing	Package type: Multi-day, Package name: Wind of Brisbane, Number of People: 3, Booking ID: 1240, Customer name: Chris Gellar,	Confirmation message shown saying the Booking was successful. All the fields are emptied. Booking ID is updated.	8,18,27	Successful	

	Confirm button, given that all the information provided is valid.	Address: 25th, Wolwin Heights, Chittagong, Contact number: 01534525688	Package Booking database table is updated for addition of new record. Vacancy information in Package Entry database is Updated.			
66.	Testing to check if all the fields on the form are emptied on pressing the Refresh button.	Click Refresh button while some of the fields in the form is filled.	All the fields on the form are emptied.	26	Successful	
67.	Testing to check if Package Booking Update form opens on selection from Main Menu and Package type combo box is loaded with Package types.	Click Package Booking Update form from Package information section of Main Menu form.	Package Booking Update form is loaded.	14	Successful	
68.	Testing to check if error message shown asking for Booking ID if Search button is pressed without giving it.	Booking ID field left empty. Click Search.	Error message shown saying that any record with the Booking ID provided does not exist in the database.	23	Successful	
69.	Testing to check if error message is shown if the Booking ID provided is not found in the database.	Booking ID:2253 Click Search.	Error message shown saying the Booking ID provided does not exist in the database.		Successful	
70.	Testing to check all the other fields are automatically loaded from the database table and filled on provision of the Booking ID and pressing Search.	Booking ID:1238 Click Search.	All the information is automatically loaded from the database.	15	Successful	
71.	Testing to check if error message shown if non – numerical data is provided	Contact number: p	Error message shown saying only numerical	16,24	Successful	

	for Contact number or Number of people.		data is accepted.			
72.	Testing to check if error message shown if Booking ID is changed.	Booking ID: 1212 Click Update.	Error message shown saying Booking ID cannot be changed.	3	Successful	
73.	Testing if error message shown if any of the fields are left empty on pressing Update button.	Contact number: "" Click Update.	Error message shown saying none of the fields can be left empty.	23	Successful	
74.	Testing to check if error message shown if number of digits for contact number is invalid.	Contact number: 015344595988 Click Update.	Error message shown saying number of digits provided for contact number is invalid.	25	Successful	
75.	Testing to check if error message shown if Number reserved exceeds number of reservations.	Previous test data used. Number of people: 3	Error message shown saying there are no vacancies for the selected Package.	18	Successful	
76.	Testing to check if Net total is calculated automatically on entering Number of people.	Booking ID: 1238 Address: 24 th , Bashundhara street, Chittagong Package type: Multi-day Package name: Sailing in Paris Number of People: 5	Net total is calculated automatically and generated in textbox	11,16	Successful	
77.	Testing to check if confirmation message is provided for successful update of the record, followed by the emptying of the fields on the form and the information held by the record is changed in the database table on making changes and pressing the Update button, provided the	Booking ID: 1238 Address: 24 th , Bashundhara street, Chittagong Package type: Multi-day Package name: Sailing in Paris Number of People: 5 Click Update.	Confirmation message that the record is updated. All the fields in the form are emptied. The database record is updated for any	9,18	Successful	

	information entered is valid.		changes made. Vacancy information is updated in Package Entry database.			
78.	Testing to check if Confirmation message for deletion is shown, all the boxes are emptied, reservation information in Package Entry database is updated and the record is deleted from database on pressing Delete button, provided all the fields are filled.	Booking ID: 1238 Click "Search". Click "Delete".	Confirmation message shown for the deletion of the record. All the boxes are emptied. Package Entry table reservation information is updated. Record is deleted from the Package Booking database table.	10,18	Successful	
79.	Testing if all the fields are emptied and the combo box information are reloaded on pressing refresh button.	Booking ID: 1237 Click "Search" button. Click "Refresh" button.	All the fields are emptied.	26	Successful	
80.	Testing to check if Vehicle entry form appears when selected from the Main Menu and a Vehicle ID is generated for the new transaction being made.	Select "Vehicle Entry" from Main Menu	The Vehicle Entry form is loaded.	12	Successful	
81.	Testing if error message appears if any of the fields are left empty on pressing Submit button.	Vehicle type: SUV Vehicle ID: 1006 Registration number: AD2589 Make: "" Model: X3 Number of seats: 5 Rate per kilometer: 2400 Click "Submit"	Error message appears saying none of the fields can be left empty.	23	Successful	

82.	Testing to check if error message appears if non-numerical data is entered in “Rate per kilometer” field.	All the previous data are used. Make: BMW Rate per kilometer: r Click “Submit”	Error message asking for numerical data.	16,24		
83.	Testing if confirmation message appears on addition of new record to the database, all the boxes in the form are emptied and the Vehicle ID updated for new transaction and the new record is added to the Vehicle Entry database table on pressing Submit button, provided all valid information is entered.	Vehicle type: SUV Vehicle ID: 1006 Registration number: AD2589 Make: BMW Model: X3 Number of seats: 5 Rate per kilometer: 2400 Click “Submit”	Confirmation message displayed for the addition of new record to the database. All the boxes are emptied. Vehicle ID is updated. The new record is added to the database.	6,12	Successful	
84.	Test to check if all the fields are emptied on pressing Refresh button and the Vehicle ID is reloaded.	Vehicle ID: 1007 Make: BMW Model: X3 Registration number: AD5466 Rate per kilometer: 2400 Click “Submit”	All the fields are emptied. Vehicle ID textbox is reloaded.	26	Successful	
85.	Testing if Vehicle Update form opens on selection from Main Menu and list of Vehicle IDs is loaded from database.	Select “Vehicle Update” form from Vehicle information section of database.	Vehicle Update form appears. List of Vehicle IDs is loaded.	14	Successful	
86.	Testing if the fields on the form are automatically filled on selecting an ID from the list.	Vehicle ID: 1003	All the fields are filled.	16	Successful	
87.	Testing if error message shown if non-numerical data is provided in fields such as rate or number of seats.	Number of seats: 9	Error message shown asking for numeric data.	16,24	Successful	
88.	Testing if error message shown if any of the fields	Vehicle type: Sedan	Error message shown saying	23,26	Successful	

	are left empty on pressing Update button.	Number of seats: 5 Registration number: AD4565 Rate: "" Make: Cadillac Model: DTS	none of the fields can be left empty.			
89.	Testing if confirmation message shown on updating of a record, followed by emptying of the fields on the form and the update of record in the database table on pressing Update button, provided none of the fields are left empty and all the information provided is valid.	Vehicle ID: 1003 Registration number: AD4582 Rate: 2000 Click "Update" button	-Confirmation message shown on update of data. -All the fields in the database are cleared. -The record is updated in the database table.	6,23	Successful	
90.	Testing if confirmation message shown on deletion of a record from the database, followed by emptying of the fields on the form and updating of the record on pressing the Delete button, provided none of the fields are left empty.	Vehicle ID: 1003 Click "Delete" button	-Confirmation message for the deletion of the record. -All the fields are emptied. -The record is updated in the database.	6	Successful	
91.	Testing to check if Vehicle Booking form is opened on selection from Main Menu and Vehicle types are loaded.	Select Vehicle Booking form from Main Menu.	Vehicle Booking form opens.		Successful	
92.	Testing to check if error message shown if non-numerical data is entered as contact number, advance paid or starting kilometer.	Advance paid: p	Error message asking for numerical data input instead.	16,24	Successful	
93.	Testing to check if error message shown if any of the fields are left empty on pressing Confirm button	Some of the fields are left empty. Click "Confirm"	Error message appears saying none of the fields can be left empty.	23	Successful	
94.	Testing to check that a Vehicle that is unavailable	Select Vehicle type: Sedan	The vehicle does not appear in the	17		

	does not appear in the list of Vehicle Booking form.	Make: Toyota	list.			
95.	Testing if error message shown if number of digits of contact number is invalid.	Contact number: 018155856644	Error message asking for valid input.	25	Successful	
96.	Testing to check if Confirmation message is provided for addition of record to the database, availability of vehicle is changed in Vehicle Entry database table, the new record is added to the Vehicle Booking database table, the fields on the form are emptied, the Booking ID is updated and a token is generated on pressing Confirm button, provided none of the fields are left empty and all valid information is provided.	Customer name: Helen Greene Contact number: 01815585664 Address: 23rd, Golpahar, Chittagong Journey date: 31-03-2015 Drop-off-date: 09-04-15 Vehicle type: SUV Make: BMW Model: X3 Vehicle ID: 1006 Advance paid: 2500 Driver name: Dylan Walker Starting kilometer: 321354	Confirmation message shown for successful booking. “IsBooked?” section of Vehicle Entry database table is updated. New record added to the Vehicle Booking database table. All the fields emptied and New Booking ID generated for new transaction.	8,12,17,28	Successful	
97.	Testing to check if all the fields are emptied from the form on pressing Refresh button	Click “Refresh” button while some of the fields are filled.	All the fields are emptied.	26	Successful	
98.	Testing if Vehicle Booking Update form appears on selection from the Main Menu.	Selection of Vehicle Booking Update from Main Menu.	The form appears.		Successful	
99.	Testing if error message appears if search button pressed without providing Booking ID.	Booking ID is left empty. Click “Search”	Error message appears asking for Booking ID.	23	Successful	
100.	Testing if error message appears if Booking ID provided is not present in the database.	Booking ID: 2153 Click “Search”	Error message appears saying Booking ID being searched is not found.		Successful	

101.	Testing if all the information is loaded onto the form if correct Booking ID is provided.	Booking ID: 1423 Click "Search"	All the fields in the forms are loaded from the database and filled.	15	Successful	
102.	Testing if error message provided if Booking ID is changed and Update button is pressed.	Booking ID: 1480 Click "Update"	Error message shown saying Booking ID cannot be changed.	3	Successful	
103.	Testing if Confirmation message is provided on Updating a record, availability options of vehicles are changed in the Vehicle Entry database table(in case it is changed), the record is updated in Vehicle Booking database table and the fields on the form are automatically emptied on pressing Update button.	Booking ID: 1423 Click "Search" Vehicle type: SUV Make: BMW Model: X1 Vehicle ID: 1004 Click "Update"	Confirmation message on update of record information. Changes made to "IsBooked?" field of the Vehicle Entry database. The record in the Vehicle Booking database is updated. The fields on the form are emptied.	9,17	Successful	
104.	Testing if confirmation message appears on deleting a record, the availability of the Vehicle that was booked is changed in Vehicle Entry database table, the record is deleted from Vehicle Booking database table and the fields on the form are emptied by pressing Delete button of Vehicle Booking Update form.	Booking ID: 1424 Click "Search" Click "Delete"	Confirmation message for deletion on the record. The fields on the form are emptied. The record is deleted from Vehicle Booking database. The Vehicle that was booked is made available in Vehicle Entry database.	10,17	Successful	

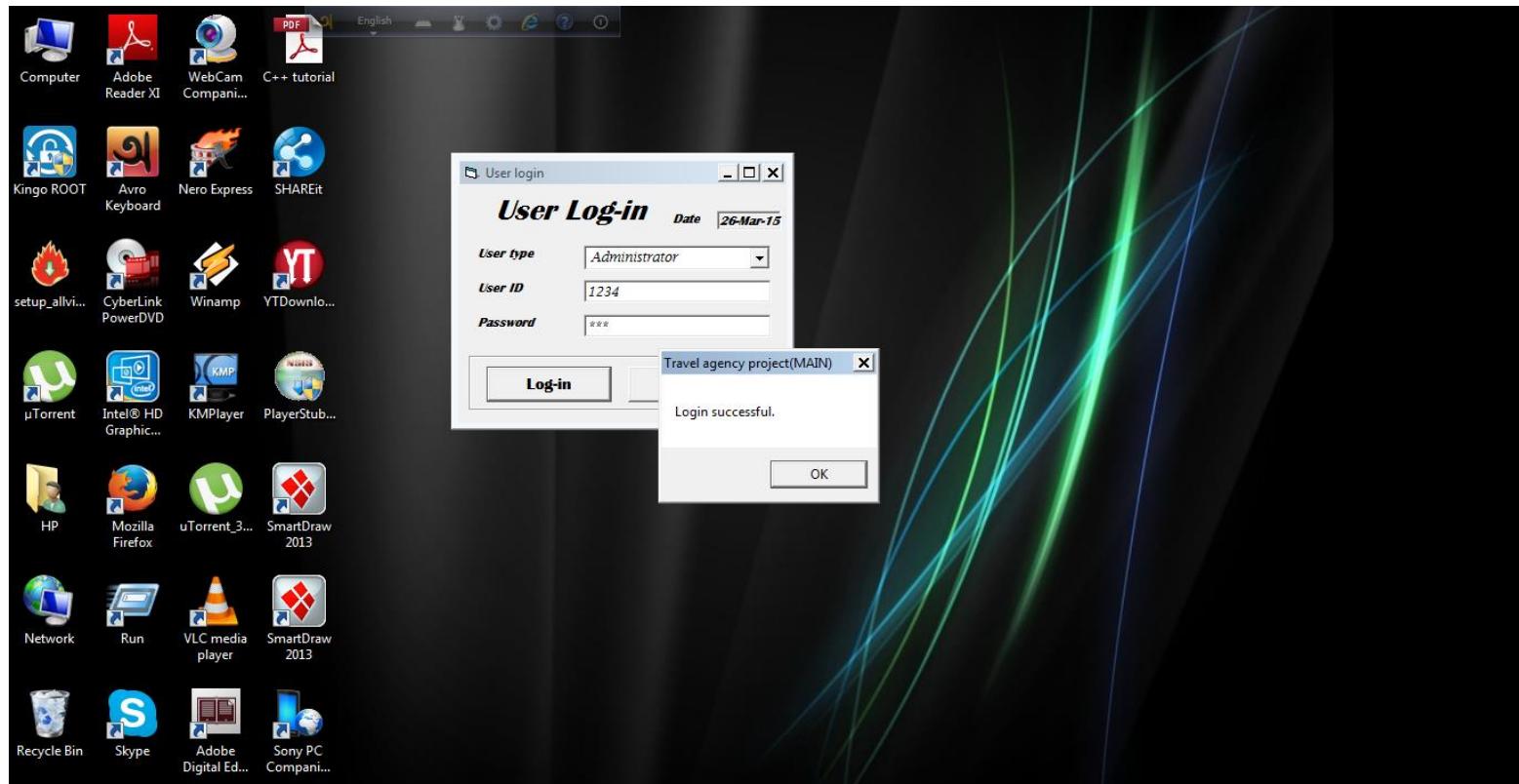
105.	Testing to check if Vehicle information is loaded along with “Late fine” calculated and Booking charge generated on provision of Booking ID in Vehicle Billing form on pressing Search button.	Booking ID: 1423 Click “Search”	Some of the fields are filled. Late fine calculated. Booking charge generated.	15,19	Successful	
106.	Testing to check if Cost of travel is calculated on provision of Ending kilometer.	Ending kilometer: 13220 Move on to next field(coding done in Lost focus event)	Cost of travel is generated.	11,16,19		
107.	Testing if Bill is generated on pressing “Generate Bill” button after providing all the necessary information of the Vehicle Billing form.	Use all the previous data. Driver charge: \$4000 Additional Expenditures: \$2600 Discount: \$500 Click “Generate Bill”	All the calculations are made and a bill is shown on the screen.	11,19	Successful	
108.	Testing if the Billing information is added to the “Vehicle Billing” database table and the Vehicle for which the bill is being made is made reavailable on pressing “Save” button, provided all the information input is valid and none of the fields are left empty.		All the necessary billing information is added to the database table. The vehicle which was booked is made reavailable.	17,19		
109.	Testing if a bill is printed on pressing “Print” button.		A bill is Printed.	19,29	Successful	
110.	Testing if the list of Vehicle IDs is loaded on selection of a Vehicle type on Vehicle servicing form, and Servicing history for a particular Vehicle ID is loaded on selection of that Vehicle ID from Vehicle Servicing form.		The servicing history is loaded.	14,16,21	Successful	

111.	Test to check if servicing details and amount are added to the Servicing history table and the database on pressing the “Save” button		Servicing details and amount are added to the table.	21	Successful	
112.	Testing if Confirmation message is shown on addition of data to the database, Fuel record data is added to the Fuel record database table, the fields are emptied on pressing “Save” button on the Fuel Record form.		Confirmation message shown for addition of data to the database.	22	Successful	
113.	Testing if Backup is created when “Backup Files” button is pressed on the backup form after selecting a suitable directory.		Backup created.	31	Successful	
114.	Testing if reports are generated when any reports of the “Reports” section of the Main menu is clicked on.	Click on names of reports.	Reports generated.	30	Successful	
115.	Testing if User documentation opened when it is selected from the main menu.	Click on User documentation	User documentation opens.	31	Successful	

➤ ***Test Evidence:***

Test #1: Testing whether the Main Menu form is loaded when correct User type, User name and Password is given in

Test data used	User type: Administrator, User ID:1234, Password: 123
Type of data	Normal
Outcome	Confirmation message saying Login successful. Main Menu is loaded.



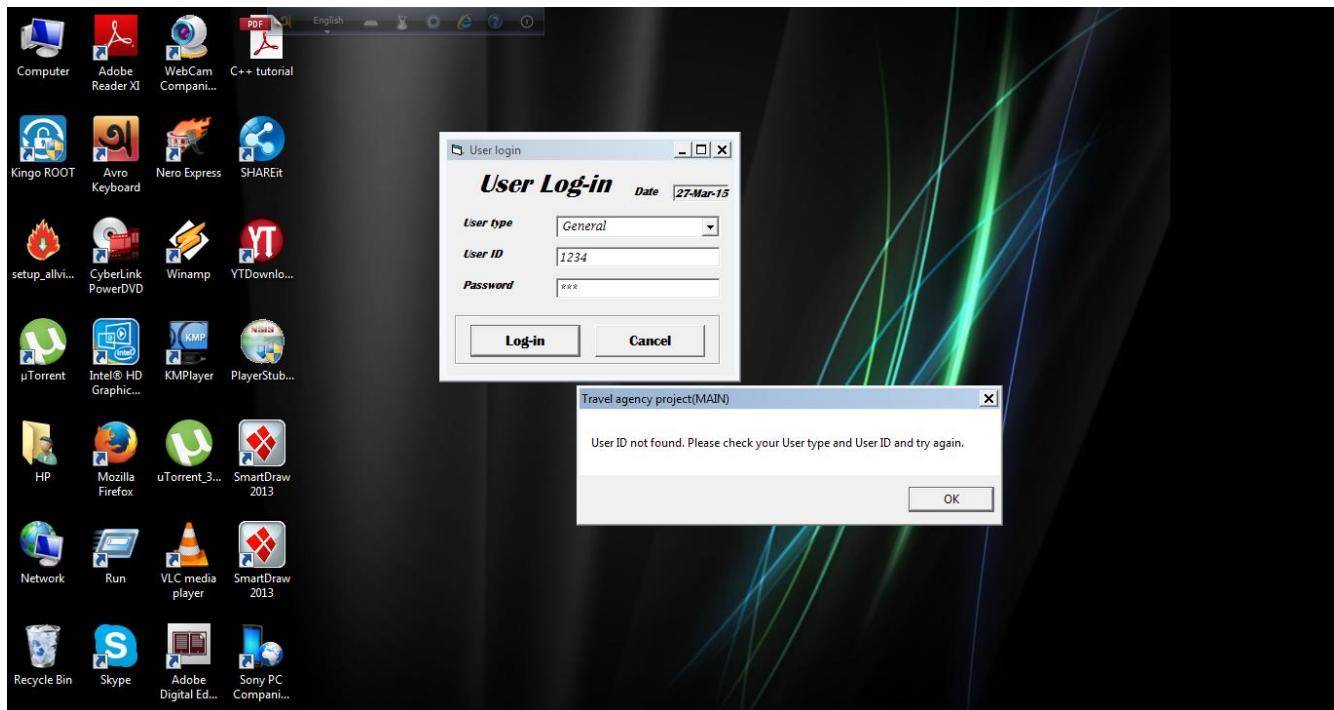
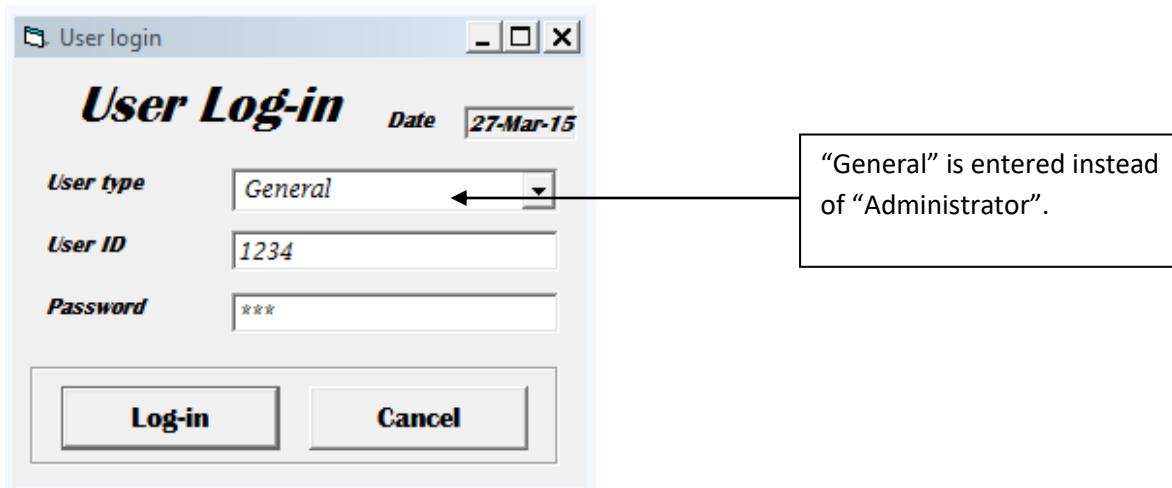
Jan. 1

Travel Agency Management System



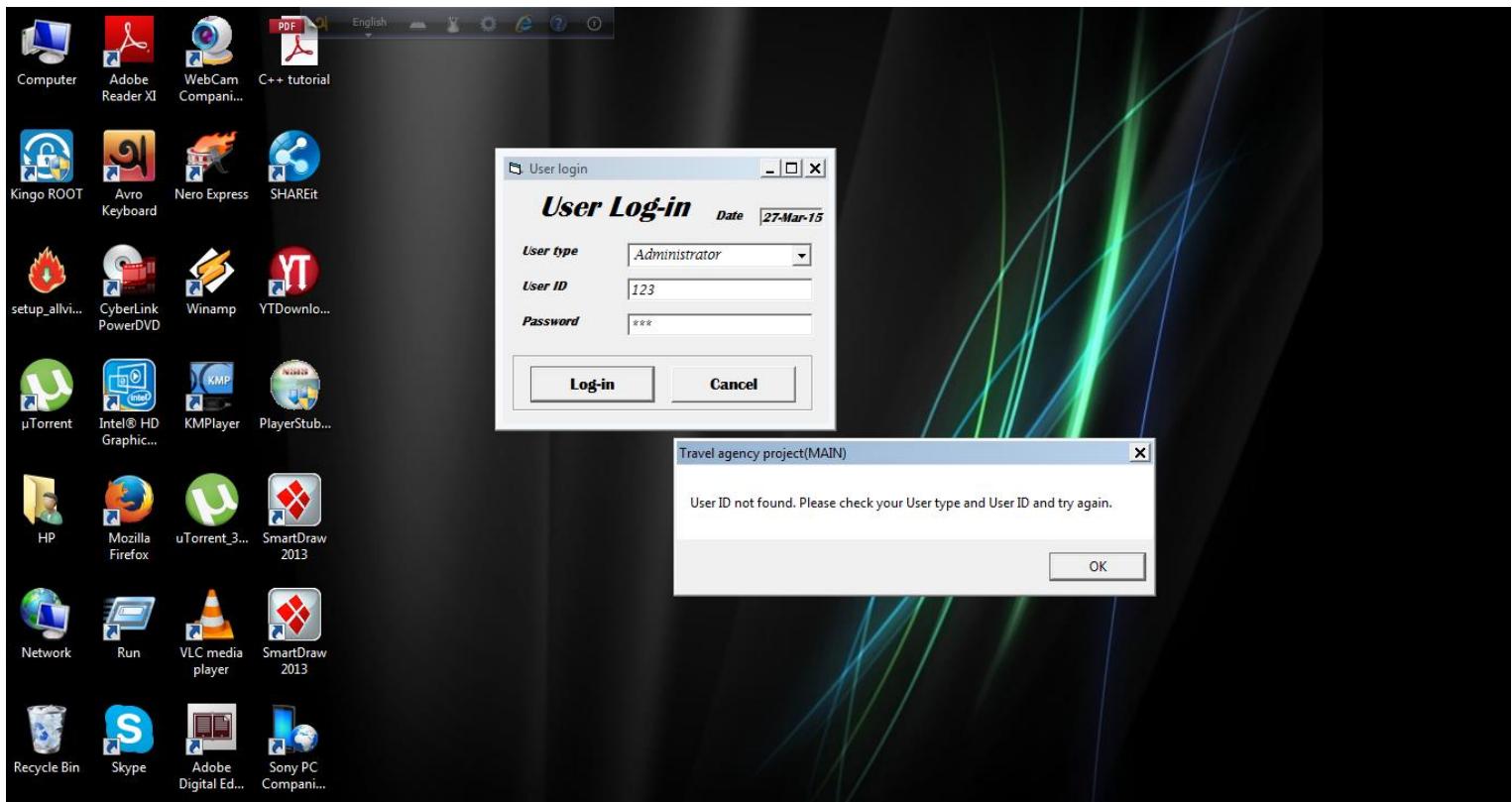
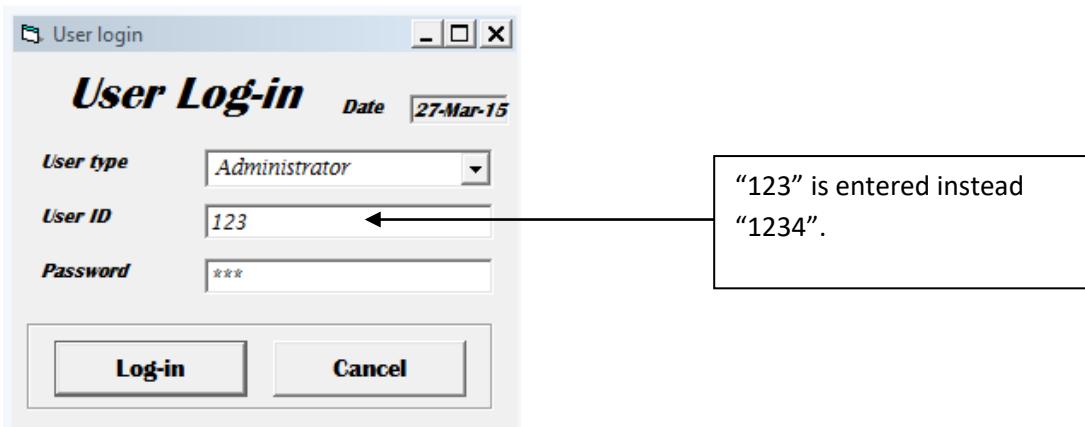
Test #2: Testing whether an error is generated when correct User ID and password but wrong user type given.

Test data used	User type: General, User ID: 1234, Password: 123
Type of data	Normal
Outcome	Error message shown saying "User ID not found. Please check User type and User ID and try again."



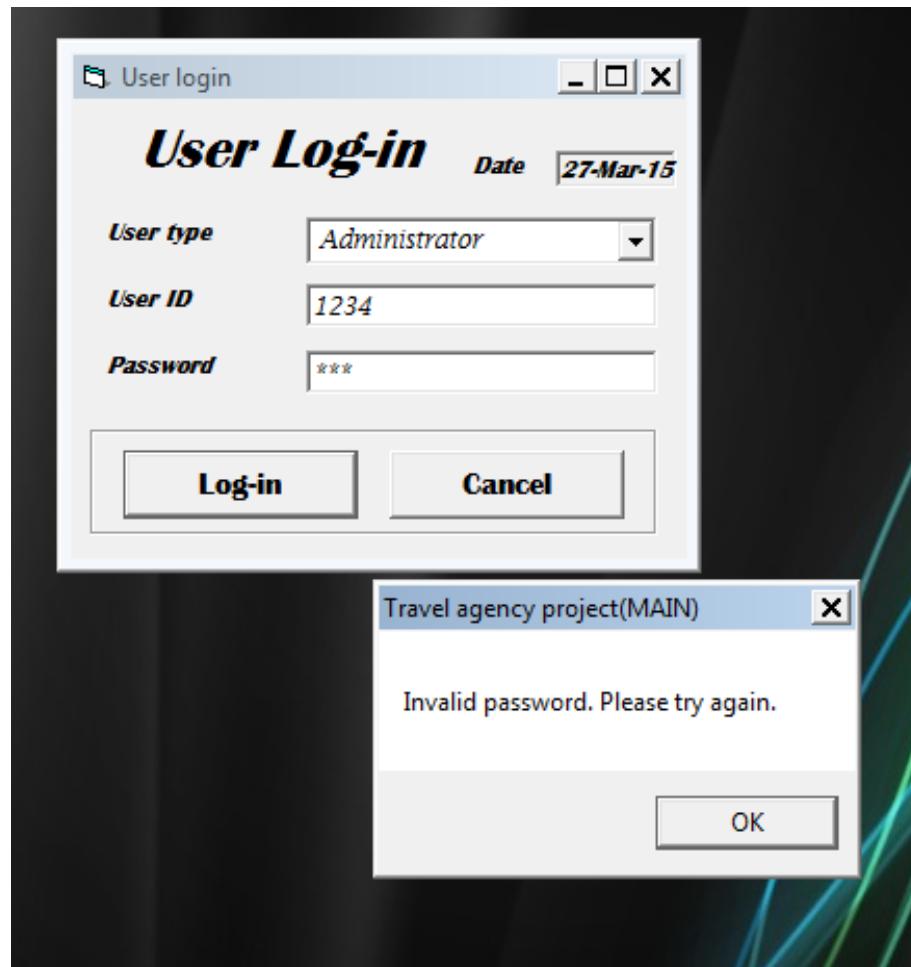
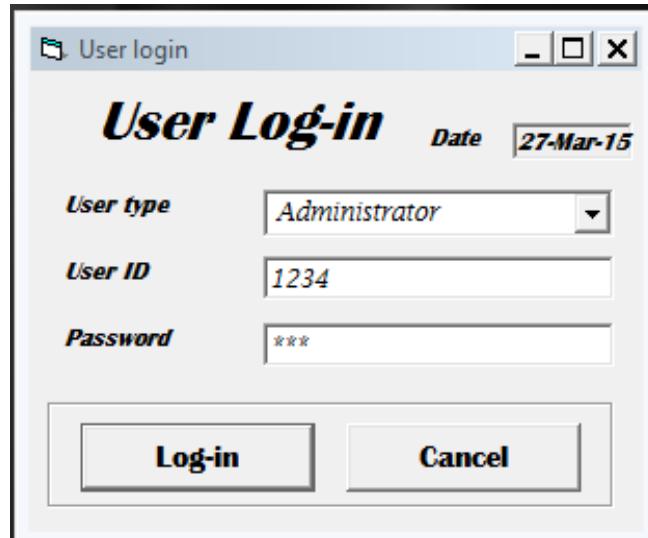
Test #3: Testing if error message is provided if correct User type and Password is provided but wrong User ID is provided.

Test data used	User type: Administrator, User ID: 123, Password: 123
Type of data	Normal
Outcome	Error message shown saying "User ID not found. Please check User type and User ID and try again."



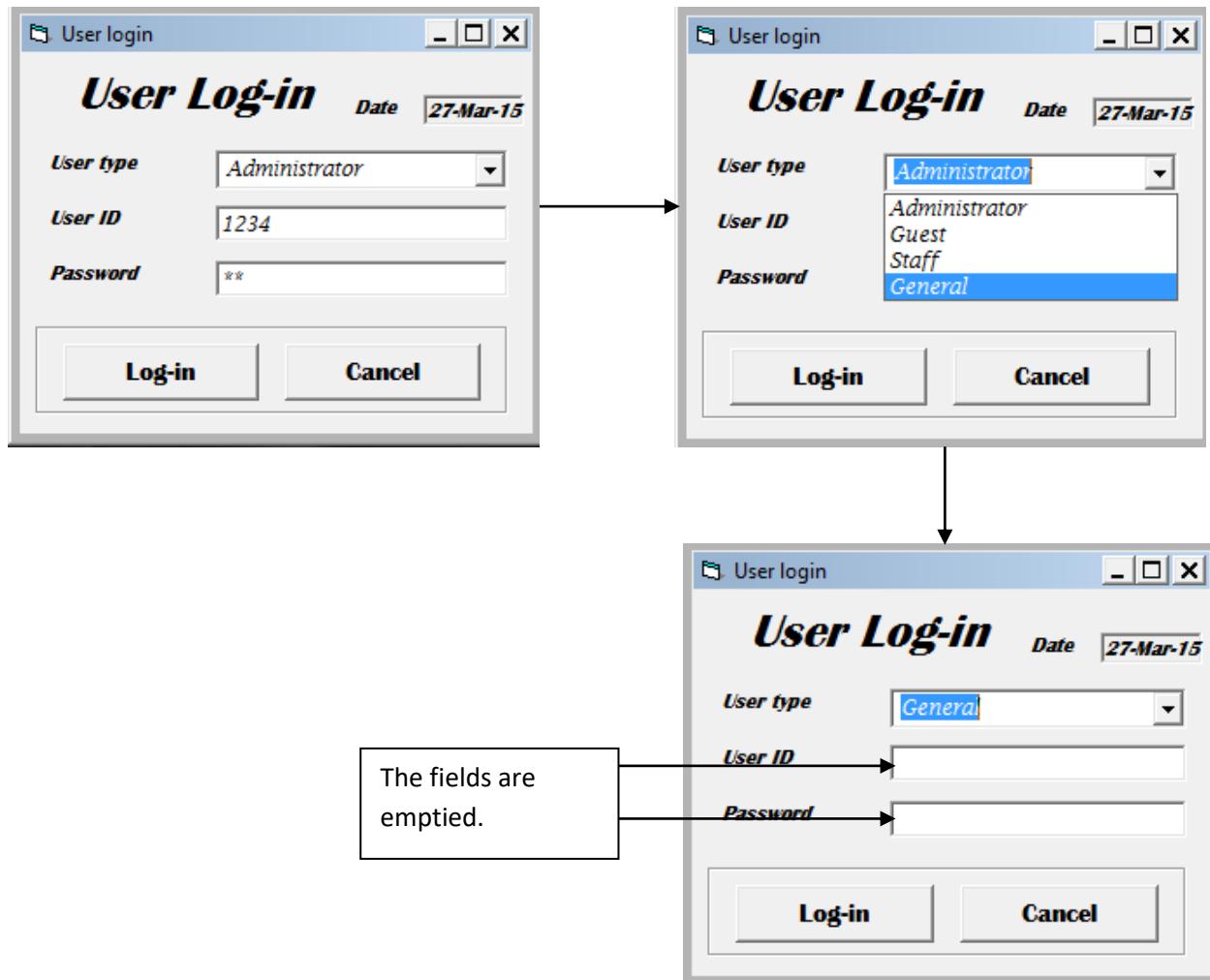
Test #4: Testing if error message is displayed if correct User type and User ID are given but wrong password is given.

Test data used	User type: Administrator, User ID: 1234, Password: 456
Type of data	Normal
Outcome	Error message shown to User notifying that the password provided is invalid.



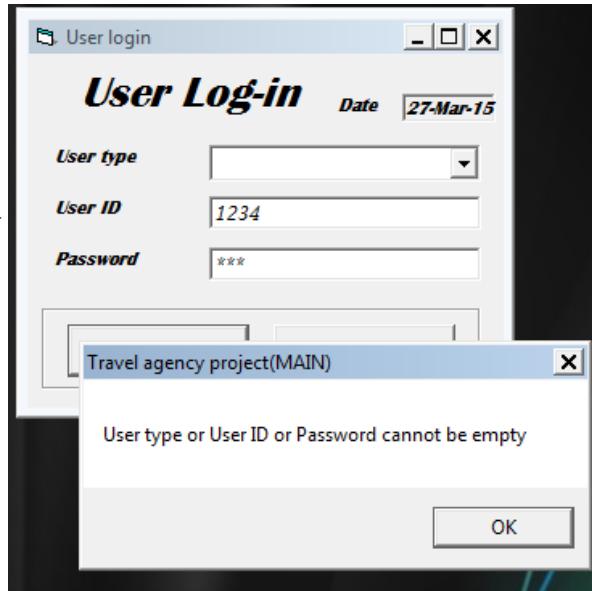
Test #5 Testing if all the other fields are cleared when the User type is changed.

Test data used	User type: General
Type of data	Normal
Outcome	All the boxes are emptied.



Test #6 Testing if error is given if User type box is left empty.

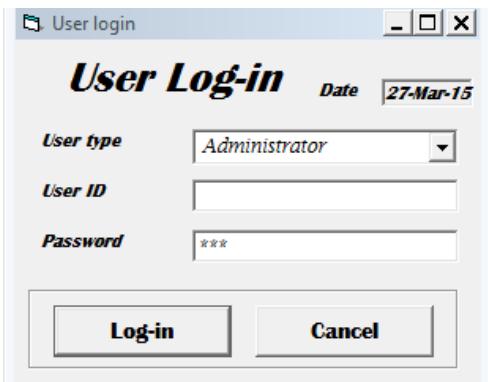
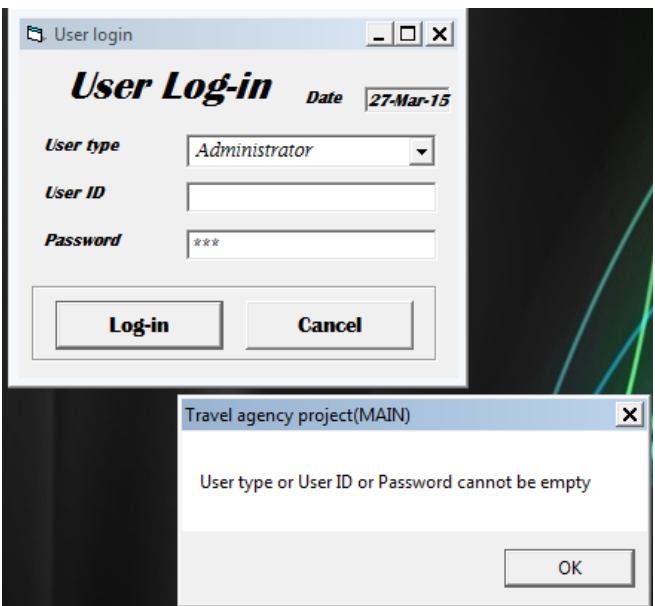
Test data used	User type: , User ID: 1234, Password: 123
Type of data	Normal
Outcome	Error message shown notifying the User that User type, User ID and Password need to be provided.

The screenshot shows two windows. The left window is titled "User Log-in" with a date of "27-Mar-15". It has three input fields: "User type" (dropdown), "User ID" (text box containing "1234"), and "Password" (text box containing "***"). Below the fields are "Log-in" and "Cancel" buttons. An arrow points from this window to the right one. The right window is also titled "User Log-in" with the same date. The "User type" field is empty. A modal dialog box titled "Travel agency project(MAIN)" contains the message "User type or User ID or Password cannot be empty". An "OK" button is at the bottom of the dialog.

Test #7 Testing if error message is shown if User ID box is left empty.

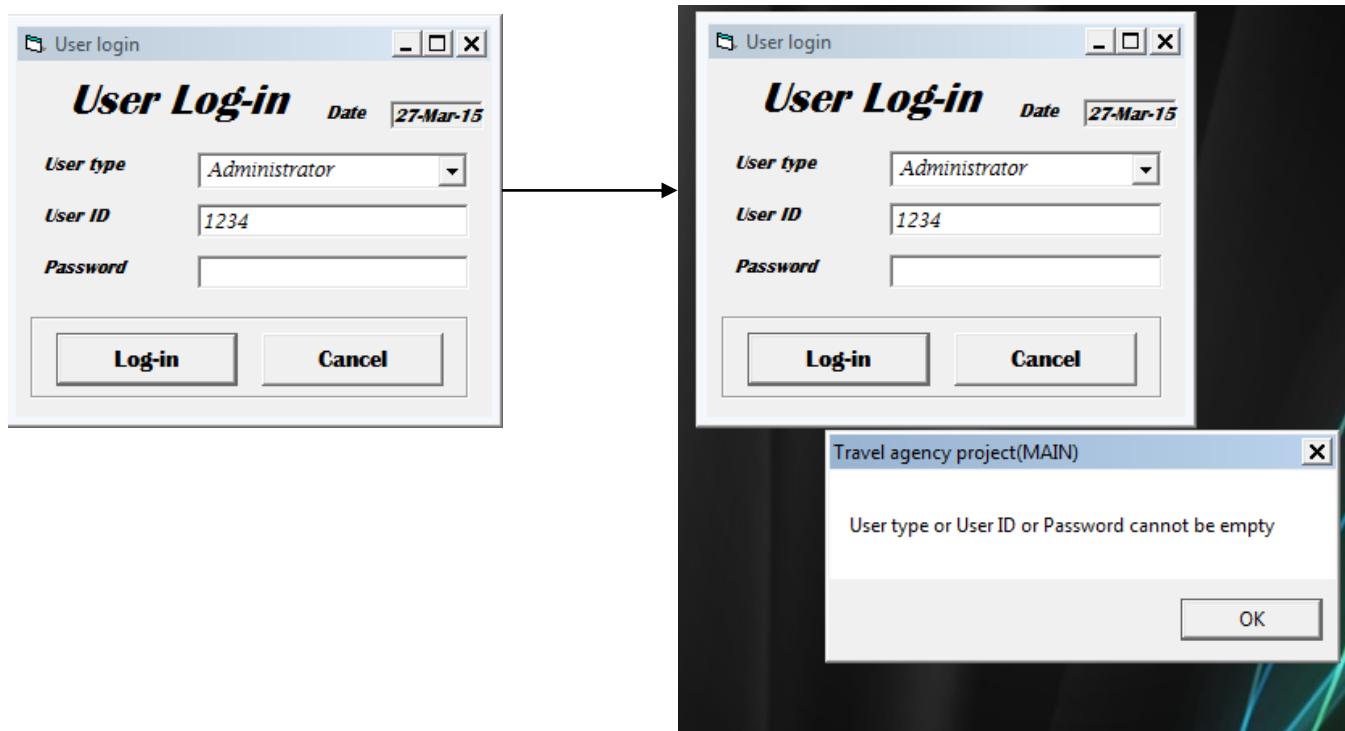
Test data used	User type: Administrator, User ID: , Password: 123
Type of data	Normal
Outcome	Error message shown notifying the User that User type, User ID and Password need to be provided.

The screenshot shows two windows. The left window is titled "User Log-in" with a date of "27-Mar-15". It has three input fields: "User type" (dropdown set to "Administrator"), "User ID" (empty text box), and "Password" (text box containing "***"). Below the fields are "Log-in" and "Cancel" buttons. An arrow points from this window to the right one. The right window is also titled "User Log-in" with the same date. The "User ID" field is empty. A modal dialog box titled "Travel agency project(MAIN)" contains the message "User type or User ID or Password cannot be empty". An "OK" button is at the bottom of the dialog.

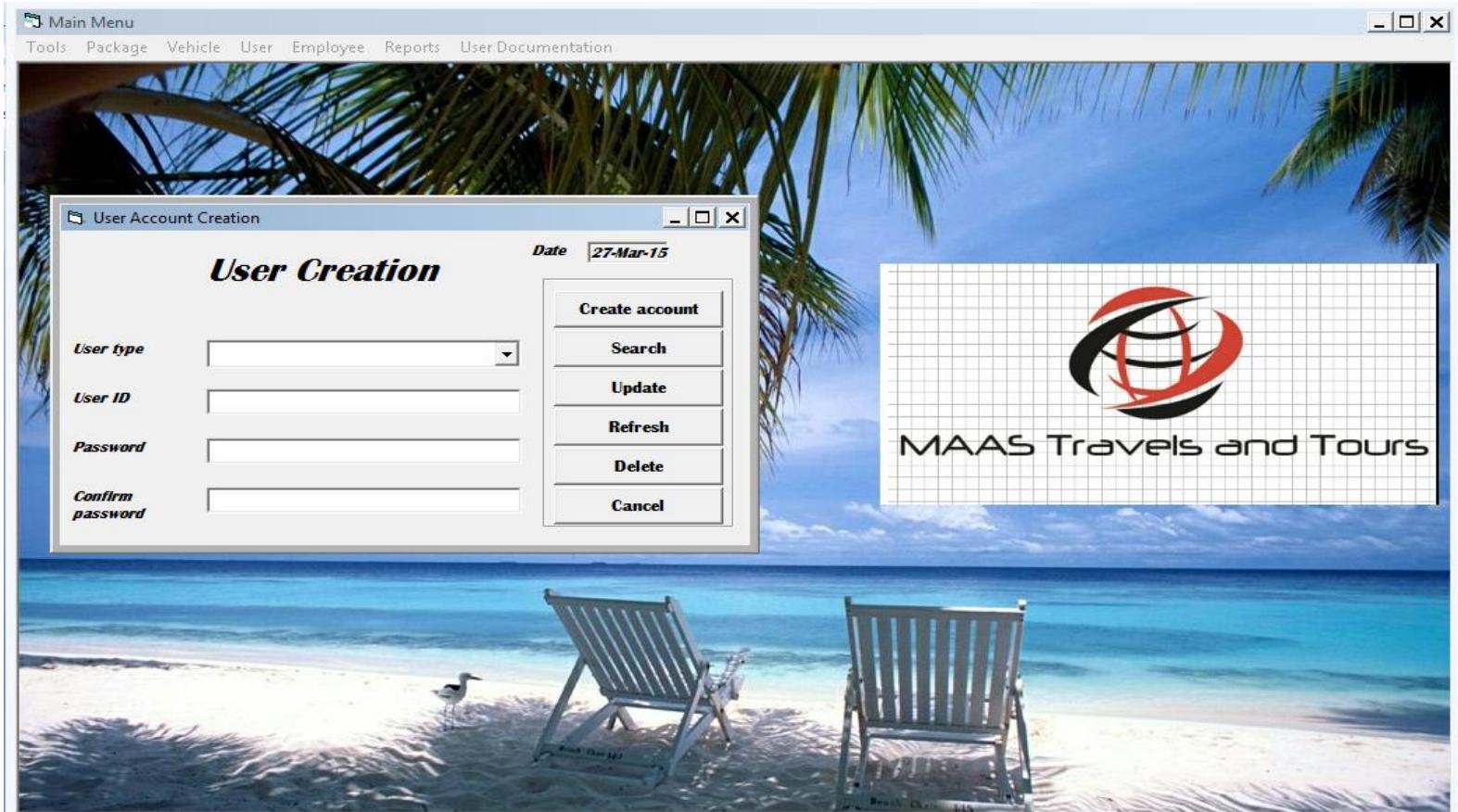
Test #8 Testing if error message is shown if Password box is left empty.

Test data used	User type: Administrator, User ID:1234, Password:
Type of data	Normal
Outcome	Error message shown notifying the User that User type, User ID and Password need to be provided.



Test #9 Testing if User Creation form is displayed when selected from the Main Menu.

Test data used	User type: Administrator, User ID:1234, Password: 123 Select "User Creation" form from Main Menu.
Type of data	
Outcome	User Creation form is displayed.



Test #10 Testing if error message is shown if User type, User ID, Password or Confirm Password fields are left empty on pressing “Create Account” button.

Test data used	User type: Staff, User ID: , Password:1321, Confirm Password:1321
Type of data	Normal
Outcome	Error message shown notifying the User that User type, User ID and Password need to be provided.

User Account Creation

Date 27-Mar-15

User Creation

User type: Staff

User ID:

Password: ***

Confirm password: ***

Create account

Search

Update

Refresh

Delete

Cancel



Test #11 Testing to check if error message shown if the User ID provided already exists in the User Login database table on pressing “Create account” button.

Test data used	User type: Administrator, User ID:1234, Password: 1321, Confirm password: 1321
Type of data	Normal
Outcome	Error message shown notifying the User that the User ID provided is already in use.

Test #12 Testing to check if error message shown if “Password” does not match “Confirm Password” on pressing “Create account” button.

Test data used	User type: Administrator, User ID: 1111, Password:1321, Confirm password: 1123
Type of data	Normal
Outcome	Error message shown to notify the User that the “Password” and

"Confirm Password" do not match.

User Account Creation

User Creation

Date 27-Mar-15

User type	Administrator
User ID	1111
Password	****
Confirm password	****

Create account
Search
Update
Refresh
Delete
Cancel

User Account Creation

User Creation

Date 27-Mar-15

User type	Administrator
User ID	1111
Password	****
Confirm password	****

Create account
Search
Update
Refresh
Delete
Cancel

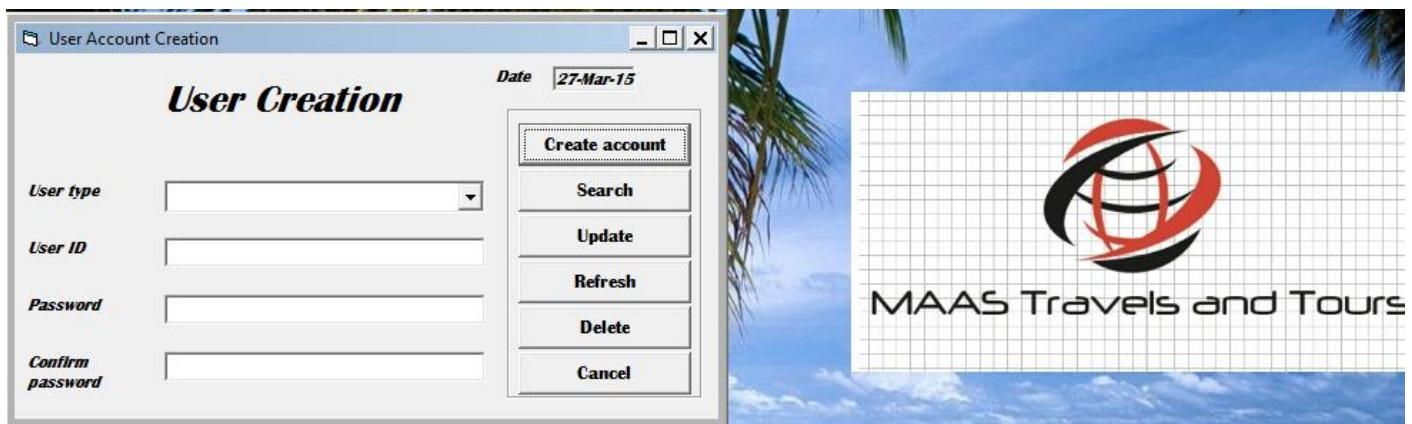
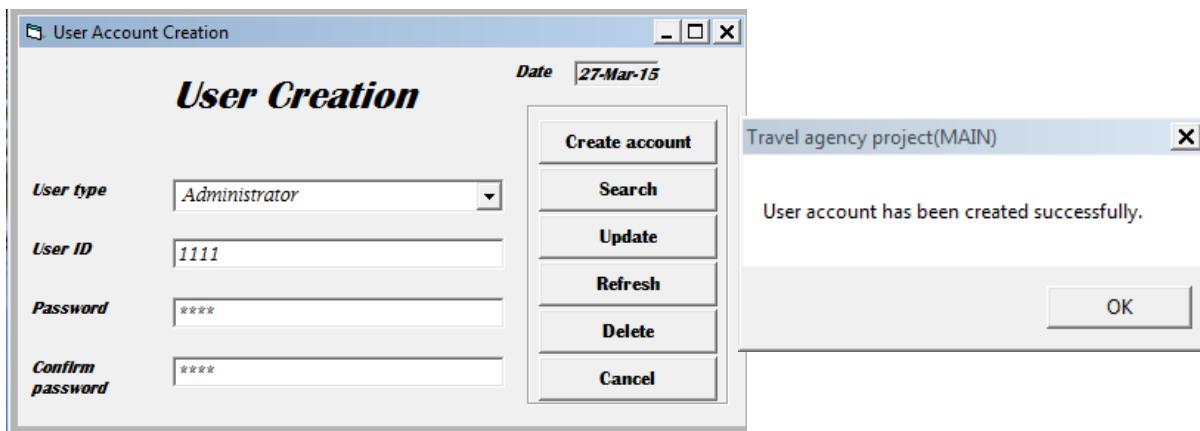
The password and the confirm password do not match. Please re-enter them and try again

OK

Test #13 Testing to check if confirmation message is shown after the new record is entered, all the fields are cleared after, the new record is added to the database table and User login database table is updated.

Test data used	User type: Administrator , User ID:1111 , Password:1321, Confirm password: 1321
Type of data	Normal
Outcome	Confirmation message shown notifying the User that the record is successfully added to the database.

Travel Agency Management System



travel agency database : Database (Access 2007) - Microsoft Access			
Table Tools			
Datasheet			
View	Cut	External Data	Database Tools
Clipboard	Copy	Datasheet	
Views	Format Painter	Calibri	11
Font	B I U	Rich Text	Refresh All
Records	New Save	Σ Totals	Spelling
Sort & Filter	Delete More	Z A	Filter Advanced Toggle Filter
Find	Replace	Z A	Find Go To Select
Security Warning Certain content in the database has been disabled Options...			
User_login			
User_type	User_ID	Password	Add New Field
Administrator	1111	1321	*

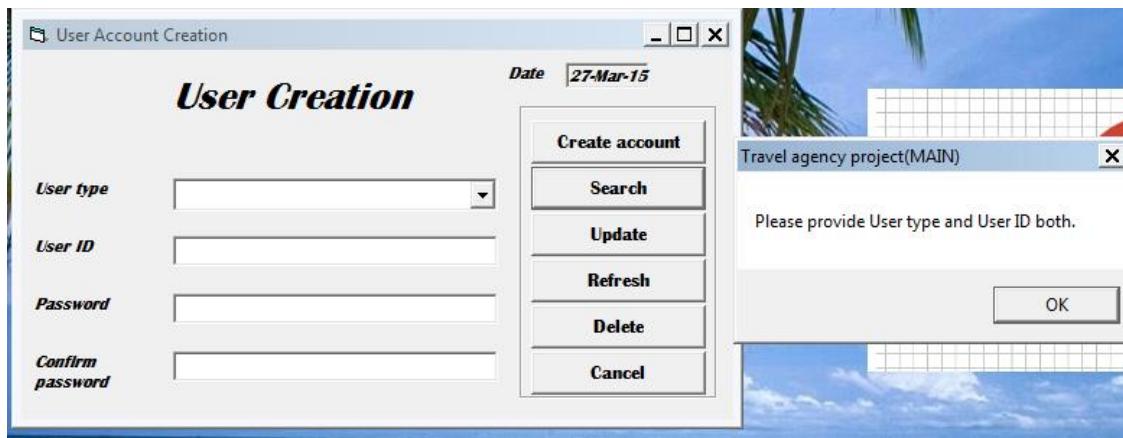
Database before Account is created.

travel agency database : Database (Access 2007) - Microsoft Access			
Table Tools			
Datasheet			
View	Cut	External Data	Database Tools
Clipboard	Copy	Datasheet	
Views	Format Painter	Calibri	11
Font	B I U	Rich Text	Refresh All
Records	New Save	Σ Totals	Spelling
Sort & Filter	Delete More	Z A	Filter Advanced Toggle Filter
Find	Replace	Z A	Find Go To Select
Security Warning Certain content in the database has been disabled Options...			
User_login			
User_type	User_ID	Password	Add New Field
Administrator	1111	1321	*
Administrator	1234	123	*

Database after Account is created.

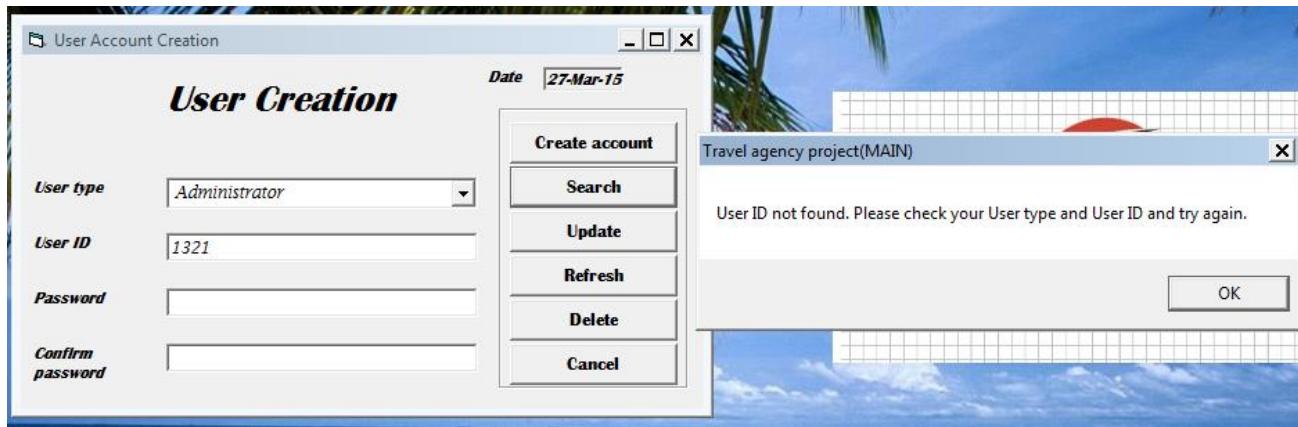
Test #14 Testing to check if error message shown if Search button is pressed without filling up User type and User ID.

Test data used	User type: , User ID:
Type of data	Normal
Outcome	Error message shown asking for User type and User ID.



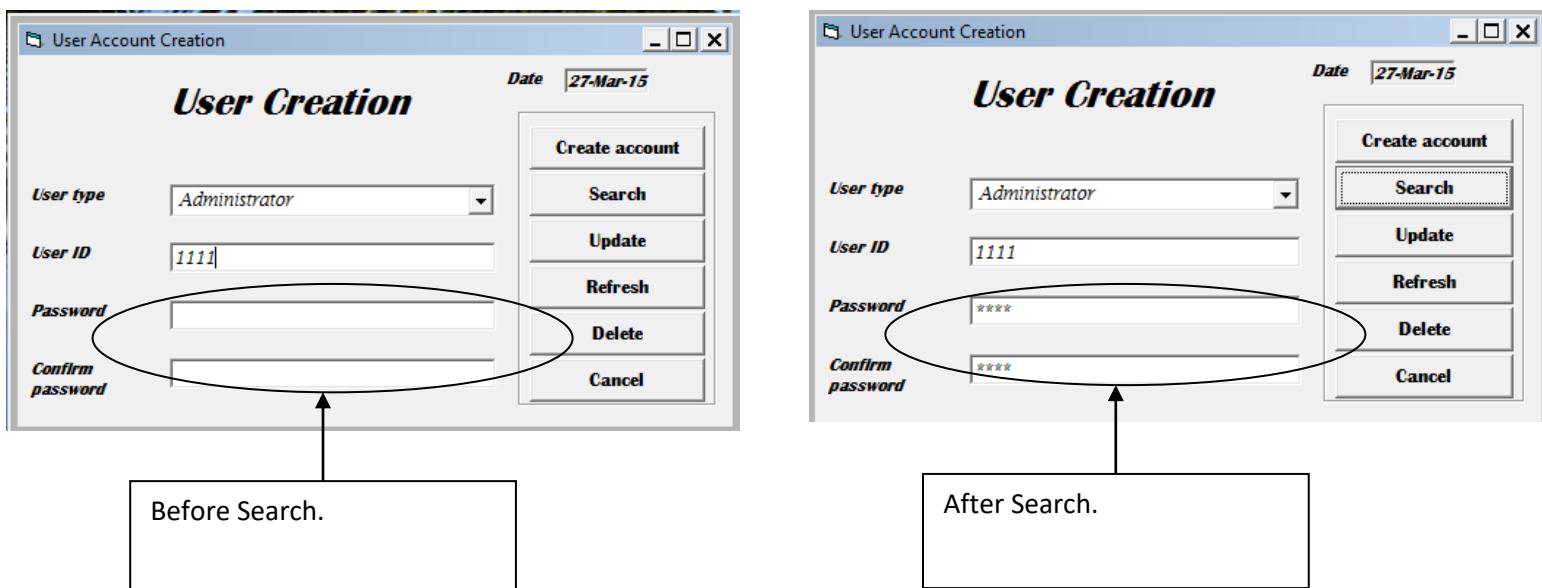
Test #15 Testing to check if error message is shown if User ID for the particular User type is not found in the database.

Test data used	User type: Administrator , User ID: 1321
Type of data	Normal
Outcome	Error message shown saying there is no User ID for the particular User type in the database.



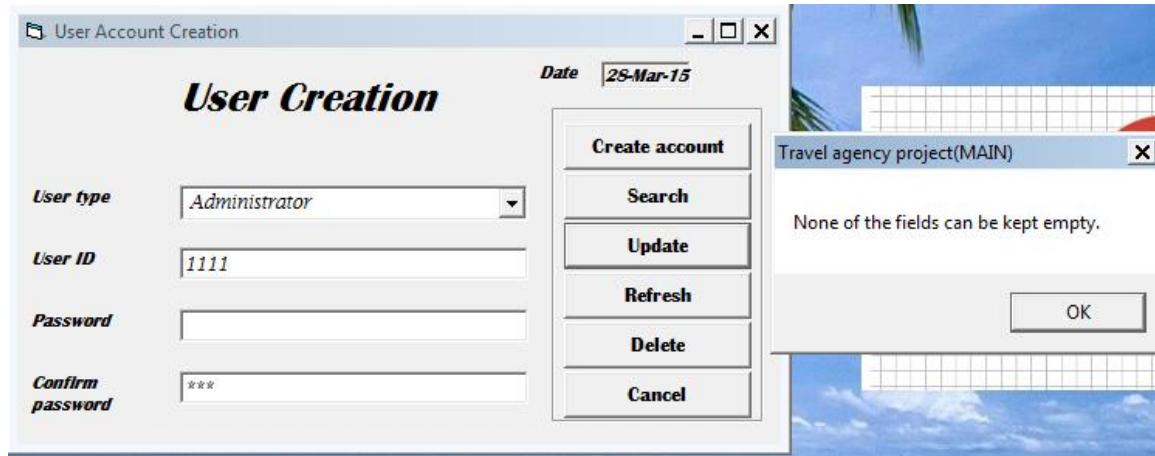
Test #16 Test to see if the required information appears if all the data is provided correctly.

Test data used	User type: Administrator , User ID: 1111
Type of data	Normal
Outcome	Error message shown saying there is no User ID for the particular User type in the database.



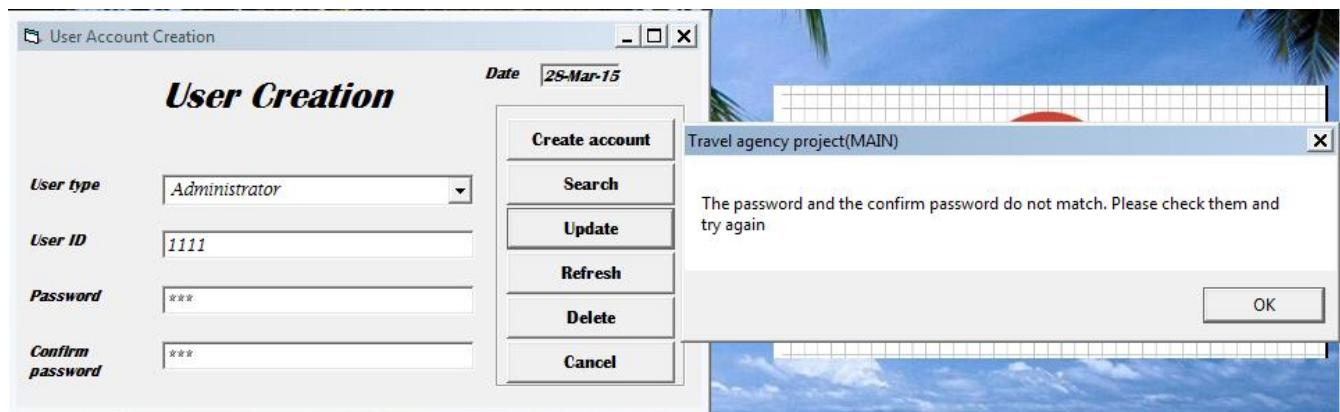
Test #17 Testing to check if error message shown if User type, User ID, password or confirm password not provided during update.

Test data used	User type: Administrator, User ID: 1111, Password: , Confirm Password: 111
Type of data	Normal
Outcome	Error message shown saying none of the fields can be left empty.



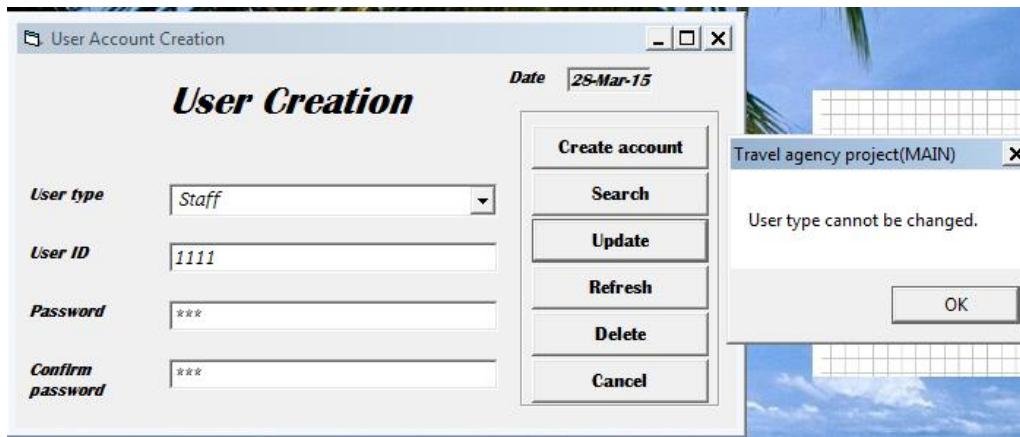
Test #18 Testing to check if error message is shown if password and confirm password don't match on pressing update button.

Test data used	User type: Administrator, User ID: 1111, Password: 112 , Confirm Password: 111
Type of data	Normal
Outcome	Error message shown saying password and confirm password do not match.



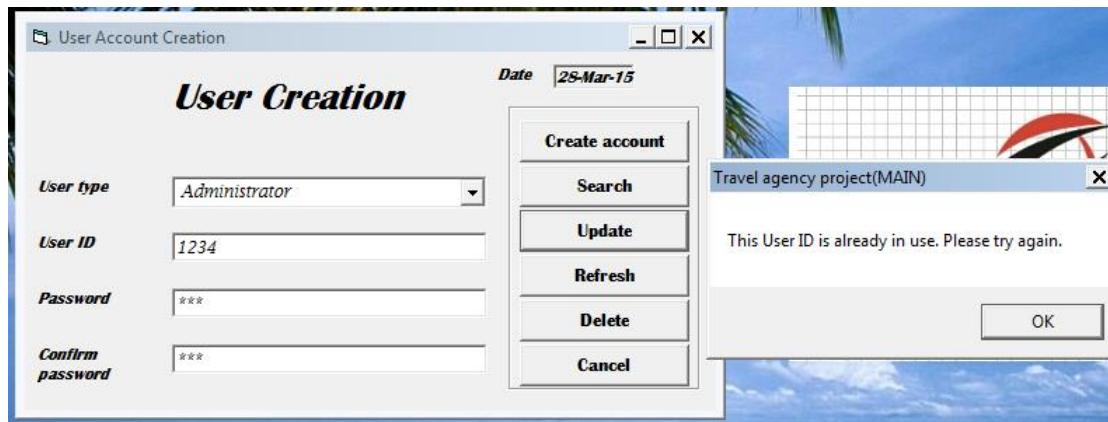
Test #19 Testing to check if error message is shown if User type is changed and update button pressed.

Test data used	User type: Staff, User ID: 1111, Password: 111, Confirm Password: 111
Type of data	Normal
Outcome	Error message shown saying User type cannot be changed.



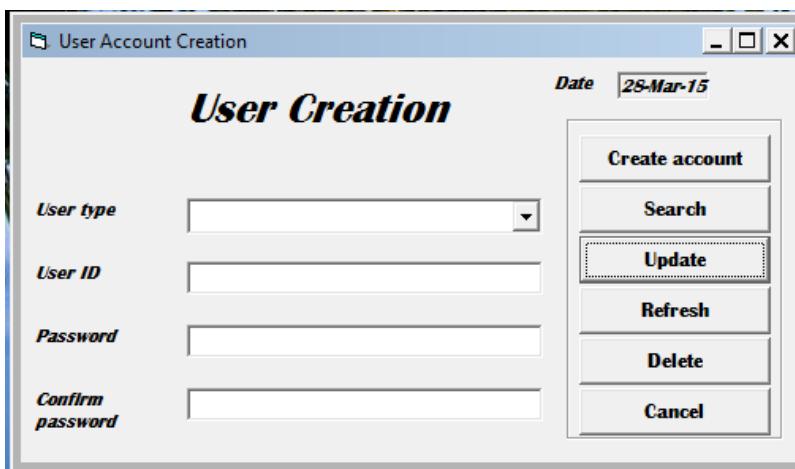
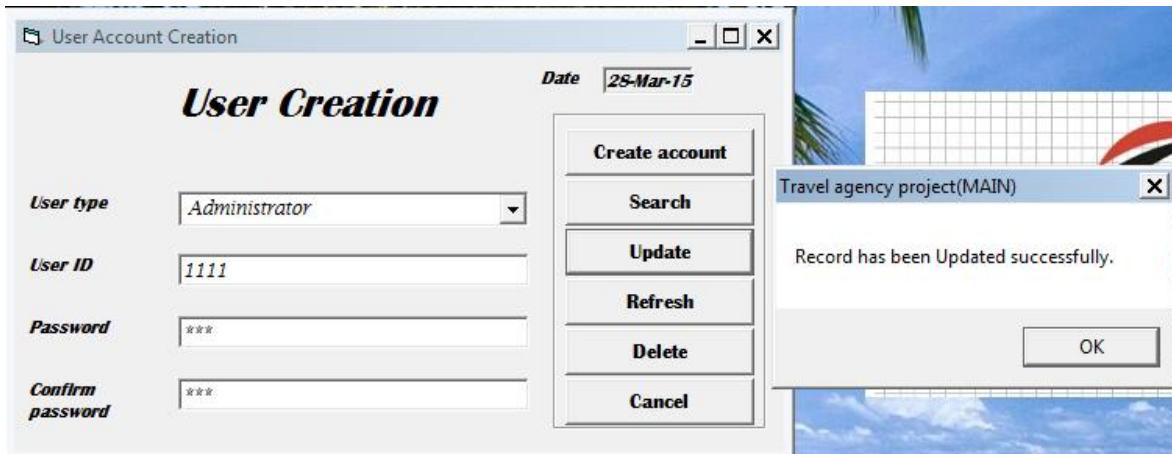
Test #20 Testing to check if error message shown if User ID is changed and the new User ID is already present in the database when update button is pressed.

Test data used	User type: Administrator, User ID: 1234, Password: 111, Confirm Password: 111
Type of data	Normal
Outcome	Error message shown saying User ID is already in use.



Test #21 Testing to check if confirmation is shown when a record is updated, all the fields are emptied after and the information on the database record is changed in the database.

Test data used	User type: Administrator, User ID: 1111, Password: 111, Confirm Password: 111
Type of data	Normal
Outcome	Confirmation message shown saying the record has been updated.



The database before the record was updated.

The screenshot shows the Microsoft Access 'Datasheet' view of the 'User_login' table. The table has columns: User_type, User_ID, and Password. The data is as follows:

User_type	User_ID	Password
Administrator	1111	1321
Administrator	1234	123

An arrow points from the bottom of the table to a box containing the text "The database before the record was updated."

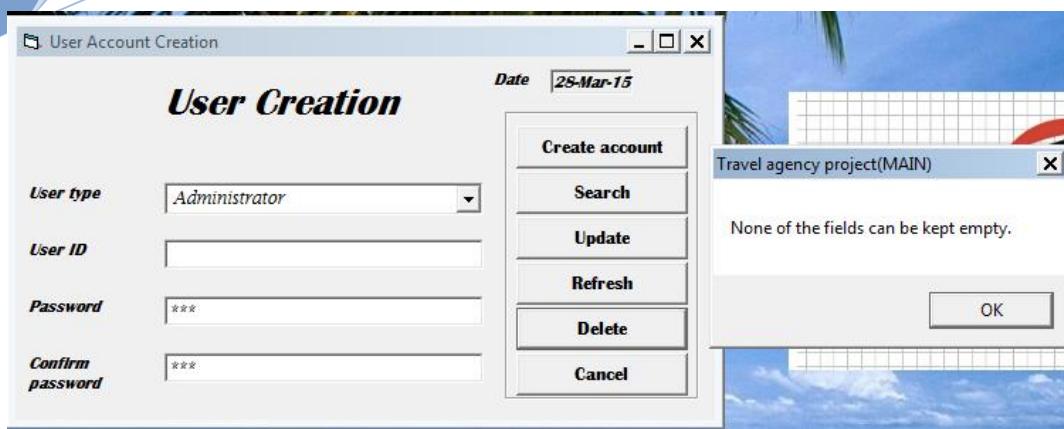
The database after the record was updated.

Test #22 Testing to check if all the fields are emptied on pressing the Refresh button.

Test data used	User type: Administrator, User ID: 1111, Password: 111, Confirm Password:
Type of data	Normal
Outcome	All the fields are emptied.

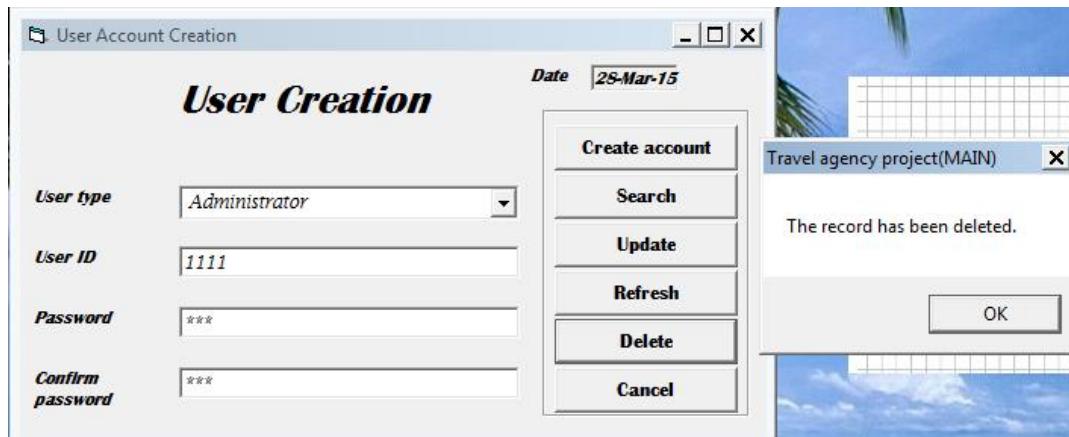
Test #23 Testing to check if error message is shown if User type, User ID, Password or Confirm Password box is left empty on pressing the Delete button.

Test data used	User type: Administrator , User ID: , Password: 111 , Confirm Password:
Type of data	Normal
Outcome	Error message shown saying none of the fields can be left empty.



Test #24 Testing to check if confirmation message shown on the deletion of a record, the fields in the form are emptied and the database is updated on the deletion of a record.

Test data used	User type: Administrator , User ID: 1111 , Password: 111 , Confirm Password:
Type of data	Normal
Outcome	Confirmation message shown saying the record has been deleted.



The screenshot shows the Microsoft Access ribbon and a table named 'User_login' in the datasheet view. The table has columns: User_type, User_ID, and Password. Two rows are visible: one for 'Administrator' with User_ID 1111 and Password 111, and another for 'Administrator' with User_ID 1234 and Password 123. An arrow points from the bottom left towards the table, and a callout box states: 'Database before record has been deleted.'

User_type	User_ID	Password
Administrator	1111	111
Administrator	1234	123

Database before record has been deleted.

Travel Agency Management System

Database after record has been deleted.

User Account Creation

User Creation

Date 25-Mar-15

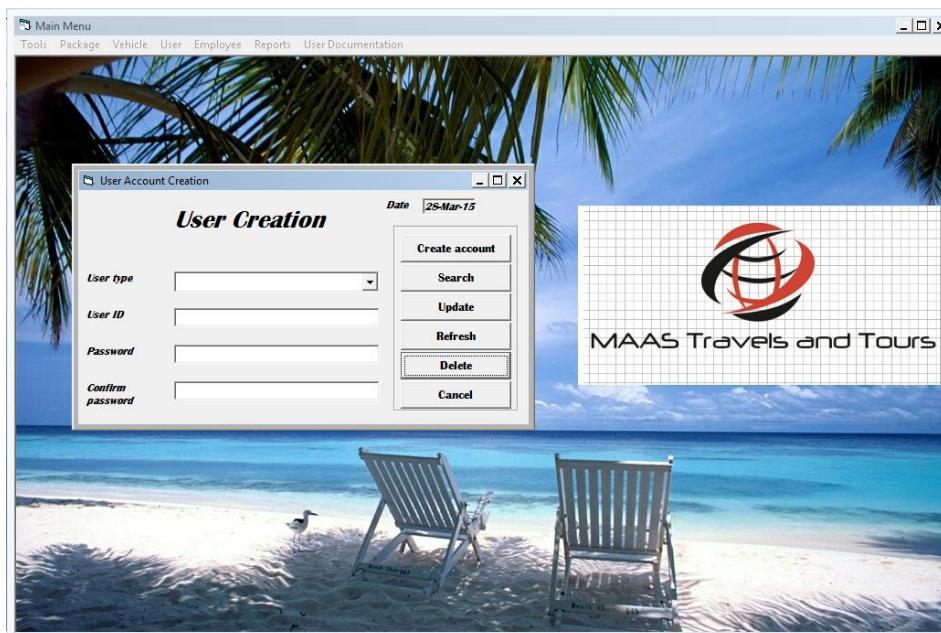
User type	<input type="text"/>
User ID	<input type="text"/>
Password	<input type="text"/>
Confirm password	<input type="text"/>

Create account
 Search
 Update
 Refresh
 Delete
 Cancel

The fields are automatically emptied after deletion of a record.

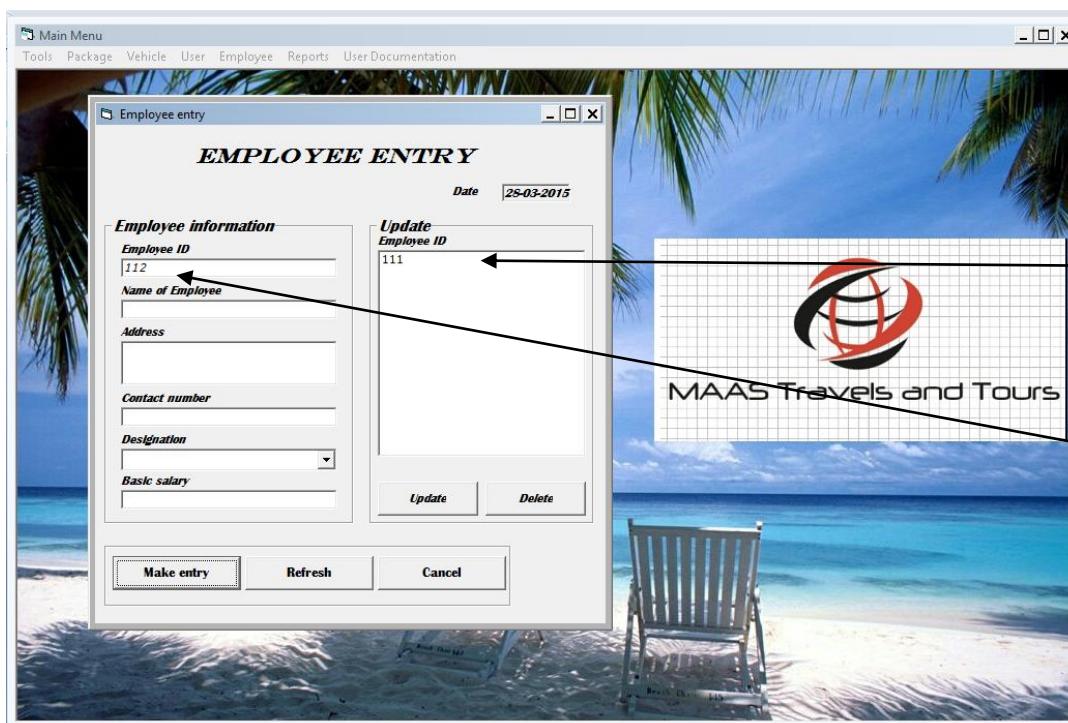
Test #25 Testing to check if form is closed on pressing cancel button.

Test data used	User type: , User ID: , Password: , Confirm Password:
Type of data	Normal
Outcome	The form closes.



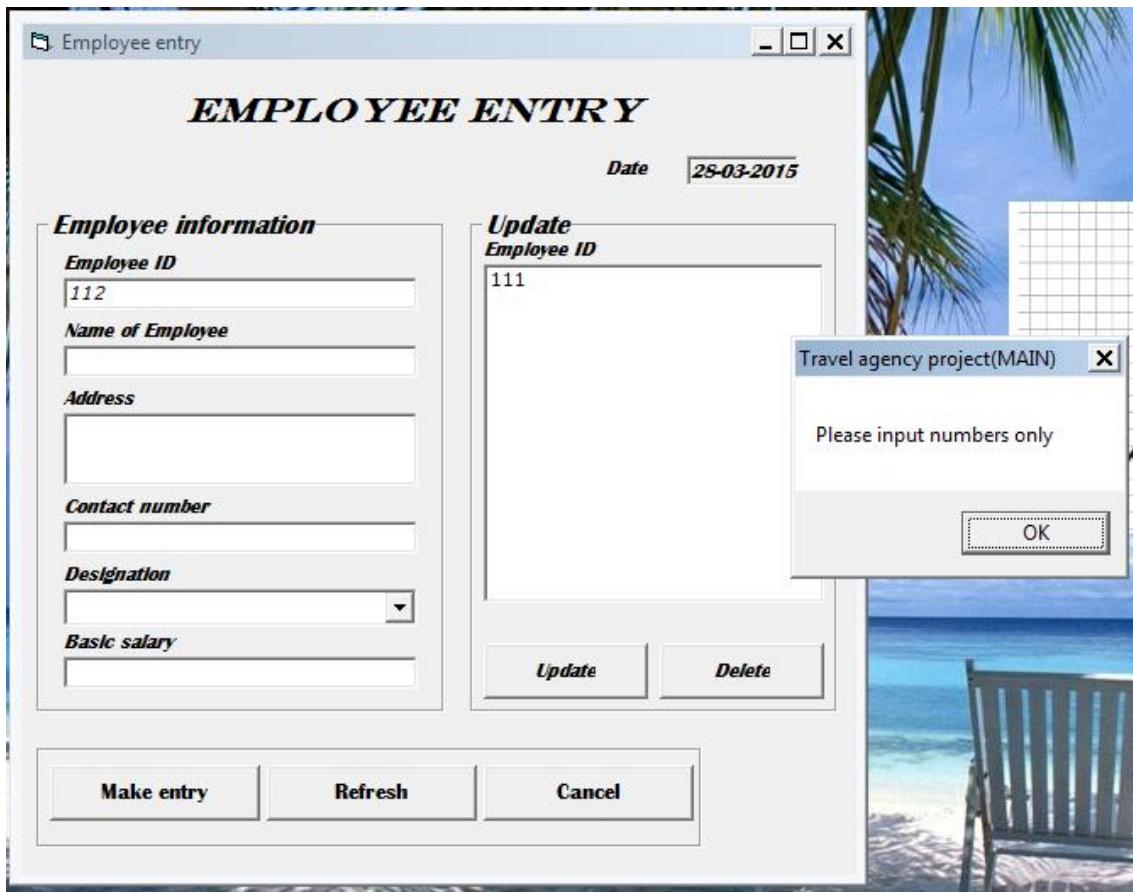
Test #26 Testing to check if Employee Entry form appears on selection of Employee Entry form from the Main Menu and the Employee ID list is loaded automatically.

Test data used	Selecting Employee Entry form from Main Menu.
Type of data	
Outcome	Employee Entry form is loaded and Employee ID list shows a number of Employee IDs.



Test #27 Test to check if error message is shown if wrong character type, other than numbers, is entered in the Contact number textbox.

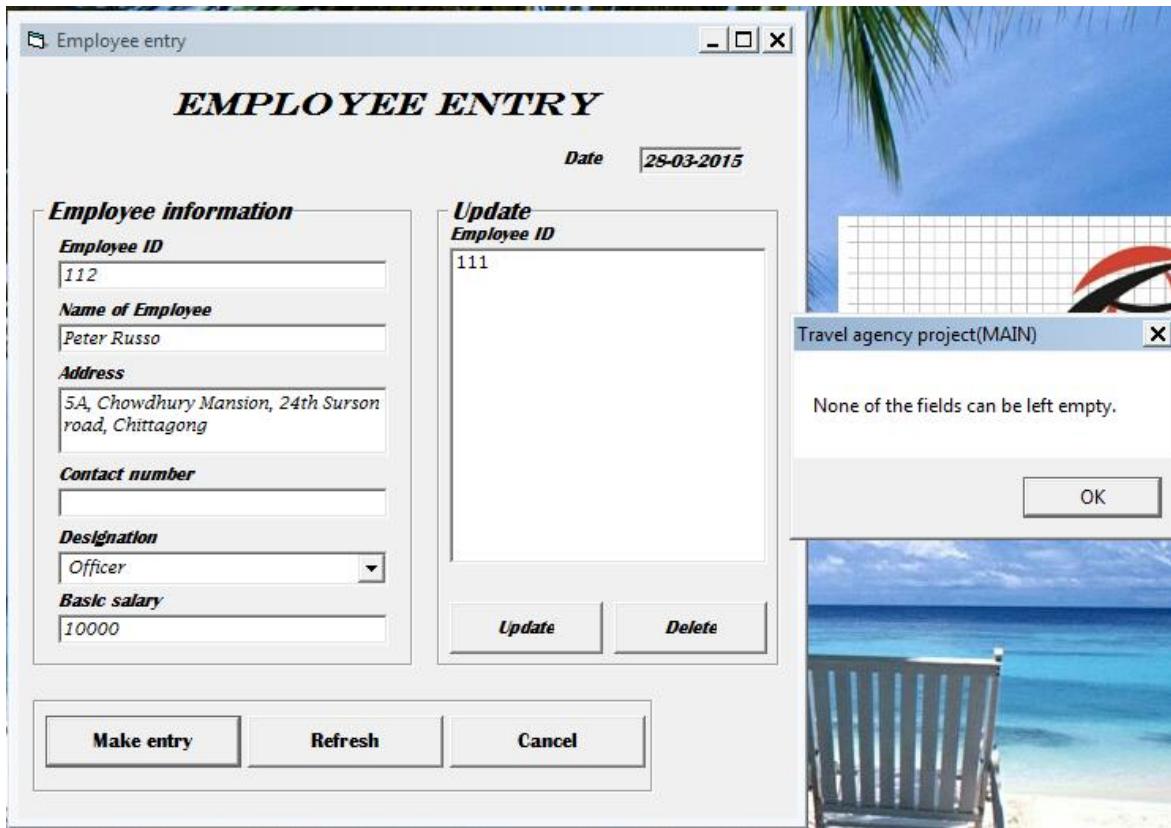
Test data used	Contact number: a
Type of data	
Outcome	Error message shown asking for Numerical data.



As the coding is done in the keypress event, the message is shown before the letter appears.

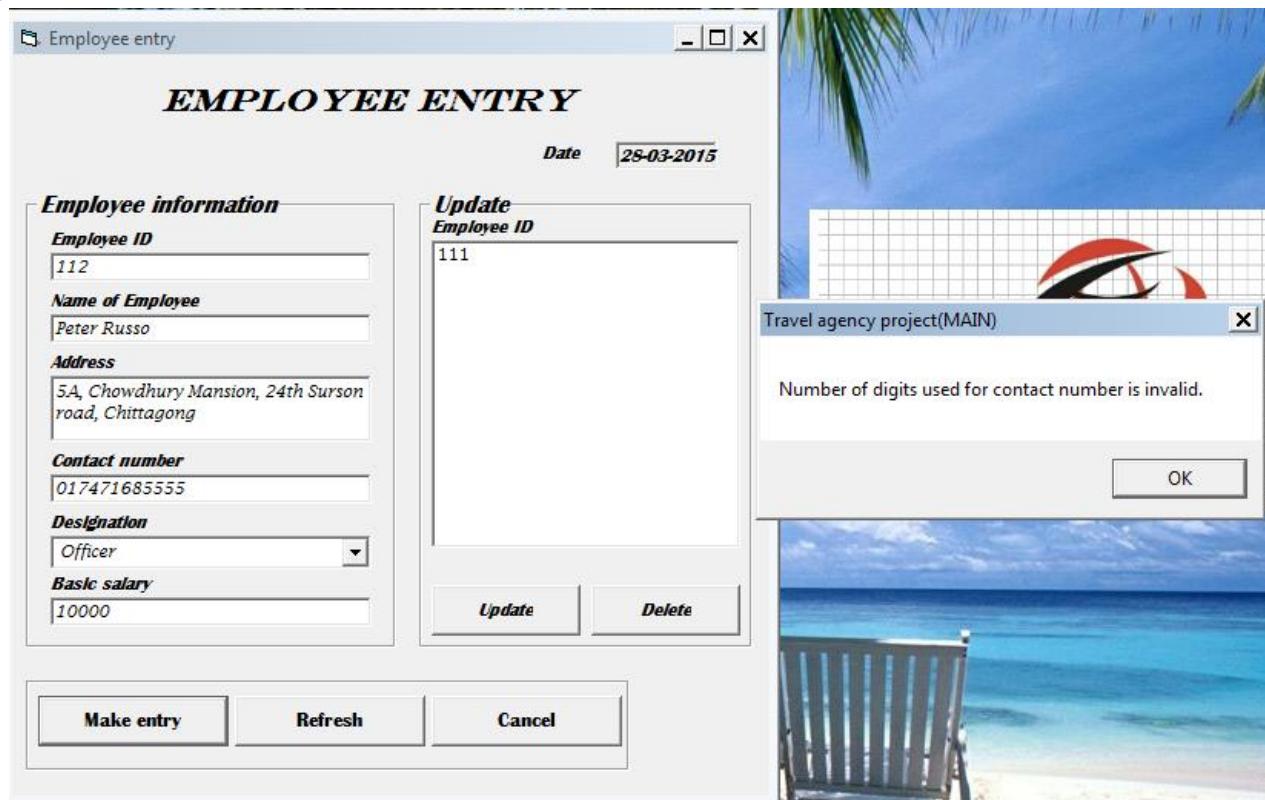
Test #28 Testing to check if error message is shown if any of the fields are left empty on pressing the Make Entry button.

Test data used	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: ,Designation: Officer, Basic salary: \$10000
Type of data	Normal
Outcome	Error message shown saying none of the fields can be left empty.



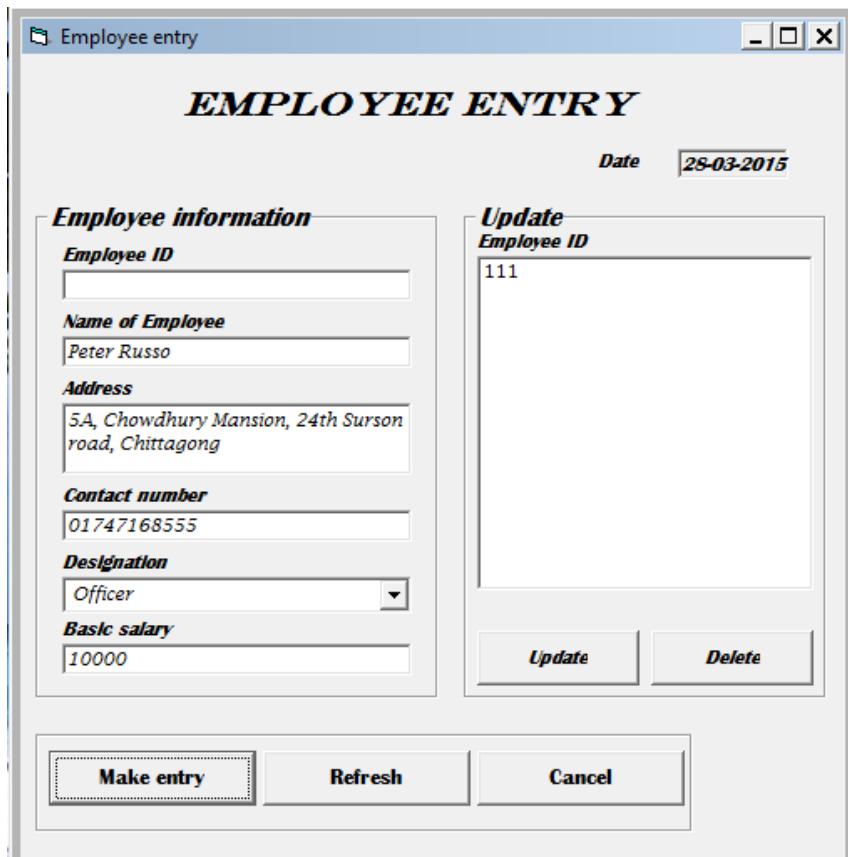
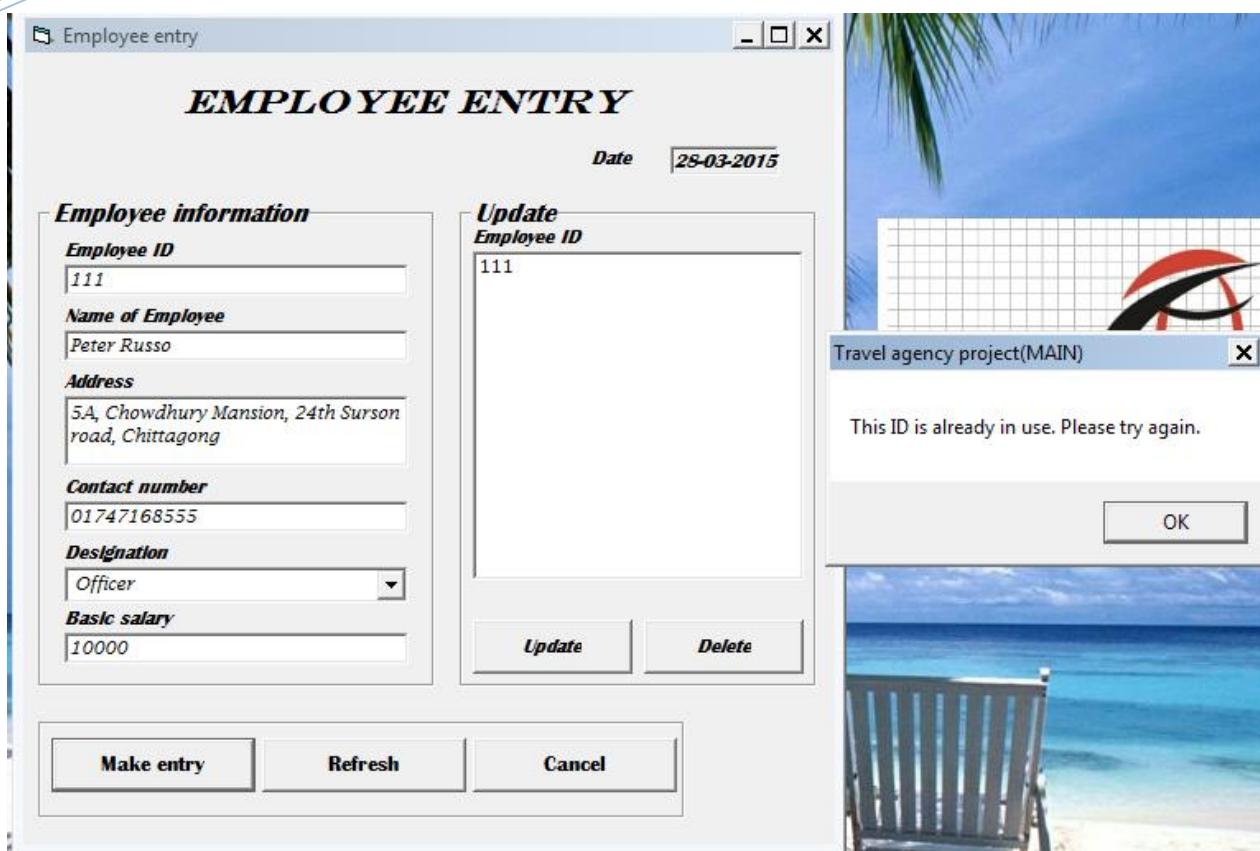
Test #29 Test to check if error message is shown if number of digits provided for Contact number is invalid.

Test data used	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 017471685555, Designation: Officer, Basic salary: \$10000
Type of data	
Outcome	Error message produced saying number of digits provided for contact number is invalid.



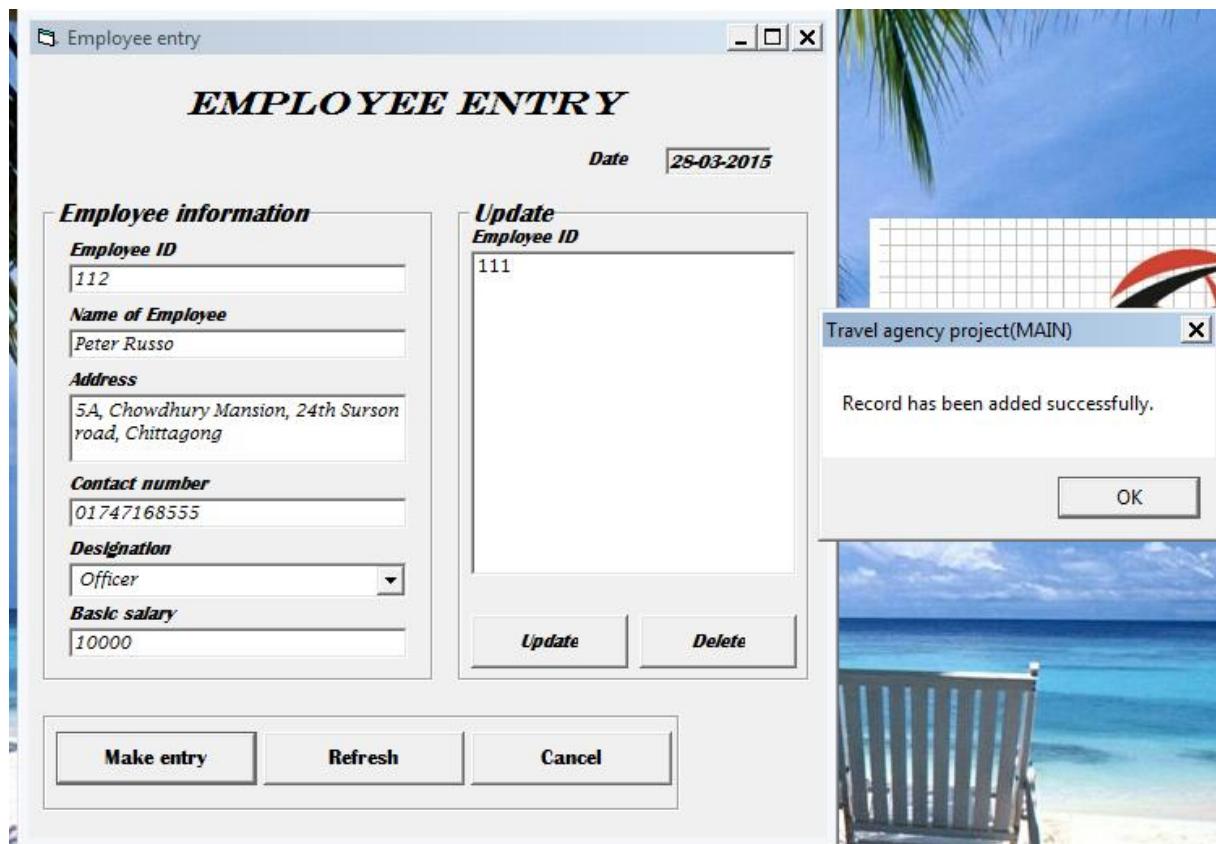
Test #30 Testing to check if error message shown if Employee ID given is already provided present in the database and Employee ID box is automatically cleared, on pressing Make Entry button.

Test data used	Employee ID: 111, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 017471685555, Designation: Officer, Basic salary: \$10000
Type of data	Normal
Outcome	Error message produced saying the employee ID provided is already present in the database.



Test #31 Testing to make sure confirmation message is shown on addition of the new record to notify the User that the record is added successfully and the Employee ID list and the database table are updated and the fields in the form are emptied automatically on pressing Make Entry button if all the information provided is valid. The Employee ID in the Employee ID textbox is also updated for next transaction.

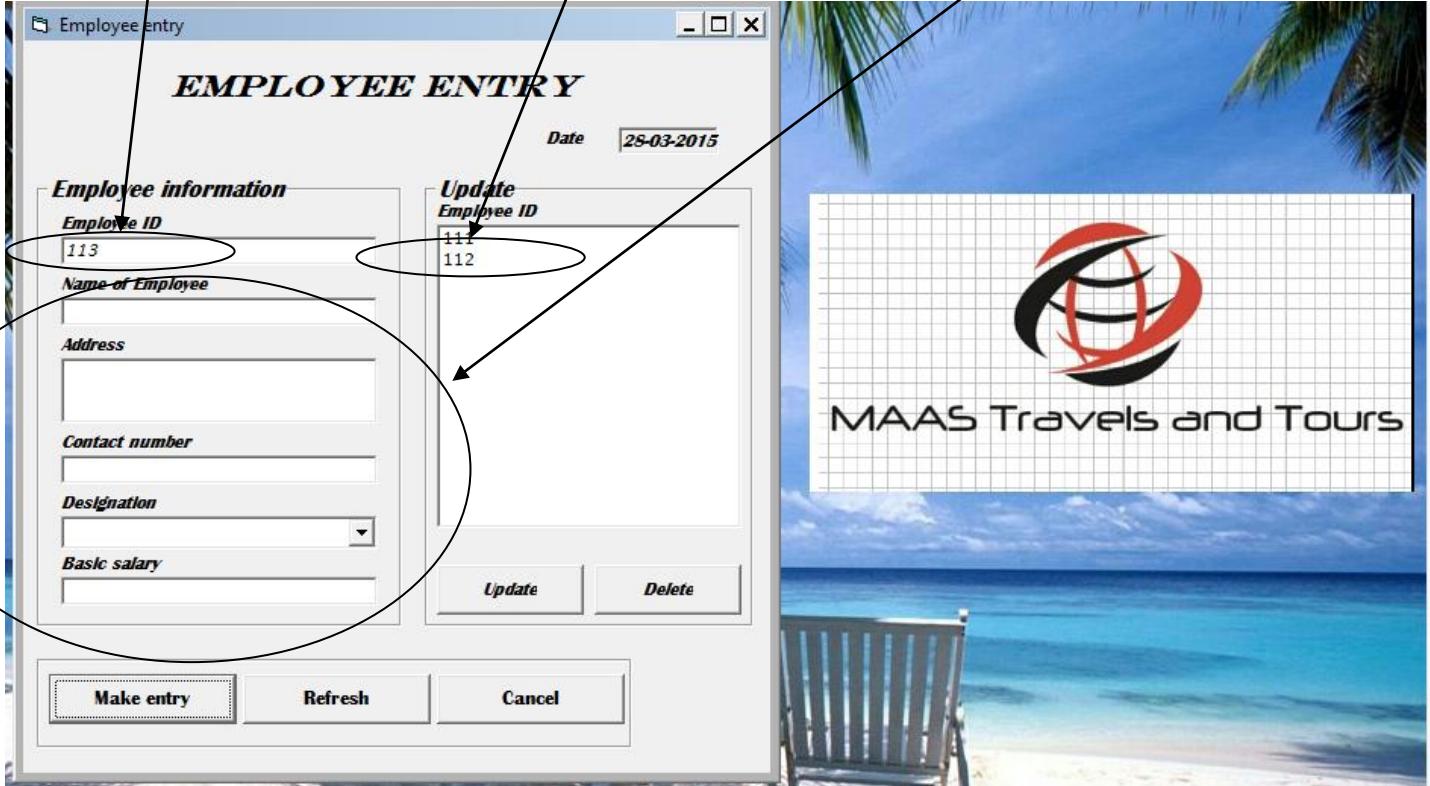
Test data used	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 01747168555, Designation: Officer, Basic salary: \$10000
Type of data	Normal
Outcome	<p>Confirmation message shown to notify the User that the record has been added successfully.</p> <p>The fields in the form are emptied.</p> <p>New Employee ID is added to the list.</p> <p>New record is added to the database.</p> <p>The Employee ID in the textbox is updated.</p>



Employee ID textbox is updated.

New Employee ID is added to the list.

All the other fields are emptied.

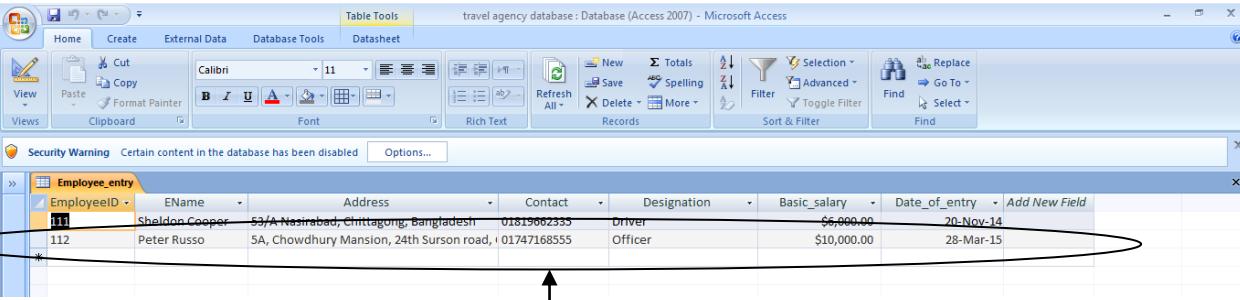


The screenshot shows the Microsoft Access 'travel agency database : Database (Access 2007) - Microsoft Access' window. The 'Employee_entry' table is open in Datasheet view. The table has columns: EmployeeID, EName, Address, Contact, Designation, Basic_salary, and Date_of_entry. One row is selected, showing the data: EmployeeID '113', EName 'Sheldon Cooper', Address '53/A Nasirabad, Chittagong, Bangladesh', Contact '01819662335', Designation 'Driver', Basic_salary '\$6,000.00', and Date_of_entry '20-Nov-14'. A large oval highlights the entire table area.

EmployeeID	EName	Address	Contact	Designation	Basic_salary	Date_of_entry	Add New Field
113	Sheldon Cooper	53/A Nasirabad, Chittagong, Bangladesh	01819662335	Driver	\$6,000.00	20-Nov-14	
*							

Employee Entry database table before new Employee was added.

Travel Agency Management System

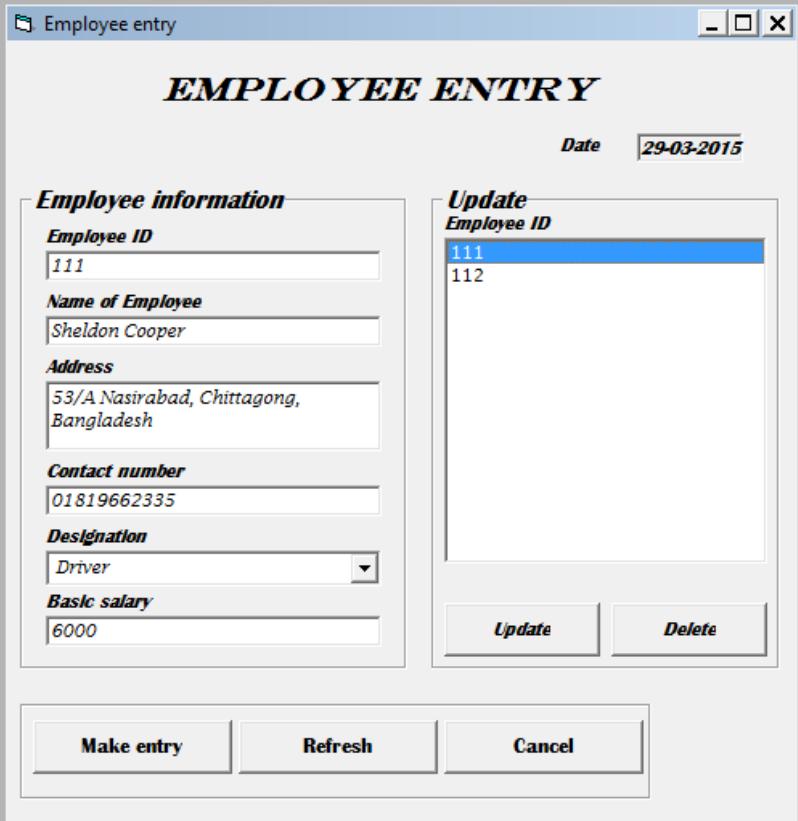


The screenshot shows a Microsoft Access window titled "Employee entry". The table has columns: EmployeeID, EName, Address, Contact, Designation, Basic_salary, and Date_of_entry. A new row is added with EmployeeID 112, EName "Peter Russo", Address "5A, Chowdhury Mansion, 24th Surson road, I", Contact "01747168555", Designation "Officer", Basic_salary "\$10,000.00", and Date_of_entry "28-Mar-15". A circled arrow points from the table back to a callout box below it.

Employee Entry database table after new Employee was added.

Test #32 Testing to check if all the fields are filled automatically on selection of an Employee ID from the Employee ID list

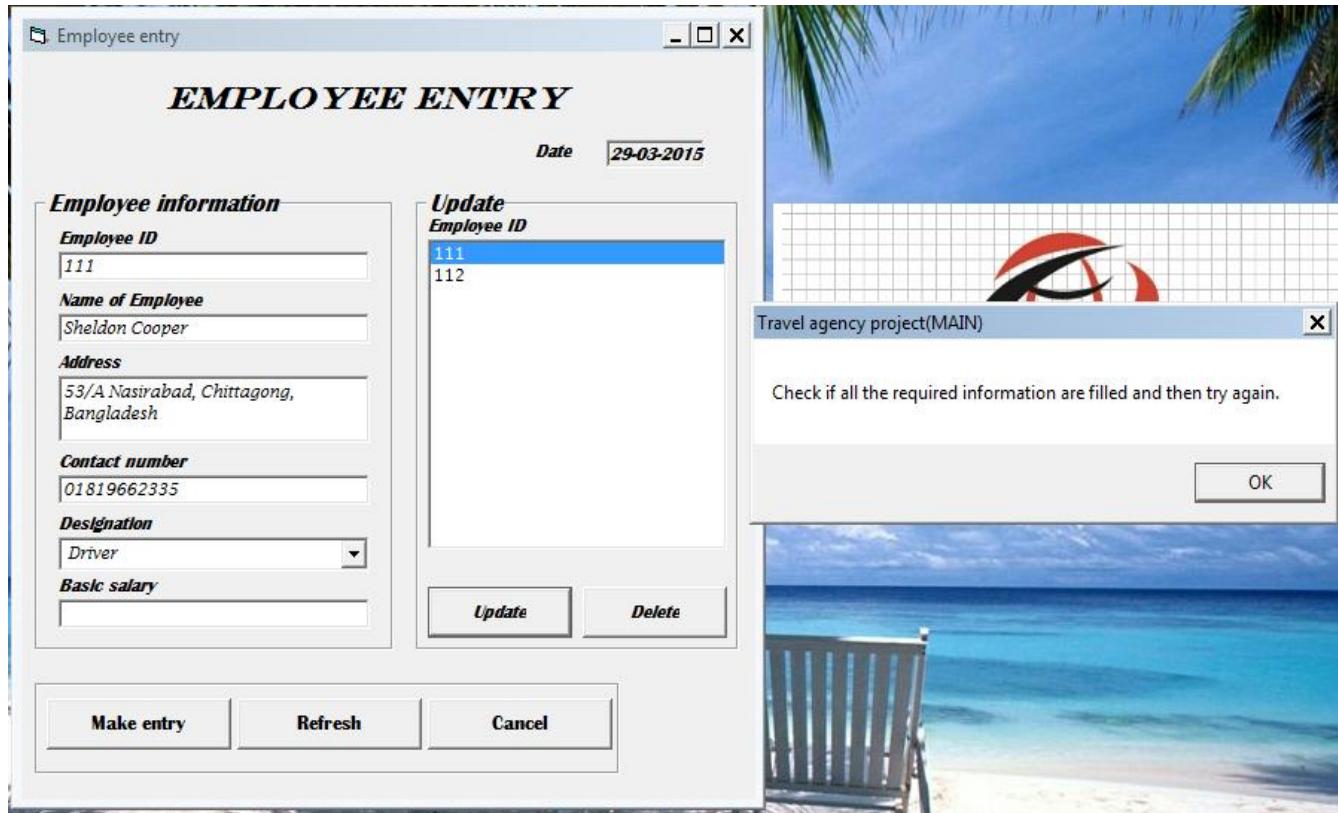
Test data used	Select an Employee ID from the Employee ID list.
Type of data	
Outcome	All the boxes are filled



The application window is titled "Employee entry" and contains the title "EMPLOYEE ENTRY". The date "29-03-2015" is displayed. On the left, there is a form titled "Employee information" with fields for Employee ID (111), Name of Employee (Sheldon Cooper), Address (53/A Nasirabad, Chittagong, Bangladesh), Contact number (01819662335), Designation (Driver), and Basic salary (6000). On the right, there is a "Update Employee ID" list box containing "111" and "112", with "111" selected. At the bottom, there are buttons for "Make entry", "Refresh", "Cancel", "Update", and "Delete".

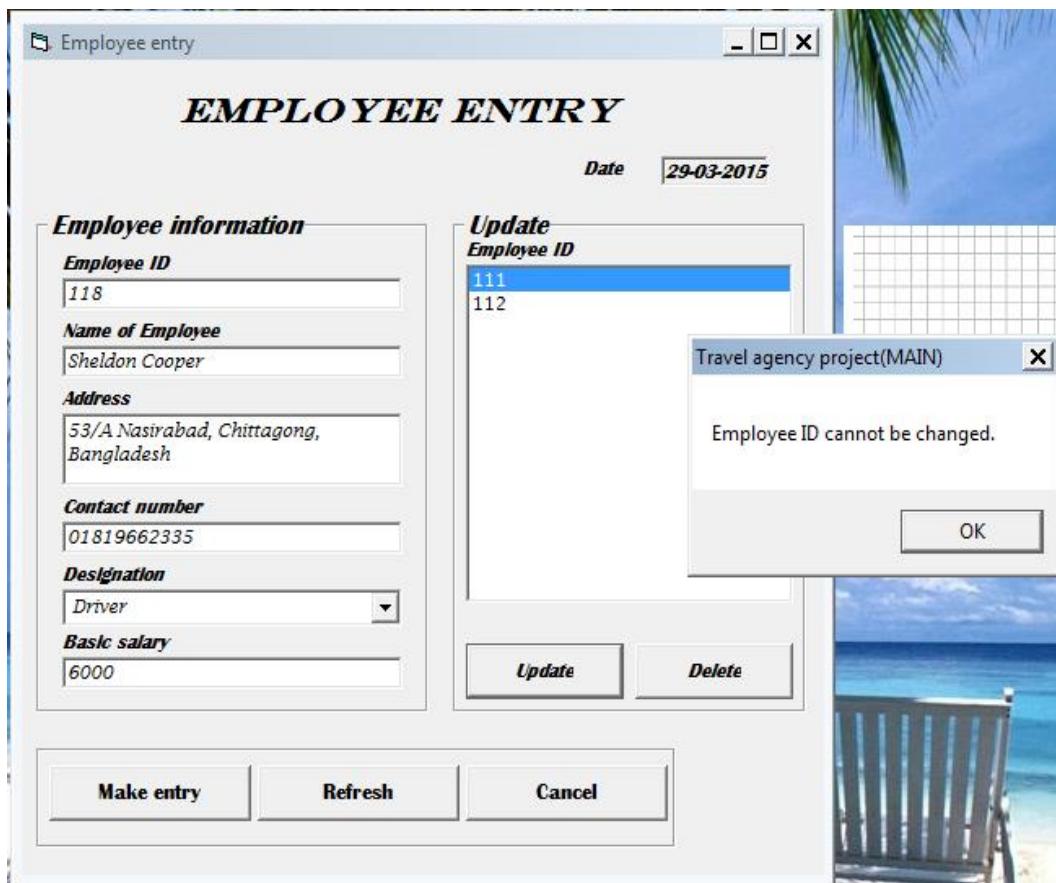
Test #33 Testing to check if error message shown if Update button is clicked with any of the forms left empty.

Test data used	Leaving basic salary empty and clicking Update button.
Type of data	
Outcome	Error message saying none of the fields can be left empty



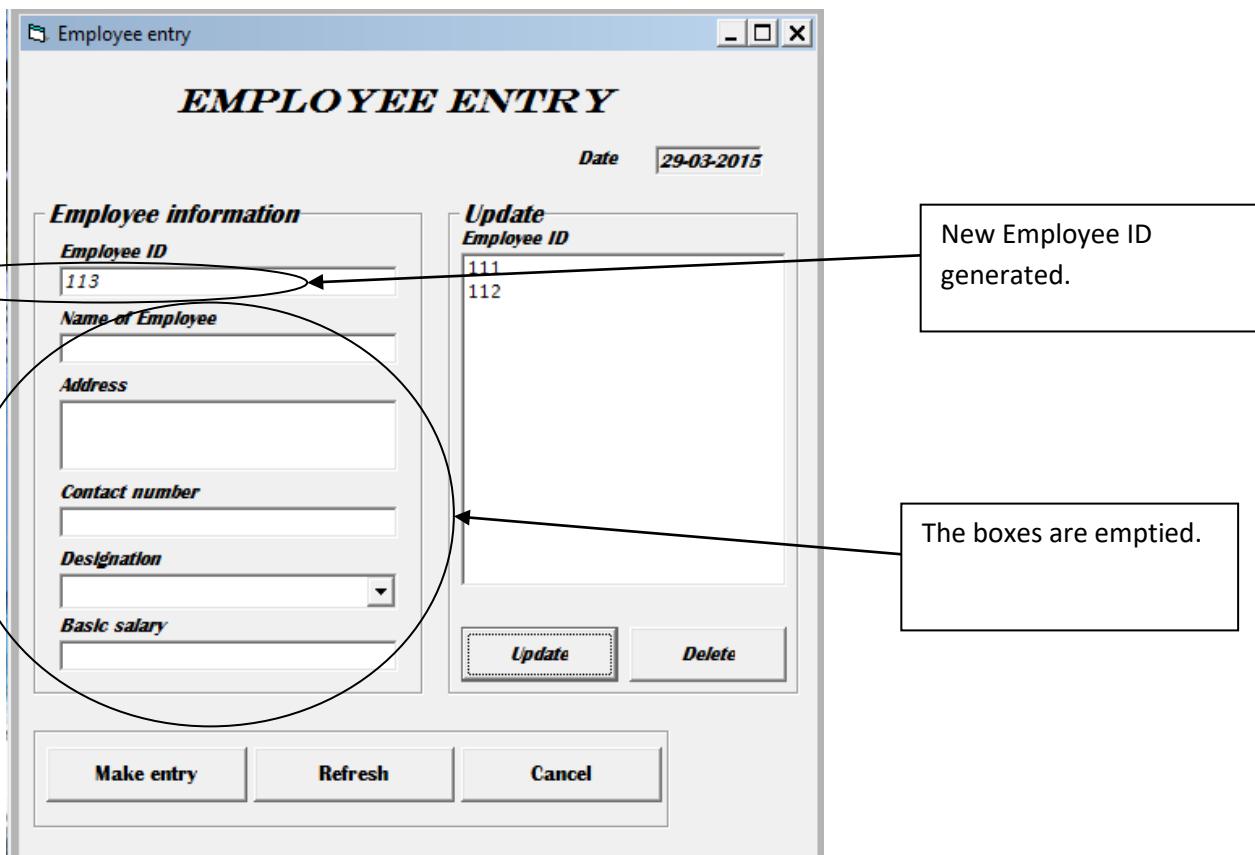
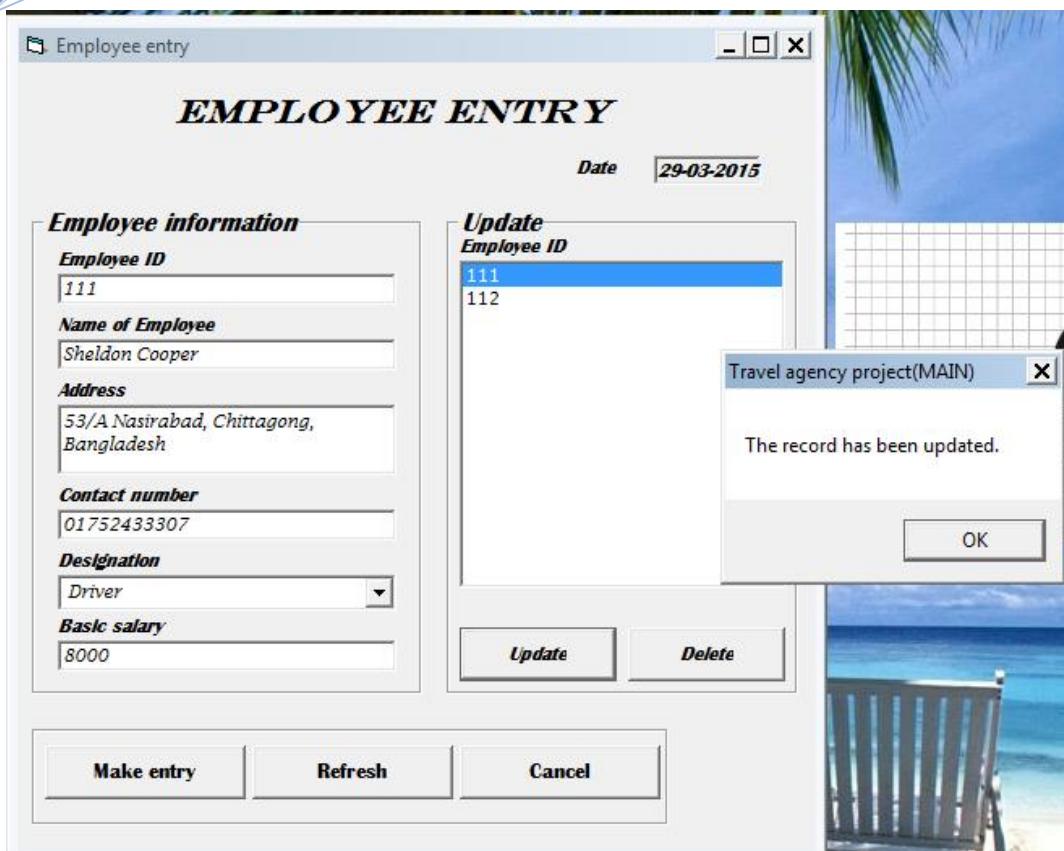
Test #34 Testing if error message shown if Employee ID is changed on pressing Update button.

Test data used	Employee ID: 118
Type of data	Normal
Outcome	Error message saying Employee ID cannot be changed.



Test #35 Testing if confirmation message is provided on updating a record, the record is updated in Employee Entry database table, the fields on the form are emptied afterwards, Employee ID list is updated and a new Employee ID is generated in the Employee ID box on pressing Update button, provided all valid information is provided and none of the fields are left empty.

Test data used	Basic Salary: \$8000, Contact number: 01752433307
Type of data	Normal
Outcome	<p>Confirmation message on Update of the record.</p> <p>The fields on the form are emptied.</p> <p>A new Employee ID is generated in the Employee ID box.</p> <p>The record in the database is updated.</p>



Travel Agency Management System

Database record before the record is updated.

EmployeeID	EName	Address	Contact	Designation	Basic_salary	Date_of_entry
111	Sheldon Cooper	53/A Nasirabad, Chittagong, Bangladesh	01819662335	Driver	\$6,000.00	20-Nov-14
112	Peter Russo	5A, Chowdhury Mansion, 24th Surson road,	01747169555	Officer	\$10,000.00	28-Mar-15

Database record after the record is updated.

EmployeeID	EName	Address	Contact	Designation	Basic_salary	Date_of_entry
111	Sheldon Cooper	53/A Nasirabad, Chittagong, Bangladesh	01752433307	Driver	\$8,000.00	20-Nov-14
112	Peter Russo	5A, Chowdhury Mansion, 24th Surson road,	01747169555	Officer	\$10,000.00	28-Mar-15

Test #35 Testing if confirmation is shown on deletion of a record, the fields on the form are emptied and the record is deleted from the database.

Test data used	Select a record from Employee ID list and click Delete button.
Type of data	Normal
Outcome	<p>Confirmation message for deletion of the record.</p> <p>Fields on the form are emptied.</p> <p>The record is deleted from the database.</p>

Employee entry

EMPLOYEE ENTRY

Date **29-03-2015**

Employee information

Employee ID
111

Name of Employee
Sheldon Cooper

Address
53/A Nasirabad, Chittagong,
Bangladesh

Contact number
01752433307

Designation
Driver

Basic salary
8000

Update
Employee ID
112

Travel agency project(MAIN)

The Record has been deleted.

OK

Update **Delete**

Make entry **Refresh** **Cancel**

Employee entry

EMPLOYEE ENTRY

Date **29-03-2015**

Employee information

Employee ID
113

Name of Employee

Address

Contact number

Designation

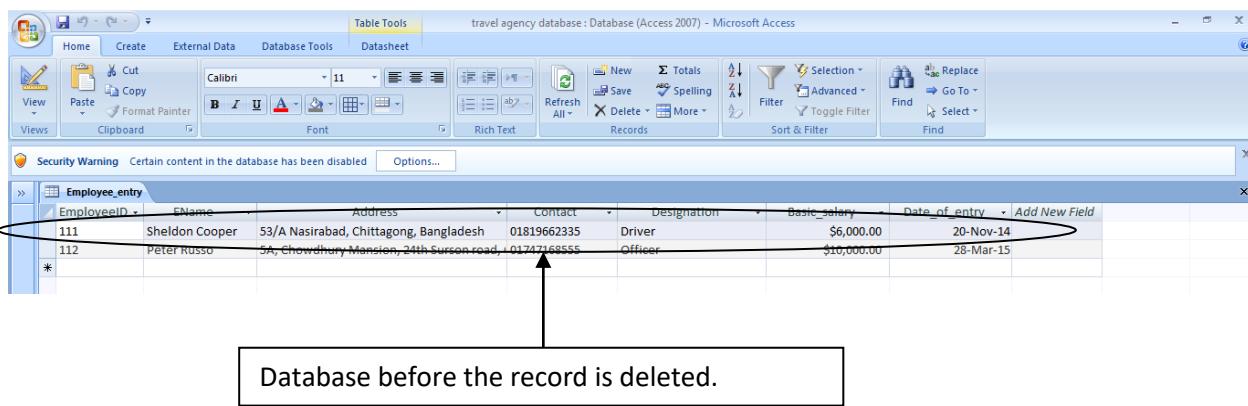
Basic salary

Update
Employee ID
112

Update **Delete**

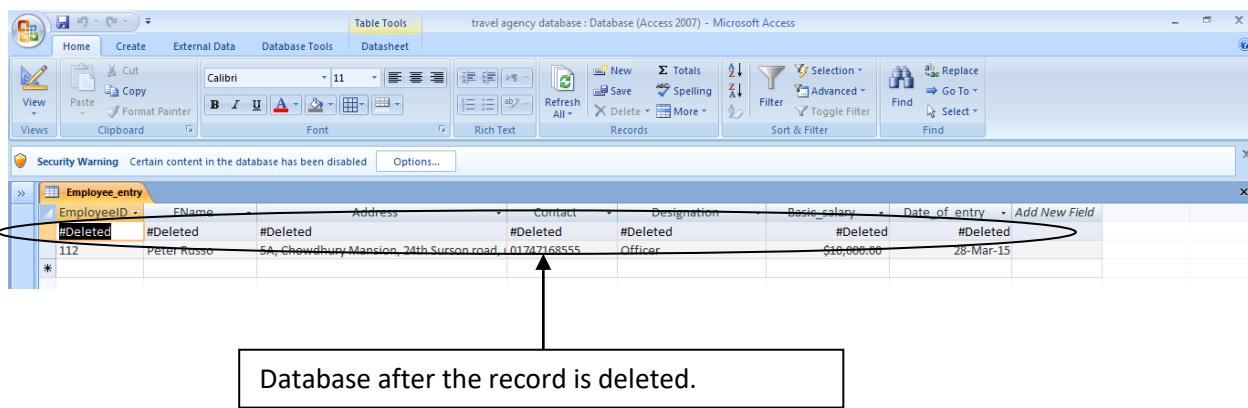
Make entry **Refresh** **Cancel**

Travel Agency Management System



Database before the record is deleted.

EmployeeID	FName	Address	Contact	Designation	Basic_salary	Date_of_entry	Add New Field
111	Sheldon Cooper	S3/A Nasirabad, Chittagong, Bangladesh	01819662335	Driver	\$6,000.00	20-Nov-14	
112	Peter Russo	5A, Chowdhury Mansion, 24th Surson road,	01747168555	Officer	\$10,000.00	28-Mar-15	
*							



Database after the record is deleted.

EmployeeID	FName	Address	Contact	Designation	Basic_salary	Date_of_entry	Add New Field
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	
112	Peter Russo	5A, Chowdhury Mansion, 24th Surson road,	01747168555	Officer	\$10,000.00	28-Mar-15	
*							

Test #37 Testing to check all the fields on the form are cleared on pressing the Refresh button and the Employee ID list is refreshed and Employee ID box is showing proper Employee ID for the next record.

Test data used	Employee ID: 112, Name of Employee: Peter Russo, Address: 5A, Chowdhury Mansion, 24 th Surson road, Chittagong, Contact number: 01747168555, Designation: Officer, Basic salary: \$10000 Click Refresh button
Type of data	Normal
Outcome	The fields in the form are emptied. Employee ID list is refreshed. The Employee ID in the textbox is updated.

Test #38 Testing to check if form is closed on clicking Cancel button.

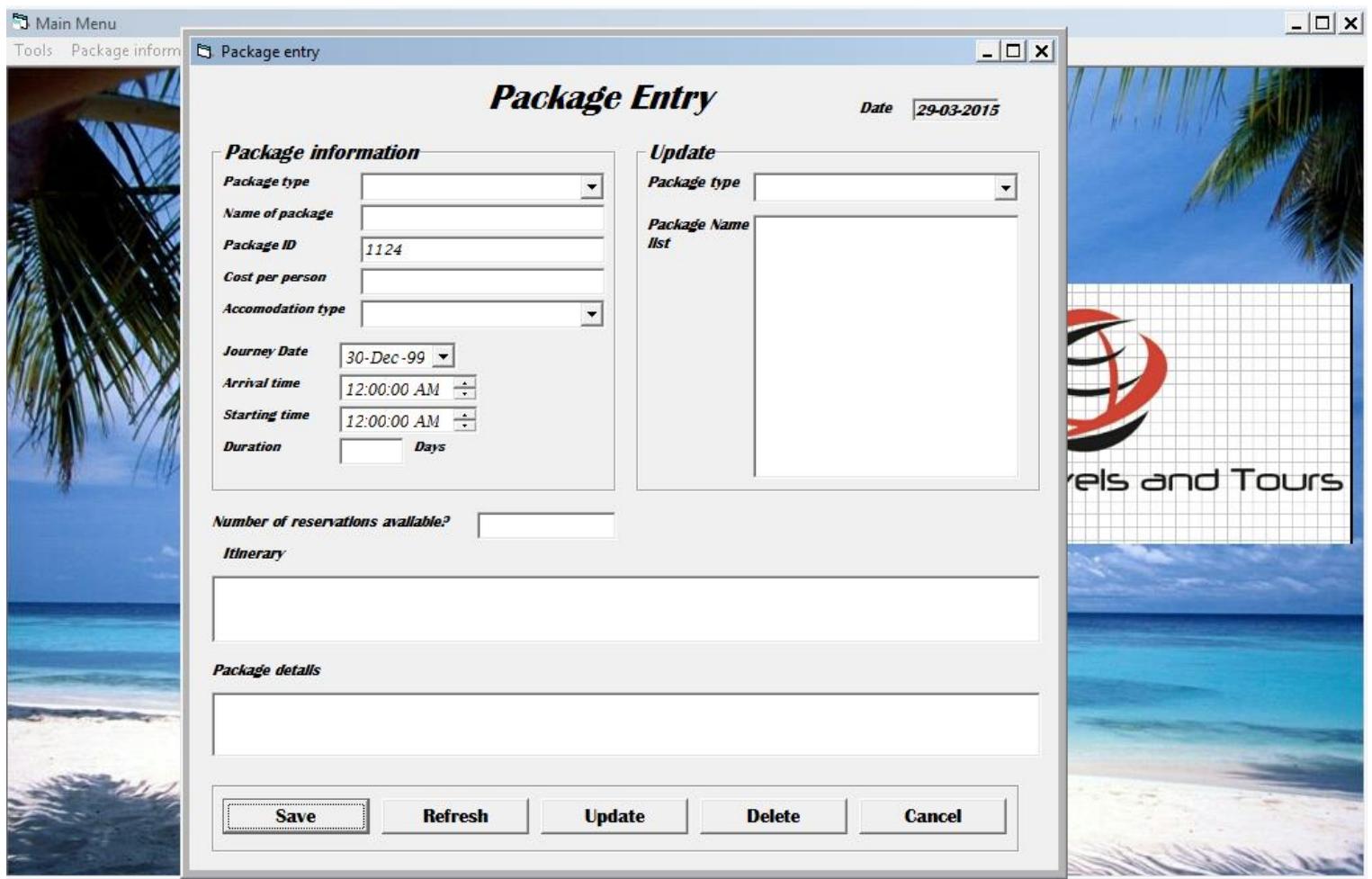
Test data used	Cancel button is clicked
Type of data	Normal
Outcome	The form closes.

Travel Agency Management System



Test #39 Testing to check if Package Entry Form is loaded on selection from the Main Menu form, a new Package ID is generated for new transaction and the Package types are automatically loaded in the update section.

Test data used	Package Entry selected from database.
Type of data	
Outcome	Package Entry form appears. Package ID generated at Package ID textbox. Package types are loaded in the combo box.



The screenshot displays the Travel Agency Management System interface. On the left, a 'Main Menu' window is open, showing a beach scene and a 'Tools' menu item. On the right, a 'Package entry' window is open, also showing a beach scene. The 'Package entry' window has a title bar 'Package entry' and a date field 'Date 29-03-2015'. It contains two main sections: 'Package information' and 'Update'. The 'Package information' section includes fields for Package type (dropdown), Name of package (text box), Package ID (text box containing '1124'), Cost per person (text box), Accommodation type (dropdown), Journey Date (dropdown with '30-Dec-99'), Arrival time (dropdown with '12:00:00 AM'), Starting time (dropdown with '12:00:00 AM'), Duration (dropdown with 'Days'), Number of reservations available? (text box), and Itinerary (text area). The 'Update' section includes a 'Package type' dropdown and a 'Package Name list' text area. At the bottom of the 'Package entry' window are buttons for Save, Refresh, Update, Delete, and Cancel.

Package entry

Package Entry

Date 29-03-2015

Package information

Package type
Name of package
Package ID 1124
Cost per person
Accommodation type
Journey Date 30-Dec-99
Arrival time 12:00:00 AM
Starting time 12:00:00 AM
Duration Days

Number of reservations available?

Itinerary

Package details

Update

Package type
Package Name list Multi-day Single day

Buttons: Save Refresh Update Delete Cancel

Test #40 Testing to check if error message shown if any of the fields are left empty on pressing Save button.

Test data used	All fields are filled except cost per person.
Type of data	
Outcome	Error message shown saying all the information need to be provided.

Test #41 Testing to check if error message shown if anything other than numerical data is provide for Cost per person or Duration.

Test data used	Cost per person: p
Type of data	
Outcome	Error message shown saying only numerical data can be provided.

Test #42 Testing to check if the Package ID provided is already in use. Error message shown if it is.

Test data used	Package ID: 1121
Type of data	Normal
Outcome	Error message shown saying the Package ID provided is already in use.

The screenshot shows a 'Package Entry' window. On the left, under 'Package information', the 'Package ID' field is set to '1121'. On the right, under 'Update', a dropdown menu shows 'Summer Isles' selected. A modal dialog box titled 'Travel agency project(MAIN)' is overlaid, displaying the message 'This Package ID is already in use. Please provide a different Package ID.' with an 'OK' button.

Test #43 Testing to check if confirmation message for addition of data to the database is shown, the fields are automatically cleared after message shown, record is added to database table, Package ID textbox is updated and Package type and Package name list is updated after addition of new record on pressing Save button, provided all the information entered is valid.

Test data used	Package type: Multi-day Package name: River Run through Venice Package ID: 1124 Cost per person: 25000 Accommodation type: Normal Itinerary: 12/2/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/2/16 Sail through Venice 15/2/16 Day of leisure Package details: 4 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options
Type of data	Normal

Jan. 1

Travel Agency Management System

Outcome	<ul style="list-style-type: none"> -Confirmation message saying the data has been added successfully. -The fields on the form are emptied. -Package type and Package name list is updated. -The new record is added to the database.
----------------	--

Before saving data:

Package Entry

Date: 29-03-2015

Package information

Package type	Multi-day
Name of package	River run in Venice
Package ID	1124
Cost per person	25000
Accommodation type	Normal

Journey Date: 12-Feb-16
Arrival time: 10:00:00 AM
Starting time: 11:00:00 AM
Duration: 3 Days

Number of reservations available?: 300

Itinerary

12/2/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/2/16 Sail through Venice

Package details

2 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options

OK

Record has been added successfully

Save Refresh Update Delete Cancel

After saving data:

Package Entry

Date: 29-03-2015

Package information

Package type	
Name of package	
Package ID	1125
Cost per person	
Accommodation type	

Journey Date: 29-Mar-15
Arrival time: 12:00:00 AM
Starting time: 12:00:00 AM
Duration: Days

Number of reservations available?:

Itinerary

Package details

Save Refresh Update Delete Cancel

Package ID updated for next transaction.

travel agency database : Database (Access 2007) - Microsoft Access

Home Create External Data Database Tools Datasheet

Font Rich Text Records Sort & Filter Find

Views Clipboard

Security Warning Certain content in the database has been disabled Options...

Package_entry

Package_type	Package_name	PackageID	Itinerary	Package_details	Cost_per_person	Accommodation_type	Journey_date
Multi-day	Nights in Madagas	1120	12/2/16 Depart Zia Airport - Emirates	Flights, Transfers, Tour host and all options	\$15,000.00	Deluxe	12-Feb-
Multi-day	Sailing in Paris	1123	12/3/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int	\$30,000.00	Deluxe	12-Mar-
Single day	Summer Isles	1121	11:30 AM Depart from Zia Airport from Dhak	Lunch	\$1,200.00	Deluxe	21-Sep-
Multi-day	Wind of Brisbane	1122	12/2/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int	\$5,000.00	Normal	12-Feb-

* **Package Entry database table before transaction.**

Travel Agency Management System

Package Entry database table after transaction.

Package_type	Package_name	PackageID	Itinerary	Package_details	Cost_per_person	Accommodation_type	Journey_date
Multi-day	Nights in Madagascar	1120	12/2/16 Depart Zia Airport - Emirates	4 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$15,000.00	Deluxe	12-Feb-16
Multi-day	River run in Venice	1124	12/2/16 Depart Zia Airport - Emirates	2 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$25,000.00	Normal	12-Feb-16
Multi-day	Sailing in Paris	1123	12/3/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$30,000.00	Deluxe	12-Mar-16
Single day	Summer Isles	1121	11:30 AM Depart from Zia Airport from Dhaka, Bangladesh	Lunch included	\$1,200.00	Deluxe	21-Sep-16
Multi-day	Wind of Brisbane	1122	12/2/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$5,000.00	Normal	12-Feb-16

Test #44 Testing to check if all the fields are emptied on pressing Refresh button, Package ID box is updated and the Package type box is refreshed.

Test data used	Click Refresh button.
Type of data	Normal
Outcome	All the fields are emptied.

Before pressing Refresh button:

After pressing Refresh button:

Test #45 Testing to check if Package name list in Update section is automatically loaded with names on selection of a Package type and all the other information in the form are automatically loaded on selection of a Package name from the list.

Test data used	Selecting a Package type from combo box of Update section. Selecting a Package name from the names that appear on the list.
Type of data	
Outcome	All the fields are automatically filled.

Package Entry

Package information

Package type	Multi-day
Name of package	
Package ID	1125
Cost per person	
Accommodation type	
Journey Date	29-Mar-15
Arrival time	12:00:00 AM
Starting time	12:00:00 AM
Duration	Days

Number of reservations available?

Itinerary

Package details

Update

Package type	Multi-day
Package Name list	Single day

Save Refresh Update Delete Cancel

Package Entry

Package information

Package type	Multi-day
Name of package	
Package ID	1123
Cost per person	
Accommodation type	
Journey Date	29-Mar-15
Arrival time	12:00:00 AM
Starting time	12:00:00 AM
Duration	Days

Number of reservations available?

Itinerary

Package details

Update

Package type	Multi-day
Package Name list	Wind of Brisbane Nights in Madagascar Sailing in Paris River run in Venice

Save Refresh Update Delete Cancel

Package entry

Package Entry Date 29-03-2015

Package information		Update	
Package type	Multi-day	Package type	Multi-day
Name of package	Nights in Madagascar	Package Name	Wind of Brisbane
Package ID	1120	lst	Nights in Madagascar
Cost per person	15000		Sailing in Paris
Accommodation type	Deluxe		River run in Venice
Journey Date	12-Feb-16		
Arrival time	10:00:00 AM		
Starting time	11:00:00 PM		
Duration	3 Days		
Number of reservations available? 500			
Itinerary			
12/2/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/2/16 African Zoo - Outback Africa Dinner show			
Package details			
4 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options			
<input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>			

Test #46 Testing to check if error message shown if Package ID is changed or any of the fields are empty on pressing the Update button.

Test data used	Package ID is changed and Update button is pressed.
Type of data	Normal
Outcome	Error message shown saying the Package ID cannot be changed.

Test #47 Testing to check if confirmation message is shown on Updating a record, all the fields are emptied afterwards, Package type combo box in the update section is updated, Package ID is updated and record is updated in the database table on pressing the Update button, provided all the information entered is valid.

Test data used	Cost per person: 20000
Type of data	Normal
Outcome	<p>Confirmation message saying the record is updated.</p> <p>All the fields are emptied.</p> <p>Package types available are updated.</p> <p>Record is updated in the database.</p>

Jan. 1

Travel Agency Management System

Left Window (Initial State):

- Package information:**
 - Package type: Multi-day
 - Name of package: Nights in Madagascar
 - Package ID: 1120
 - Cost per person: 20000
 - Accommodation type: Deluxe
 - Journey Date: 12-Feb-16
 - Arrival time: 10:00:00 AM
 - Starting time: 11:00:00 PM
 - Duration: 3 Days
- Itinerary:**

```
12/2/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/2/16 African Zoo - Outback Africa Dinner show
```
- Package details:**

4 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options

Right Window (After Update):

- Package information:**
 - Package type: Multi-day
 - Name of package: Wind of Brisbane
 - Package ID: **1125** (highlighted)
 - Cost per person: 20000
 - Accommodation type: Deluxe
 - Journey Date: 29-Mar-15
 - Arrival time: 12:00:00 AM
 - Starting time: 12:00:00 AM
 - Duration: 3 Days
- Itinerary:** (Same as left window)
- Package details:** (Same as left window)

Message Box: Package information has been updated successfully.

New Package ID generated after Update.

Database table before Update:

Package_type	Package_name	PackageID	Itinerary	Package_details	cost_per_person	Accommodation_type	Journey_date
Multi-day	Nights in Madagas	1120	12/2/16 Depart Zia Airport - Emirates	4 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$15,000.00	Deluxe	12-Feb-16
Multi-day	River run in Venici	1124	12/2/16 Depart Zia Airport - Emirates	2 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$20,000.00	Normal	12-Feb-16
Multi-day	Sailing in Paris	1123	12/3/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$30,000.00	Deluxe	12-Mar-16
Single day	Summer Isles	1121	11:30 AM Depart from Zia Airport from Dhaka	Lunch	\$1,200.00	Deluxe	21-Sep-16
Multi-day	Wind of Brisbane	1122	12/2/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$5,000.00	Normal	12-Feb-16

Database table before Update

Database table before Update:

Package_type	Package_name	PackageID	Itinerary	Package_details	cost_per_person	Accommodation_type	Journey_date
Multi-day	Nights in Madagas	1120	12/2/16 Depart Zia Airport - Emirates	4 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$20,000.00	Deluxe	12-Feb-16
Multi-day	River run in Venici	1124	12/2/16 Depart Zia Airport - Emirates	2 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$20,000.00	Normal	12-Feb-16
Multi-day	Sailing in Paris	1123	12/3/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$30,000.00	Deluxe	12-Mar-16
Single day	Summer Isles	1121	11:30 AM Depart from Zia Airport from Dhaka	Lunch	\$1,200.00	Deluxe	21-Sep-16
Multi-day	Wind of Brisbane	1122	12/2/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l Flights, Transfers, Tour host and all options	\$5,000.00	Normal	12-Feb-16

Database table before Update

Test #48 Testing to check if error message shown if any of the fields are left empty on pressing the Delete button.

Test data used	Cost per person is left empty Click Delete button.
Type of data	Normal
Outcome	Error message shown saying all the information need to be provided.

Test #49 Testing to check if Deletion confirmation message is shown after deletion of a record, all the fields on the form are emptied automatically, Package ID is updated and the database table and Package type and Package list combo box are updated on pressing the Delete button.

Test data used	Select a Package type and Package name from the Update section. Click Delete button.
Type of data	Normal
Outcome	-Confirmation message on deletion of record. -All the fields are emptied. -The Package name is removed from the list and Package type is updated.

Package entry

Package Entry

Date 29-03-2015

Package information		Update
Package type	Package type	
Name of package	Package Name	
Package ID	list	
Cost per person		
Accommodation type		
Journey Date	30-Dec-99	
Arrival time	12:00:00 AM	
Starting time	12:00:00 AM	
Duration	Days	
Number of reservations available?		
Itinerary		
Package details		
<input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>		

The record in the former test is used.

The field is empty because all the fields are emptied before the message is shown.

Package entry

Package Entry

Date 29-03-2015

Package information		Update
Package type	Package type	
Name of package	Package Name	
Package ID	1125	
Cost per person		
Accommodation type		
Journey Date	29-Mar-15	
Arrival time	12:00:00 AM	
Starting time	12:00:00 AM	
Duration	Days	
Number of reservations available?		
Itinerary		
Package details		
<input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>		

New Package ID generated.

Travel Agency Management System

Package_type	Package_name	PackageID	Itinerary	Package_details	Cost_per_person	Accommodation_type	Journey_date
Multi-day	Nights In Madagas	1120	12/2/16 Depart Zia Airport - Emirates	4 star hotel for 3 nights with Breakfast, Int'l flight	\$15,000.00	Deluxe	12-Feb-17
Multi-day	River run in Vento	1124	12/3/16 Depart Zia Airport - Emirates	2 star hotel for 3 nights with Breakfast, Int'l flight	\$25,000.00	Normal	12-Feb-17
Multi-day	Sailing in Paris	1123	12/3/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l flight	\$30,000.00	Deluxe	12-Mar-17
Single day	Summer Isles	1121	11:30 AM Depart from Zia Airport from Dhaka, Lunch		\$1,200.00	Deluxe	21-Sep-16
Multi-day	Wind of Brisbane	1122	12/2/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l flight	\$5,000.00	Normal	12-Feb-17

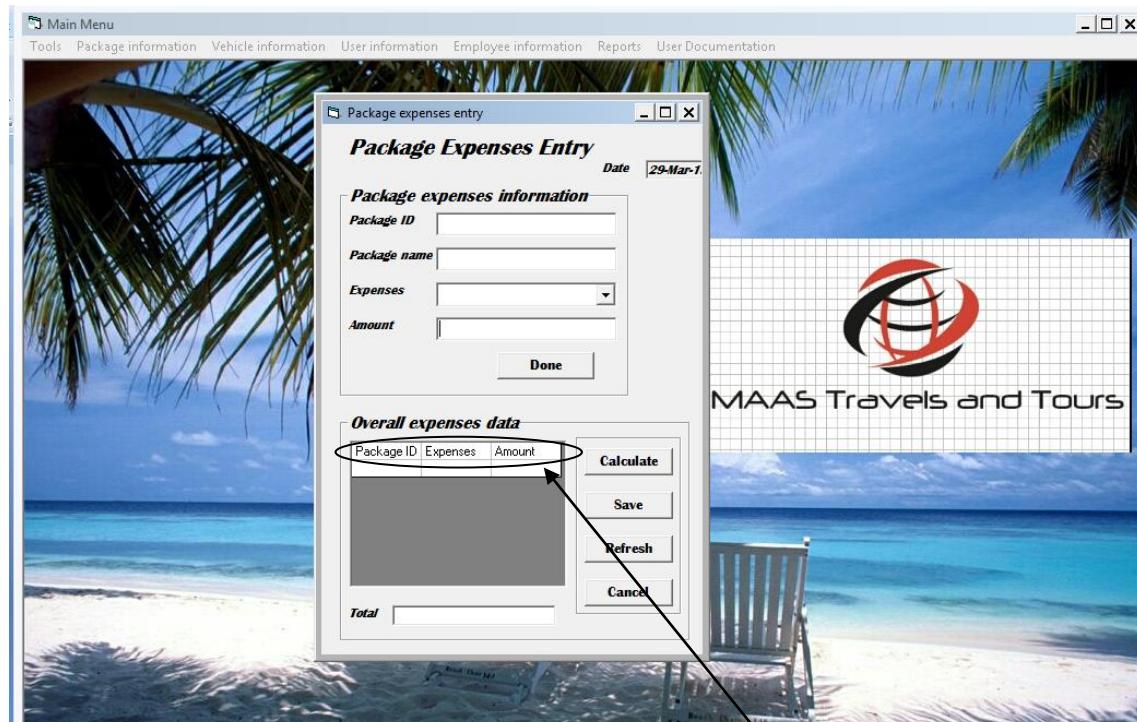
Package Entry database before
Deletion.

Package_type	Package_name	PackageID	Itinerary	Package_details	Cost_per_person	Accommodation_type	Journey_date
Multi-day	River run in Vento	1124	12/2/16 Depart Zia Airport - Emirates	2 star hotel for 3 nights with Breakfast, Int'l flight	\$25,000.00	Normal	12-Feb-17
Multi-day	Sailing in Paris	1123	12/3/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l flight	\$30,000.00	Deluxe	12-Mar-17
Single day	Summer Isles	1121	11:30 AM Depart from Zia Airport from Dhaka, Lunch		\$1,200.00	Deluxe	21-Sep-16
Multi-day	Wind of Brisbane	1122	12/2/16 Depart Zia Airport - Emirates	3 star hotel for 3 nights with Breakfast, Int'l flight	\$5,000.00	Normal	12-Feb-17

Package Entry database after Deletion.

Test #50 Testing to check if Package Expenses Entry form opens on selection from Main Menu and names are automatically assigned to the columns of “Overall expenses data” section.

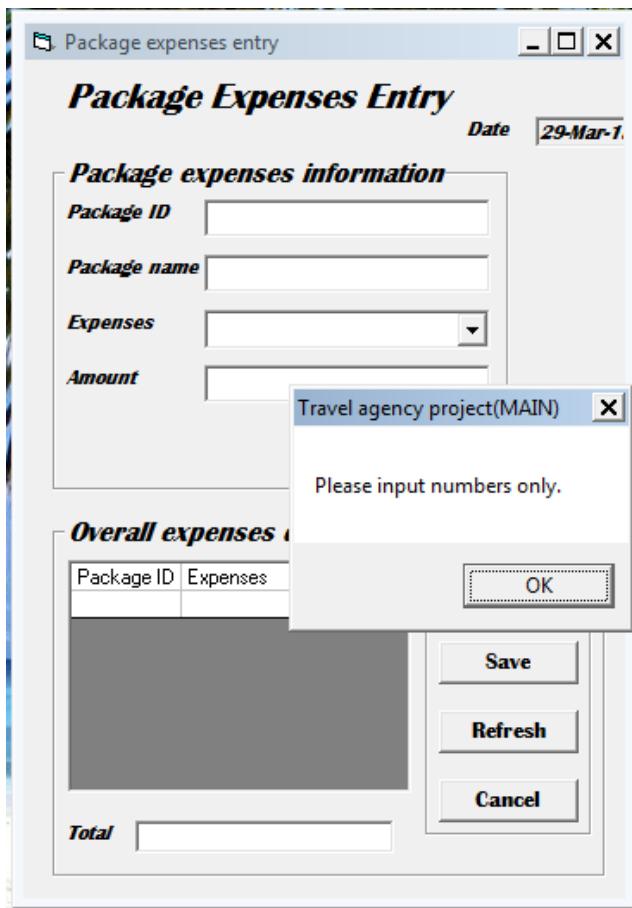
Test data used	Click Package Expenses Entry from Main Menu.
Type of data	
Outcome	Package Expenses Entry form is loaded.



The columns are assigned names automatically.

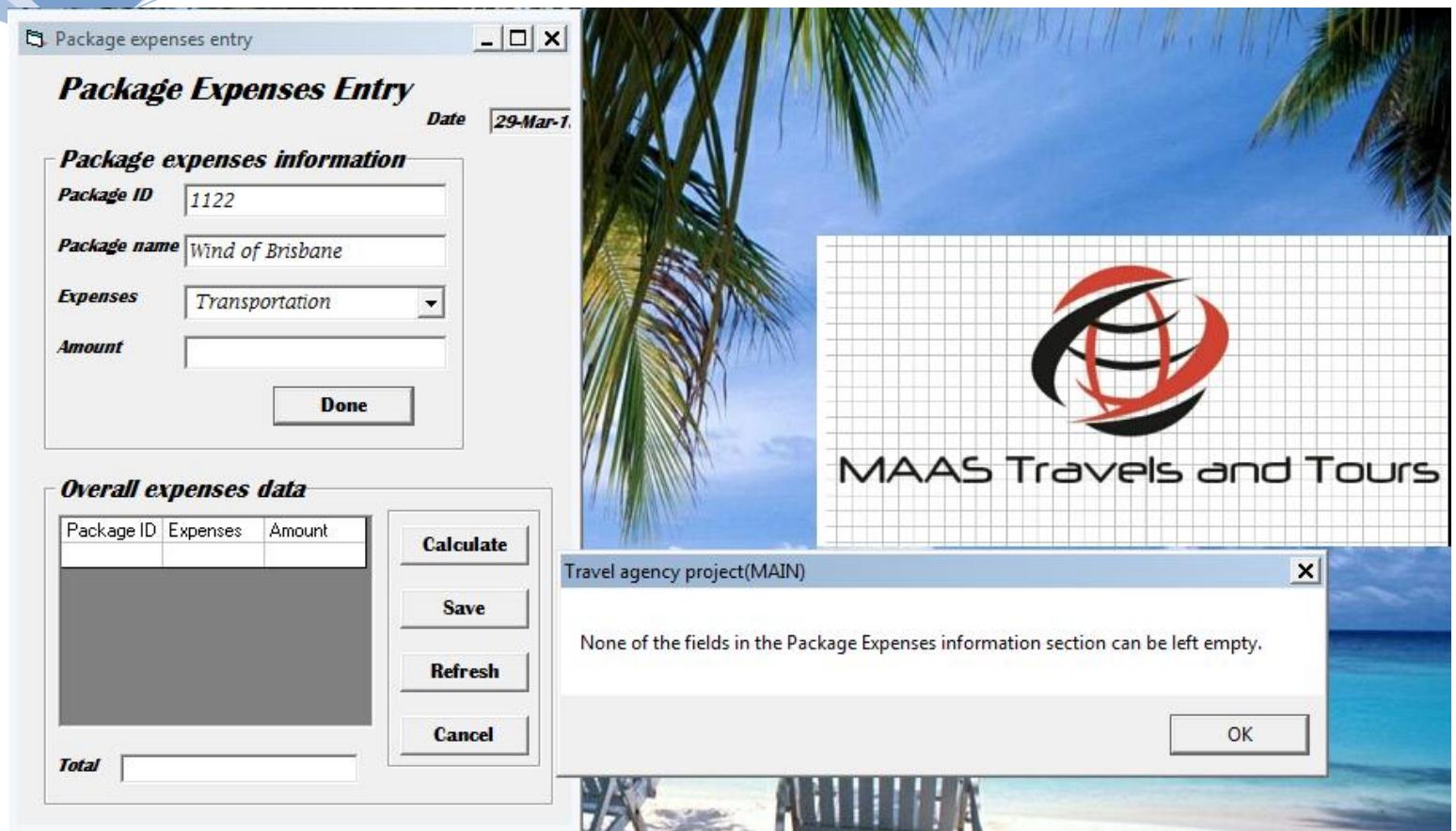
Test #51 Testing to check if error message is shown if non – numerical data is entered in the Amount section.

Test data used	Amount: s
Type of data	Abnormal
Outcome	Error message shown saying only numerical data is accepted.



Test #52 Testing to check if error message shown if any of the fields in the “Package Expenses information” section is left empty on pressing Done button.

Test data used	“Amount” Field is left empty.
Type of data	
Outcome	Error message shown saying none of the fields in the “Package Expenses information” can be left empty.



Test #53 Testing to check if Package Expenses data is automatically added to the flexgrid in the “Overall expenses data” section of the form and to the Package Expenses Details database table on pressing Done button, provided all the information entered is valid.

Test data used	Package ID: 1122 Package name: Wind of Brisbane Expenses: Transportation, Amount: 5000 Expenses: Accommodation, Amount: 15000 Expenses: Entry fees, Amount: 3000 Expenses: Accommodation, Amount: 20000 Expenses: Transportation, Amount: 10000
Type of data	Normal
Outcome	-Data is added to the “Overall Expenses data” table in the form. -Data is added to the Package Expenses Details database table.

Package Expenses Entry

Date 29-Mar-15

Package expenses information

Package ID	1122
Package name	Wind of Brisbane
Expenses	Accommodation
Amount	20000

Overall expenses data

Package ID	Expenses	Amount
1122	Transportation	5000
1122	Accommodate	15000
1122	Entry fee	3000
	Total	

Buttons: Done, Calculate, Save, Refresh, Cancel.

On pressing
“Done”
button

“Expenses”
and
“Amount”
fields cleared

The New
record is
added to the
table.

Package Expenses Entry

Date 29-Mar-15

Package expenses information

Package ID	1122
Package name	Wind of Brisbane
Expenses	Accommodation
Amount	20000

Overall expenses data

Package ID	Expenses	Amount
1122	Transportation	5000
1122	Accommodate	15000
1122	Entry fee	3000
1122	Accommodation	20000
	Total	

Buttons: Done, Calculate, Save, Refresh, Cancel.

travel agency database : Database (Access 2007) - Microsoft Access

Home Create External Data Database Tools Datasheet

Views Paste Cut Copy Format Painter Clipboard

Font Rich Text

New Save Spelling Refresh All More Advanced Filter Selection Advanced More Sort & Filter Find Replace Go To Select Find

Security Warning Certain content in the database has been disabled Options...

Package_expenses_details

PackageID	Expenses	Amount
1121	Transportation	\$5,000.00
1121	Transportation	\$5,000.00
1121	Entry fee	\$2,000.00
1121	Accommodate	\$6,650.00
1121	Transportation	\$9,000.00

The database table
before addition of
all the data.

travel agency database : Database (Access 2007) - Microsoft Access

Home Create External Data Database Tools Datasheet

Views Paste Cut Copy Format Painter Clipboard

Font Rich Text

New Save Spelling Refresh All More Advanced Filter Selection Advanced More Sort & Filter Find Replace Go To Select Find

Security Warning Certain content in the database has been disabled Options...

Package_expenses_details

PackageID	Expenses	Amount
1121	Transportation	\$5,000.00
1121	Transportation	\$5,000.00
1121	Entry fee	\$2,000.00
1121	Accommodate	\$6,650.00
1121	Transportation	\$9,000.00
1122	Transportation	\$5,000.00
1122	Accommodate	\$15,000.00
1122	Entry fee	\$3,000.00
1122	Accommodate	\$20,000.00
1122	Transportation	\$10,000.00

The database after
the addition of all
the new data.

Test #54 Testing to check if all the amounts in the Overall Expenses data table are added, a total is calculated and the total value is shown at the Total textbox on pressing the calculate button.

Test data used	The data from the previous test is used.
Type of data	Normal
Outcome	A total amount appears on the total textbox.

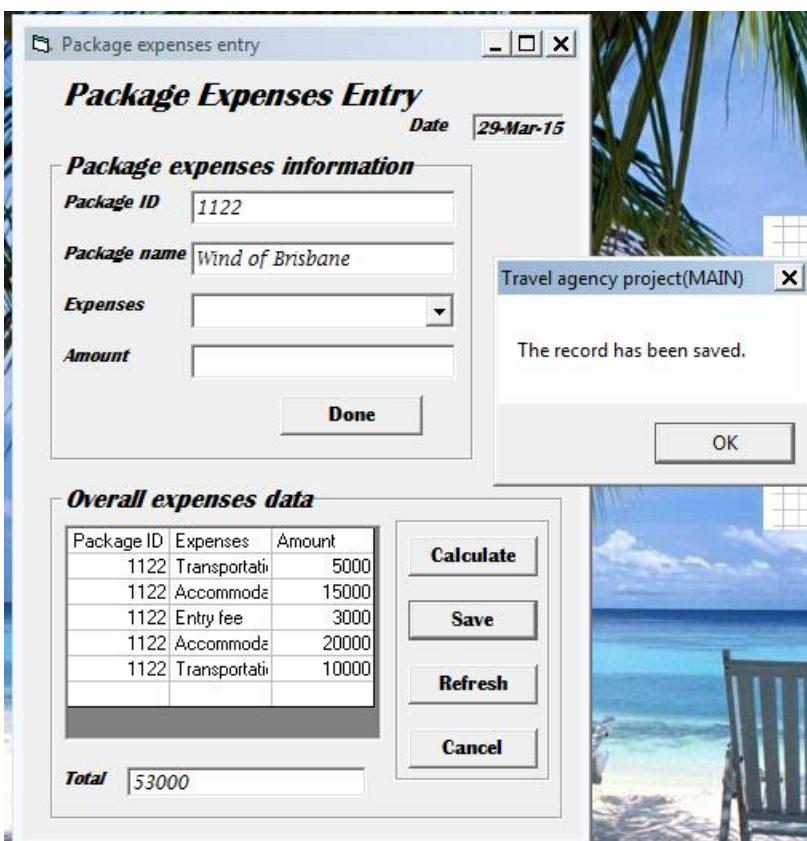
The screenshot shows the 'Package expenses entry' application window. At the top, it displays 'Package Expenses Entry' and the date '29-Mar-15'. Below this, there's a section titled 'Package expenses information' containing fields for 'Package ID' (1122), 'Package name' (Wind of Brisbane), 'Expenses' (dropdown menu), and 'Amount' (text input). A 'Done' button is also present. The main area is titled 'Overall expenses data' and contains a grid of five rows with columns for Package ID, Expenses, and Amount. The grid data is as follows:

Package ID	Expenses	Amount
1122	Transportati	5000
1122	Accommoda	15000
1122	Entry fee	3000
1122	Accommoda	20000
1122	Transportati	10000

To the right of the grid is a vertical toolbar with 'Calculate', 'Save', 'Refresh', and 'Cancel' buttons. A callout box with an arrow points to the 'Total' text input field at the bottom left of the grid, which contains the value '53000'. The callout box contains the text: 'A total is calculated and added to the "Total" textbox.'

Test #55 Testing to check if a message is shown for confirmation that the data is saved and if Package ID, Package name and the total amount is added to the Package Expenses master table in the database on pressing Save button, provided all the information entered is valid.

Test data used	The data from the previous test is used.
Type of data	Normal
Outcome	<p>Confirmation message that the data is saved.</p> <p>New record is added to the database table.</p>



travel agency database : Database (Access 2007) - Microsoft Access		
Table Tools		
Home Create External Data Database Tools Datasheet		
	Cut	Copy
		Clipboard
	Calibri	11

Test #56 Testing to check if all the fields in the form are emptied and the flexgrid table is refreshed on pressing the refresh button.

Test data used	The Refresh button is clicked.
Type of data	
Outcome	All the fields are emptied.

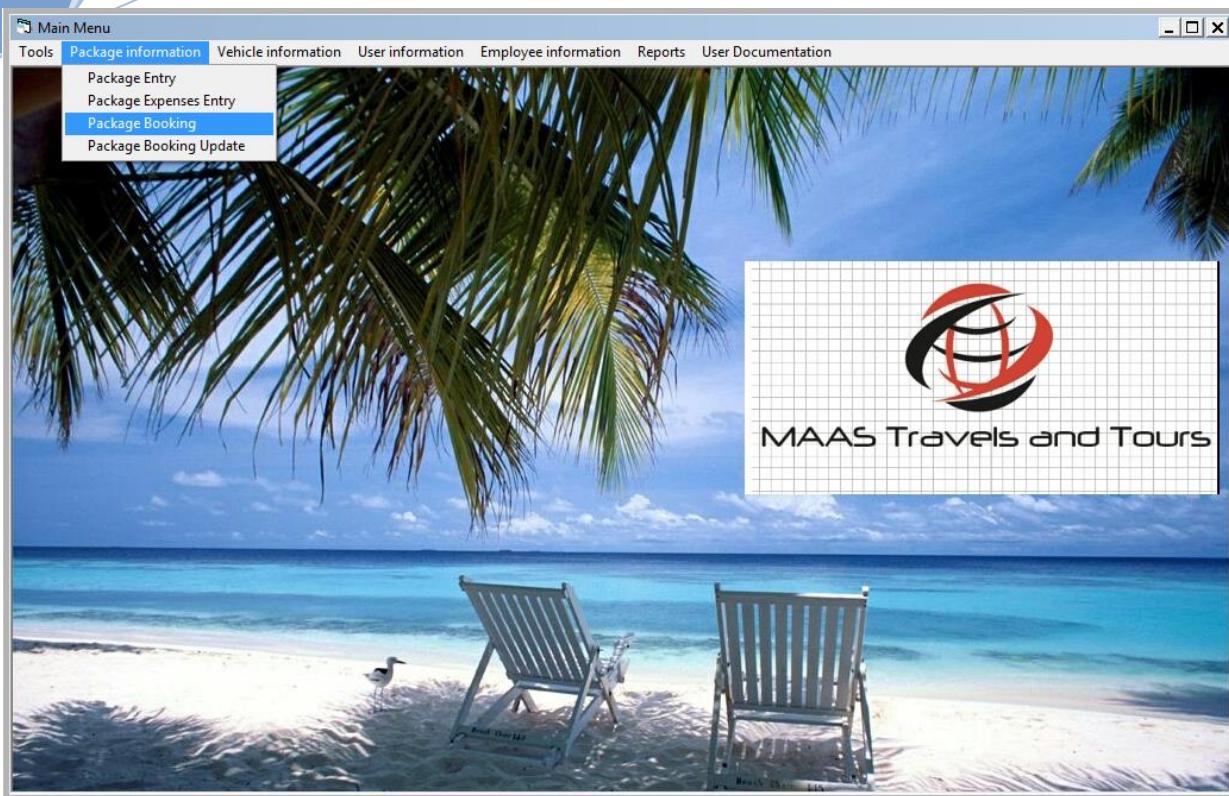
On pressing Refresh button.

Package ID	Expenses	Amount
1122	Transportation	5000
1122	Accommodation	15000
1122	Entry fee	3000
1122	Accommodation	20000
1122	Transportation	10000

Total: 53000

Test #57 Testing to check if Package Booking form is loaded on selection from the Main Menu and the Package types are automatically loaded.

Test data used	Selecting Package Booking form from Main Menu.
Type of data	
Outcome	Package Booking form is loaded. Package types are loaded on the form.



The screenshot shows the "Package booking" window. The title bar says "Main Menu" and "Package booking". The main area is titled "Package Booking". It has three main sections: "Package Specifications", "Information on Package", and "Customer Details".

- Package Specifications:**
 - Package Type:
 - Package name:
 - PackageID:
 - Cost per person:
 - Number of people:
 - Net total:
- Information on Package:**
 - Itinerary:
 - Package Details:
 - Accommodation type:
 - Journey Date:
 - Arrival time:
 - Starting time:
 - Duration: Days
 - Number of Vacancies:
- Customer Details:**
 - Date: 29-03-2015
 - Booking ID: 1235
 - Customer Name:
 - Address:
 - Contact number:

At the bottom right are buttons for "Confirm", "Refresh", and "Cancel". The background of the window features a scenic image of a beach with palm trees and a logo for "MAAS Travels and Tours".

Package types are loaded automatically.

Booking ID generated for new transaction.

Test #58 Testing to check if the Package names list is loaded with names on selection of a Package type and all the required Package information is filled on the form on selection of a Package name.

Test data used	Select a Package type. Select a Package name.
Type of data	Normal
Outcome	Package names appear on the list on selection of a Package type. Fields dependent on the Package names are filled.

Package booking

Package Booking

Date 30-03-2015

Package Specifications		Information on Package		Customer Details	
Package Type <input type="button" value="Multi-day"/>		Itinerary 		Booking ID <input type="text" value="1239"/> Customer Name <input type="text"/> Address <input type="text"/> Contact number <input type="text"/>	
Package name <input type="text" value="Wind of Brisbane"/> <input type="text" value="Sailing in Paris"/> <input type="text" value="River run in Venice"/>		Package Details 		<input type="button" value="Confirm"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>	
PackageID <input type="text"/>		Accommodation type <input type="text"/>			
Cost per person <input type="text"/>		Journey Date <input type="text"/>			
Number of people <input type="text"/>		Arrival time <input type="text"/>			
Net total <input type="text"/>		Starting time <input type="text"/>			
		Duration Number of Vacancies <input type="text"/> Days <input type="text"/>			

Package booking

Package Booking

Date 30-03-2015

Package Specifications		Information on Package		Customer Details	
Package Type <input type="button" value="Multi-day"/>		Itinerary 12/2/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/2/16 Australia Zoo - Outback Australia Dinner show 15/2/16 Day of leisure		Booking ID <input type="text" value="1239"/> Customer Name <input type="text"/> Address <input type="text"/> Contact number <input type="text"/>	
Package name <input type="text" value="Wind of Brisbane"/> <input type="text" value="Sailing in Paris"/> <input type="text" value="River run in Venice"/>		Package Details 3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options		<input type="button" value="Confirm"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>	
PackageID <input type="text" value="1122"/>		Accommodation type <input type="text" value="Normal"/>			
Cost per person <input type="text" value="5000"/>		Journey Date <input type="text" value="12-Feb-16"/>			
Number of people <input type="text"/>		Arrival time <input type="text" value="10:00:00 AM"/>			
Net total <input type="text"/>		Starting time <input type="text" value="11:00:00 PM"/>			
		Duration Number of Vacancies <input type="text" value="3"/> Days <input type="text" value="491"/>			

Test #59 Testing to check if the fields that depend on the Package name are automatically emptied if Package type is changed.

Test data used	Select a Package type.
Type of data	Normal
Outcome	Fields dependent on Package name are emptied.

The figure consists of two side-by-side screenshots of a Windows application window titled "Package booking". Both screenshots show the "Package Booking" form. In the left screenshot, the "Package Type" dropdown is set to "Multi-day" and the package name is "Wind of Brisbane" with the description "Sailing in Paris River run in Venice". The right screenshot shows the same form after the "Package Type" has been changed to "Single day", at which point all the dependent fields (itinerary, accommodation, contact, etc.) have been cleared or reset to their default state.

All the fields in the form that are dependent on Package type are emptied on changing it.

Test #60 Testing to check if error message shown on provision of non-numerical data for fields such as Contact number, Number of people etc.

Test data used	Number of people: p
Type of data	Abnormal
Outcome	Message shown asking to provide only numerical data.

Travel Agency Management System

Package Specifications

Information on Package

Customer Details

Date: 30-03-2015

Please input numbers only

OK

Test #61 Testing to check if error message shown if any of the fields are left empty on pressing Confirm button.

Test data used	"Number of People" and "Net total" is left empty.
Type of data	Abnormal
Outcome	Error message shown saying all Package information and Customer information need to be provided.

Package Specifications

Information on Package

Customer Details

Date: 30-03-2015

All the Package information and customer information need to provided in order to proceed with the booking.

OK

Test #62 Texting if Net total is automatically calculated on provision of Cost per person and Number of People.

Test data used	Package type: Multi-day Package name: Sailing in Paris Number of People: 5
Type of data	Abnormal
Outcome	Error message shown to notify that the number of digits for contact number is invalid.

Before moving to new field:

Package Booking

Package Specifications

- Package Type: Multi-day
- Package name: Wind of Brisbane
Sailing in Paris
River run in Venice

Information on Package

- Itinerary: 12/2/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/2/16 Australia Zoo - Outback Australia Dinner show
15/2/16 Day of leisure
- Package Details: 3 star hotel for 3 nights with Breakfast, International flights, Transfers, Tour host and all options

Customer Details

- Booking ID: 1241
- Customer Name: Peter Regal Whittam
- Address: [empty]
- Contact number: [empty]

PackageID: 1122

Cost per person: 5000

Number of people: 5

Net total: [empty]

Accommodation type: Normal

Journey Date: 12-Feb-16

Arrival time: 10:00:00 AM

Starting time: 11:00:00 PM

Duration: 3 Days

Number of Vacancies: 488

Date: 30-03-2015

Buttons: Confirm, Refresh, Cancel

After moving to new field:

Package Booking

Package Specifications

- Package Type: Multi-day
- Package name: Wind of Brisbane
Sailing in Paris
River run in Venice

Information on Package

- Itinerary: 12/2/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/2/16 Australia Zoo - Outback Australia Dinner show
15/2/16 Day of leisure
- Package Details: 3 star hotel for 3 nights with Breakfast, International flights, Transfers, Tour host and all options

Customer Details

- Booking ID: 1241
- Customer Name: Peter Regal Whittam
- Address: [empty]
- Contact number: [empty]

PackageID: 1122

Cost per person: 5000

Number of people: 5

Net total: 25000

Accommodation type: Normal

Journey Date: 12-Feb-16

Arrival time: 10:00:00 AM

Starting time: 11:00:00 PM

Duration: 3 Days

Number of Vacancies: 488

Date: 30-03-2015

Buttons: Confirm, Refresh, Cancel

Net total is generated automatically.

Test #63 Testing to check if error message shown if number of digits of the contact number provided is invalid on pressing Confirm button.

Test data used	Contact number: 0153452568888
Type of data	Abnormal
Outcome	Error message shown to notify that the number of digits for contact number is invalid.

The screenshot shows the 'Package Booking' application interface. On the left, under 'Package Specifications', the 'Package Type' is set to 'Multi-day' and the 'Package name' is listed as 'Wind of Brisbane', 'Sailing in Paris', and 'River run in Venice'. In the center, 'Information on Package' includes an 'Itinerary' section with several travel details. On the right, 'Customer Details' show a 'Booking ID' of '1239', a 'Customer Name' of 'Chris Gellar', and an 'Address' of '25th, Wolwin Heights, Chittagong'. A modal dialog box is displayed in the center-right area, titled 'Package Details'. It contains the text 'Number of digits used for contact number is invalid.' and has 'OK', 'Confirm', 'Refresh', and 'Cancel' buttons. The overall interface is a standard Windows-style application with a light gray background and various input fields and dropdown menus.

Test #64 Testing to check if error message shown if Number Reserved exceeds Number of Reservations limit on pressing Confirm button.

Test data used	Number of people: 492
Type of data	Normal
Outcome	Error message saying there are no vacancies left for the particular Package.

Package Specifications

Package Type: Multi-day

Package name: Wind of Brisbane
Sailing in Paris
River run in Venice

PackageID: 1122

Cost per person: 5000

Number of people: 492

Net total: 2460000

Information on Package

Itinerary

- 12/2/16 Depart Zia Airport - Emirates
- 13/2/16 Meet and Greet on arrival and transfer to hotels
- 14/2/16 Australia Zoo - Outback Australia Dinner show
- 15/2/16 Day of leisure

Customer Details

Booking ID: 1240

Customer Name: Chris Gellar

Address: 25th, Wolwin Heights, Chittagong

Package Details

Travel agency project(MAIN)

Sorry, there are no vacancies for this package. All the slots have been booked.
Please try another package.

OK Cancel

Duration	Number of Vacancies
3 Days	491

Test #65 Testing to check if confirmation message is provided on the addition of a new record to the Package Booking database table, the receipt is generated, all the fields on the form are emptied, the Booking ID and the database table are updated on pressing Confirm button, given that all the information provided is valid.

Test data used	Package type: Multi-day, Package name: Wind of Brisbane, Number of People: 3, Booking ID: 1240, Customer name: Chris Gellar, Address: 25th, Wolwin Heights, Chittagong, Contact number: 01534525688
Type of data	Normal
Outcome	Confirmation message shown saying the Booking was successful. All the fields are emptied. Booking ID is updated. Package Booking database table is updated for addition of new data. Vacancy information of Package Entry database table is updated.

Jan. 1

Travel Agency Management System

New Booking ID generated for next transaction.

Package booking

Package Booking

Date 30-03-2015

Package Specifications

Package Type

Package name

Wind of Brisbane
Sailing in Paris
River run in Venice

Information on Package

Itinerary

12/2/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/2/16 Australia Zoo - Outback Australia Dinner show
15/2/16 Day of leisure

Customer Details

Booking ID

Customer Name

Address

Package Details

3 star hotel for 3 night Breakfast, Internatic Flights, Transfers, To and all options

Travel agency project(MAIN) X

Booking has been made successfully.

Accommodation type

Journey Date

Arrival time

Starting time

Duration

Days

Number of Vacancies

Package booking		
<h1 style="text-align: center;">Package Booking</h1>		
Date 30-03-2015		
Package Specifications		
Information on Package		
Customer Details		
Package Type <input type="text"/>	Itinerary <input type="text"/>	Booking ID 1241
Package name <input type="text"/>	Package Details <input type="text"/>	Customer Name <input type="text"/>
PackageID <input type="text"/>	Accommodation type <input type="text"/>	Address <input type="text"/>
Cost per person <input type="text"/>	Journey Date <input type="text"/>	Contact number <input type="text"/>
Number of people <input type="text"/>	Arrival time <input type="text"/>	
Net total <input type="text"/>	Starting time <input type="text"/>	<input type="button" value="Confirm"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>
Duration <input type="text"/> Days Number of Vacancies <input type="text"/> 491		

The screenshot shows the Microsoft Access application window with the title "travel agency database : Database (Access 2007) - Microsoft Access". The ribbon bar at the top has "Table Tools" selected. The "Home" tab is active, displaying various toolbar icons for operations like Cut, Copy, Paste, Font, and Records. Below the toolbar is a ribbon of buttons for Refresh, Save, Spelling, Selection, Filter, and Sort & Filter. A message box at the top left says "Security Warning: Certain content in the database has been disabled" with an "Options..." button. The main area displays the "Package_booking" table in Datasheet view. The table has columns: PackageBookingID, Customer_name, Customer_address, Contact, Package_type, Package_name, PackageID, Cost_per_pe, Number_of_, Net_total, and Date_of_boo. The data is as follows:

PackageBookingID	Customer_name	Customer_address	Contact	Package_type	Package_name	PackageID	Cost_per_pe	Number_of_	Net_total	Date_of_boo
1235	Grace Muller	45th, M.M. Ali road, Chitt	01674451039	Multi-day	Sailing In Paris	1123	\$30,000.00	5	\$150,000.00	30-09-2023
1236	Jone Jenkins	83rd, Sholoshohor, Chitt	01819695582	Multi-day	Wind of Brisbane	1122	\$5,000.00	6	\$30,000.00	30-09-2023
1237	Bella Blossoms	31st, War Cemetery Road,	01578859085	Single day	Summer Isles	1121	\$1,200.00	10	\$12,000.00	30-09-2023
1238	Tyler Crowing	25th, Bashundhara street	01819625123	Multi-day	Wind of Brisbane	1122	\$5,000.00	3	\$15,000.00	30-09-2023

Package Booking database table before the record is added.

Package Booking database table after the record is added.

Travel Agency Management System

Cost_per_person	Accommodation_type	Journey_date	Arrival_time	Starting_time	Duration/days	Date_of_entry	NumberReserved	NumberOfReservations	Add New
\$25,000.00	Normal	12-Feb-16	10:00:00 AM	11:00:00 AM	3	29-Mar-15	0	300	
\$30,000.00	Deluxe	12-Mar-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	7	500	
\$1,200.00	Deluxe	21-Sep-15	11:15:00 AM	11:30:00 AM	10	29-Mar-15	98	100	
\$5,000.00	Normal	12-Feb-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	9	500	
*									

Package Entry database table before the record is added.

Cost_per_person	Accommodation_type	Journey_date	Arrival_time	Starting_time	Duration/days	Date_of_entry	NumberReserved	NumberOfReservations	Add New
\$25,000.00	Normal	12-Feb-16	10:00:00 AM	11:00:00 AM	3	29-Mar-15	0	300	
\$30,000.00	Deluxe	12-Mar-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	7	500	
\$1,200.00	Deluxe	21-Sep-15	11:15:00 AM	11:30:00 AM	10	29-Mar-15	98	100	
\$5,000.00	Normal	12-Feb-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	12	500	
*									

Package Entry database table after the record is added.

Test #66 Testing to check if all the fields on the form are emptied on pressing the Refresh button.

Test data used	Click Refresh button while some of the fields in the form is filled.
Type of data	Normal
Outcome	All the fields on the form are emptied.

Before pressing Refresh button:

Package Booking

Package Specifications

- Package Type: Multi-day
- Package name: Wind of Brisbane
Sailing in Paris
River run in Venice
- PackageID: 1123
- Cost per person: 30000
- Number of people: 9
- Net total: 270000

Information on Package

Itinerary

12/3/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/3/16 Eiffel tower - Outback Paris Dinner show
15/3/16 Day of leisure

Package Details

3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options

Accommodation type: Deluxe

Journey Date: 12-Mar-16

Arrival time: 10:00:00 AM

Starting time: 11:00:00 PM

Duration: 3 Days **Number of Vacancies**: 495

Customer Details

Date: 30-03-2015

Booking ID: 1241

Customer Name:

Address:

Contact number:

Buttons: Confirm, Refresh, Cancel

After pressing Refresh button:

Package Booking

Package Specifications

- Package Type: (empty dropdown)
- Package name: (empty text box)
- PackageID: (empty text box)
- Cost per person: (empty text box)
- Number of people: (empty text box)
- Net total: (empty text box)

Information on Package

Itinerary: (empty text area)

Package Details: (empty text area)

Accommodation type: (empty dropdown)

Journey Date: (empty text box)

Arrival time: (empty text box)

Starting time: (empty text box)

Duration: (empty dropdown) Days **Number of Vacancies**: (empty text box)

Customer Details

Date: 30-03-2015

Booking ID: 1241

Customer Name: (empty text box)

Address: (empty text box)

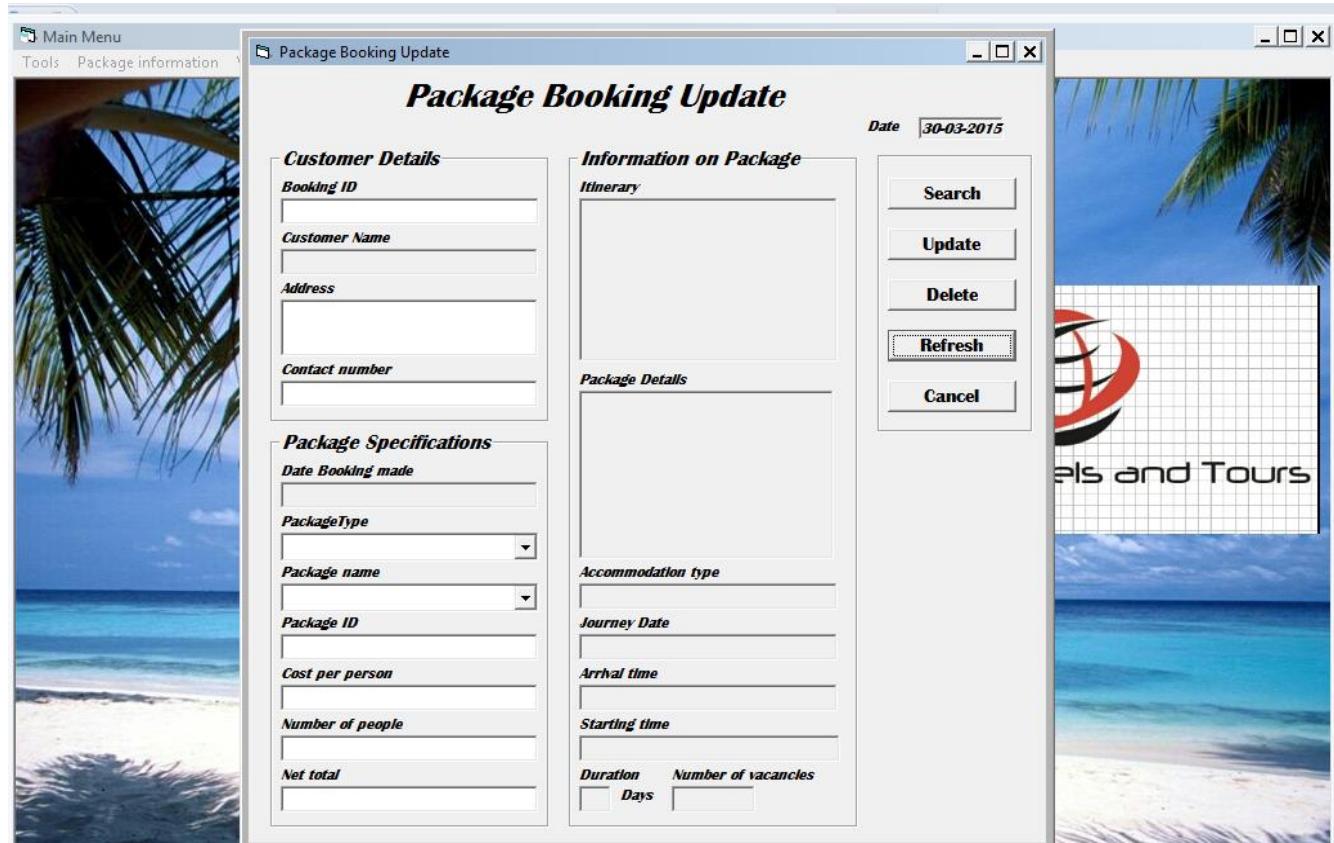
Contact number: (empty text box)

Buttons: Confirm, Refresh, Cancel

Test #67 Testing to check if Package Booking Update form opens on selection from Main Menu and Package type combo box is loaded with Package types.

Test data used	Click Package Booking Update form from Package information section of Main Menu form.
Type of data	Normal
Outcome	Package Booking Update form is loaded.

Travel Agency Management System



Package Booking Update

Package Booking Update

Date 30-03-2015

Customer Details

Booking ID
Customer Name
Address
Contact number

Information on Package

Itinerary
Package Details

Accommodation type
Journey Date
Arrival time
Starting time
Duration Days Number of vacancies

Search
Update
Delete
Refresh
Cancel

Package type loaded.

Multi-day
Single day

The screenshot shows a Windows application window titled "Package Booking Update". The window has a title bar with standard window controls. In the top right corner, the date "30-03-2015" is displayed. The main interface is divided into several sections: "Customer Details" on the left containing fields for Booking ID, Customer Name, Address, and Contact number; "Information on Package" in the center containing "Itinerary" and "Package Details" sections; and a column of buttons on the right labeled Search, Update, Delete, Refresh, and Cancel. On the far left, there is a separate rectangular box containing the text "Package type loaded." with an arrow pointing to the "Multi-day" option in a dropdown menu under "Package Specifications". The "Multi-day" option is highlighted with a red oval.

Test #68 Testing to check if error message shown asking for Booking ID if Search button is pressed without giving it.

Test data used	Booking ID field left empty. Click Search.
Type of data	
Outcome	Error message shown asking for Booking ID.

Test #69 Testing to check if error message is shown if the Booking ID provided is not found in the database.

Test data used	Booking ID:2253 Click Search.
Type of data	
Outcome	Error message shown saying that any record with the Booking ID provided does not exist in the database.

The screenshot shows the 'Package Booking Update' window. In the 'Customer Details' section, 'Booking ID' is set to 2253. The 'Information on Package' section displays an 'Itinerary' box containing travel details: '12/2/16 Depart Zia Airport - Emirates', '13/2/16 Meet and Greet on arrival and transfer to hotels', '14/2/16 Australia Zoo - Outback Australia Dinner show', and '15/2/16 Day of leisure'. A modal dialog box titled 'Accom' is overlaid on the window, displaying the message 'There is no record for this Booking ID.' with an 'OK' button. On the right side of the window, there are buttons for 'Search', 'Update', 'Delete', 'Refresh', and 'Cancel'.

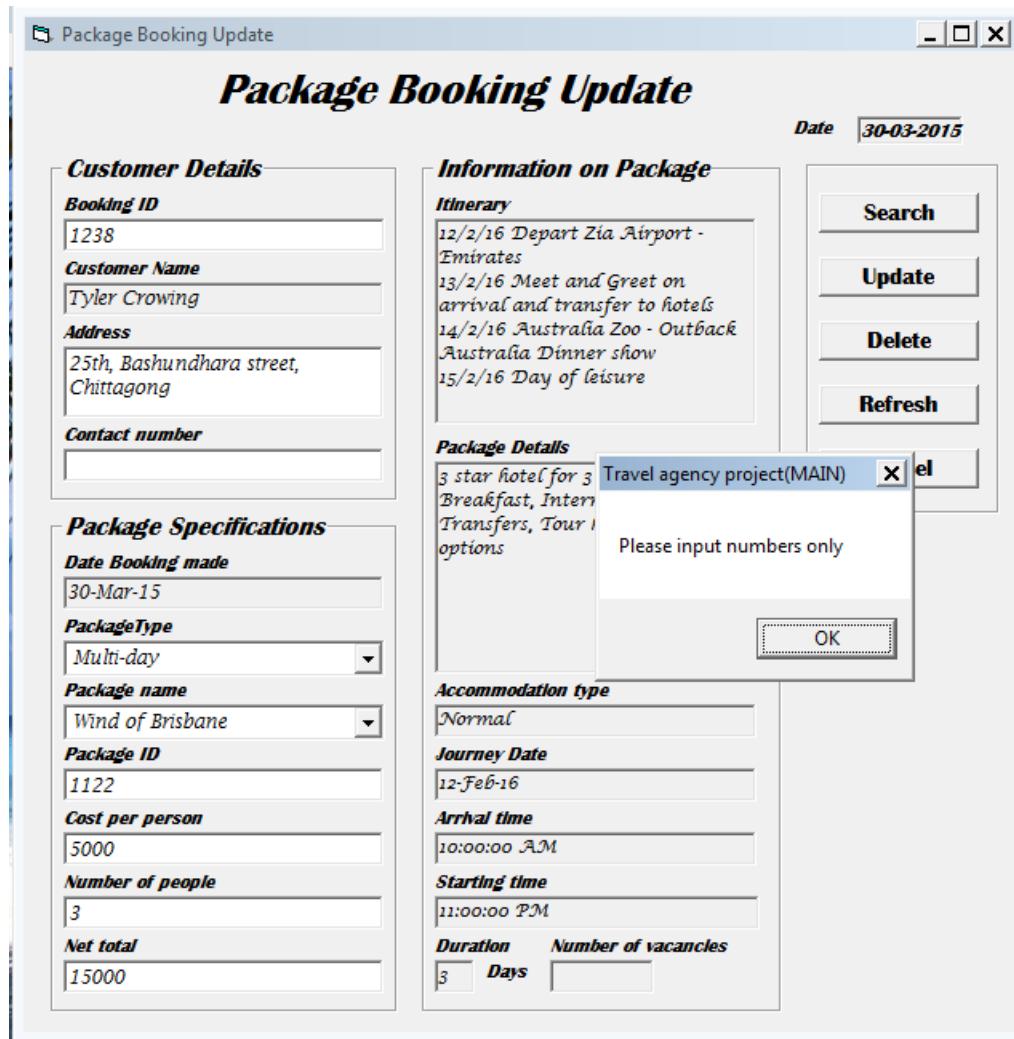
Test #70 Testing to check all the other fields are automatically loaded from the database table and filled on provision of the Booking ID and pressing Search.

Test data used	Booking ID:1238 Click Search.
Type of data	Normal
Outcome	All the information is automatically loaded from the database.

The screenshot shows the 'Package Booking Update' window. In the 'Customer Details' section, 'Booking ID' is set to 1238. The 'Information on Package' section displays an 'Itinerary' box containing travel details: '12/2/16 Depart Zia Airport - Emirates', '13/2/16 Meet and Greet on arrival and transfer to hotels', '14/2/16 Australia Zoo - Outback Australia Dinner show', and '15/2/16 Day of leisure'. The 'Package Details' section contains a note: '3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options'. On the right side of the window, there are buttons for 'Search', 'Update', 'Delete', 'Refresh', and 'Cancel'.

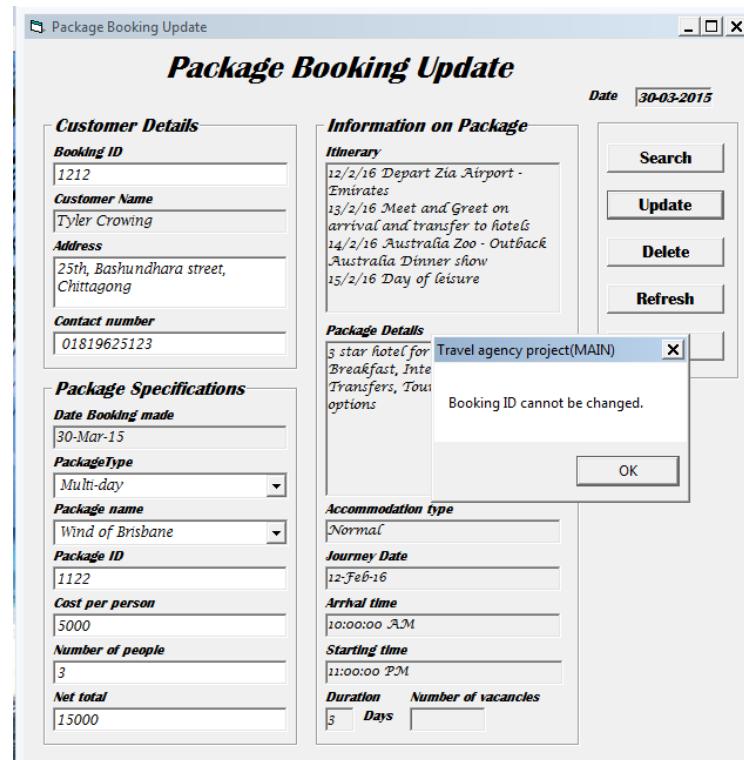
Test #71 Testing to check if error message shown if non – numerical data is provided for Contact number or Number of people.

Test data used	Contact number: p
Type of data	Abnormal
Outcome	Error message shown saying only numerical data is accepted.



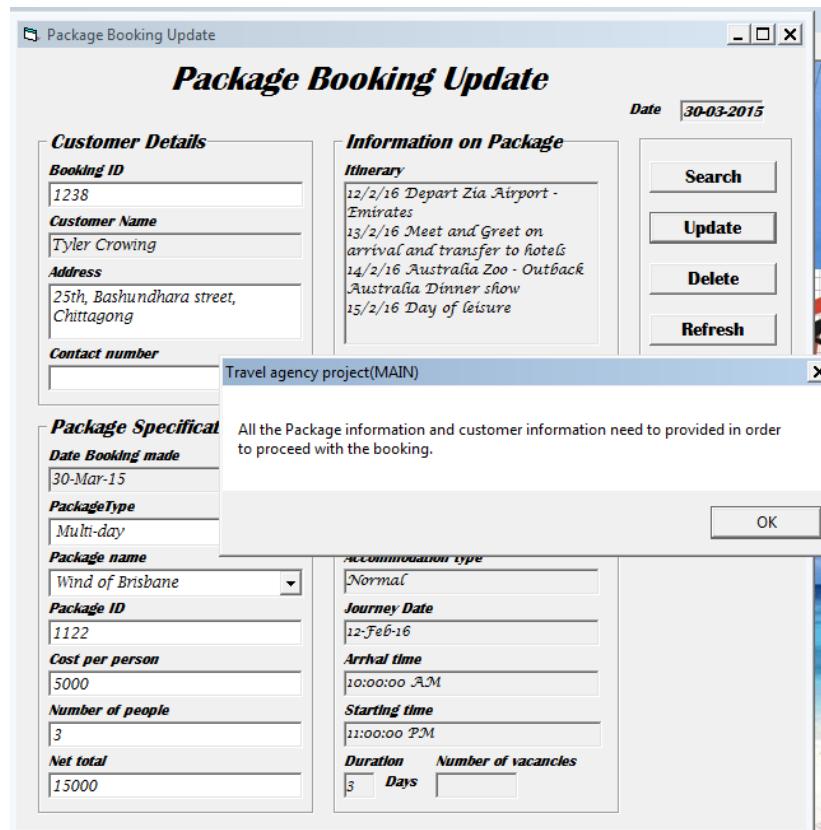
Test #72 Testing to check if error message shown if Booking ID is changed.

Test data used	Booking ID: 1212 Click Update.
Type of data	Normal
Outcome	Error message shown saying Booking ID cannot be changed.



Test #73 Testing if error message shown if any of the fields are left empty on pressing Update button.

Test data used	Contact number: "" Click Update.
Type of data	Normal
Outcome	Error message shown saying none of the fields can be left empty.



Test #74 Testing to check if error message shown if number of digits for contact number is invalid.

Test data used	Contact number: 015344595988 Click Update.
Type of data	Normal
Outcome	Error message shown saying number of digits provided for contact number is invalid.

Test #75 Testing to check if error message shown if Number reserved exceeds number of reservations.

Test data used	Number of reservations: 3 Click "Update".
Type of data	Normal
Outcome	Error message shown saying number of digits provided for contact number is invalid.

Package Booking Update

Date **30-03-2015**

Customer Details		Information on Package	
Booking ID <input type="text" value="1238"/> Customer Name <input type="text" value="Tyler Crowng"/> Address <input type="text" value="25th, Bashundhara street, Chittagong"/> Contact number <input type="text" value="01819625123"/>		Itinerary 11:30 AM Depart from Zia Airport from Dhaka to Chittagong 01:00 PM Meet and Greet on arrival transfers to hotels 02:00 PM Cox's Bazar Beach 09:00 PM Complementary dinner 11:30 PM Return to Dhaka	
Package Specifications		Package Details Lunch Complementary Dinner Beach tour with Guide	
Date Booking made <input type="text" value="30-Mar-15"/> PackageType <input type="text" value="Single day"/> Package name <input type="text" value="Summer Isles"/> Package ID <input type="text" value="1121"/> Cost per person <input type="text" value="1200"/> Number of people <input type="text" value="3"/> Net total <input type="text" value="3600"/>		Accommoda <input type="text" value="Deluxe"/> Journey Dat <input type="text" value="21-Sep-15"/> Arrival time <input type="text" value="11:15:00 AM"/> Starting time <input type="text" value="11:30:00 AM"/> Duration Number of vacancies 10 Days 2	

Travel agency project(MAIN) X

There are not enough vacancies to make this reservation.

OK

Test #76 Testing to check if Net total is calculated automatically on entering Number of people.

Test data used	Booking ID: 1238 Address: 24 th , Bashundhara street, Chittagong Package type: Multi-day Package name: Sailing in Paris Number of People: 5
Type of data	Normal
Outcome	Number of people is calculated automatically and generated in textbox.

Customer Details

Booking ID	1238
Customer Name	Tyler Crowing
Address	25th, Bashundhara street, Chittagong
Contact number	01819625123

Information on Package

Itinerary

12/3/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/3/16 Eiffel tower - Outback Paris Dinner show
15/3/16 Day of leisure

Package Details

3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options

Package Specifications

Date Booking made
30-Mar-15

PackageType
Multi-day

Package name
Sailing in Paris

Package ID
1123

Cost per person
30000

Number of people
5

Net total
[Initial Value]

Customer Details

Booking ID	1238
Customer Name	Tyler Crowing
Address	25th, Bashundhara street, Chittagong
Contact number	01819625123

Information on Package

Itinerary

12/3/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/3/16 Eiffel tower - Outback Paris Dinner show
15/3/16 Day of leisure

Package Details

3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options

Package Specifications

Date Booking made
30-Mar-15

PackageType
Multi-day

Package name
Sailing in Paris

Package ID
1123

Cost per person
30000

Number of people
5

Net total
150000

Accommodation type

Deluxe

Journey Date

12-Mar-16

Arrival time

10:00:00 AM

Starting time

11:00:00 PM

Duration **Number of vacancies**

3 Days 495

Accommodation type

Deluxe

Journey Date

12-Mar-16

Arrival time

10:00:00 AM

Starting time

11:00:00 PM

Duration **Number of vacancies**

3 Days 495

Search **Update** **Delete** **Refresh** **Cancel**

Search **Update** **Delete** **Refresh** **Cancel**

After providing Number of People

Test #77 Testing to check if confirmation message is provided for successful update of the record, followed by the emptying of the fields on the form and the information held by the record is changed in the database table on making changes and pressing the Update button, provided the information entered is valid.

Test data used	Booking ID: 1238 Address: 24 th , Bashundhara street, Chittagong Package type: Multi-day Package name: Sailing in Paris Number of People: 5 Click Update.
Type of data	Normal
Outcome	-Confirmation message that the record is updated. -All the fields in the form are emptied. -Package Booking database record is updated for any changes made. -Vacancy information is updated in Package Entry database.

Package Booking Update

Date **30-03-2015**

Customer Details		Information on Package	
Booking ID <input type="text" value="1238"/> Customer Name <input type="text" value="Tyler Crowing"/> Address <input type="text" value="25th, Bashundhara street, Chittagong"/> Contact number <input type="text" value="01819625123"/>		Itinerary 12/3/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/3/16 Eiffel tower - Outback Paris Dinner show 15/3/16 Day of leisure	
Package Specifications		Package Details 3 std Travel agency project(MAIN) Booked Train option Booking information has been Updated successfully.	
Date Booking made <input type="text" value="30-Mar-15"/> PackageType <input type="text" value="Multi-day"/> Package name <input type="text" value="Sailing in Paris"/> Package ID <input type="text" value="1123"/> Cost per person <input type="text" value="30000"/> Number of people <input type="text" value="5"/> Net total <input type="text" value="150000"/>		Accommodation type <input type="text" value="Deluxe"/> Journey Date <input type="text" value="12-May-16"/> Arrival time <input type="text" value="10:00:00 AM"/> Starting time <input type="text" value="11:00:00 PM"/> Duration Number of vacancies <input type="text" value="3"/> Days <input type="text" value="495"/>	

Package Booking Update

Date **30-03-2015**

Customer Details		Information on Package	
Booking ID <input type="text"/> Customer Name <input type="text"/> Address <input type="text"/> Contact number <input type="text"/>		Itinerary <input type="text"/>	
Package Specifications		Package Details <input type="text"/>	
Date Booking made <input type="text"/> PackageType <input type="text"/> Package name <input type="text"/> Package ID <input type="text"/> Cost per person <input type="text"/> Number of people <input type="text"/> Net total <input type="text"/>		Accommodation type <input type="text"/> Journey Date <input type="text"/> Arrival time <input type="text"/> Starting time <input type="text"/> Duration Number of vacancies <input type="text"/> Days <input type="text" value="495"/>	

Travel Agency Management System

Package_booking

PackageBookingID	Customer_name	Customer_address	Contact	Package_type	Package_name	PackageID	Cost_per_pe	Number_of_	Net_total	Date_of_boo
1235	Tyler Crowing	25th, Bashundhara street	01819625123	Single day	Summer Isles	1123	\$1,200.00	3	\$3,600.00	30-
1236	Jone Jenkins	83rd, Sholoshohor, Chitta	01819695582	Multi-day	Wind of Brisbane	1122	\$5,000.00	6	\$30,000.00	30-
1237	Bella Blossoms	31st, War Cemetery Road	01578859085	Single day	Summer Isles	1121	\$1,200.00	10	\$12,000.00	30-
1238	Tyler Crowing	25th, Bashundhara street	01819625123	Multi-day	Wind of Brisbane	1122	\$5,000.00	3	\$15,000.00	30-
1240	Chris Gellar	25th, Wolwin Heights, Ch	01534525688	Multi-day	Wind of Brisbane	1122	\$5,000.00	3	\$15,000.00	30-

Package Booking database table record before it is Updated.

Package_booking

PackageBookingID	Customer_name	Customer_address	Contact	Package_type	Package_name	PackageID	Cost_per_pe	Number_of_	Net_total	Date_of_boo
1236	Jone Jenkins	83rd, Sholoshohor, Chitta	01819695582	Multi-day	Wind of Brisbane	1122	\$5,000.00	6	\$30,000.00	30-
1237	Bella Blossoms	31st, War Cemetery Road	01578859085	Single day	Summer Isles	1121	\$1,200.00	10	\$12,000.00	30-
1238	Tyler Crowing	24th, Bashundhara street	01819625123	Multi-day	Sailing in Paris	1123	\$30,000.00	5	\$150,000.00	30-
1240	Chris Gellar	25th, Wolwin Heights, Ch	01534525688	Multi-day	Wind of Brisbane	1122	\$5,000.00	3	\$15,000.00	30-

Package Booking database table after the record is updated. The changes are circled.

Package_entry

Accommodation_type	Journey_date	Arrival_time	Starting_time	Duration/days	Date_of_entry	NumberReserved	NumberOfReservations	Add New Field
Normal	12-Feb-16	10:00:00 AM	11:00:00 AM	3	29-Mar-15	0	300	
Deluxe	12-Mar-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	2	500	
Deluxe	21-Sep-15	11:15:00 AM	11:30:00 AM	10	29-Mar-15	98	100	
Normal	12-Feb-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	12	500	

Vacancy information before update.

Package_entry

Accommodation_type	Journey_date	Arrival_time	Starting_time	Duration/days	Date_of_entry	NumberReserved	NumberOfReservations	Add New Field
Normal	12-Feb-16	10:00:00 AM	11:00:00 AM	3	29-Mar-15	0	300	
Deluxe	12-Mar-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	7	500	
Deluxe	21-Sep-15	11:15:00 AM	11:30:00 AM	10	29-Mar-15	98	100	
Normal	12-Feb-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	12	500	

Vacancy information after update.

Test #78 Testing to check if Confirmation message for deletion is shown, all the boxes are emptied, reservation information in Package Entry database is updated and the record is deleted from database on pressing Delete button, provided all the fields are filled.

Test data used	Booking ID: 1238 Click "Search". Click "Delete".
Type of data	Normal
Outcome	-Confirmation message shown for the deletion of the record. -All the boxes are emptied. -Package Entry table reservation information is updated. -Record is deleted from the Package Booking database table.

On pressing "Delete" button:

The screenshot shows the 'Package Booking Update' application window. On the right side, there is a vertical stack of buttons: Search, Update, Delete, Refresh, and Cancel. The 'Delete' button is highlighted. A small modal dialog box titled 'Travel agency project(MAIN)' is displayed in the center, containing the message 'The Booking has been deleted.' with an 'OK' button at the bottom.

On pressing "OK" button:

The screenshot shows the same 'Package Booking Update' window after the 'OK' button has been clicked. All the input fields on the left side of the window are now empty, indicating that the record has been successfully deleted from the database.

Travel Agency Management System

Package Entry and Package Booking database tables before deletion:

Accommodation_type	Journey_date	Arrival_time	Starting_time	Duration/days	Date_of_entry	NumberReserved	NumberOfReservations	Add New Field
Normal	12-Feb-16	10:00:00 AM	11:00:00 AM	3	29-Mar-15	0	300	
Deluxe	12-Mar-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	7	500	
Deluxe	21-Sep-15	11:15:00 AM	11:30:00 AM	10	29-Mar-15	98	100	
Normal	12-Feb-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	12	500	

Package Entry database vacancy information before deletion.

PackageBookingID	Customer_name	Customer_address	Contact	Package_type	Package_name	PackageID	Cost_per_pe	Number_of	Net_total	Date_of_boo
1236	Jone Jenkins	83rd, Sholoshohor, Chitta	01819695582	Multi-day	Wind of Brisbane	1122	\$5,000.00	6	\$30,000.00	30
1237	Bella Blossoms	31st, War Cemetery Road	01578859085	Single day	Summer Isles	1121	\$1,200.00	10	\$12,000.00	30
1238	Tyler Crowing	24th, Bashundhara street	01819625123	Multi-day	Sailing in Paris	1123	\$30,000.00	5	\$150,000.00	30
1240	Chris Gellar	25th, Wolwin Heights, Ch	01534525688	Multi-day	Wind of Brisbane	1122	\$5,000.00	3	\$15,000.00	30

Package Booking database table before deletion.

Travel Agency Management System

Package Entry and Package Booking database tables before deletion:

Cost_per_person	Accommodation_type	Journey_date	Arrival_time	Starting_time	Duration/days	Date_of_entry	NumberReserved	NumberOfReservations	Add New
\$25,000.00	Normal	12-Feb-16	10:00:00 AM	11:00:00 AM	3	29-Mar-15	0	300	
\$30,000.00	Deluxe	12-Mar-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	2	500	
\$1,200.00	Deluxe	21-Sep-15	11:15:00 AM	11:30:00 AM	10	29-Mar-15	8	100	
\$5,000.00	Normal	12-Feb-16	10:00:00 AM	11:00:00 PM	3	29-Mar-15	12	500	

Package Entry database vacancy information after deletion.

PackageBookingID	Customer_name	Customer_address	Contact	Package_type	Package_name	PackageID	Cost_per_pe	Number_of_	Net_total	Date_of_boo
1236	Jone Jenkins	83rd, Sholoshohor, Chitta	01819695582	Multi-day	Wind of Brisbane	1122	\$5,000.00	6	\$30,000.00	30-
1237	Bella Blossoms	31st, War Cemetery Road,	01578859085	Single day	Summer Isles	1121	\$1,200.00	10	\$12,000.00	30-
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted
1240	Chris Gellar	25th, Wolwin Heights, Ch	01534525688	Multi-day	Wind of Brisbane	1122	\$5,000.00	3	\$15,000.00	30-

Package Booking database table after deletion.

Test #79 Testing to Testing if all the fields are emptied and the combo box information are reloaded on pressing Refresh button.

Test data used	Booking ID: 1237 Click "Search" button. Click "Refresh" button.
Type of data	Normal
Outcome	All the fields are emptied.

On pressing "Search" button:

The screenshot shows the 'Package Booking Update' window with the following data entered:

- Customer Details:**
 - Booking ID: 1237
 - Customer Name: Bella Blossoms
 - Address: 31st, War Cemetery Road, Chittagong
 - Contact number: 01378859085
- Information on Package:**
 - Itinerary:** 11:30 AM Depart from Zia Airport from Dhaka to Chittagong, 01:00 PM Meet and Greet on arrival transfers to hotel, 02:00 PM Cox's Bazar Beach, 09:00 PM Complementary dinner, 11:30 PM Return to Dhaka.
 - Package Details:** Lunch, Complementary Dinner, Beach tour with Guide.
 - Accommodation type:** Deluxe.
- Package Specifications:**
 - Date Booking made: 30-Mar-15
 - PackageType: Single day
 - Package name: Summer Isles
 - Package ID: 1121
 - Cost per person: 1200
 - Number of people: 10
 - Net total: 12000

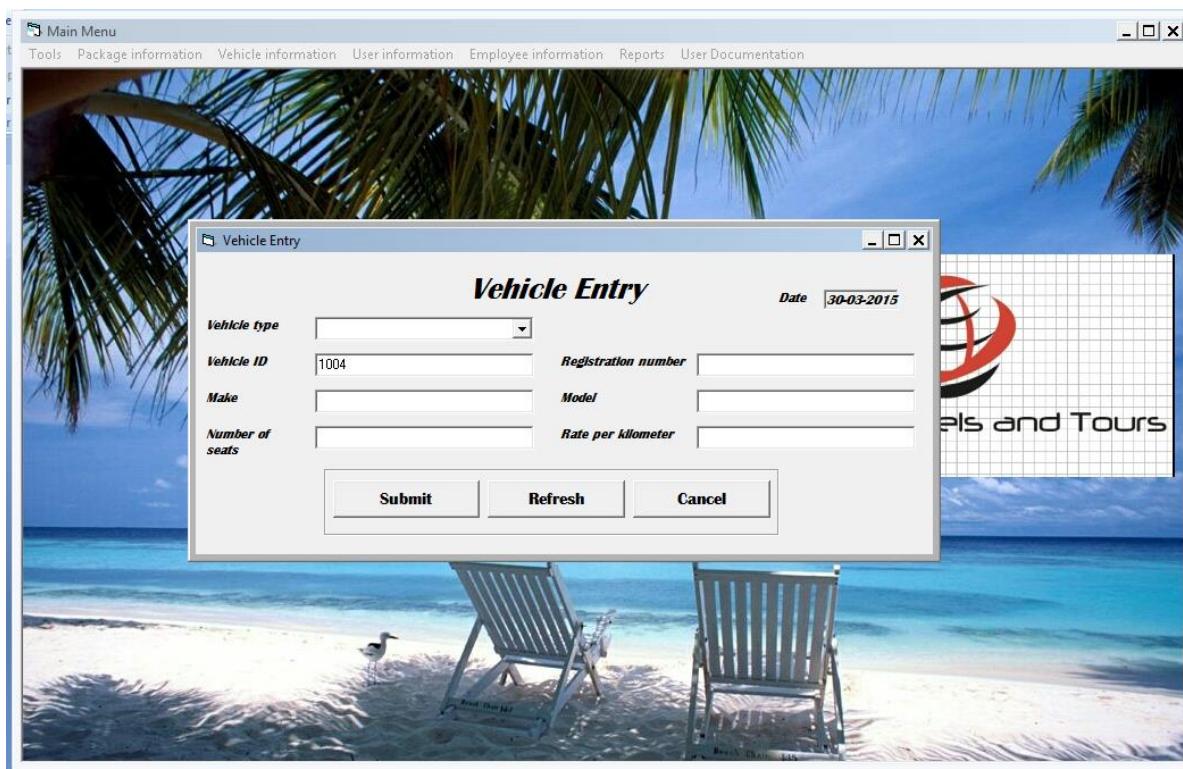
The screenshot shows the 'Package Booking Update' window with all fields empty or cleared:

- Customer Details:** All fields are empty.
- Information on Package:**
 - Itinerary:** All details are removed.
 - Package Details:** All details are removed.
 - Accommodation type:** All details are removed.
- Package Specifications:**
 - Date Booking made: 30-Mar-15
 - PackageType: (empty dropdown)
 - Package name: (empty dropdown)
 - Package ID: (empty)
 - Cost per person: (empty)
 - Number of people: (empty)
 - Net total: (empty)

Test #80 Testing to check if Vehicle entry from appears when selected from the Main Menu and a Vehicle ID is generated for the new transaction being made.

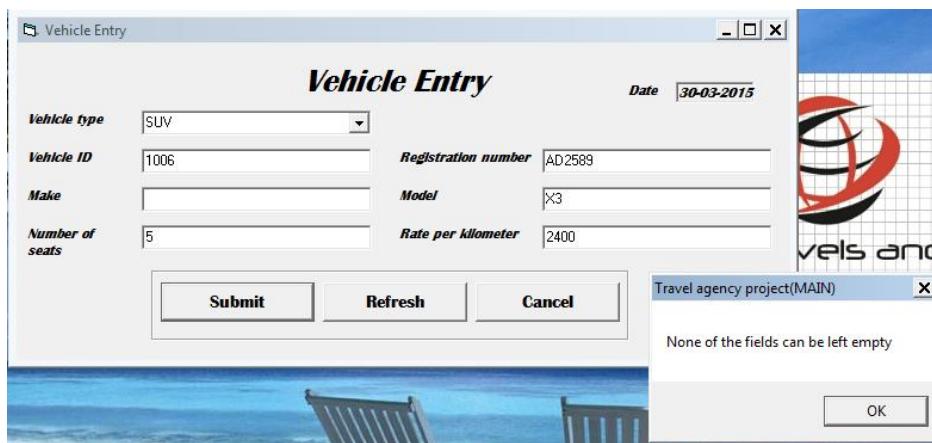
Test data used	Select "Vehicle Entry" from Main Menu
Type of data	Normal
Outcome	The Vehicle Entry form is loaded.

Travel Agency Management System



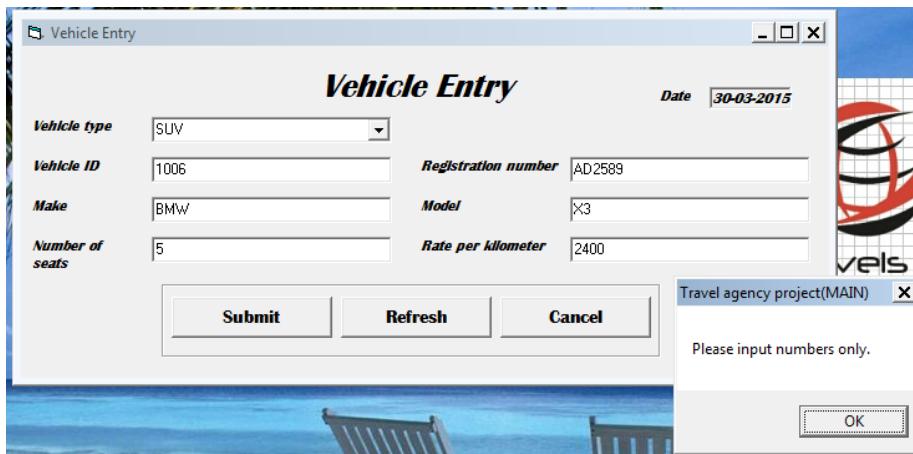
Test #81 Testing if error message appears if any of the fields are left empty on pressing Submit button.

Test data used	Vehicle type: SUV Vehicle ID: 1006 Registration number: AD2589 Make: "" Model: X3 Number of seats: 5 Rate per kilometer: 2400 Click "Submit"
Type of data	Normal
Outcome	Error message appears saying none of the fields can be left empty.



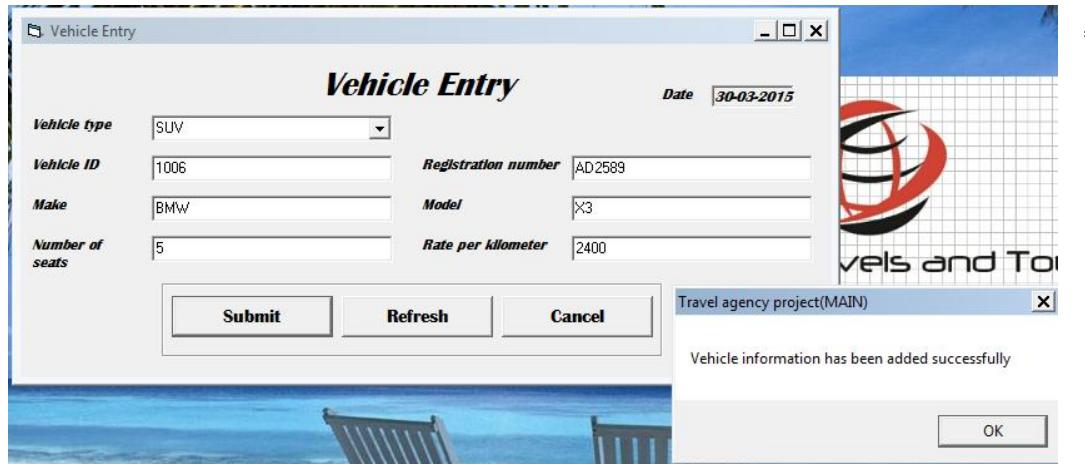
Test #82 Testing to check if error message appears if non-numerical data is entered in "Rate per kilometer" field.

Test data used	All the previous data are used. Make: BMW Rate per kilometer: r Click "Submit"
Type of data	Abnormal
Outcome	Error message asking for numerical data.



Test #83 Testing if confirmation message appears on addition of new record to the database, all the boxes in the form are emptied and the Vehicle ID updated for new transaction and the new record is added to the Vehicle Entry database table on pressing Submit button, provided all valid information is entered.

Test data used	Vehicle type: SUV Vehicle ID: 1006 Registration number: AD2589 Make: BMW Model: X3 Number of seats: 5 Rate per kilometer: 2400 Click "Submit"
Type of data	Normal
Outcome	Confirmation message displayed for the addition of new record to the database. All the boxes are emptied. Vehicle ID is updated. The new record is added to the database.



The screenshot shows the 'Vehicle Entry' application window. The 'Vehicle ID' field contains '1007'. A callout box with an arrow points to this field with the text 'New Vehicle ID generated.'. Another callout box with an arrow points to the other input fields with the text 'All the fields are emptied.'.

Travel Agency Management System

Vehicle_type	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input type="checkbox"/>	
Sedan	1003	AD4565	Cadillac	DTS	4	1800	03-Feb-15	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
*									

Vehicle Entry database table before addition record.

Vehicle_type	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input type="checkbox"/>	
Sedan	1003	AD4565	Cadillac	DTS	4	1800	03-Feb-15	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	
*									

Vehicle Entry database table after addition record.

Test #84 Test to check if all the fields are emptied on pressing Refresh button and the Vehicle ID is reloaded.

Test data used	Vehicle ID: 1007 Make: BMW Model: X3 Registration number: AD5466 Rate per kilometer: 2400 Click "Submit"
Type of data	Normal
Outcome	All the fields are emptied. Vehicle ID textbox is reloaded.

Vehicle Entry

Date 30-03-2015

Vehicle type	SUV
Vehicle ID	1006
Make	BMW
Number of seats	5
Registration number	AD5466
Model	X3
Rate per kilometer	2400

Submit Refresh Cancel

On pressing “Refresh” button:

Vehicle Entry

Date 30-03-2015

Vehicle type	
Vehicle ID	1007
Make	
Number of seats	
Registration number	
Model	
Rate per kilometer	

Submit Refresh Cancel

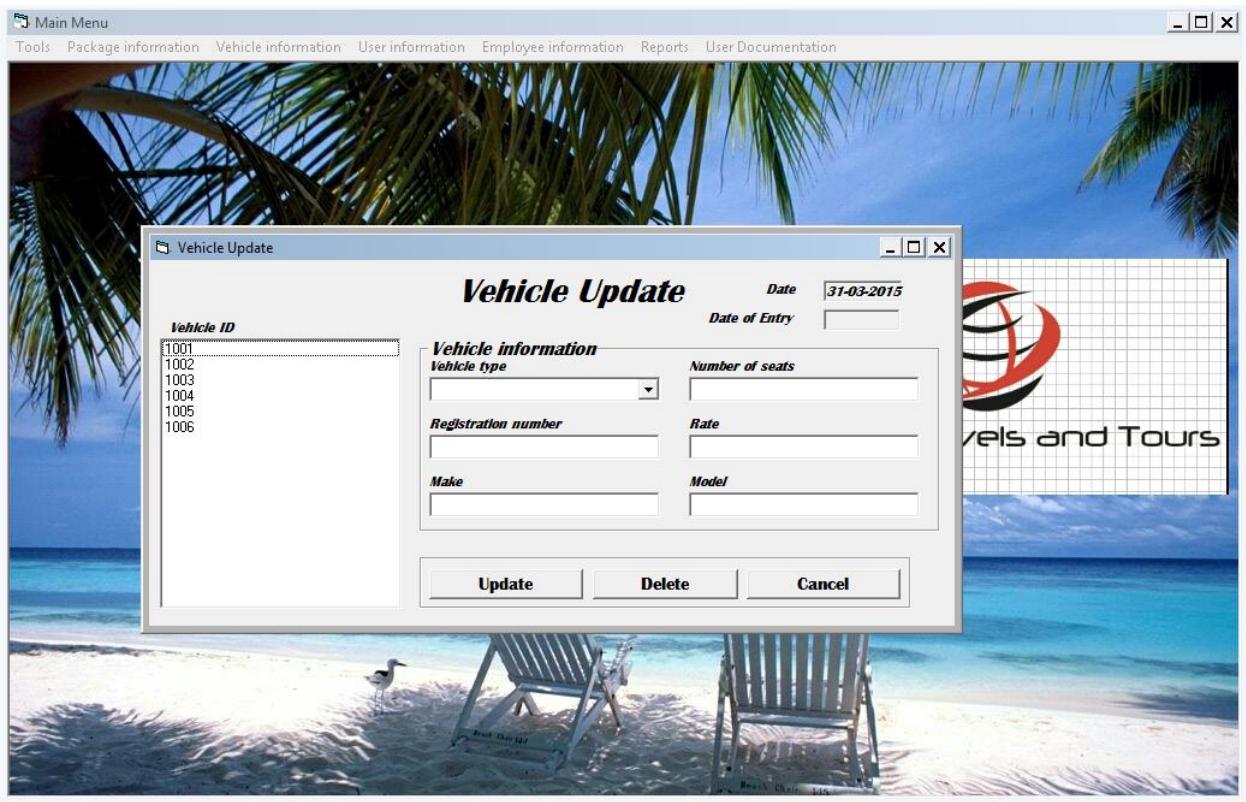
All the fields are emptied.

Vehicle ID is reloaded.

Test #85 Testing if Vehicle Update form opens on selection from Main Menu and list of Vehicle IDs is loaded from database.

Test data used	Select “Vehicle Update” form from Vehicle information section of database.
Type of data	
Outcome	Vehicle Update form appears. List of Vehicle IDs is loaded.

Travel Agency Management System



The screenshot shows a window titled "Vehicle Update". On the left, there is a list box labeled "Vehicle ID" containing the numbers 1001 through 1006. An arrow points from a callout box labeled "Vehicle IDs are loaded." to this list. To the right of the list is a form with the title "Vehicle Update". The form includes fields for "Vehicle type", "Number of seats", "Registration number", "Rate", "Make", and "Model". At the bottom are buttons for "Update", "Delete", and "Cancel". Above the form, there are two date fields: "Date" set to "31-03-2015" and "Date of Entry" set to an empty field.

Test #86 Testing if the fields on the form are automatically filled on selecting an ID from the list.

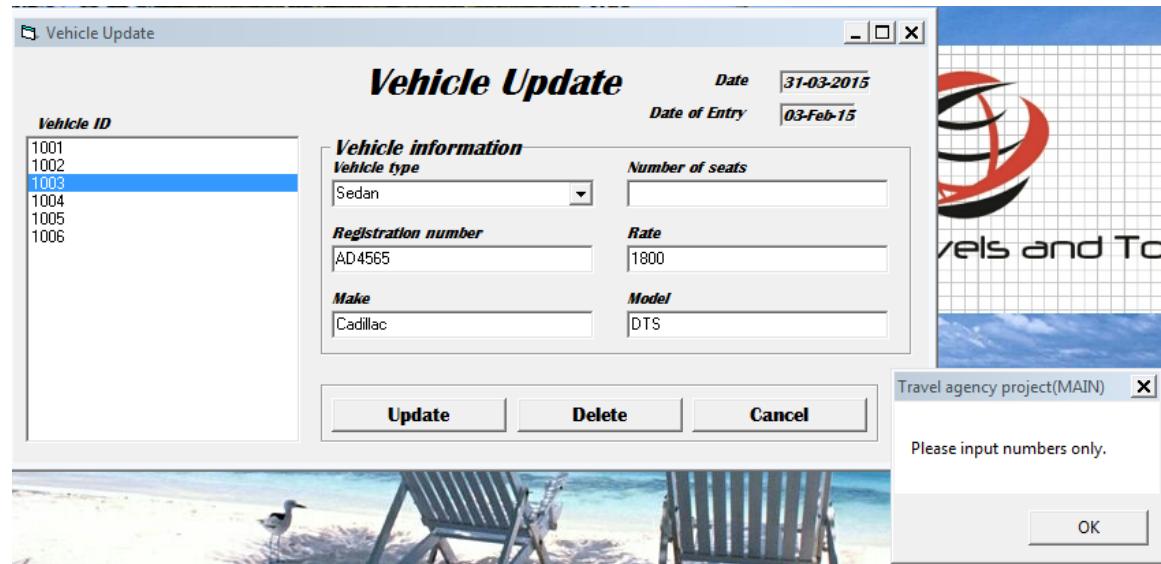
Test data used	Vehicle ID: 1003
Type of data	Normal
Outcome	All the fields are filled.

The screenshot shows the same "Vehicle Update" window as before. The "Vehicle ID" list now has item 1003 selected, highlighted with a blue selection bar. Arrows point from this selected ID to the corresponding entries in the "Vehicle information" form: "Vehicle type" (Sedan), "Number of seats" (4), "Registration number" (AD4565), "Rate" (1800), "Make" (Cadillac), and "Model" (DTS). A callout box on the right states: "The fields are automatically filled on selection of a Vehicle ID from the list."

Test #87 Testing if error message shown if non-numerical data is provided in fields such as rate or number of seats.

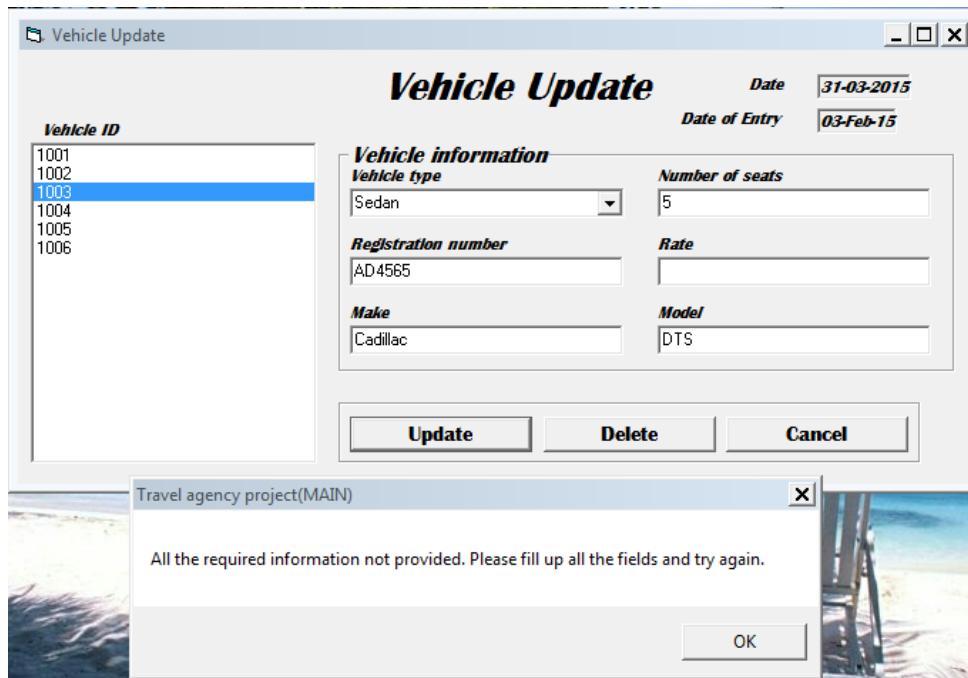
Test data used	Number of seats: d
Type of data	Abnormal
Outcome	All the fields are filled.

The error message appears before the input as the programming is done in the keypress event of the textbox



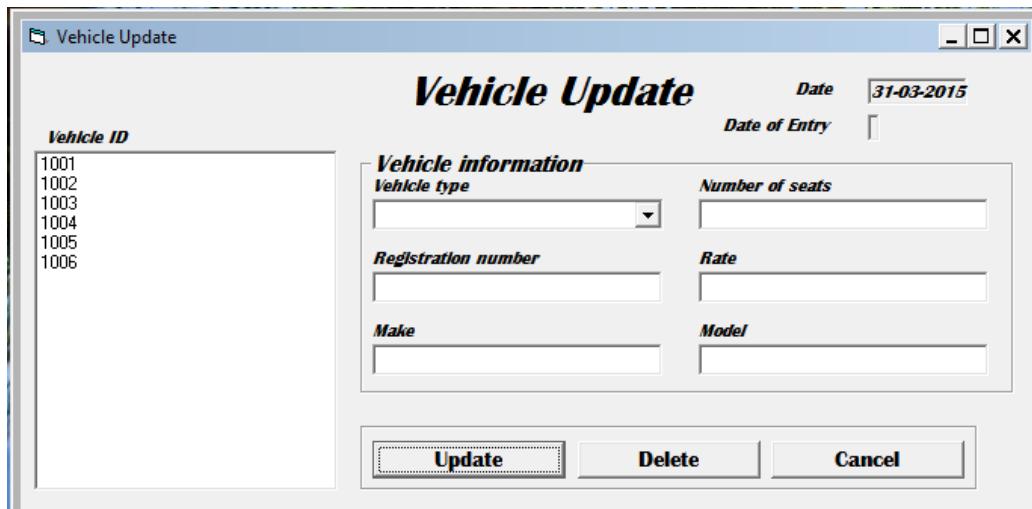
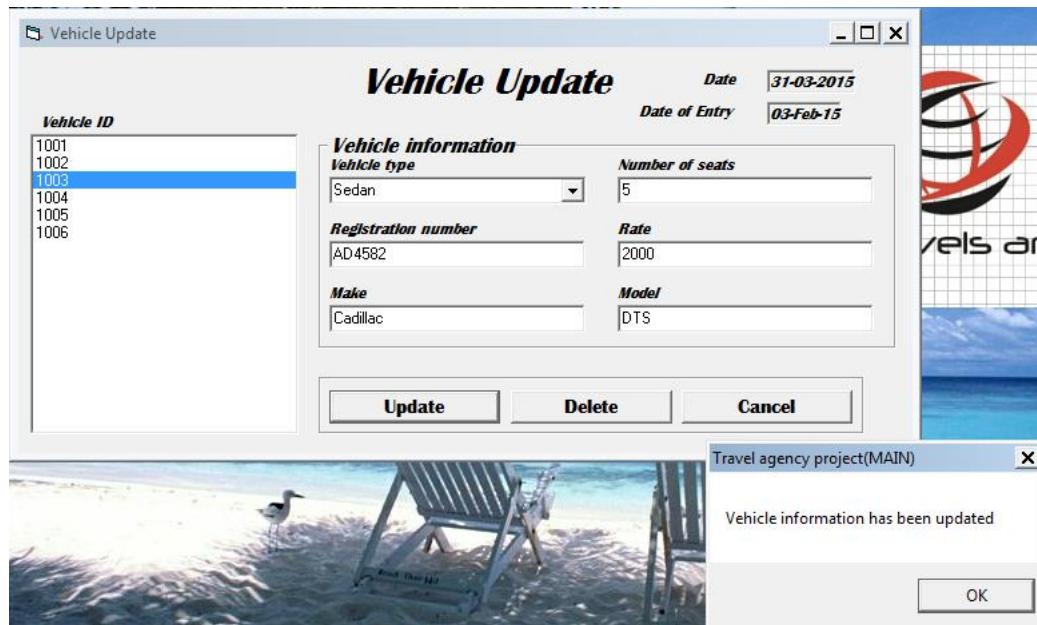
Test #88 Testing if error message shown if any of the fields are left empty on pressing Update button.

Test data used	Vehicle type: Sedan Number of seats: 5 Registration number: AD4565 Rate: "" Make: Cadillac Model: DTS
Type of data	Normal
Outcome	Error message shown saying none of the fields can be left empty.



Test #89 Testing if confirmation message shown on updating of a record, followed by emptying of the fields on the form and the update of record in the database table on pressing Update button, provided none of the fields are left empty and all the information provided is valid.

Test data used	Vehicle ID: 1003 Registration number: AD4582 Rate: 2000 Click "Update" button
Type of data	Normal
Outcome	Confirmation message shown on update of data. All the fields in the database are cleared. The record is updated in the database table.



Travel Agency Management System

Vehicle Entry

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input type="checkbox"/>	
Sedan	1003	AD4565	Cadillac	DTS	4	1800	03-Feb-15	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	

Vehicle Entry

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input type="checkbox"/>	
Sedan	1003	AD4582	Cadillac	DTS	5	2000	03-Feb-15	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	

Test #90 Testing if confirmation message shown on deletion of a record from the database, followed by emptying of the fields on the form and updating of the record on pressing the Delete button, provided none of the fields are left empty.

Test data used	Vehicle ID: 1003 Click "Delete" button
Type of data	Normal
Outcome	Confirmation message for the deletion of the record. All the fields are emptied. The record is updated in the database.

Travel Agency Management System

Vehicle Update

Vehicle ID	Vehicle type	Number of seats
1001	Sedan	5
1002		
1003		
1004		
1005		
1006		

Vehicle information

Registration number	Rate
AD4582	2000
Make	Model
Cadillac	DTS

Buttons: Update, Delete, Cancel

Message: Record has been deleted

OK

Vehicle Update

Vehicle ID	Vehicle type	Number of seats
1001		
1002		
1004		
1005		
1006		

Vehicle information

Registration number	Rate
Make	Model

Buttons: Update, Delete, Cancel

travel agency database : Database (Access 2007) - Microsoft Access

Vehicle_entry

Vehicle_type	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input type="checkbox"/>	
Sedan	1003	AD4582	Cadillac	DTS	5	2000	03-Feb-15	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	
*									

Test #91 Testing to check if Vehicle Booking form is opened on selection from Main Menu, Booking ID generated for new transaction and Vehicle types are loaded.

Test data used	Select Vehicle Booking form from Main Menu.
Type of data	Normal
Outcome	Vehicle Booking form opens.



Vehicle Booking

Date

Customer information		
Booking ID <input style="outline: 2px solid red;" type="text" value="1424"/> Address <input type="text"/>		
Customer Name <input type="text"/>		
Contact <input type="text"/>		
Journey date <input type="text" value="31-Mar-15"/> Drop off date <input type="text" value="31-Mar-15"/>		
Vehicle information		
Vehicle type <input type="text"/>	Registration number <input type="text"/>	
Make <input type="text"/>	Model <input type="text"/>	VehicleID <input type="text"/>
Rate per kilometer <input type="text"/>	Advance paid <input type="text"/>	
Number of seats <input type="text"/>	Driver name <input type="text"/>	
Starting kilometer <input type="text"/>		
<input type="button" value="Confirm"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>		

Vehicle Booking

Customer information

Booking ID: 1425 Address: []

Customer Name: []

Contact: []

Journey date: 11-Apr-15 Drop off date: 11-Apr-15

Vehicle information

Vehicle type: [Sedan, SUV] Registration number: []

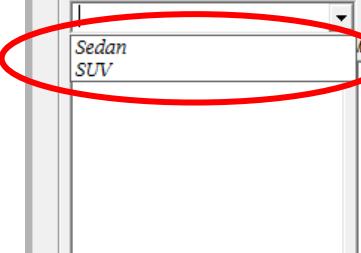
VehicleID: []

Rate per kilometer: [] Advance paid: []

Number of seats: [] Driver name: []

Starting kilometer: []

Buttons: Confirm | Refresh | Cancel



Test #92 Testing to check if error message shown if non-numerical data is entered as contact number, advance paid or starting kilometer.

Test data used	Advance paid: p
Type of data	Abnormal
Outcome	Testing to check if error message shown if non-numerical data is entered as contact number, advance paid or starting kilometer.

The screenshot shows the 'Vehicle Booking' application window. In the 'Customer information' section, the 'Contact' field contains the letter 'p'. A modal dialog box titled 'Travel agency project(MAIN)' displays the message 'Please input numbers only.' with an 'OK' button. The application interface includes fields for Booking ID (1424), Customer Name, Address, Journey date (31-Mar-15), Drop off date (31-Mar-15), Vehicle type, Registration number, Make, Model, VehicleID, Rate per kilometer, Advance paid, Number of seats, Driver name, and Starting kilometer. At the bottom are buttons for Confirm, Refresh, and Cancel.

Test #93 Testing to check if error message shown if any of the fields are left empty on pressing Confirm button

Test data used	Some of the fields are left empty. Click "Confirm"
Type of data	
Outcome	Error message appears saying none of the fields can be left empty.

The screenshot shows a 'Vehicle Booking' application window. On the left, there's a 'Customer information' section with fields for Booking ID (1424), Customer Name (Steven Jones), Contact, Journey date (31-Mar-15), and Drop off date (31-Mar-15). Below it is a 'Vehicle information' section with dropdowns for Vehicle type (SUV), Make (BMW), Model (X1/X3), Registration number (AD2589), VehicleID (1006), Rate per kilometer (2400), Advance paid, Number of seats (5), Driver name (Lu White), and Starting kilometer. At the bottom are 'Confirm', 'Refresh', and 'Cancel' buttons. A status bar at the bottom left shows 'B4 Words: 24,305'. On the right, a modal dialog box titled 'Travel agency project(MAIN)' contains the message 'Some of the required fields are not filled. Please fill up all of the fields and try again.' with an 'OK' button.

Test #94 Testing to check that a Vehicle that is unavailable does not appear in the list of Vehicle Booking form.

Test data used	Select Vehicle type: Sedan Make: Toyota
Type of data	Normal
Outcome	The Vehicle does not appear in the list.

Travel Agency Management System

While none of them are booked:

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input type="checkbox"/>	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	
*									

When one of the Vehicles is booked:

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	
*									

Travel Agency Management System

Vehicle Booking

Customer information

Booking ID: 1424 Address:

Customer Name:

Contact:

Journey date: 31-Mar-15 Drop off date: 31-Mar-15

Vehicle information

Vehicle type: Sedan Registration number:

Make: Toyota Model: Premio

Rate per kilometer: Advance paid:

Number of seats: Driver name:

Starting kilometer:

Buttons: Confirm, Refresh, Cancel

The vehicle that is now booked does not appear in the list

Test #95 Testing if error message shown if number of digits of contact number is invalid.

Test data used	Contact number: 018155856644
Type of data	Normal
Outcome	Error message asking for valid input.

Vehicle Booking

Customer information

Booking ID: 1424 Address: 23rd, Golpahar, Chittagong

Customer Name: Helen Greene Contact: 018155856644

Journey date: 31-Mar-15 Drop off date: 09-Apr-15

Vehicle information

Vehicle type: Sedan Registration number: AD2133

Make: Toyota Model: Premio VehicleID: 1001

Rate per kilometer: 1200 Advance paid: 2500

Number of seats: 5 Driver name: Dylan Walker

Starting kilometer: 321354

Buttons: Confirm, Refresh, Cancel

A modal dialog box is displayed with the title "Travel agency project(MAIN)". The message inside says "Number of digits used for contact number is invalid." There is an "OK" button at the bottom right of the dialog.

Test #96 Testing to check if Confirmation message is provided for addition of record to the database, availability of vehicle is changed in Vehicle Entry database table, the new record is added to the Vehicle Booking database table, the fields on the form are emptied, the Booking ID is updated and a token is generated on pressing Confirm button, provided none of the fields are left empty and all valid information is provided.

Test data used	Customer name: Helen Greene Contact number: 01815585664 Address: 23rd, Golpahar, Chittagong Journey date: 31-03-2015 Drop-off-date: 09-04-15 Vehicle type: SUV Make: BMW Model: X3 Vehicle ID: 1006 Advance paid: \$2500 Driver name: Dylan Walker Starting kilometer: 321354
Type of data	Normal
Outcome	-Confirmation message shown for successful booking. -“IsBooked?” section of Vehicle Entry database table is updated. -New record added to the Vehicle Booking database table. -All the fields emptied and New Booking ID generated for new transaction. -Token generated.

The screenshot shows the 'Vehicle Booking' application window. The 'Customer information' section contains fields for Booking ID (1424), Customer Name (Helen Greene), Contact (01815585664), Journey date (31-Mar-15), and Drop off date (09-Apr-15). The 'Vehicle information' section includes dropdowns for Vehicle type (SUV), Make (BMW), Model (X3), Registration number (AD2589), and VehicleID (1006). At the bottom, there are fields for Rate per kilometer (2400), Advance paid (2500), Number of seats (5), Driver name (Dylan Walker), and Starting kilometer (321354). A confirmation message box is overlaid on the window, stating 'Booking has been made successfully.' with an 'OK' button. The application interface features a blue header bar and a decorative background image of a beach and palm trees.

Travel Agency Management System

Vehicle Booking

Customer information

Booking ID: 1425 (highlighted)

Customer Name: []

Contact: []

Journey date: 31-Mar-15

Drop off date: 31-Mar-15

Vehicle information

Vehicle type: []

Make: []

Model: []

Registration number: []

VehicleID: []

Rate per kilometer: []

Advance paid: []

Number of seats: []

Driver name: []

Starting kilometer: []

Buttons: Confirm, Refresh, Cancel

Vehicle Entry and Vehicle Booking database tables before transaction:

travel agency database : Database (Access 2007) - Microsoft Access

Vehicle_entry

Vehicle_type	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
*	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	

Travel Agency Management System

This screenshot shows the Microsoft Access application window with the 'Vehicle_booking' table selected. The table has columns: VehicleBookingID, VehicleID, Customer_name, Customer_address, Contact, Journey_date, Drop_off_da, Driver_name, Date_of_bor, Registration_number, and Veh. A row for booking ID 1423 is highlighted, showing details for Michelle Winger. An oval highlights the entire row.

VehicleBookingID	VehicleID	Customer_name	Customer_address	Contact	Journey_date	Drop_off_da	Driver_name	Date_of_bor	Registration_number	Veh
1423	1002	Michelle Winger	23rd, Sholoshohor, Chittagong	0182554558	21-Feb-15		Tywell Bittersman	23-Jan-15	AD8646	Sedan

Vehicle Booking table
before booking is made.

Vehicle Entry and Vehicle Booking database tables after transaction:

This screenshot shows the Microsoft Access application window with the 'Vehicle_entry' table selected. The 'IsBooked?' column for vehicle ID 1006 is checked, indicating it is booked. An arrow points from this checked box to a callout box explaining its meaning.

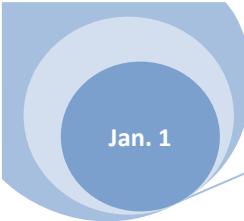
Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input checked="" type="checkbox"/>	

The “IsBooked?” section is ticked which means the Vehicle is now unavailable

This screenshot shows the Microsoft Access application window with the 'Vehicle_booking' table selected. Two rows are visible: one for booking ID 1423 (VehicleID 1002) and another new row for booking ID 1424 (VehicleID 1006). An oval highlights the second row.

VehicleBookingID	VehicleID	Customer_name	Customer_address	Contact	Journey_date	Drop_off_da	Driver_name	Date_of_bor	Registration_number	Veh
1423	1002	Michelle Winger	23rd, Sholoshohor, Chittagong	0182554558	21-Feb-15		Tywell Bittersman	23-Jan-15	AD8646	Sedan
1424	1006	Helen Greene	23rd, Golpahar, Chittagong	01815585664	31-Mar-15	09-Apr-15	Dylan Walker	31-Mar-15	AD2589	SUV

Vehicle Booking table after booking is made.



Jan. 1

Travel Agency Management System

Test #97 Testing to check if all the fields are emptied from the form on pressing Refresh button

Test data used	Click “Refresh” button while some of the fields are filled.
Type of data	Normal
Outcome	All the fields are emptied.

Before pressing “Refresh” button:

Customer information

Booking ID: 1425 Address: []

Customer Name: Thomas Goodwin

Contact: 01212253451

Journey date: 31-Mar-15 Drop off date: 31-Mar-15

Vehicle information

Vehicle type	Registration number
Sedan	AD2133
Make	VehicleID
Toyota	1001
Rate per kilometer	Advance paid
Number of seats	Driver name
Starting kilometer	

Buttons: Confirm, Refresh, Cancel

After pressing “Refresh” button:

Customer information

Booking ID: 1425 Address: []

Customer Name: []

Contact: []

Journey date: 31-Mar-15 Drop off date: 31-Mar-15

Vehicle information

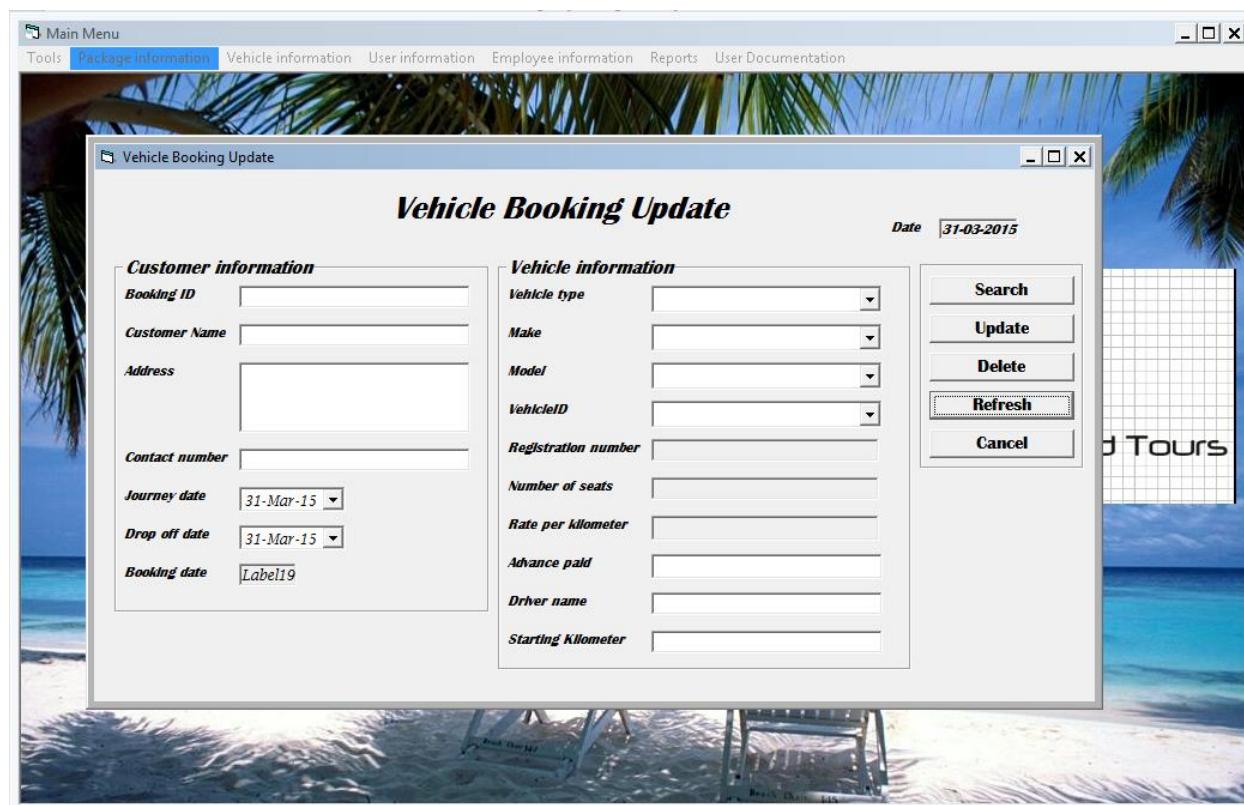
Vehicle type	Registration number
[]	[]
Make	VehicleID
[]	[]
Rate per kilometer	Advance paid
Number of seats	Driver name
Starting kilometer	

Buttons: Confirm, Refresh, Cancel

Test #98 Testing if Vehicle Booking Update form appears on selection from the Main Menu.

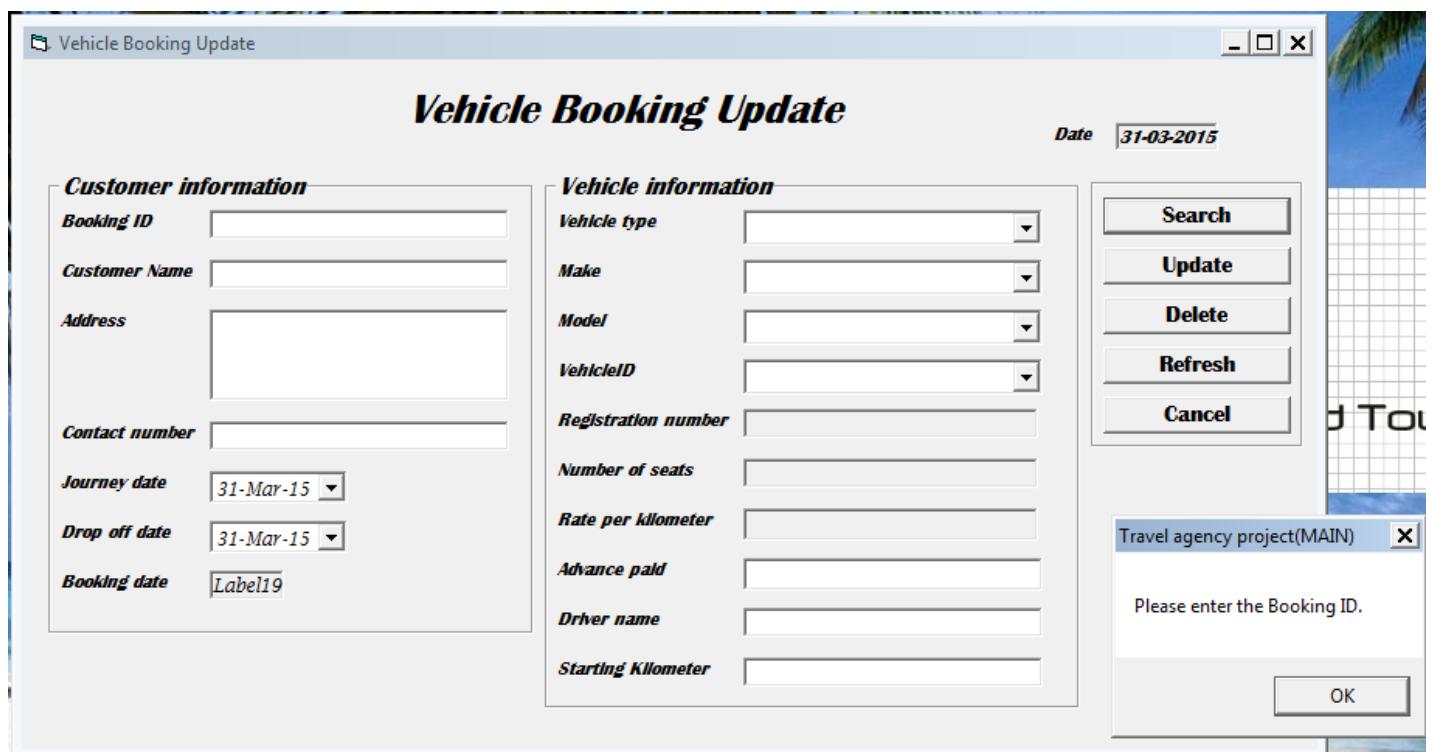
Test data used	Selection of Vehicle Booking Update from Main Menu.
Type of data	Normal
Outcome	The form appears.

Travel Agency Management System



Test #99 Testing if error message appears if search button pressed without providing Booking ID.

Test data used	Booking ID is left empty. Click "Search"
Type of data	
Outcome	Error message appears asking for Booking ID.



Test #100 Testing if error message appears if Booking ID provided is not present in the database.

Test data used	Booking ID: 2153 Click "Search"
Type of data	Normal
Outcome	Error message appears saying Booking ID being searched is not found.

The screenshot shows the 'Vehicle Booking Update' window. On the right, there is a vertical stack of buttons labeled 'Search', 'Update', 'Delete', 'Refresh', and 'Cancel'. Below these buttons is a message box with the title 'Travel agency project(MAIN)'. The message in the box reads: 'Booking ID being searched is not found. Please enter correct Booking ID and try again.' At the bottom right of the message box is an 'OK' button.

Customer information		Vehicle information	
Booking ID	2153	Vehicle type	<input type="text"/>
Customer Name	<input type="text"/>	Make	<input type="text"/>
Address	<input type="text"/>	Model	<input type="text"/>
Contact number	<input type="text"/>	VehicleID	<input type="text"/>
Journey date	31-Mar-15	Registration number	<input type="text"/>
Drop off date	31-Mar-15	Number of seats	<input type="text"/>
Booking date	Label19	Rate per kilometer	<input type="text"/>
		Advance paid	<input type="text"/>
		Driver name	<input type="text"/>
		Starting Kilometer	<input type="text"/>

Test #101 Testing if all the information is loaded onto the form if correct Booking ID is provided.

Test data used	Booking ID: 1423 Click "Search"
Type of data	Normal
Outcome	All the fields in the forms are loaded from the database and filled.

The screenshot shows the 'Vehicle Booking Update' window with the correct Booking ID (1423) entered. All fields in both the 'Customer information' and 'Vehicle information' sections are now populated with data, matching the test results table above.

Customer information		Vehicle information	
Booking ID	1423	Vehicle type	Sedan
Customer Name	Michelle Winger	Make	Toyota
Address	23rd, Sholoshohor, Chittagong	Model	Vista
Contact number	0182554558	VehicleID	1002
Journey date	21-Feb-15	Registration number	AD8646
Drop off date	04-Aug-15	Number of seats	5
Booking date	23-Jan-15	Rate per kilometer	1400
		Advance paid	2500
		Driver name	Tywell Bittersman
		Starting Kilometer	13213

Test #102 Testing if error message provided if Booking ID is changed and Update button is pressed.

Test data used	Booking ID: 1480 Click "Update"
Type of data	Normal
Outcome	Error message shown saying Booking ID cannot be changed.

The screenshot shows the 'Vehicle Booking Update' application window. On the left, there is a 'Customer information' section with fields for Booking ID (1480), Customer Name (Michelle Winger), Address (23rd, Sholashohor, Chittagong), Contact number (01825545581), Journey date (21-Feb-15), Drop off date (04-Aug-15), and Booking date (23-Jan-15). On the right, there is a 'Vehicle information' section with fields for Vehicle type (Sedan), Make (Toyota), Model (Vista), VehicleID (1002), Registration number (AD8646), Number of seats (5), Rate per kilometer (1400), Advance paid (2500), Driver name (Tywell Bittersman), and Starting Kilometer (13213). To the right of these sections is a vertical toolbar with buttons for Search, Update, Delete, Refresh, and Cancel. A date field at the top right shows 31-03-2015. A small error message box titled 'Travel agency project(MAIN)' appears in the bottom right corner, stating 'Booking ID cannot be changed.' with an 'OK' button.

Test #103 Testing if Confirmation message is provided on Updating a record, availability options of vehicles are changed in the Vehicle Entry database table(in case it is changed), the record is updated in Vehicle Booking database table and the fields on the form are automatically emptied on pressing Update button.

Test data used	Booking ID: 1423 Click "Search" Vehicle type: SUV Make: BMW Model: X1 Vehicle ID: 1004 Click "Update"
Type of data	Normal
Outcome	Confirmation message on update of record information. Changes made to "IsBooked?" field of the Vehicle Entry database. The record in the Vehicle Booking database is updated. The fields on the form are emptied.

Vehicle Booking Update

Date **31-03-2015**

Customer information		Vehicle information	
Booking ID	1423	Vehicle type	SUV
Customer Name	Michelle Winger	Make	BMW
Address	23rd, Sholoshohor, Chittagong	Model	X1
Contact number	01825545581	VehicleID	1004
Journey date	21-Feb-15	Registration number	AD9755
Drop off date	04-Aug-15	Number of seats	5
Booking date	23-Jan-15	Rate per kilometer	2000
		Advance paid	2500
		Driver name	Tywell Bittersman
		Starting Kilometer	13213

Search

Update

Delete

Refresh

Cancel

Travel agency project(MAIN) **OK**

Booking has been updated successfully.

Vehicle Booking Update

Date **31-03-2015**

Customer information		Vehicle information	
Booking ID		Vehicle type	
Customer Name		Make	
Address		Model	
Contact number		VehicleID	
Journey date	31-Mar-15	Registration number	
Drop off date	31-Mar-15	Number of seats	
Booking date	23-Jan-15	Rate per kilometer	
		Advance paid	
		Driver name	
		Starting Kilometer	

Search

Update

Delete

Refresh

Cancel

Jan. 1

Travel Agency Management System

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input checked="" type="checkbox"/>	

Toyota Vista unavailable before update.

VehicleBookingID	VehicleID	Customer_name	Customer_address	Contact	Journey_date	Drop_off_da	Driver_name	Date_of_bor	Registration_number	Vehicle
1423	1002	Michelle Winger	23rd, Sholoshohor, Chittagong	01825545581	21-Feb-15	04-Aug-15	Tywell Bittersman	31-Mar-15	AD8646	Sedan
1424	1006	Helen Greene	23rd, Golpahar, Chittagong	0181558564	31-Mar-15	09-Apr-15	Dylan Walker	31-Mar-15	AD2589	SUV

Vehicle Booking database before update.

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input checked="" type="checkbox"/>	

Toyota Vista becomes available and BMW X1 becomes unavailable after update.

VehicleBookingID	VehicleID	Customer_name	Customer_address	Contact	Journey_date	Drop_off_da	Driver_name	Date_of_bor	Registration_number	Vehicle
1423	1004	Michelle Winger	23rd, Sholoshohor, Chittagong	01825545581	21-Feb-15	04-Aug-15	Tywell Bittersman	31-Mar-15	AD9755	
1424	1006	Helen Greene	23rd, Golpahar, Chittagong	0181558564	31-Mar-15	09-Apr-15	Dylan Walker	31-Mar-15	AD2589	SUV

Vehicle Booking database after update.

Test #104 Testing if confirmation message appears on deleting a record, the availability of the Vehicle that was booked is changed in Vehicle Entry database table, the record is deleted from Vehicle Booking database table and the fields on the form are emptied by pressing Delete button of Vehicle Booking Update form.

Test data used	Booking ID: 1424 Click "Search" Click "Delete"
Type of data	Normal
Outcome	Confirmation message for deletion on the record. The fields on the form are emptied. The record is deleted from Vehicle Booking database. The Vehicle that was booked is made available in Vehicle Entry database.

The screenshot shows the 'Vehicle Booking Update' application window. On the right, a vertical toolbar has 'Delete' highlighted. A confirmation dialog box titled 'Travel agency project(MAIN)' displays the message 'Booking has been deleted.' with an 'OK' button. The main form contains two sections: 'Customer information' and 'Vehicle information'. The 'Vehicle information' section includes dropdown menus for 'Vehicle type', 'Make', 'Model', and 'VehicleID', and text input fields for 'Registration number', 'Number of seats', 'Rate per kilometer', 'Advance paid', 'Driver name', and 'Starting Kilometer'. The 'Customer information' section includes text input fields for 'Booking ID' (1424), 'Customer Name' (Helen Greene), 'Address' (23rd, Golpahar, Chittagong), 'Contact number' (01815585664), 'Journey date' (31-Mar-15), 'Drop off date' (09-Apr-15), and 'Booking date' (31-Mar-15). The date field at the top right shows '31-03-2015'.

This screenshot shows the same 'Vehicle Booking Update' application window after the delete operation. The 'Delete' button in the toolbar is now grayed out, indicating it is disabled. The confirmation dialog box is no longer present. The form fields are now empty, matching the outcome described in the test case.

Travel Agency Management System

Vehicle_booking table data:

VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?
1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>
1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>
1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>
1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>
1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input checked="" type="checkbox"/>

The Vehicle is unavailable before
deletion. It is booked.

Vehicle_booking table data:

VehicleBookingID	VehicleID	Customer_name	Customer_address	Contact	Journey_date	Drop_off_da	Driver_name	Date_of_bor	Registration_number	Veh
1423	1002	Michelle Winger	23rd, Shososhohor, Chittagong	01825545581	21-Feb-15		Tywell Bittersman	31-Mar-15	AD8646	Sedan
1424	1006	Helen Greene	23rd, Golpahar, Chittagong	01815585664	31-Mar-15	09-Apr-15	Dylan Walker	31-Mar-15	AD2589	SUV

Vehicle Booking database before
deletion of the record.

Vehicle_booking table data:

VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?
1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>
1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>
1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>
1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>
1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>

The vehicle is made available
after deletion.

Vehicle_booking table data:

VehicleBookingID	VehicleID	Customer_name	Customer_address	Contact	Journey_date	Drop_off_da	Driver_name	Date_of_bor	Registration_number	Veh
1423	1002	Michelle Winger	23rd, Shososhohor, Chittagong	01825545581	21-Feb-15		Tywell Bittersman	31-Mar-15	AD8646	Sedan
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted

Vehicle Booking database after
deletion of the record.

Test #105 Testing to check if Vehicle information is loaded along with “Late fine” calculated and Booking charge generated on provision of Booking ID in Vehicle Billing form on pressing Search button.

Test data used	Booking ID: 1423 Click “Search”
Type of data	Normal
Outcome	Some of the fields are filled. Late fine calculated. Booking charge generated.

Test #106 Testing to check if Cost of travel is calculated automatically on provision of Ending kilometer.

Test data used	Ending kilometer: 13220 Move on to next field(Programming done in Lostfocus event)
Type of data	Normal
Outcome	Cost of travel is generated.

Test #107 Testing if Bill is generated on pressing “Generate Bill” button after providing all the necessary information of the Vehicle Billing form.

Test data used	Using data from last test. Driver charge: 4000 Additional charges: 2600 Discount: 500 Click “Generate Bill”
Type of data	Normal
Outcome	All the calculations are made and a bill is shown on the screen.

Test #108 Testing if the Billing information is added to the “Vehicle Billing” database table and the Vehicle for which the bill is being made is made reavailable on pressing “Save” button, provided all the information input is valid and none of the fields are left empty.

Test data used	Using data from last test.
Type of data	Normal
Outcome	All the necessary billing information is added to the database table. The vehicle which was booked is made reavailable.

Vehicle Billing database table before addition of record.

Jan. 1

Travel Agency Management System

This screenshot shows the Microsoft Access Datasheet view for the 'Vehicle_Billing' table. The table has columns: BookingID, Customer_name, Customer_address, VehicleID, Kilometer_travelled, Rate_per_kilometer, Cost_of_travel, Journey_date, and Drop_off_date. A new record has been added with values: BookingID 1423, Customer_name Michelle Winger, Customer_address 23rd, Sholoshohor, Chit 1002, VehicleID null, Kilometer_travelled 7, Rate_per_kilometer \$1,400.00, Cost_of_travel \$9,800.00, Journey_date 21-Feb-15, and Drop_off_date 29-Mar-15.

BookingID	Customer_name	Customer_address	VehicleID	Kilometer_travelled	Rate_per_kilometer	Cost_of_travel	Journey_date	Drop_off_date
1423	Michelle Winger	23rd, Sholoshohor, Chit 1002		7	\$1,400.00	\$9,800.00	21-Feb-15	29-Mar-15

Vehicle Billing database table after addition of record.

This screenshot shows the Microsoft Access Datasheet view for the 'Vehicle_entry' table. The table has columns: Vehicle_typ, VehicleID, RegistrationNo, Make, Model, NSeats, Rate_per_km, Date_of_entry, IsBooked?, and Add New Field. The 'IsBooked?' column contains checkboxes. The checkbox for the row where VehicleID is 1002 and Date_of_entry is 28-Mar-15 is checked.

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input checked="" type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	

Vehicle Entry database before Billing.

This screenshot shows the Microsoft Access Datasheet view for the 'Vehicle_entry' table. The table has columns: Vehicle_typ, VehicleID, RegistrationNo, Make, Model, NSeats, Rate_per_km, Date_of_entry, IsBooked?, and Add New Field. The 'IsBooked?' column contains checkboxes. The checkboxes for the rows where VehicleID is 1002 and 1004 are now unchecked.

Vehicle_typ	VehicleID	RegistrationNo	Make	Model	NSeats	Rate_per_km	Date_of_entry	IsBooked?	Add New Field
Sedan	1001	AD2133	Toyota	Premio	5	1200	14-Mar-15	<input type="checkbox"/>	
Sedan	1002	AD8646	Toyota	Vista	5	1400	28-Mar-15	<input type="checkbox"/>	
SUV	1004	AD9755	BMW	X1	5	2000	08-Jan-15	<input type="checkbox"/>	
SUV	1005	AD3313	BMW	X1	5	2200	30-Mar-15	<input type="checkbox"/>	
SUV	1006	AD2589	BMW	X3	5	2400	30-Mar-15	<input type="checkbox"/>	

Vehicle Entry database after Billing.

Test #109 Testing if a bill is printed on pressing “Print” button.

Test data used	Click “Print” button
Type of data	Normal
Outcome	A bill is printed.

Test #110 Testing if the list of Vehicle IDs is loaded on selection of a Vehicle type on Vehicle servicing form and “Servicing history” for a particular Vehicle ID is loaded on selection of that Vehicle ID from Vehicle Servicing form.

Test data used	Vehicle type: SUV Vehicle ID: 1004
Type of data	Normal
Outcome	“Servicing History” table is loaded.

Vehicle Servicing

Vehicle type: (dropdown menu open, showing 'SUV' selected)

Date: (06-Apr-15)

Vehicle ID: (dropdown menu open, showing 'SUV' selected)

Servicing details:

Amount:

Save | **Refresh** | **Cancel**

Servicing History Preview

Date	Servicing Details	Amount

Vehicle Servicing

Vehicle type: (dropdown menu open, showing 'SUV')

Date: (06-Apr-15)

Vehicle ID: (dropdown menu open, showing '1004' selected)

Servicing details:

Amount:

Save | **Refresh** | **Cancel**

Servicing History Preview

Date	Servicing Details	Amount
06-Apr-15	Painting	500
06-Apr-15	New exhausts	1500

Test #111 Test to check if servicing details and amount are added to the Servicing history table and the database on pressing the save button

Test data used	Vehicle ID: 1004 Servicing Details: Engine repairs Amount: \$3000
Type of data	Normal
Outcome	Servicing details and amount are added to the table.

Vehicle servicing

Vehicle Servicing

Vehicle type: SUV **Date**: 06-Apr-15

Vehicle ID		
1004	Servicing details	Engine repairs
1005		
1006		

Amount: 3000

Save **Refresh** **Cancel**

Servicing History Preview

Date	Servicing Details	Amount
06-Apr-15	Painting	500
06-Apr-15	New exhausts	1500

Vehicle servicing

Vehicle Servicing

Vehicle type: SUV **Date**: 06-Apr-15

Vehicle ID		
1004	Servicing details	
1005		
1006		

Amount:

Save **Refresh** **Cancel**

Servicing History Preview

Date	Servicing Details	Amount
06-Apr-15	Painting	5
06-Apr-15	New exhausts	15
06-Apr-15	Engine repairs	30

New record added to the table.

Travel Agency Management System

Vehicle_typ	VehicleID	Servicing_dt	Amount	Date
SUV	1004	Painting	\$500.00	06-Apr-15
SUV	1004	New exhausts	\$1,000.00	06-Apr-15
*				

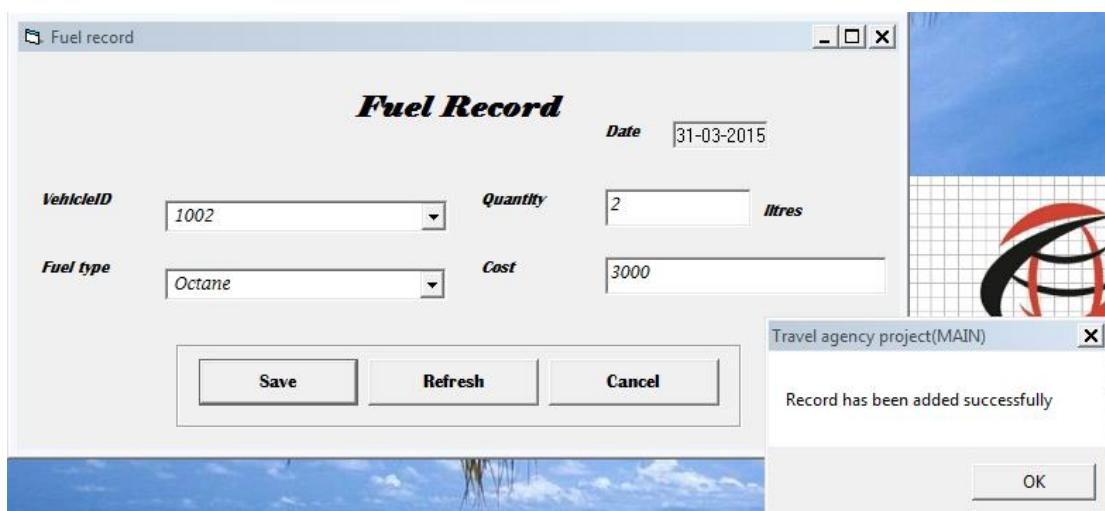
"Vehicle Servicing" database before addition of record.

Vehicle_typ	VehicleID	Servicing_dt	Amount	Date
SUV	1004	Painting	\$500.00	06-Apr-15
SUV	1004	New exhausts	\$1,000.00	06-Apr-15
*		Engine repairs	\$3,000.00	06-Apr-15

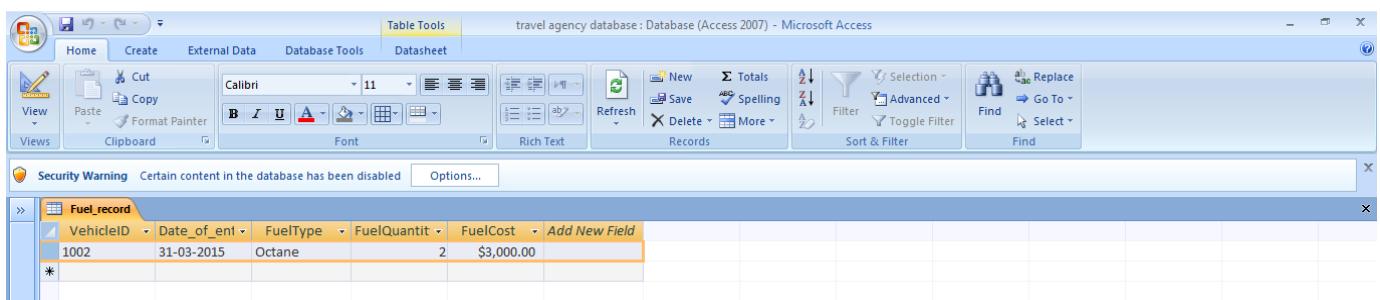
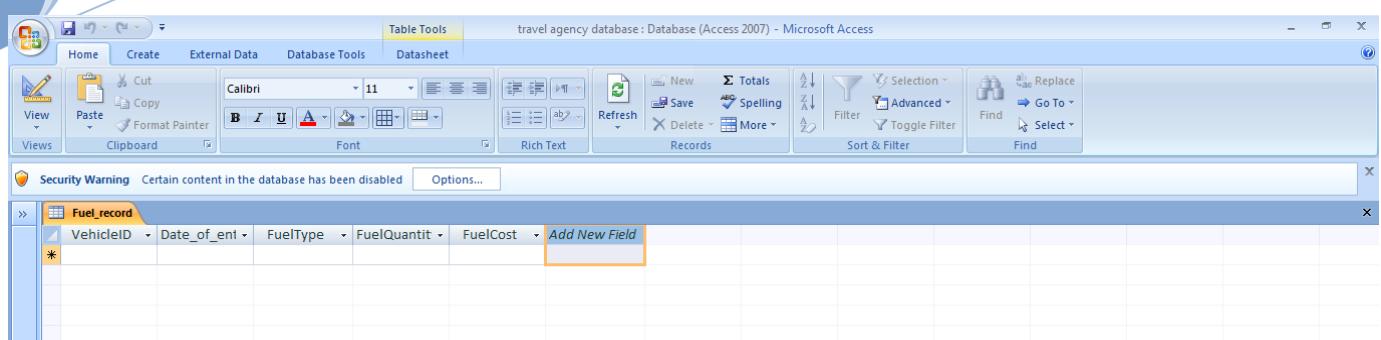
"Vehicle Servicing" database table after addition of record.

Test #112 Testing if Confirmation message is shown on addition of data to the database, Fuel record data is added to the Fuel record database table, the fields are emptied on pressing "Save" button on the Fuel Record form.

Test data used	Vehicle ID: 1002 Fuel type: Octane Quantity: 2 Cost 3000
Type of data	Normal
Outcome	Confirmation message shown for addition of data to the database.

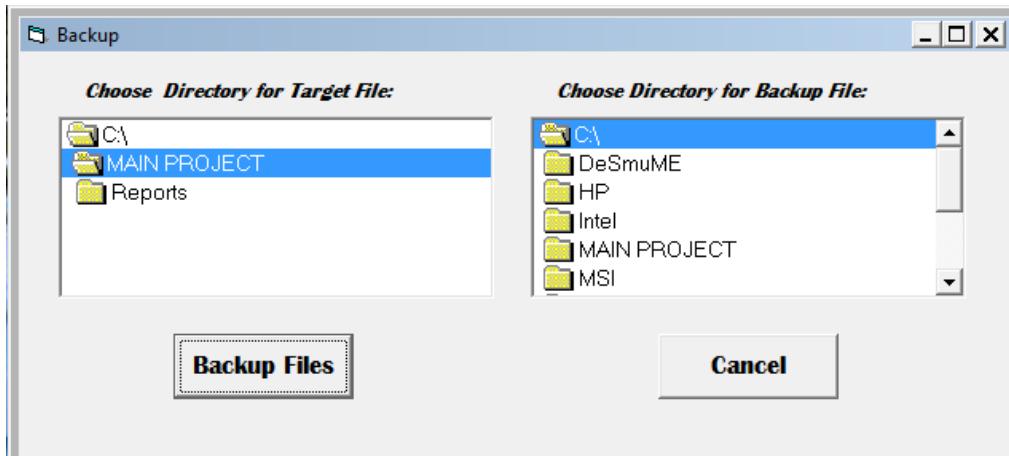


Travel Agency Management System

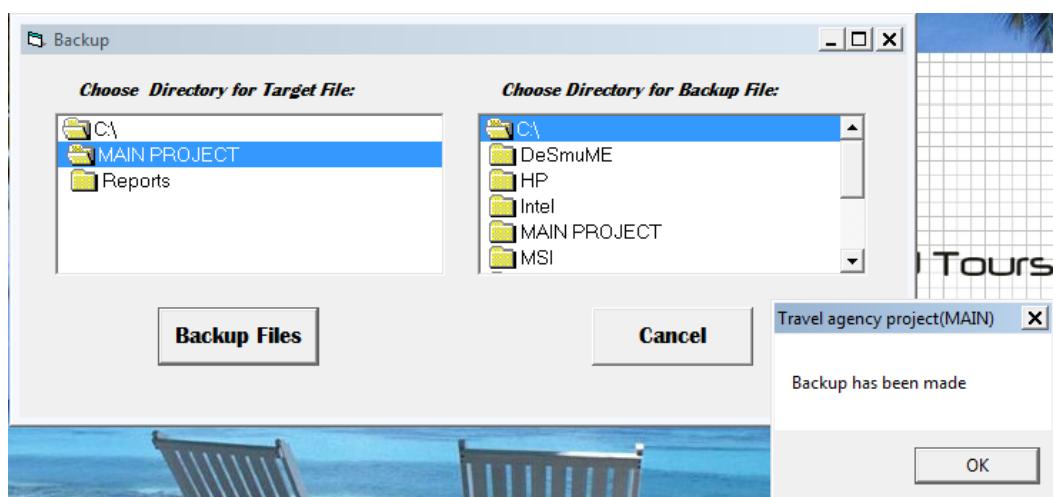
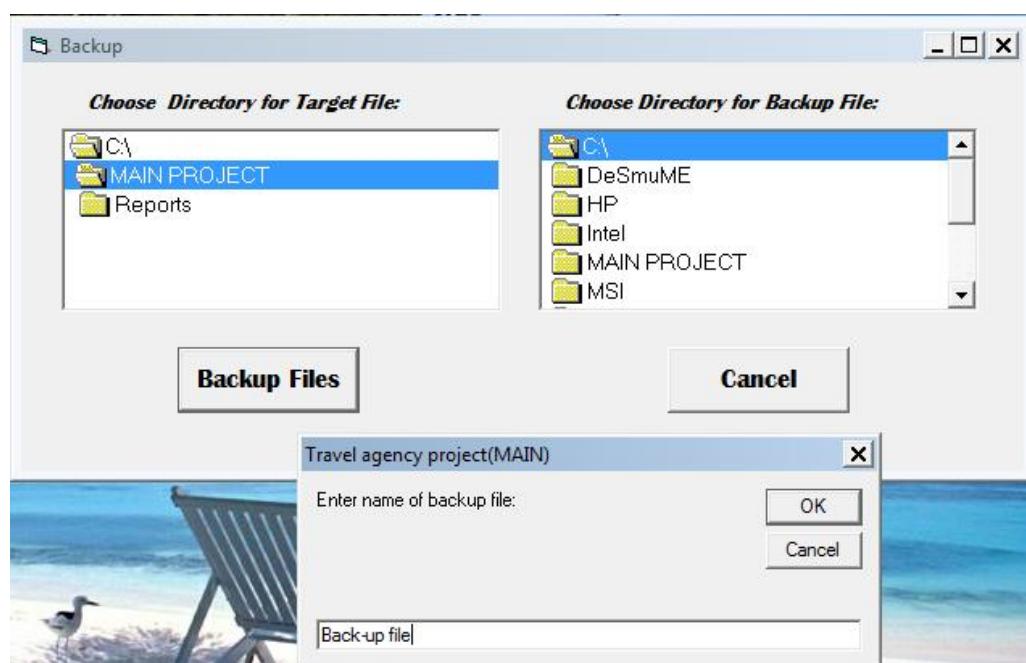
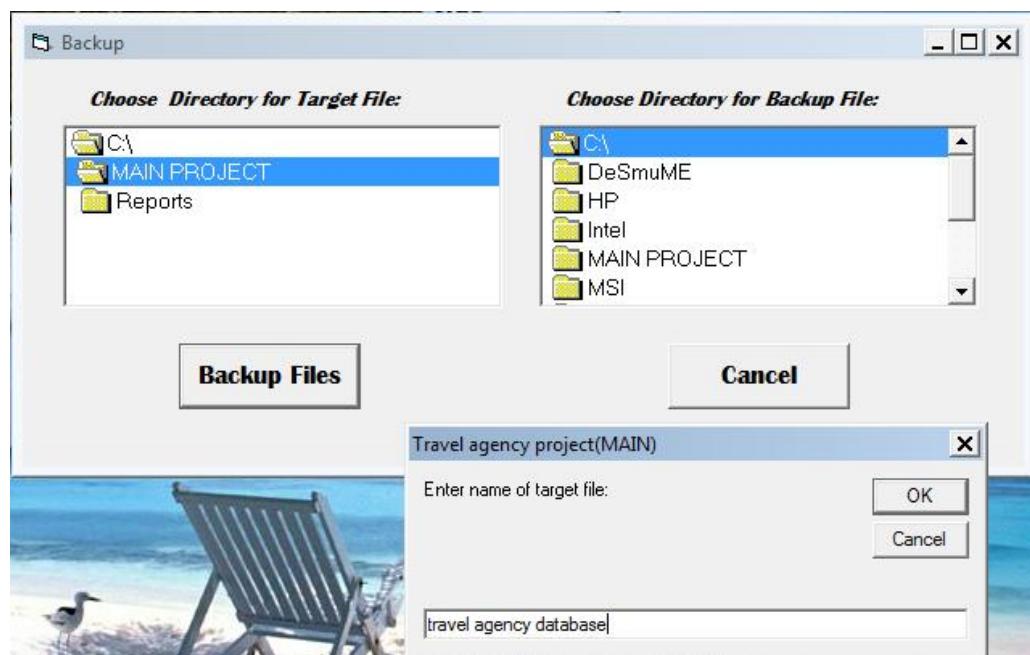


Test #113 Testing if Backup is created when “Backup Files” button is pressed on the backup form after selecting a suitable directory.

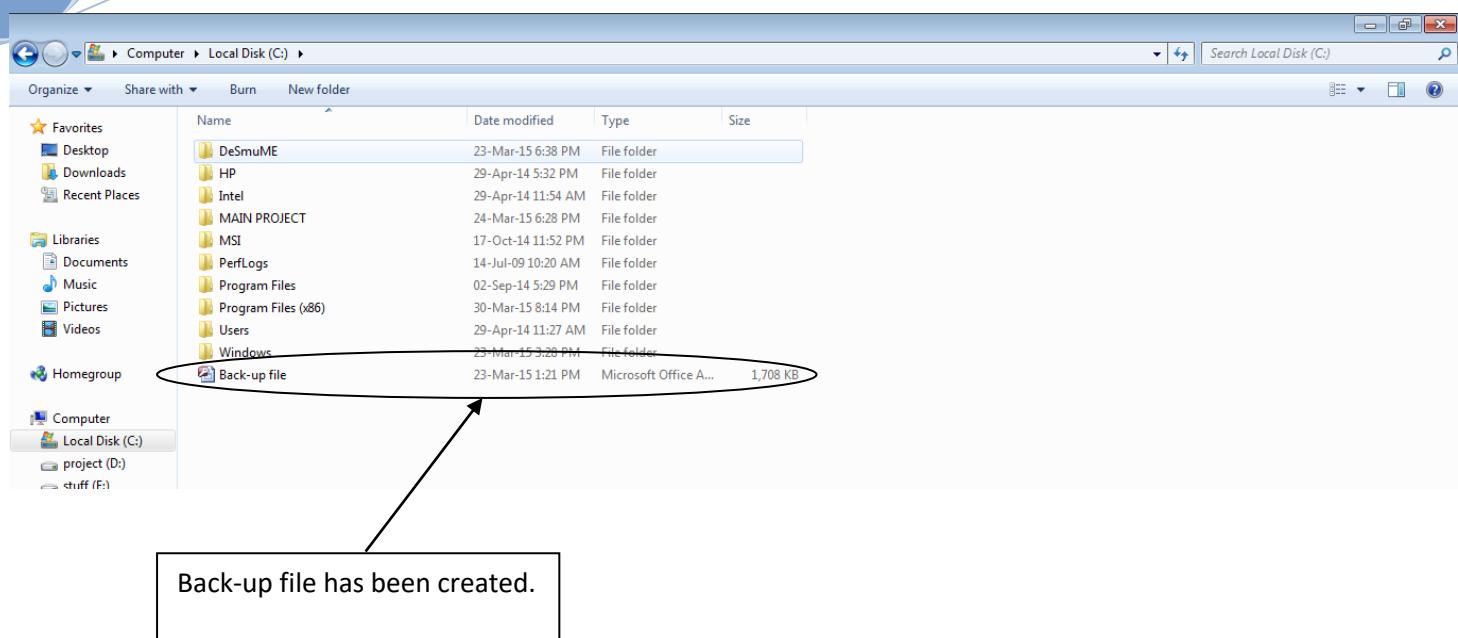
Test data used	Click “Backup” after entering name of target file and backup file
Type of data	Normal
Outcome	Message displayed that backup has been made



Travel Agency Management System



Travel Agency Management System



Test #114 Testing if Backup is created when “Backup Files” button is pressed on the backup form after selecting a suitable directory.

Test data used	Click on names of reports.
Type of data	
Outcome	Reports generated.

1. Employee Details report

A screenshot of a Microsoft Word document titled 'Employee Details'. The document features a logo for 'MAAS Travels and Tours' at the top. Below the logo, the title 'Employee Details' is centered. A table follows, displaying the employee's information:

Employee ID	Employee Name	Address	Contact	Designation	Basic salary	Date of entry
112	Peter Russo	5A, Chowdhury Mansion, 24th Surson road, Chittagong	0174716855	Officer	10,000.00	28-Mar-15 12:00:00AM

2. Fuel record

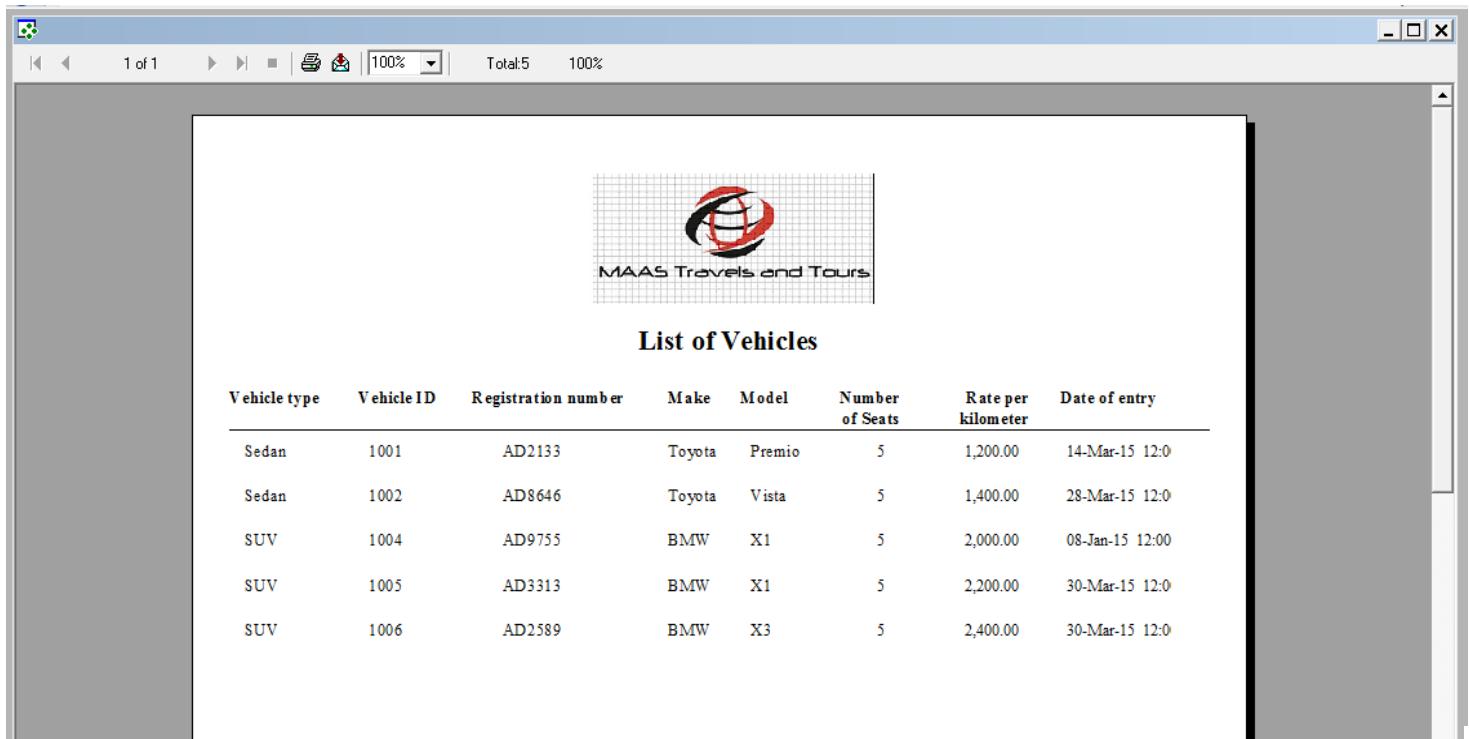
The screenshot shows a software window titled "Fuel Record". At the top, there is a logo for "MAAS Travels and Tours" featuring a stylized globe. Below the logo, the title "Fuel Record" is centered. A table follows, with columns: "Vehicle ID", "Fuel Type", "Quantity/litres", "Cost", and "Date of entry". There is one entry: Vehicle ID 1002, Fuel Type Octane, Quantity/litres 2.00, Cost 3,000.00, and Date of entry 31-03-2015.

Vehicle ID	Fuel Type	Quantity/litres	Cost	Date of entry
1002	Octane	2.00	3,000.00	31-03-2015

3. List of Packages

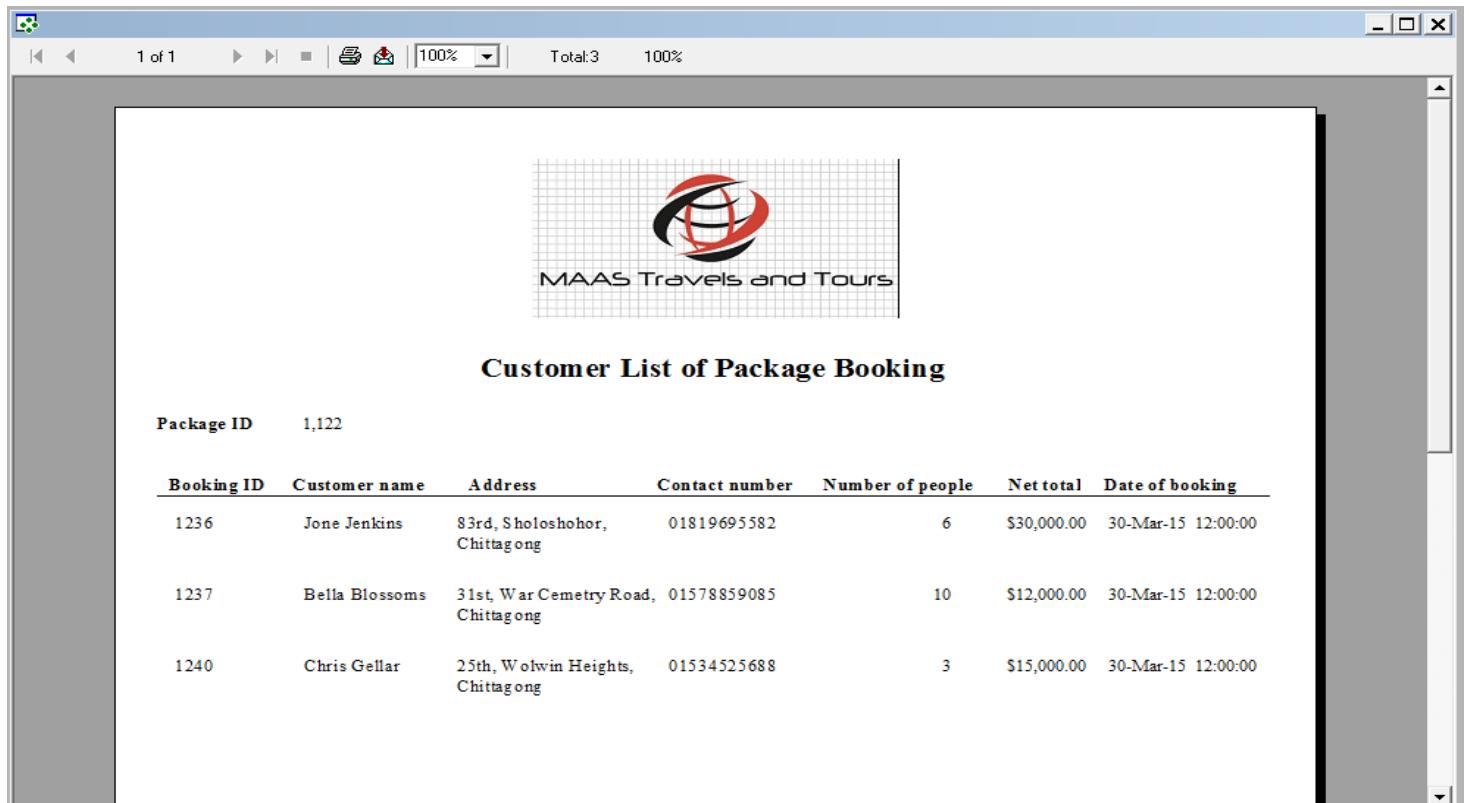
The screenshot shows a software window titled "List of Packages". At the top, there is a logo for "MAAS Travels and Tours" featuring a stylized globe. Below the logo, the title "List of Packages" is centered. A table follows, with columns: "Package type", "Package name", "Package ID", "Package details", "Accommodation type", "Cost per person", and "Journey date". There are three entries:

Package type	Package name	Package ID	Package details	Accommodation type	Cost per person	Journey date
Single day	Summer Isles	1121	Lunch Complementary Dinner Beach tour with Guide	Deluxe	1,200.00	21-Sep-15 12:00:0
Multi-day	Wind of Brisbane	1122	3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options	Normal	5,000.00	12-Feb-16 12:00:0
Multi-day	Sailing in Paris	1123	3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options	Deluxe	30,000.00	12-Mar-16 12:00:0

4. List of Vehicles


The screenshot shows a Windows application window titled "List of Vehicles". At the top, there is a logo for "MAAS Travels and Tours" featuring a stylized globe icon. Below the logo, the title "List of Vehicles" is centered. A table follows, displaying the following data:

Vehicle type	Vehicle ID	Registration number	Make	Model	Number of Seats	Rate per kilometer	Date of entry
Sedan	1001	AD2133	Toyota	Premio	5	1,200.00	14-Mar-15 12:0
Sedan	1002	AD8646	Toyota	Vista	5	1,400.00	28-Mar-15 12:0
SUV	1004	AD9755	BMW	X1	5	2,000.00	08-Jan-15 12:00
SUV	1005	AD3313	BMW	X1	5	2,200.00	30-Mar-15 12:0
SUV	1006	AD2589	BMW	X3	5	2,400.00	30-Mar-15 12:0

5. Customer list of Package Booking


The screenshot shows a Windows application window titled "Customer List of Package Booking". At the top, there is a logo for "MAAS Travels and Tours" featuring a stylized globe icon. Below the logo, the title "Customer List of Package Booking" is centered. A table follows, displaying the following data:

Package ID	Customer List of Package Booking						
Booking ID	Customer name	Address	Contact number	Number of people	Net total	Date of booking	
1236	Jone Jenkins	83rd, Sholoshohor, Chittagong	01819695582	6	\$30,000.00	30-Mar-15 12:00:00	
1237	Bella Blossoms	31st, War Cemetery Road, Chittagong	01578859085	10	\$12,000.00	30-Mar-15 12:00:00	
1240	Chris Gellar	25th, Wolwin Heights, Chittagong	01534525688	3	\$15,000.00	30-Mar-15 12:00:00	

6. Customer list of Vehicle Booking

1 of 1 | 100% | Total:1 100%



Customer List of Vehicle Booking

Vehicle ID	1002								
Booking ID	Customer name	Address	Contact	Vehicle type	Advance paid	Date of Booking	Journey date	Drop off date	Driver name
1423	Michelle Winger	23rd, Sholoshohor, Chittagong	01825545581	Sedan	\$2,500.00	31-Mar-15	21-Feb-15	29-Mar-15	Tywell Bitters

7. Package Expenses list

1 of 1 | 100% | Total:10 100%



Package Expenses List

PackageID	Expenses	Amount
1121	Transportation	5,000.00
1121	Transportation	5,000.00
1121	Entry fee	2,000.00
1121	Accommodation	6,650.00
1121	Transportation	9,000.00
1122	Transportation	5,000.00
1122	Accommodation	15,000.00
1122	Entry fee	3,000.00
1122	Accommodation	20,000.00
1122	Transportation	10,000.00

8. List of total Package Expenses

11-Apr-15



List of Total Package Expenses

Package ID	Package name	Total
1121	Summer Isles	\$22,650.00
1122	Wind of Brisbane	\$53,000.00

9. List of Takings from Vehicle Rentals

11-Apr-15



List of Takings from Vehicle Rentals

Booking ID	Customer name	Vehicle ID	Net total
1424	Helen Greene	1006	\$2,086,361.00

10. Vehicle Servicing Details

1 of 1 | 100% | Total:2 100%



Vehicle Servicing Details List

Vehicle type	Vehicle ID	Servicing details	Amount	Date of entry
SUV	1004	Painting	500.00	06-Apr-15
SUV	1004	New exhausts	1,500.00	06-Apr-15

1 of 1 | 100% | Total:1 100%



11-Apr-15

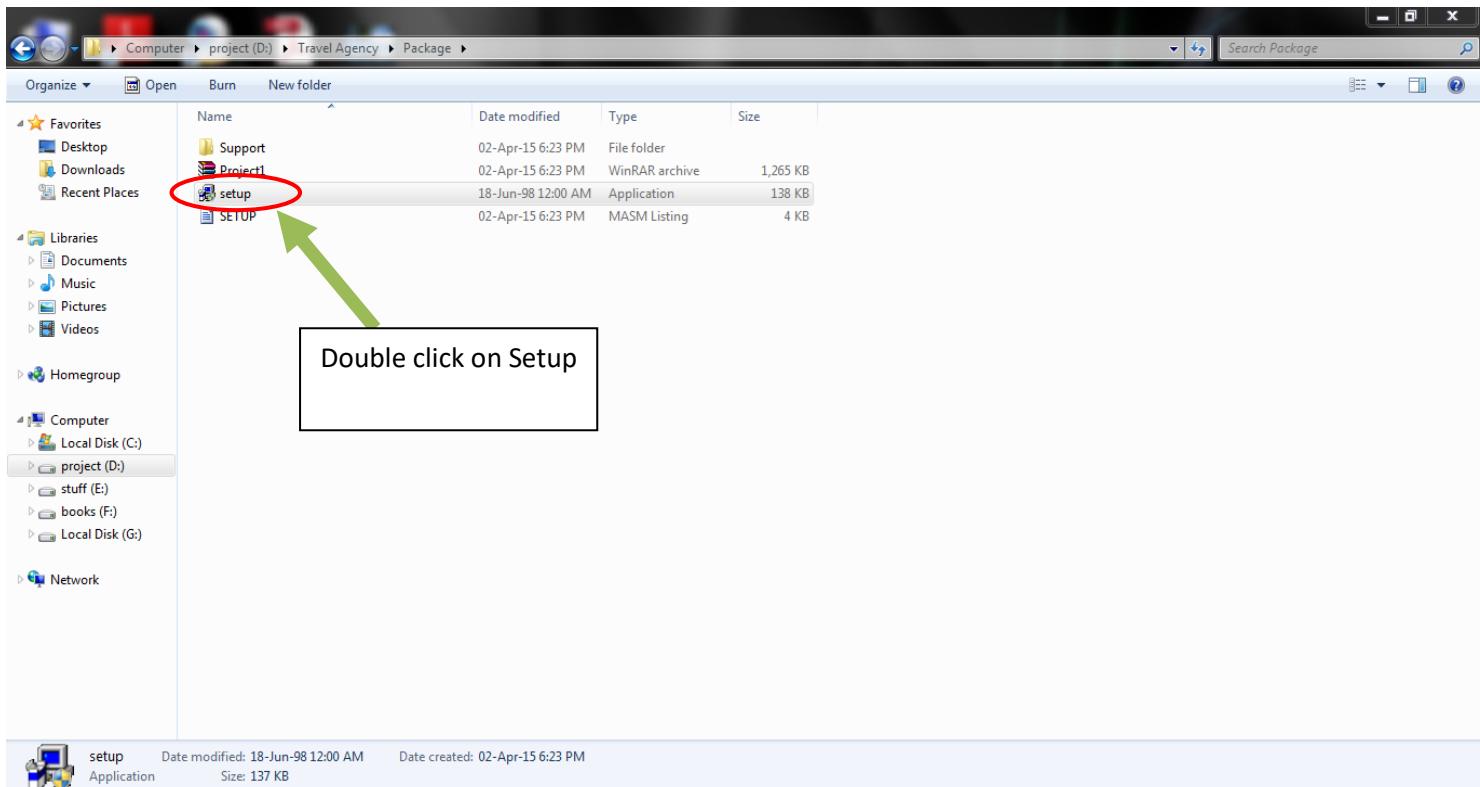
List of Takings from Vehicle Rentals

Booking ID	Customer name	Vehicle ID	Net total
1424	Helen Greene	1006	\$2,086,361.00

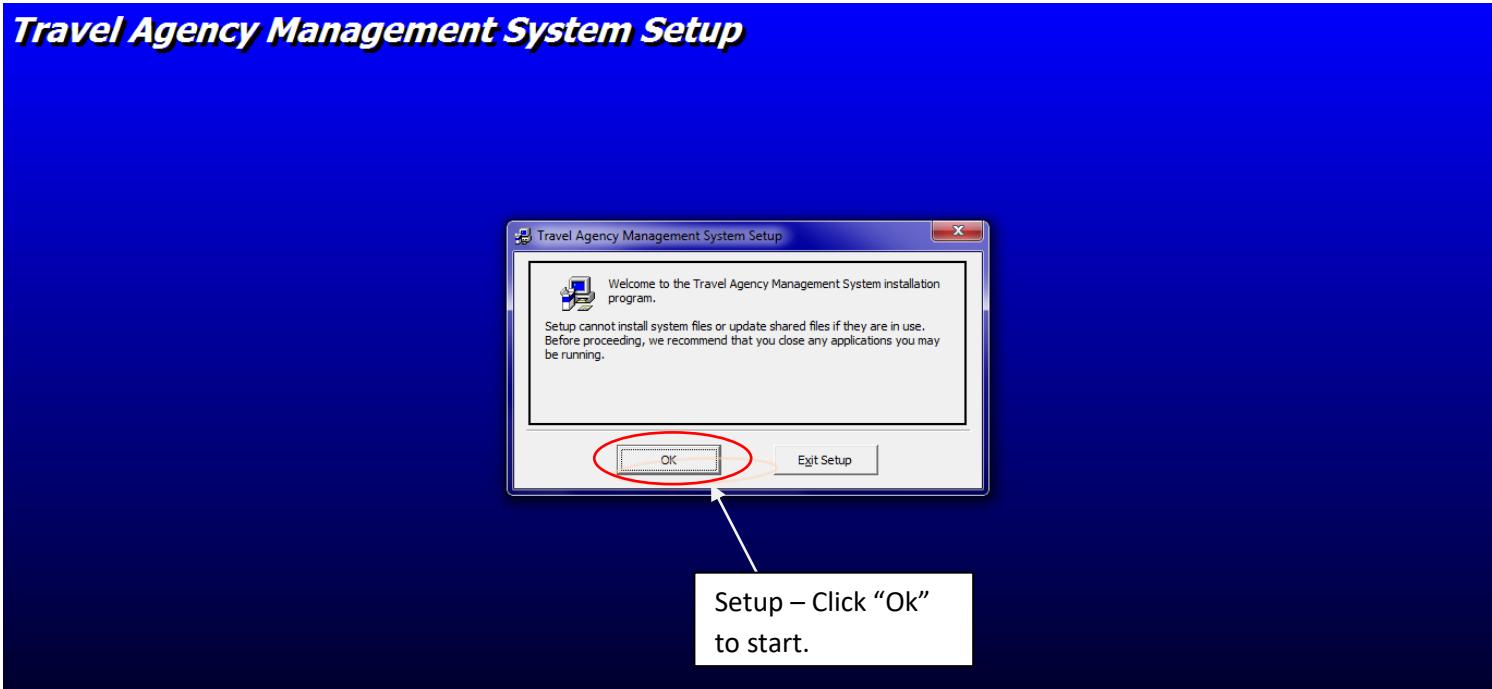
Installation instructions

The steps that need to be taken for the installation of the software are:

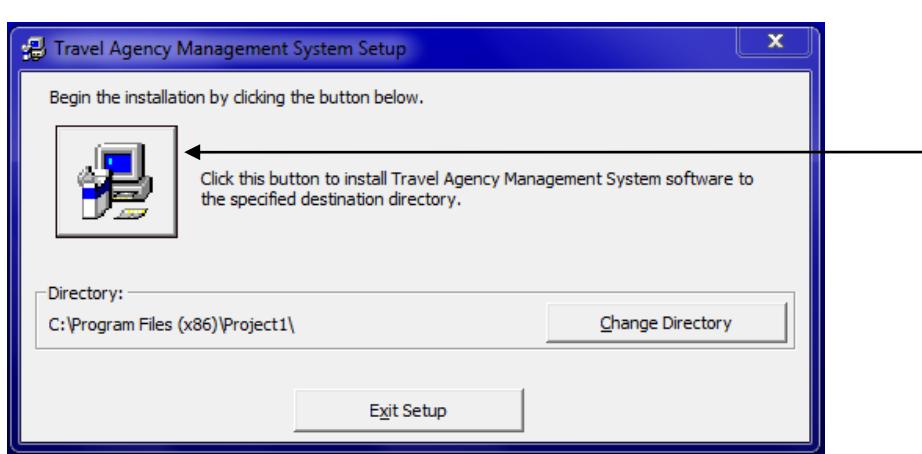
1. i. Open “Travel Agency” folder from Project (D:) drive.
ii. Open folder named “Package”.
iii. Double click on Setup.exe file.



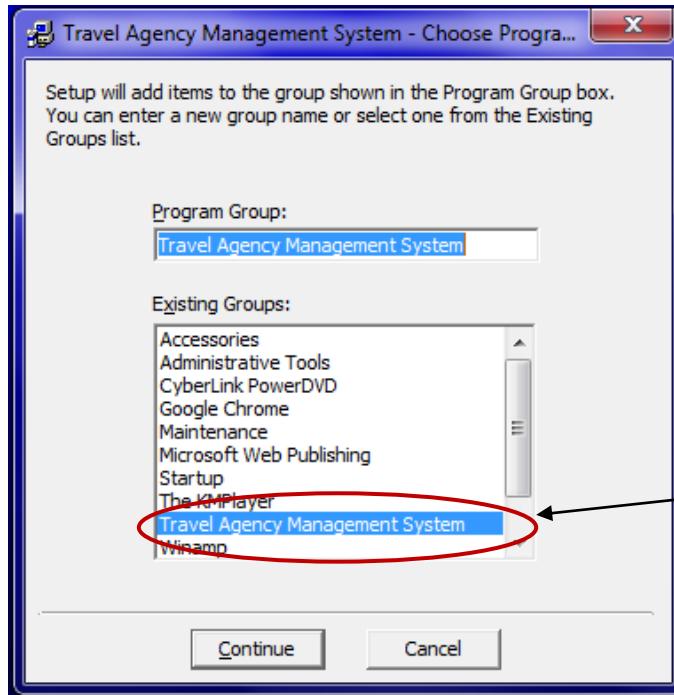
2. Click "Ok"



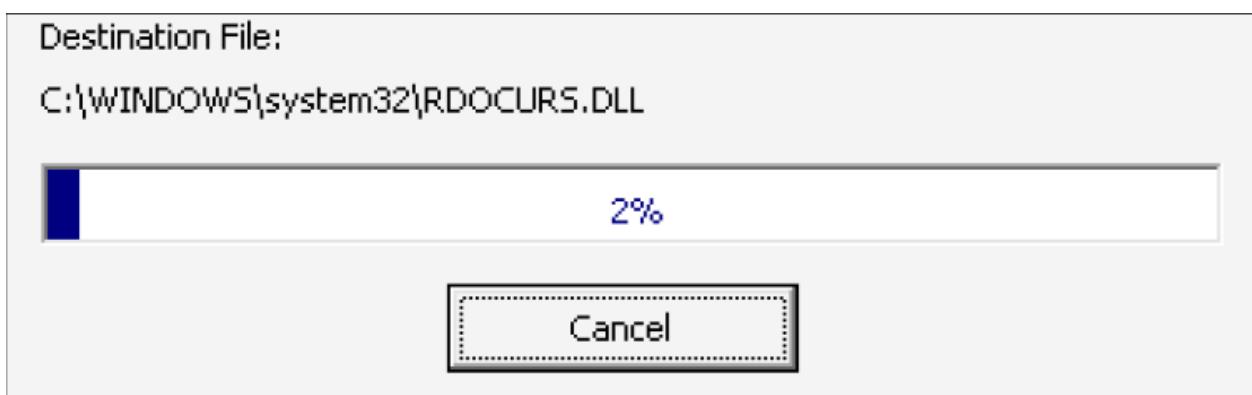
3. Click the button shown by the arrow.



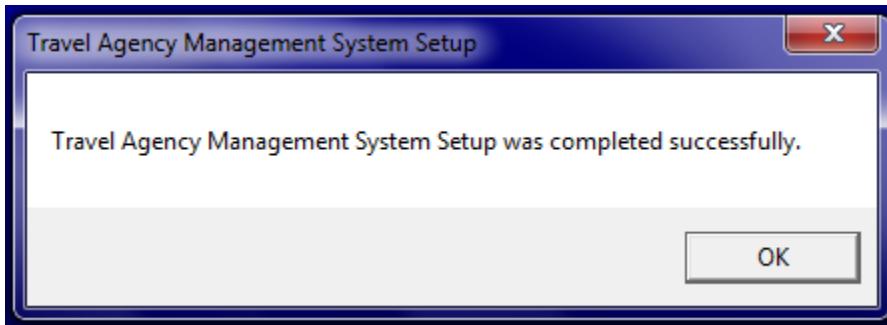
4. Click “Continue”.



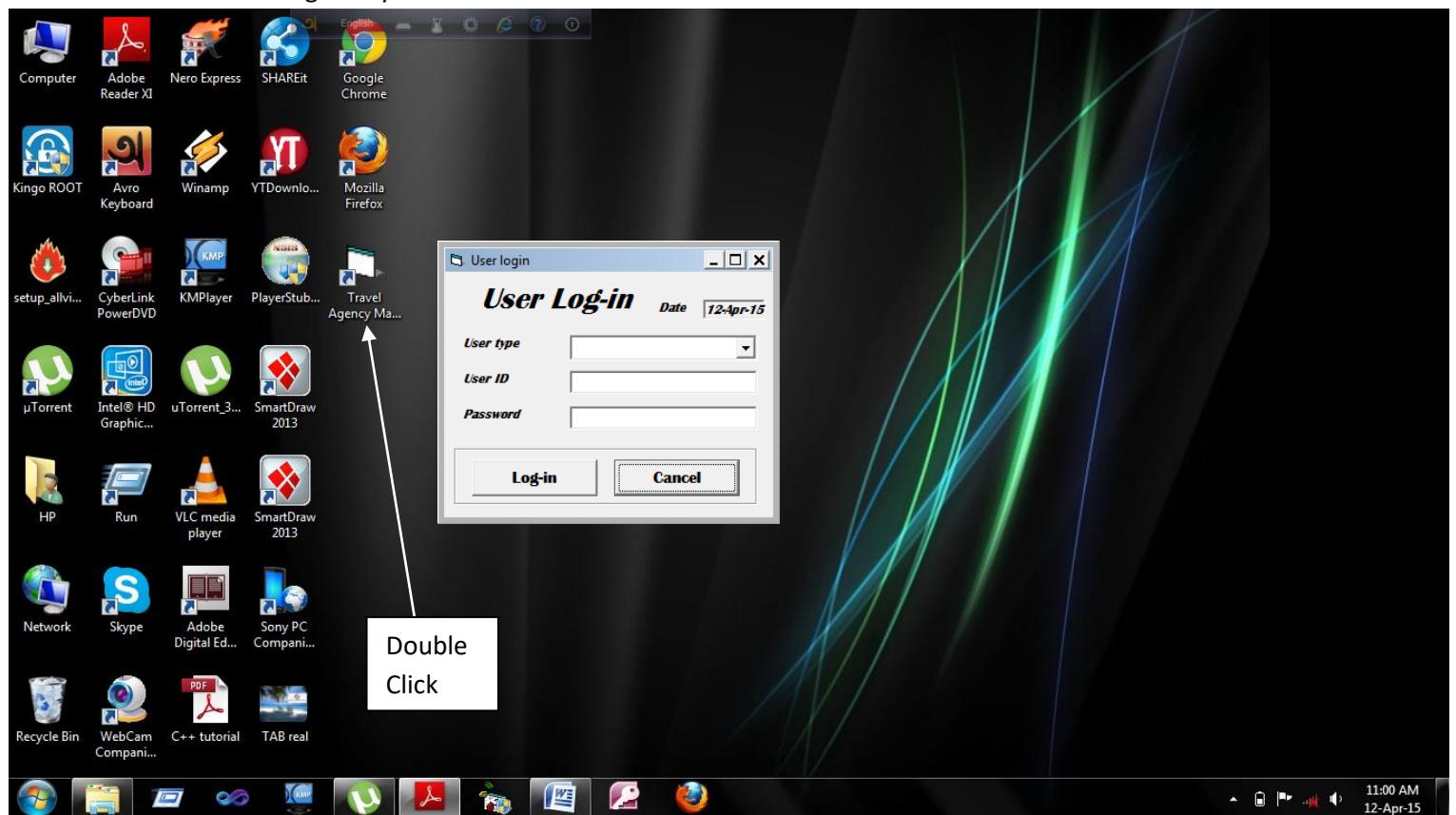
5. A progress bar is shown



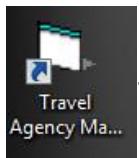
6. The installation is finished when the message shown below appears.



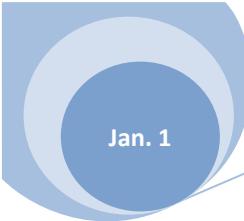
7. Starting the system



The Log-in form appears when the following icon is clicked from the desktop window. The access to Main Menu form can only be gained if the correct User type, User ID and Password are provided.



"Travel Agency Management System.exe" will be shown in the desktop when a shortcut of setup file is created. Double click on the icon to begin program as shown above.



Jan. 1

Travel Agency Management System

(d)

Documentation

(i) System maintenance documentation:

- System requirements:*

Hardware requirements:

1. Intel core i3 processor with speed 2.0 GHz: 2.0 GHz is recommended to ensure that OS loads up quickly and the computer does not lag while running heavy applications.
2. 1 GB RAM: This is the minimum temporary memory storage required for quick results on loading the system files which were to be processed and keeping data temporarily while processing.
3. 160 GB hard disk: To keep the system files after installing my system, the database and backup of the database.
4. 15/19" inch LED Monitor: This is required for a proper visual display and a proper Graphical User Interface (GUI).
5. Printer (Laser, Inkjet or Dot matrix): To print bills, sales statement, quotation, re-order list, customer list etc.
6. Combo drive: To write the backup copy of the database into a CD and to access files from the CD or DVD.
7. UPS: Provides temporary power supply thus preventing loss of unsaved data if power failure occurs.
8. Removable flash drive (pen drive, 32 GB): To keep the back up of the database.
9. Mouse and keyboard: For enter data into the system using forms and to navigate the whole system; i.e. select menus, draw controls etc.

Software requirements:

1. Windows 7 – Ideal for desktop computers, it provides a Graphical User Interface between user and hardware through which the user can easily access different applications and programs. To install MS Office, an OS is required to provide a platform to run the program. Windows 7 is recommended because it uses graphical user interface and is prone to less chances of errors and crashes.
2. MS Access: This is an RDBMS software used to create and maintain a database easily and efficiently. This software is found in the integrated package of MS Office 2007.
3. Visual Basic: It is a well-known programming language. It is used to develop the interfaces of the software and to run the software.
4. Crystal Reports: This is a report making software where any output report can be designed very easily.

❖ Database design:

There are 11 tables in the proposed system

- **User Login**
- **Employee Entry**
- **Package Entry**
- **Package Booking**
- **Package Expenses Details**
- **Package Expenses Master**
- **Vehicle Entry**
- **Vehicle Booking**
- **Vehicle Billing**
- **Vehicle Servicing**
- **Fuel Record**

❖ Description of data structures, records and files:

1) User Login table (Design view)

User Login table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
User_type	Text	10	User type of user	Administrator/ General	Presence check
User_ID	Text	10	A ID for identification of user, unique for each user.	JohnyB2112	Presence check
Password	Text	20	The password of the user.		Presence check

2) Employee Entry database table (Design view)

The screenshot shows the Microsoft Access 2007 interface in Design view for the 'Employee_entry' table. The table structure is as follows:

Field Name	Data Type	Description
EmployeeID	Text	
EName	Text	
Address	Text	
Contact	Text	
Designation	Text	
Basic_salary	Currency	
Date_of_entry	Date/Time	

Field Properties (EmployeeID):

- General tab: Field Size = 255, Format = Text, Input Mask = , Caption = , Default Value = , Validation Rule = , Validation Text = , Required = No, Allow Zero Length = Yes, Indexed = Yes (No Duplicates), Unicode Compression = No, IME Mode = No Control, IME Sentence Mode = None, Smart Tags = .
- Lookup tab: Description = A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Employee Entry database (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
EmployeeID	Text	6	ID of an employee working for the agency, unique to each employee.	Bessy1212	Presence check
EName	Text	20	Name of the employee.	Bessy Graham	Presence check, character type check
Address	Text	30	Address of employee	Road 4, house 8, khulshi, Chittagong	Presence check
Contact	Text	16	Contact number of employee.	+8801819637911/ +917204255875	Presence check, character type check
Designation	Text	10	Category of classification of an employee. Role or position of employee within agency	General manager/Chief executive officer/chief of staff/ executive officer	Presence check
Basic salary	Currency	8	Salary of an employee.	Salary of an employee.	Presence check
Date_of_entry	Date/Time	10	The date on which the entry was made.	10/03/2015	Presence check

3) Package Entry table (Design view)

The screenshot shows the Microsoft Access 2007 interface in Design view for the 'Package_entry' table. The ribbon at the top has tabs for Home, Create, External Data, Database Tools, and Design. The 'Create' tab is selected, showing icons for Table, SharePoint Lists, Table Design, Form, Split Form, Multiple Items, More Forms, Report, Labels, Blank Report, PivotChart, Form Design, Reports, Report Wizard, Report Design, Query, Query Design, Macro, and Other.

A security warning message is displayed: "Security Warning Certain content in the database has been disabled Options..."

The 'Package_entry' table structure is shown in the main pane:

Field Name	Data Type	Description
Package_type	Text	
Package_name	Text	
Package_details	Text	
Cost_per_person	Currency	
Date_of_entry	Date/Time	
PackageID	Text	
Itinerary	Text	
Journey_date	Date/Time	
Arrival_time	Date/Time	
Starting_time	Date/Time	
Duration/days	Number	

The 'Field Properties' pane on the right shows the 'General' tab for the 'Duration/days' field, which is set to 'Long Integer' as the field size and 'Auto' as the decimal places. A tooltip 'Alignment of text in control' is visible over the properties pane.

At the bottom left, the status bar reads: "Design view. F6 = Switch panes. F1 = Help." At the bottom right, there are standard window control buttons.

Package Entry table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
Package_type	Text	20	Type of the package.	Special package/deluxe package/best on budget	Presence check
Package_name	Text	20	Name of the package.	Beach travel/ Asia tour	Presence check, character type check
Package_details	Text	70	Information on the package.		Presence check
Cost_per_person	Currency	8	Amount charged per person.	\$215.55	Presence check
Date_of_entry	Date/Time	10	Date on which an entry was made.	10/03/2015	Presence check
PackageID	Text	6	An ID for identification of a package, unique to each package.	Qw1152	Presence check
Itinerary	Text	70	Complete travel schedule.		Presence check
Journey_date	Date/Time	10	Date on which the start of the journey is due.	21/08/2015	Presence check
Arrival_time	Date/Time	10	Time of arrival of the customers.	10:00:00 AM	Presence check
Starting_time	Date/Time	10	The time at which the journey is going to start.	10:30:00 AM	Presence check
Duration/days	Number	4	The number of days the journey is going to occur.	5	Presence check
Accomodation_type	Text	10	The type of accommodation the package plan provides.	Deluxe/luxury/low budget	Presence check
NumberReserved	Number	4	The number of bookings made of a particular package.	25	Presence check
NumberOfReservations	Number	4	The number of bookings than can be made for a particular package.	30	Presence check

4) Package Booking table (Design view)

The screenshot shows the Microsoft Access 2007 interface in Design view for the 'Package_booking' table. The table structure is as follows:

Field Name	Data Type	Description
PackageBookingID	Text	
Customer_name	Text	
Customer_address	Text	
Contact	Text	
Package_name	Text	
Cost_per_person	Currency	
Number_of_people	Number	
Net_total	Currency	
Date_of_booking	Date/Time	
Package_type	Text	
Accommodation_type	Text	

The 'PackageBookingID' field is selected, and its properties are displayed in the Field Properties pane:

General	Lookup
Field Size: 255	
Format:	
Input Mask:	
Caption:	
Default Value:	
Validation Rule:	
Validation Text:	
Required:	No
Allow Zero Length:	Yes
Indexed:	Yes (No Duplicates)
Unicode Compression:	No
DDE Mode:	No Control
IME Sentence Mode:	None
Smart Tags:	

A note in the pane states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names."

Package Booking table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
PackageBookingID	Text	6	An ID to distinguish between bookings, unique to each customer.	PG9485	Presence check
Customer_name	Text	20	The name of the customer.	Brett Snow	Presence check, character type check
Customer_address	Text	30	Address of the customer.	Road 6, house 8,Nasirabad housing society, Chittagong	Presence check
Contact	Text	16	The contact number of the customer.	+8801819637911	Presence check, character type check
Package_name	Text	20	The name of the package.	Beach travel/Asia tour	Presence check, character type check
Cost_per_person	Currency	8	Amount charged per person for the package.	\$124.99	Presence check
Number_of_people	Number	4	The number of people for which a booking is being made.	3	Presence check
Net_total	Currency	8	Net amount that a customer needs to make.	\$499.96	Presence check
Date_of_booking	Date/Time	20	The date on which the booking is made.	10/03/2015	Presence check
Package_type	Text	20	The type of the package.	Special package/deluxe package/best on budget	Presence check
Accommodation_type	Text	10	The type of accommodation provided by the package plan.	Deluxe/Luxury/low Budget	Presence check
Journey_date	Date/Time	10	The date on which the beginning of the journey is due.	15/06/2015	Presence check
Arrival_time	Date/Time	10	The time of arrival of the customer.	10:00:00 AM	Presence check
Starting_time	Date/Time	10	The time at which the journey will start.	10:30:00 AM	Presence check
Duration/days	Number	4	The number of days for which the tour will carry on	5	Presence check
Itinerary	Text	70	Complete schedule of the journey		Presence check
Package_details	Text	70	Detailed information about the package.		Presence check
PackageID	Text	6	An ID for identification of package, unique to each.	PG3215	Presence check

5) Package Expenses Details table (Design view)

Package_expenses_details

Field Name	Data Type	Description
PackageID	Text	
Expenses	Text	
Amount	Currency	

Field Properties

General

Field Size: 255
 Format:
 Input Mask:
 Caption:
 Default Value:
 Validation Rule:
 Validation Text:
 Required: No
 Allow Zero Length: Yes
 Indexed: No
 Unicode Compression: No
 IME Mode: No Control
 IME Sentence Mode: None
 Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Package Expenses Details table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
PackageID	Text	6	An ID that serves as identification for each package, unique to each package.	PG8945	Presence check
Expenses	Text	20	The names of the expenses.	Travel food costs	Presence check
Amount	currency	8	The amount charged for a particular expense.	\$555.86	Presence check

6) Package Expenses Master table (Design view)

Microsoft Access 2007 - travelagency database : Database (Access 2007) - Microsoft Access

Package_expenses_master

Field Name	Data Type	Description
PackageID	Text	
Package_name	Text	
Total	Currency	

Field Properties

General

- Field Size: 255
- Format:
- Input Mask:
- Caption:
- Default Value:
- Validation Rule:
- Validation Text:
- Required: No
- Allow Zero Length: Yes
- Indexed: No
- Unicode Compression: No
- IME Mode: No Control
- IME Sentence Mode: None
- Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Package Expenses Master table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
PackageID	Text	6	Unique ID that serves as identification for each package.	PG2313	Presence check
Package_name	Text	20	The name of the package.	Beach travels/Asia tour	Presence check, character type check
Total	Currency	8	The total numerical amount spent on package expenses.	\$1400.85	Presence check

7) Vehicle Entry table (Design view)

The screenshot shows the Microsoft Access application window in Design view. The title bar reads "travel agency database: Database (Access 2007) - Microsoft Access". The ribbon tabs are Home, Create, External Data, Database Tools, and Design, with Design selected. The main area displays the "Vehicle_entry" table structure. The table has the following fields:

Field Name	Data Type	Description
Vehicle_type	Text	
VehicleID	Text	
RegistrationNo	Text	
Make	Text	
Model	Text	
NSeats	Number	
Rate_per_km	Number	
Date_of_entry	Date/Time	
IsBooked?	Yes/No	

A navigation pane on the left lists "Views", "Primary Key", "Validation Rules", "Lookup Columns", and "Tools". A "Field Properties" pane is open for the first field, "Vehicle_type", showing properties like Field Size (255), Format (Text), Input Mask (None), and Validation Rule (Required). A note in the pane states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names." The status bar at the bottom indicates "Design view. F6 = Switch panes. F1 = Help."

Vehicle Entry table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
Vehicle_type	Text	10	Type of the vehicle.	SUV/Sedan/Convertible/minivan	Presence check
VehicleID	Text	6	An ID for identification of vehicle, unique to each vehicle.	Ad2150	Presence check
RegistrationNo	Text	7	Legal registration number of vehicle.	22-2122	Presence check
Make	Text	10	The name of the company that made the vehicle.	toyota/mercedes/Bentley/chevrolet	Presence check
Model	Text	6	The model of the vehicle.	Vista/arnage/aveo	Presence check
Nseats	Number	4	Number of seats in the vehicle.	5	Presence check,character type check
Rate_per_km	Number	8	Amount charged per kilometer travelled.	\$125.99	Presence check,character type check
Date_of_entry	Date/Time	10	Date on which the entry was made.	10/03/2015	Presence check
IsBooked?	Yes/No	1	A check to see if a vehicle is available for booking.	Yes or No, where No means vehicle is available.	Presence check

8) Vehicle Booking table (Design view)

The screenshot shows the Microsoft Access 2007 application window. The title bar reads "travel agency database : Database (Access 2007) - Microsoft Access". The ribbon is visible with the "Table Tools" tab selected, and the "Design" tab is active. In the main pane, the "Vehicle_booking" table is displayed in Design view. The table has the following fields:

Field Name	Data Type	Description
VehicleBookingID	Text	
VehicleID	Text	
Customer_name	Text	
Customer_address	Text	
Contact	Text	
Vehicle_type	Text	
Journey_date	Date/Time	
Driver_name	Text	
Date_of_booking	Date/Time	
Registration_number	Text	
Drop_off_date	Date/Time	

The "Field Properties" pane on the right shows the properties for the "VehicleBookingID" field, which is set as the primary key. The "Format" property is set to 255. A note in the pane states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names." The status bar at the bottom left indicates "Design view. F6 = Switch panes. F1 = Help."

Vehicle Booking table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
VehicleBookingID	Text	6	Booking ID unique to each customer, for differentiation.	789754	Presence check
VehicleID	Text	6	A ID number for differentiation of vehicles, unique to each vehicle.	VC2323	Presence check
Customer_name	Text	20	The name of the customer.	Gary P. Gale	Presence check, character type check
Customer_address	Text	30	Address of the customer.	Road 4, house 5, Khulshi, Chittagong	Presence check
Contact	Text	16	Contact number of the customer.	+8801819637911	Presence check, character type check
Vehicle_type	Text	10	Type of the vehicle.	Sedan/SUV/Convertible/minivan	Presence check
Journey_date	Date/Time	10	The date on which the journey was due.	12/06/2015	Presence check
Driver_name	Text	20	Name of driver.	Peter Gatsby	Presence check, character type check
Date_of_booking	Date/Time	10	Date on which booking was made.	13/05/2015	Presence check
Registration_number	Text	6	Legal registration number of the vehicle.	22-3215	Presence check
Drop_off_date	Date/Time	10	The day on which the vehicle is supposed to be returned according to the deal with the customer.	06/07/2015	Presence check
Make	Text	10	The name of the company that made the vehicle.	Toyota/Mercedes/Bentley	Presence check
Model	Text	6	The model of the vehicle.	Premio/Vista/Arnage/Prius	Presence check
Rate_per_kilometer	Currency	8	The amount charged per kilometer travelled.	\$125.99	Presence check
Number_of_seats	Number	4	Number of seats in the vehicle.	4	Presence check
Advance_paid	Currency	8	The amount that needs to be paid at booking.	\$1400.00	Presence check, character type check

9) Vehicle Billing table (Design view)

The screenshot shows the Microsoft Access 2007 application window. The title bar reads "Travel Agency Management System" and "travel agency database : Database (Access 2007) - Microsoft Access". The ribbon is visible at the top with tabs like Home, Create, External Data, Database Tools, and Design. The Design tab is selected. The main area displays the "Vehicle_Billing" table in Design view. The table has 11 fields: BookingID, Customer_name, Customer_address, VehicleID, Kilometer_travelled, Rate_per_kilometer, Cost_of_travel, Journey_date, Drop_off_date, Booking_charge, and Driver_charge. The "BookingID" field is highlighted and set as the Primary Key. The "Field Properties" pane on the right shows the "General" tab for the "BookingID" field, with "Field Size" set to 255. A note in the pane states: "A field name can be up to 64 characters long, including spaces. Press F1 for help on field names." The status bar at the bottom indicates "Design view. F6 = Switch panes. F1 = Help."

Field Name	Data Type	Description
BookingID	Text	
Customer_name	Text	
Customer_address	Text	
VehicleID	Text	
Kilometer_travelled	Number	
Rate_per_kilometer	Currency	
Cost_of_travel	Currency	
Journey_date	Date/Time	
Drop_off_date	Date/Time	
Booking_charge	Currency	
Driver_charge	Currency	

Field Properties (General tab):

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Vehicle Billing table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
BookingID	Text	6	An identification number for a booking, unique for each customer.	320305	Presence check, character check
Customer_name	Text	20	The name of the customer	Lu Kyang	Presence check
Customer_address	Text	50	The address of the customer.	45 th , avenue street, Chittagong	Presence check
VehicleID	Text	6	The identification number for a Vehicle, unique for each Vehicle.	Ad4646	Presence check, Character check
Kilometer_travelled	Number	4	The number of kilometers travelled during the journey with the vehicle.	10	Presence check
Rate_per_kilometer	Currency	8	The amount of money charged for each kilometer travelled.	\$50	Presence check
Cost_of_travel	Currency	8	The total cost of the travel.	\$500	Presence check
Journey_date	Date/Time	10	The date on which the journey is due.	10/10/2015	Presence check
Drop_off_date	Date/Time	10	The date on which the vehicle is supposed to be dropped off.	20/10/2015	Presence check
Booking_charge	Currency	8	The amount charged for making a booking.	\$300	Presence check
Driver_charge	Currency	8	The amount charged for hiring a driver.	\$150	Presence check
Additional_expenditures	Currency	8	Any additional expenditures on the vehicle booking by the customer.	\$30	Presence check
Gross_total	Currency	8	The gross total on the vehicle.	\$980	Presence check
Discount	Currency	8	Any discount provided to the customer.	\$40	Presence check
Advance_paid	Currency	8	The advance paid by the customer.	\$200	Presence check
Net_total	Currency	8	The net total for the booking.	\$780	Presence check
Billing_date	Date/Time	10	The date on which the billing is being done.	20/10/2015	Presence check

Vehicle Servicing table (Design view)

Vehicle_servicing

Field Name	Data Type	Description
Vehicle_type	Text	
VehicleID	Text	
Servicing_details	Text	
Amount	Currency	
Date	Date/Time	

Field Properties

General

Field Size: 255
 Format:
 Input Mask:
 Caption:
 Default Value:
 Validation Rule:
 Validation Text:
 Required: No
 Allow Zero Length: Yes
 Indexed: Yes (No Duplicates)
 Unicode Compression: No
 IME Mode: No Control
 IME Sentence Mode: None
 Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Vehicle Servicing table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
Vehicle_type	Text	10	The type of the vehicle being serviced.	SUV	Presence check
VehicleID	Text	6	And identification for the recognition of the vehicle.	Ad2656	Presence check, character check
Servicing_details	Text	70	The information on the servicing of the vehicle.		Presence check
Amount	Currency	8	The amount paid in servicing of the vehicle.	\$1500	Presence check
Date	Date/time	10	The date on which the servicing transaction was made.	10/2/2013	Presence check

10) Fuel Record table (Design view)

Fuel_record

Field Name	Data Type	Description
VehicleID	Text	
Date_of_entry	Text	
FuelType	Text	
FuelQuantity/litres	Number	
FuelCost	Currency	

Field Properties

General

Field Size: 255
Format: 255
Input Mask:
Caption:
Default Value:
Validation Rule:
Validation Text:
Required: No
Allow Zero Length: Yes
Indexed: Yes (No Duplicates)
Unicode Compression: No
IME Mode: No Control
IME Sentence Mode: None
Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

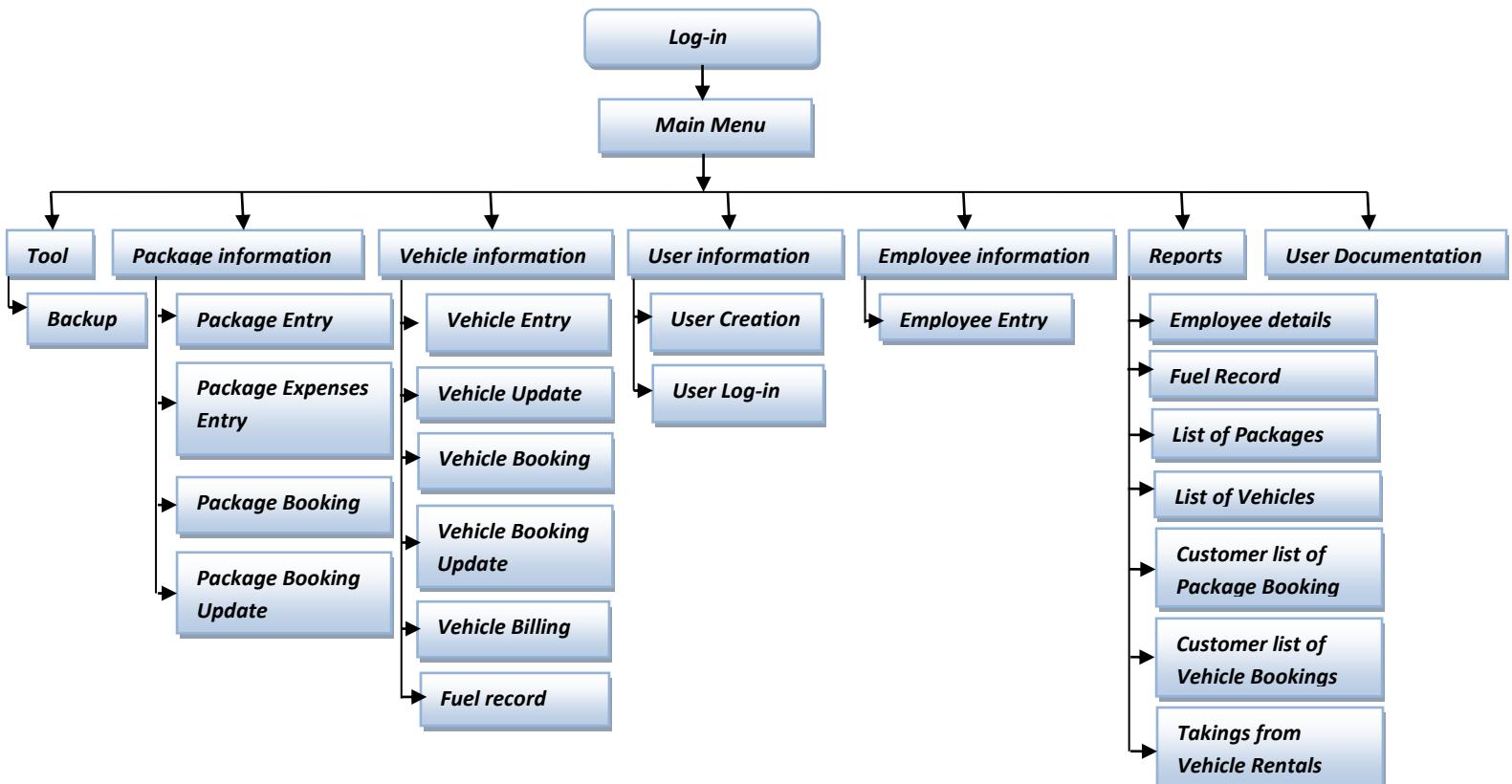
Fuel Record table (Datasheet view)

Field name	Data type	Size	Note	Example	Validation
VehicleID	Text	6	Unique ID to identify a vehicle.	Ad2133	Presence check
Date_of_enty	Text	10	Date on which entry was made.	10/03/2015	Presence check
FuelType	Text	10	Type of the fuel.	Octane/diesel	Presence check
FuelQuantity/litres	Number	8	Quantity of fuel.	5	Presence check
FuelCost	Currency	8	Cost of fuel.	\$125.45	Presence check

❖ Form Design:

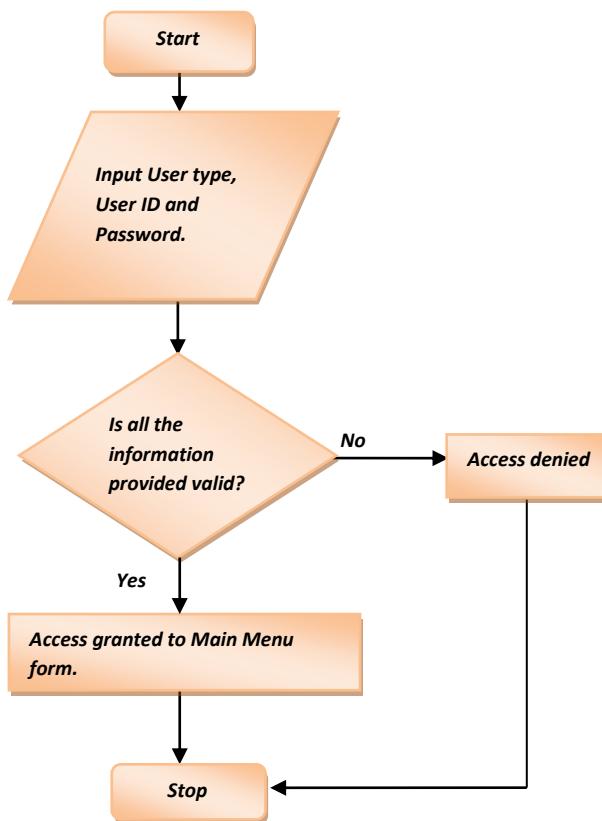
➤ There are fourteen (14) forms. The form names are listed below-

1. *Log-in form*
2. *Main (Navigation) form*
3. *User Creation form*
4. *Employee Entry form*
5. *Package Entry*
6. *Package Booking*
7. *Package Booking Update*
8. *Package Expenses Entry*
9. *Vehicle Entry*
10. *Vehicle Update*
11. *Vehicle Booking*
12. *Vehicle Booking Update*
13. *Vehicle Billing*
14. *Vehicle servicing*
15. *Fuel Record*
16. *Backup*

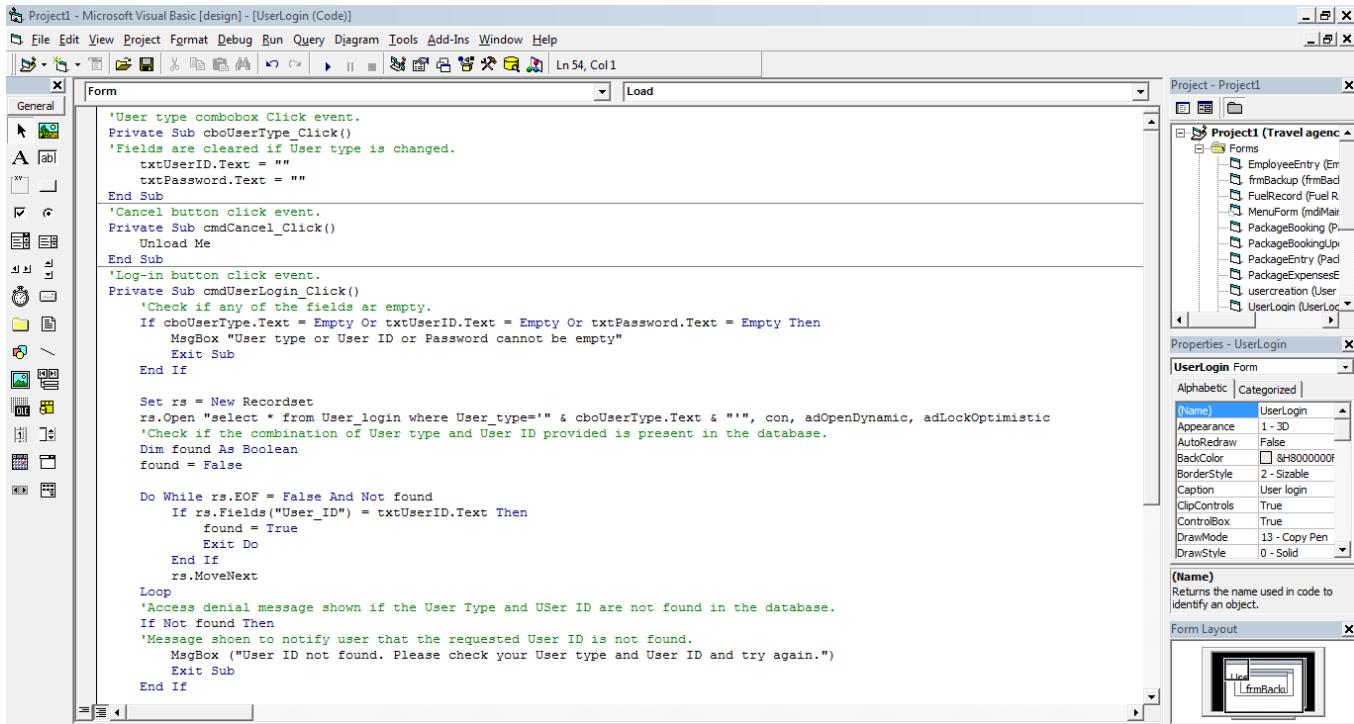


a. User Log-in form:

The screenshot shows the actual layout of the log-in form.



The screenshot below shows the coding of the Log-in form.



The coding of the User log-in form is shown below.

```

'User type combobox Click event.
Private Sub cboUserType_Click()
'Fields are cleared if User type is changed.
    txtUserID.Text = ""
    txtPassword.Text = ""
End Sub

'Cancel button click event.
Private Sub cmdCancel_Click()
    Unload Me
End Sub

'Log-in button click event.
Private Sub cmdUserLogin_Click()
    'Check if any of the fields ar empty.
    If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Then
        MsgBox "User type or User ID or Password cannot be empty"
        Exit Sub
    End If

    Set rs = New Recordset
    rs.Open "select * from User_login where User_type=''" & cboUserType.Text & "'", con,
    adOpenDynamic, adLockOptimistic
    'Check if the combination of User type and User ID provided is present in the database.
    Dim found As Boolean

```

```
found = False

Do While rs.EOF = False And Not found
    If rs.Fields("User_ID") = txtUserID.Text Then
        found = True
        Exit Do
    End If
    rs.MoveNext
Loop
'Access denial message shown if the User Type and USer ID are not found in the database.
If Not found Then
    'Message shoen to notify user that the requested User ID is not found.
    MsgBox ("User ID not found. Please check your User type and User ID and try again.")
    Exit Sub
End If
'Check for validation of password if combination of User type and User ID are accepted.
If txtPassword.Text = rs.Fields("Password") Then
    MsgBox "Login successful."
Else
    'Access denail message shown if password is not accepted.
    MsgBox "Invalid password. Please try again."
End If

MenuForm.Show
End Sub

'Form load event.
Private Sub Form_Load()
'Calling the function to connect the database.
    Call dblink
    lblDate.Caption = Date
End Sub
```

b. Main Menu form

The screenshot below shows the coding of the Main Menu form.

The screenshot shows the Microsoft Visual Basic IDE interface. The code editor window displays the following VBA code for the "MenuForm" form:

```

Private Sub mnFile_Click()
    FuelRecord.Show
End Sub

Private Sub mnBackup_Click()
    frmBackup.Show
End Sub

Private Sub mnEmployeeentry_Click()
    EmployeeEntry.Show
End Sub

Private Sub mnFuelRecord_Click()
    FuelRecord.Show
End Sub

Private Sub mnlogin_Click()
    UserLogin.Show
End Sub

Private Sub mnpackagebooking_Click()
    PackageBooking.Show
End Sub

Private Sub mnpackagebookingupdate_Click()
    PackageBookingUpdate.Show
End Sub

Private Sub mnpackageentry_Click()
    PackageEntry.Show
End Sub

Private Sub mnpackageexpensesentry_Click()
    PackageExpensesEntry.Show
End Sub

Private Sub mnusercreation_Click()
    UserCreation.Show
End Sub

```

The Project Explorer window on the right lists various forms in the "Project1 (Travel agency)" project, including "EmployeeEntry", "frmBackup", "FuelRecord", "MenuForm", "PackageBooking", "PackageBookingUpdate", "PackageEntry", "PackageExpenses", "UserCreation", and "UserLoan". The Properties window shows the properties for "MenuForm" (MDIForm), and the Form Layout window shows a diagram of the "MenuForm" window.

```
Private Sub mnuFile_Click()
    FuelRecord.Show
End Sub
```

```
Private Sub mnBackup_Click()
    frmBackup.Show
End Sub
```

```
Private Sub mnemployeeentry_Click()
    EmployeeEntry.Show
End Sub
```

```
Private Sub mnFuelRecord_Click()
    FuelRecord.Show
End Sub
```

```
Private Sub mnlogin_Click()
    UserLogin.Show
End Sub
```

```
Private Sub mnpackagebooking_Click()
    PackageBooking.Show
End Sub
```

```
Private Sub mnpackagebookingupdate_Click()
    PackageBookingUpdate.Show
End Sub
```

```
Private Sub mnpackageentry_Click()
    PackageEntry.Show
End Sub
```

```
Private Sub mnpackageexpensesentry_Click()
    PackageExpensesEntry.Show
End Sub
```

```
Private Sub mnusercreation_Click()
    usercreation.Show
End Sub
```

```
Private Sub mnvehiclebilling_Click()
    VehicleBilling.Show
End Sub
```

```
Private Sub mnvehiclebooking_Click()
    VehicleBooking.Show
End Sub
```

```
Private Sub mnvehiclebookingupdate_Click()
    VehicleBookingUpdate.Show
End Sub
```

```
Private Sub mnvehicleentry_Click()
    VehicleEntry.Show
End Sub
```

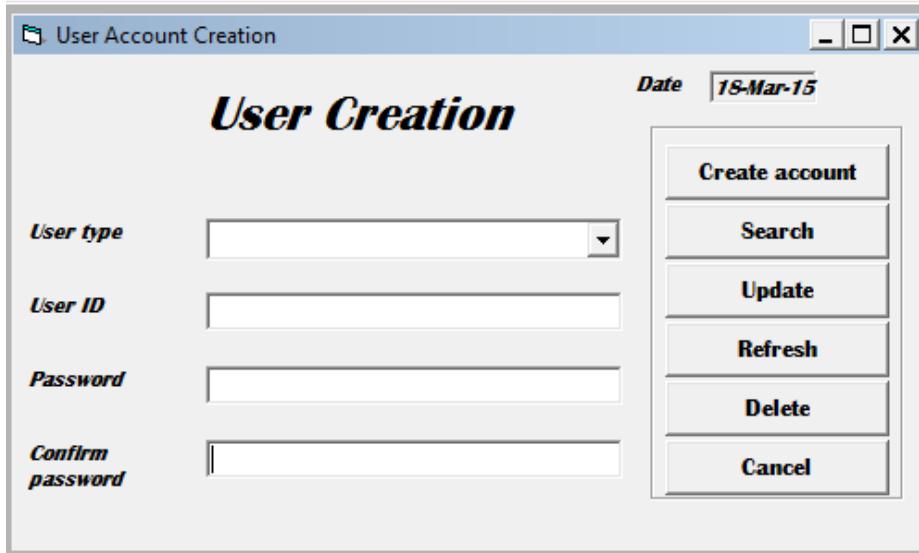
```
Private Sub mnvehicleservicing_Click()
    VehicleServicing.Show
End Sub
```

```
Private Sub mnvehicleupdate_Click()
    VehicleUpdate.Show
End Sub
```

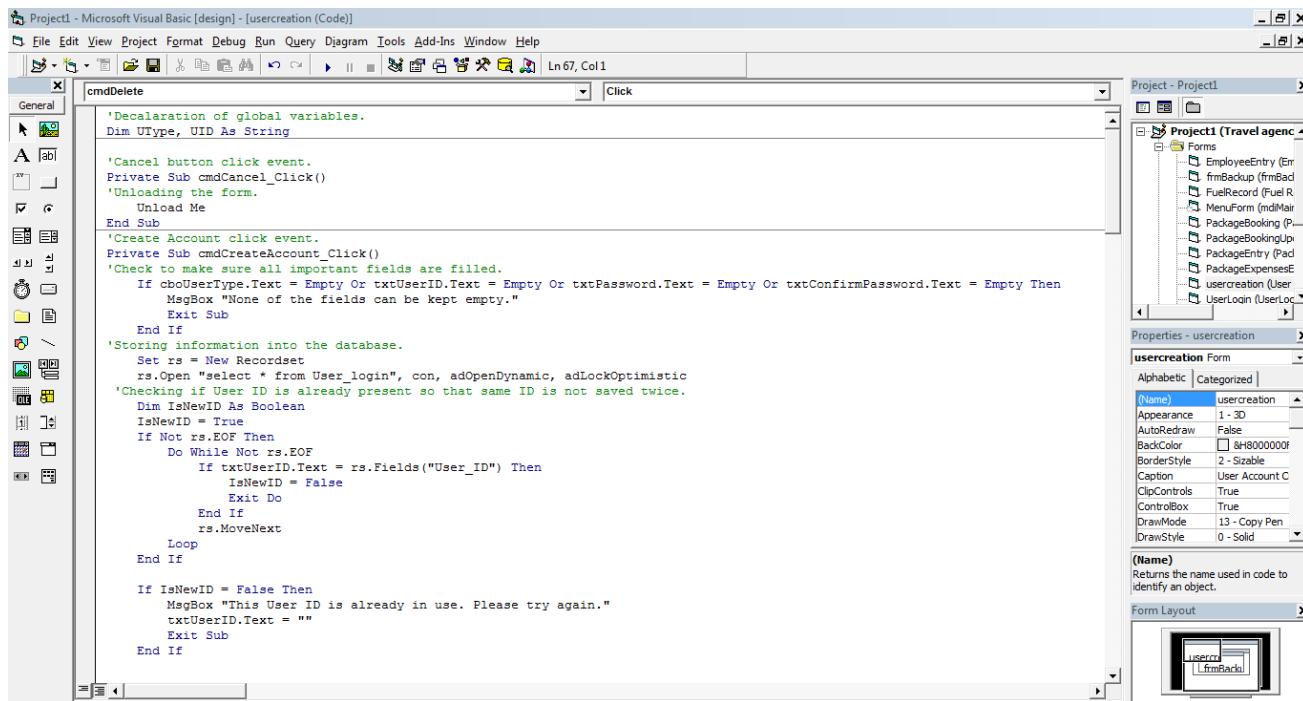
```
Private Sub rptEmployeeDetails_Click()
    rpt1.ReportFileName = App.Path & "\Reports\Employee Details.rpt"
    'CrystalReport1.SelectionFormula = "{guestentry.ticketnumber}=" & txtticketno.Text & ""
    rpt1.Action = 2
End Sub
```

c. **User Creation form**

The screenshot below shows the actual layout of the User Creation form.



The screenshot below shows the coding of the User Creation form.



Below is the coding of the User Creation form.

```
'Declaration of global variables.
Dim UType, UID As String

'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
Unload Me
End Sub

>Create Account click event.
Private Sub cmdCreateAccount_Click()
'Check to make sure all important fields are filled.
If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Or
txtConfirmPassword.Text = Empty Then
    MsgBox "None of the fields can be kept empty."
    Exit Sub
End If

'Storing information into the database.
Set rs = New Recordset
rs.Open "select * from User_login", con, adOpenDynamic, adLockOptimistic
'Checking if User ID is already present so that same ID is not saved twice.
Dim IsNewID As Boolean
IsNewID = True
If Not rs.EOF Then
    Do While Not rs.EOF
        If txtUserID.Text = rs.Fields("User_ID") Then
            IsNewID = False
            Exit Do
        End If
    rs.MoveNext
Loop
End If

If IsNewID = False Then
    MsgBox "This User ID is already in use. Please try again."
    txtUserID.Text = ""
    Exit Sub
End If
```

```
If Not rs.EOF Then
    Do While Not rs.EOF
        If txtUserID.Text = rs.Fields("User_ID") Then
            IsNewID = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID = False Then
    MsgBox "This User ID is already in use. Please try again."
    txtUserID.Text = ""
    Exit Sub
End If

rs.AddNew
'Check made to see if password and confirm password match if User type and User ID are Valid.
If txtPassword.Text = txtConfirmPassword.Text Then
    rs.Fields("Password") = txtPassword.Text
Else
    MsgBox ("The password and the confirm password do not match. Please re-enter them and try
again")
    txtPassword.Text = ""
    txtConfirmPassword.Text = ""
    Exit Sub
End If

rs.Fields("User_type") = cboUserType.Text
rs.Fields("User_ID") = txtUserID.Text

rs.Update

MsgBox ("User account has been created successfully.")
'The fields are emptied after the information is stored.
cboUserType.Text = ""
txtUserID.Text = ""
txtPassword.Text = ""
txtConfirmPassword = ""

End Sub

'Delete button click event.
Private Sub cmdDelete_Click()
'Check to make sure all important fields are filled.
```

```
If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Or  
txtConfirmPassword.Text = Empty Then  
    MsgBox "None of the fields can be kept empty."  
    Exit Sub  
End If  
'The selected record is deleted.  
rs.Delete  
MsgBox "The record has been deleted."  
'The fields on the form are cleared after deletion.  
cboUserType.Text = ""  
txtUserID.Text = ""  
txtPassword.Text = ""  
txtConfirmPassword = ""  
End Sub  
'Refresh button click event.  
Private Sub cmdRefresh_Click()  
'The fields are cleared.  
cboUserType.Text = ""  
txtUserID.Text = ""  
txtPassword.Text = ""  
txtConfirmPassword = ""  
End Sub  
'Search button click event.  
Private Sub cmdSearch_Click()  
'Fields are checked to see if any of the fields are empty.  
If cboUserType.Text = Empty Or txtUserID.Text = Empty Then  
    MsgBox "Please provide User type and User ID both."  
    Exit Sub  
End If  
'Information is loaded from the database.  
Set rs = New Recordset  
rs.Open "select * from User_login where User_type ='" & cboUserType.Text & "'", con,  
adOpenDynamic, adLockOptimistic  
  
Dim found As Boolean  
found = False  
'The database is checked to see if the User ID is present.  
Do While rs.EOF = False And Not found  
    If rs.Fields("User_ID") = txtUserID.Text Then  
        found = True  
        Exit Do  
    End If  
    rs.MoveNext  
Loop  
'Message if database is not found.  
If Not found Then  
    MsgBox ("User ID not found. Please check your User type and User ID and try again.")
```

```
Exit Sub
End If

txtPassword.Text = rs.Fields("Password")
txtConfirmPassword.Text = txtPassword.Text

UType = cboUserType.Text
UID = txtUserID.Text

End Sub

Private Sub cmdUpdate_Click()
'The fields are checked to see if they are all filled.
If cboUserType.Text = Empty Or txtUserID.Text = Empty Or txtPassword.Text = Empty Or
txtConfirmPassword.Text = Empty Then
    MsgBox "None of the fields can be kept empty."
    Exit Sub
End If

'Validation to ensure User type is not changed.
If cboUserType.Text <> UType Then
    MsgBox "User type cannot be changed."
    cboUserType.Text = UType
    Exit Sub
End If

'Validation to check if User ID is already present if User plans on changing the User ID.
If txtUserID <> UID Then
    If Not rs.EOF Then
        Dim IsNewID As Boolean
        IsNewID = True
        Do While Not rs.EOF
            If txtUserID.Text = rs.Fields("User_ID") Then
                IsNewID = False
                Exit Do
            End If
            rs.MoveNext
        Loop
    End If

    If IsNewID = False Then
        MsgBox "This User ID is already in use. Please try again."
        txtUserID.Text = ""
        Exit Sub
    End If
End If

'User information in the database is updated.
```

```

rs.MoveFirst

'Validation to make sure password and confirm password match.
If txtPassword.Text = txtConfirmPassword.Text Then
    rs.Fields("Password") = txtPassword.Text
Else
    'Message shown if password and confirm password don't match
    MsgBox ("The password and the confirm password do not match. Please check them and try again")
    Exit Sub
End If

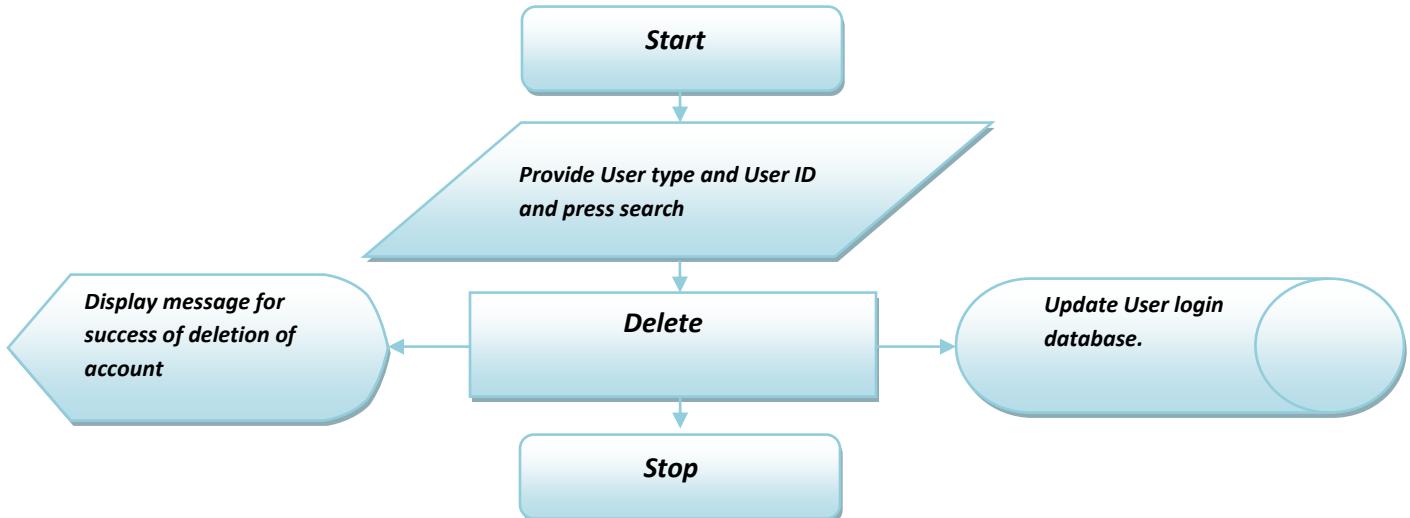
rs.Fields("User_ID") = txtUserID.Text

rs.Update
'Message shown to notify the User that the Update was successful.
MsgBox ("Record has been Updated successfully.")
'The fields are cleared after the Update is successful.
cboUserType.Text = ""
txtUserID.Text = ""
txtPassword.Text = ""
txtConfirmPassword = ""

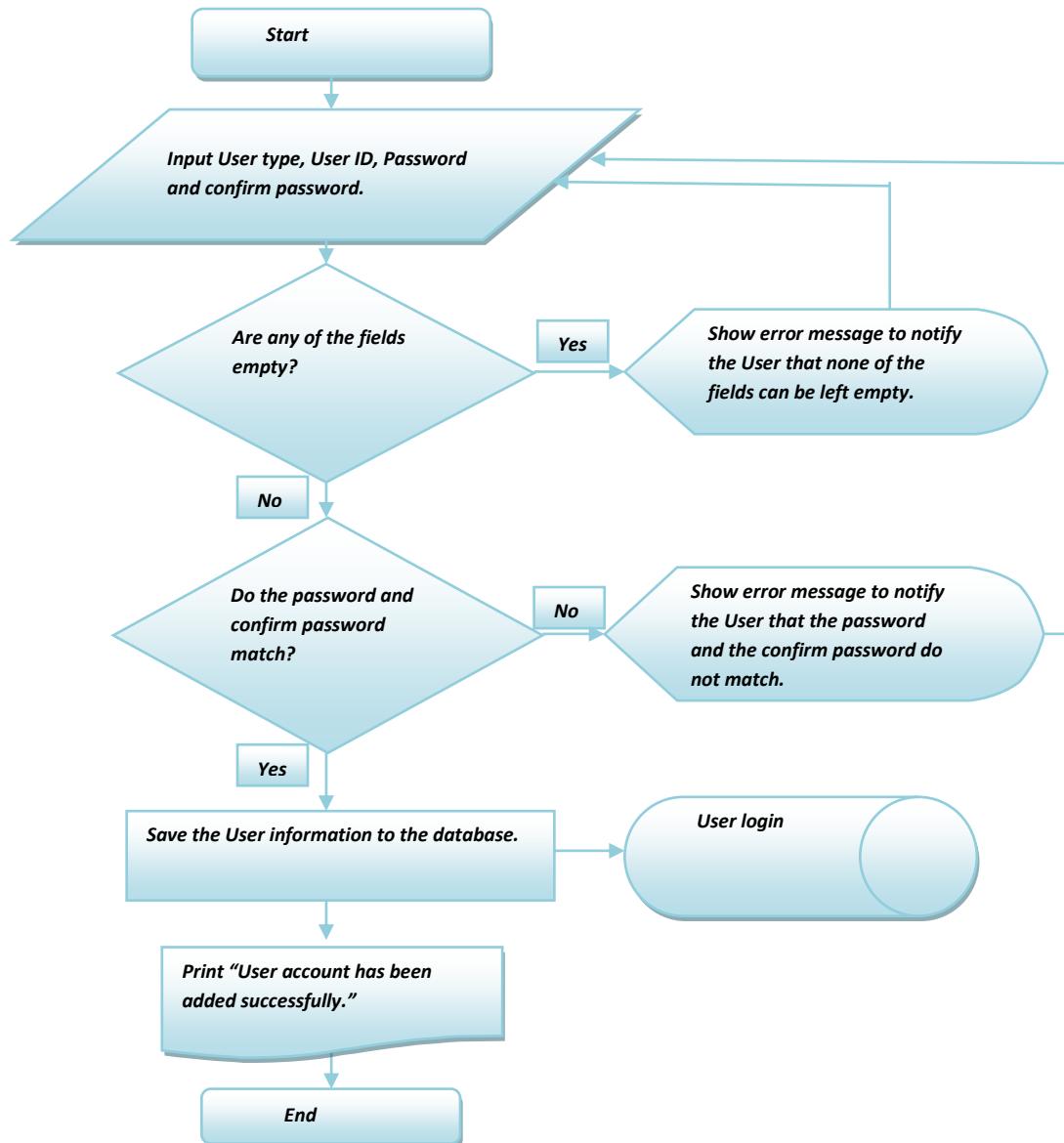
End Sub
'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
    Call dblLink
    lblDate.Caption = Date
End Sub

```

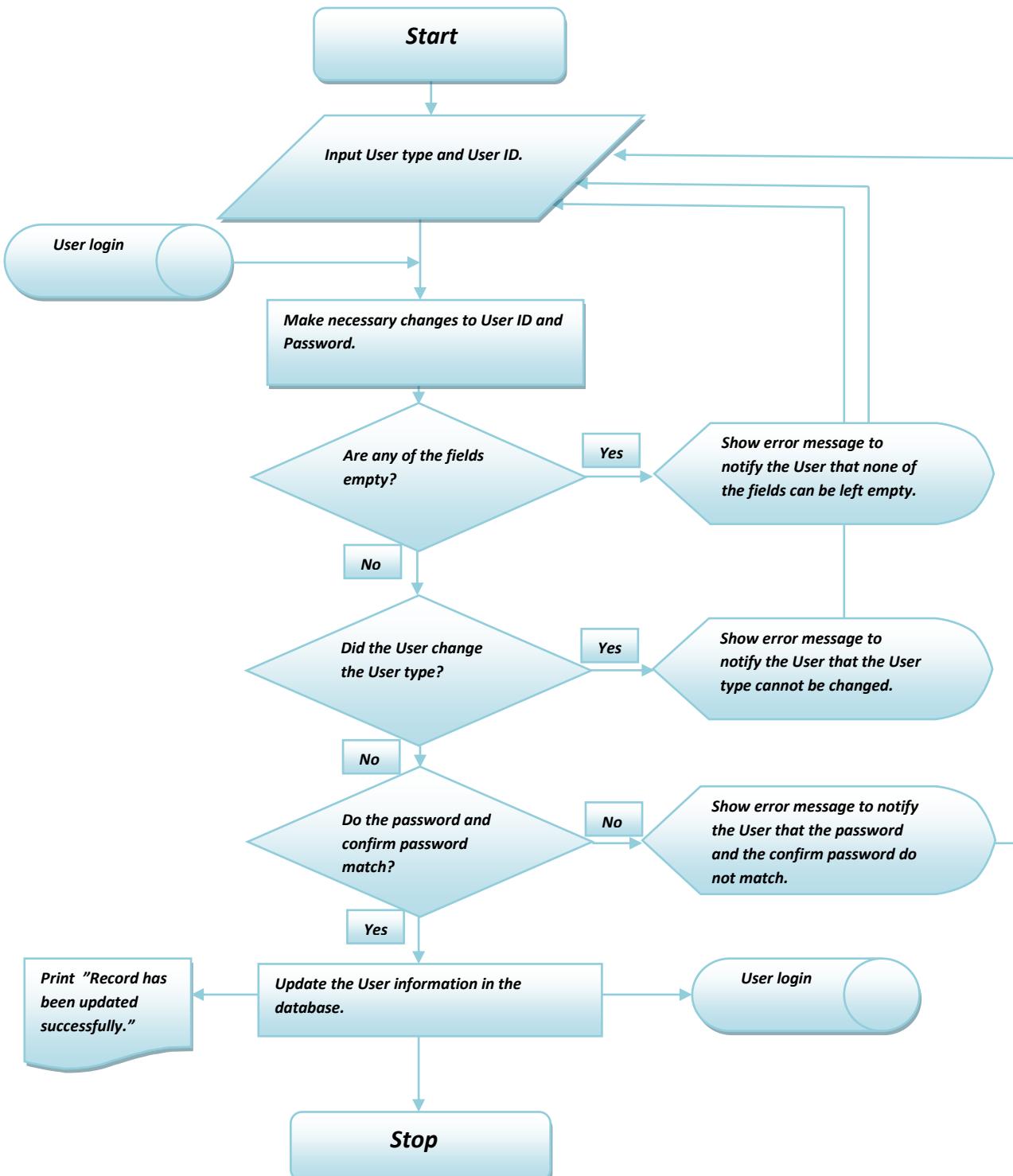
1. *Deletion of existing User:*



2. Adding a new User account:



3. Updating User Information:



4. Employee Entry form

The screenshot below shows the actual layout of the Employee Entry form.

The screenshot of the coding of the Employee Entry form is shown below.

```

Project1 - Microsoft Visual Basic [design] - [EmployeeEntry (Code)]
File Edit View Project Format Debug Run Query Diagram Tools Add-ins Window Help
Ln10, Col1

cmdDelete_Click
Dim EID As String

'Cancel button click event
Private Sub cmdCancel_Click()
'The form is unloaded.
Unload Me
End Sub

'Delete button click event.
Private Sub cmdDelete_Click()
If txtEmployeeID.Text = "" Or txtEmployeeName.Text = "" Or txtEmployeeAddress.Text = "" Or txtContactNumber.Text = "" Then
    MsgBox "Select the record to be deleted."
    Exit Sub
End If
'The selected record is deleted.
rs.Delete
'The Employee ID list is updated.
lstEmployeeID.RemoveItem (lstEmployeeID.ListIndex)
lstEmployeeID.Refresh
'Message shown to notify the user that the record is deleted from the database.
MsgBox "The Record has been deleted."
'The fields are cleared after deletion of a record.
txtEmployeeID.Text = ""
txtEmployeeName.Text = ""
txtEmployeeAddress.Text = ""
txtContactNumber.Text = ""
.cboDesig.Text = ""
.txtBasicSalary.Text = ""
lstEmployeeID.Clear
Call Form_Load
End Sub

'Make Entry button click event.
Private Sub cmdMakeEntry_Click()

'The fields are checked to make sure none of them are empty.
If txtEmployeeID.Text = "" Or txtEmployeeName.Text = "" Or txtEmployeeAddress.Text = "" Or txtContactNumber.Text = "" Then
    MsgBox "Select the record to be deleted."
End If
End Sub

```

The coding of the Employee Entry form is shown below.

```
Dim EID As String

'Cancel button click event
Private Sub cmdCancel_Click()
'The form is unloaded.
    Unload Me
End Sub

'Delete button click event.
Private Sub cmdDelete_Click()
    If txtEmployeeID.Text = Empty Or txtEmployeeName.Text = Empty Or txtEmployeeAddress.Text =
Empty Or txtContactNumber.Text = Empty Or cboDesig.Text = Empty Or txtBasicSalary.Text = Empty
Then
    'Message shown to notify the user that none of the fields can be left empty.
        MsgBox "Select the record to be deleted."
        Exit Sub
    End If
'The selected record is deleted.
    rs.Delete
'The Employee ID list is updated.
    lstEmployeeID.RemoveItem (lstEmployeeID.ListIndex)
    lstEmployeeID.Refresh
'Message shown to notify the user that the record is deleted from the database.
    MsgBox "The Record has been deleted."
'The fields are cleared after deletion of a record.
    txtEmployeeID.Text = ""
    txtEmployeeName.Text = ""
    txtEmployeeAddress.Text = ""
    txtContactNumber.Text = ""
    cboDesig.Text = ""
    txtBasicSalary.Text = ""
    lstEmployeeID.Clear
    Call Form_Load
End Sub

'Make Entry button click event.
Private Sub cmdMakeEntry_Click()

'The fields are checked to make sure none of them are empty.
    If txtEmployeeID.Text = Empty Or txtEmployeeName.Text = Empty Or txtEmployeeAddress.Text =
Empty Or txtContactNumber.Text = Empty Or cboDesig.Text = Empty Or txtBasicSalary.Text = Empty
Then
    'Message shown to notify the user that none of the fields can be left empty.
        MsgBox "None of the fields can be left empty."
        Exit Sub
    End If
```

```
'Length check to make sure contact number is of 11 digits.  
If Len(txtContactNumber.Text) <> 11 Then  
    MsgBox "Number of digits used for contact number is invalid."  
    Exit Sub  
End If  
  
'The data of the new employee are added to the Employee Entry database.  
Set rs = New Recordset  
rs.Open "select * from Employee_entry", con, adOpenDynamic, adLockOptimistic  
'Validation check to make sure same Employee is not used twice.  
If Not rs.EOF Then  
    Dim IsNewID As Boolean  
    IsNewID = True  
    Do While Not rs.EOF  
        If txtEmployeeID.Text = rs.Fields("EmployeeID") Then  
            IsNewID = False  
            Exit Do  
        End If  
        rs.MoveNext  
    Loop  
End If  
  
If IsNewID = False Then  
    MsgBox "This ID is already in use. Please try again."  
    txtEmployeeID.Text = ""  
    Exit Sub  
End If  
  
rs.AddNew  
rs.Fields("EmployeeID") = txtEmployeeID.Text  
rs.Fields("EName") = txtEmployeeName.Text  
rs.Fields("Address") = txtEmployeeAddress.Text  
rs.Fields("Contact") = txtContactNumber.Text  
rs.Fields("Designation") = cboDesig.Text  
rs.Fields("Basic_salary") = txtBasicSalary.Text  
rs.Fields("Date_of_entry") = lblDate.Caption  
rs.Update  
  
'Message shown to notify the user that the record has been added successfully.  
MsgBox "Record has been added successfully."  
'The fields are cleared after the information is added to the database.  
txtEmployeeID.Text = ""  
txtEmployeeName.Text = ""  
txtEmployeeAddress.Text = ""  
txtContactNumber.Text = ""  
cboDesig.Text = ""  
txtBasicSalary.Text = ""
```

```
IstEmployeeID.Clear  
IstEmployeeID.Refresh
```

```
Call Form_Load
```

```
End Sub
```

```
'Refresh button click event.
```

```
Private Sub cmdRefresh_Click()
```

```
'The fields are cleared at the click of the button.
```

```
IstEmployeeID.Clear
```

```
IstEmployeeID.Refresh
```

```
txtEmployeeID.Text = ""
```

```
txtEmployeeName.Text = ""
```

```
txtEmployeeAddress.Text = ""
```

```
txtContactNumber.Text = ""
```

```
cboDesig.Text = ""
```

```
txtBasicSalary.Text = ""
```

```
Call Form_Load
```

```
End Sub
```

```
'Update button click event.
```

```
Private Sub cmdUpdate_Click()
```

```
'Validation check to make sure none of the fields are kept empty.
```

```
If txtEmployeeID.Text = "" Or txtEmployeeName.Text = "" Or txtEmployeeAddress.Text = "" Or  
txtContactNumber.Text = "" Or cboDesig.Text = "" Or txtBasicSalary.Text = "" Then
```

```
MsgBox "Check if all the required information are filled and then try again."
```

```
Exit Sub
```

```
End If
```

```
'Ensuring the Employee ID is not changed as it serves as the basis of identification of the employee.
```

```
If txtEmployeeID.Text <> EID Then
```

```
MsgBox "Employee ID cannot be changed."
```

```
txtEmployeeID.Text = EID
```

```
Exit Sub
```

```
End If
```

```
'Updating Employee Entry database information for a particular Employee record.
```

```
rs.Fields("EName") = txtEmployeeName.Text
```

```
rs.Fields("Address") = txtEmployeeAddress.Text
```

```
rs.Fields("Contact") = txtContactNumber.Text
```

```
rs.Fields("Designation") = cboDesig.Text
```

```
rs.Fields("Basic_salary") = txtBasicSalary.Text
```

```
rs.Update
```

```
'Message shown to notify the user that the update was successful.
```

```
MsgBox "The record has been updated."
```

'The fields on the form are emptied to ready the form for next transaction.

```
IstEmployeeID.Clear
IstEmployeeID.Refresh
txtEmployeeID.Text = ""
txtEmployeeName.Text = ""
txtEmployeeAddress.Text = ""
txtContactNumber.Text = ""
cboDesig.Text = ""
txtBasicSalary.Text = ""
```

Call Form_Load

End Sub

Private Sub Form_Load()

'Calling the function to connect to the database.

Call dblink

'Updating the date.

lblDate.Caption = Format(Now, "dd/mm/yyyy")

'The Employee IDs present in the Employee database are loaded onto the Employee ID list.

```
Set rs = New Recordset
rs.Open "select * from Employee_entry", con, adOpenDynamic, adLockOptimistic
Do While Not rs.EOF
    IstEmployeeID.AddItem (rs.Fields("EmployeeID"))
    rs.MoveNext
Loop
```

rs.MoveLast

txtEmployeeID.Text = CInt(rs!EmployeeID) + 1

End Sub

'Employee ID list click event.

Private Sub IstEmployeeID_Click()

Set rs = New Recordset

'Employee information loaded from database on selection of Employee ID from Employee list.

```
rs.Open "select * from Employee_entry where EmployeeID ='" & IstEmployeeID.Text & "'", con,
adOpenDynamic, adLockOptimistic
```

```
txtEmployeeID.Text = rs.Fields("EmployeeID")
txtEmployeeName.Text = rs.Fields("EName")
txtEmployeeAddress.Text = rs.Fields("Address")
txtContactNumber.Text = rs.Fields("Contact")
cboDesig.Text = rs.Fields("Designation")
txtBasicSalary.Text = rs.Fields("Basic_salary")
```

EID = txtEmployeeID.Text

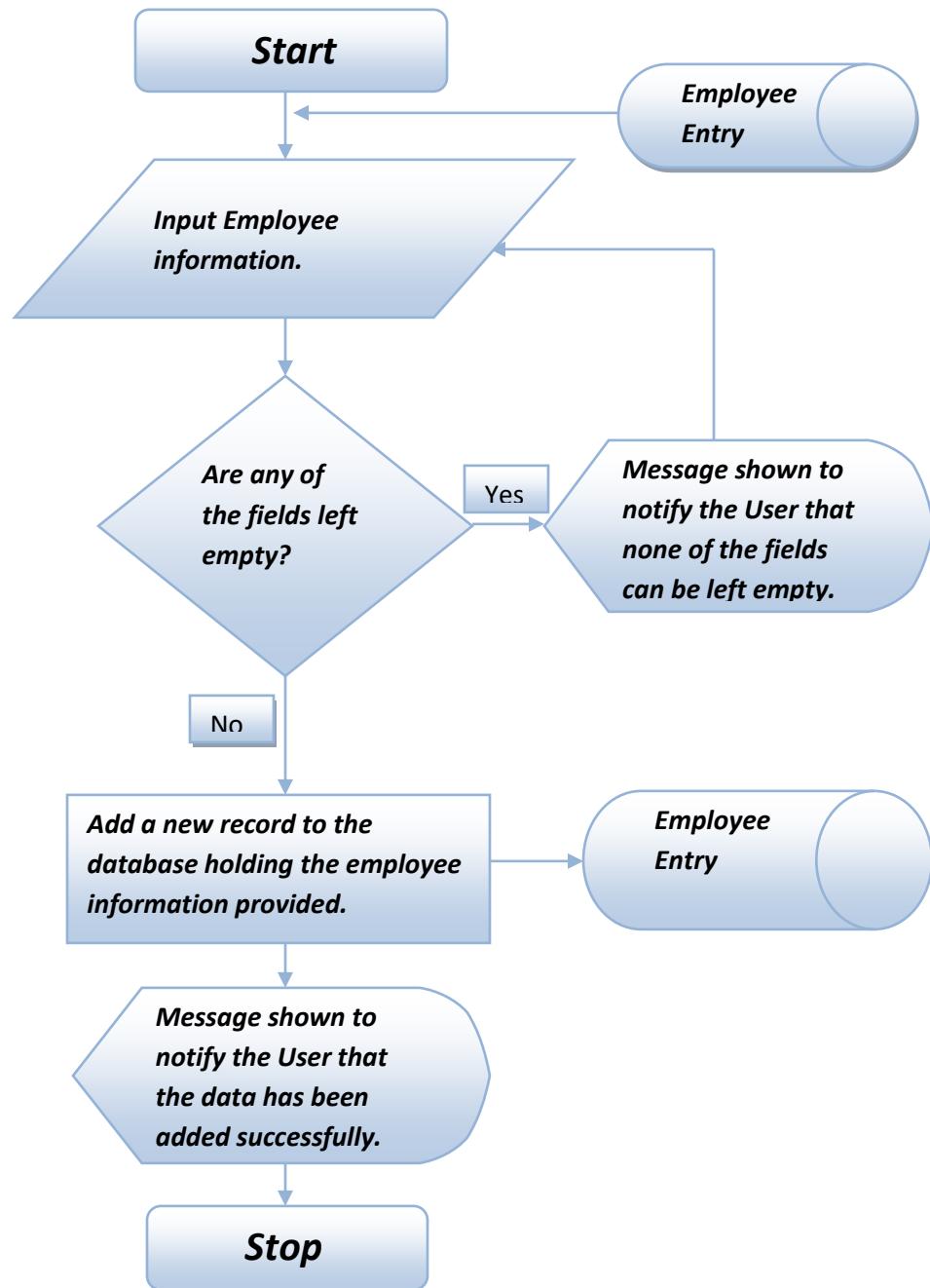
End Sub

```

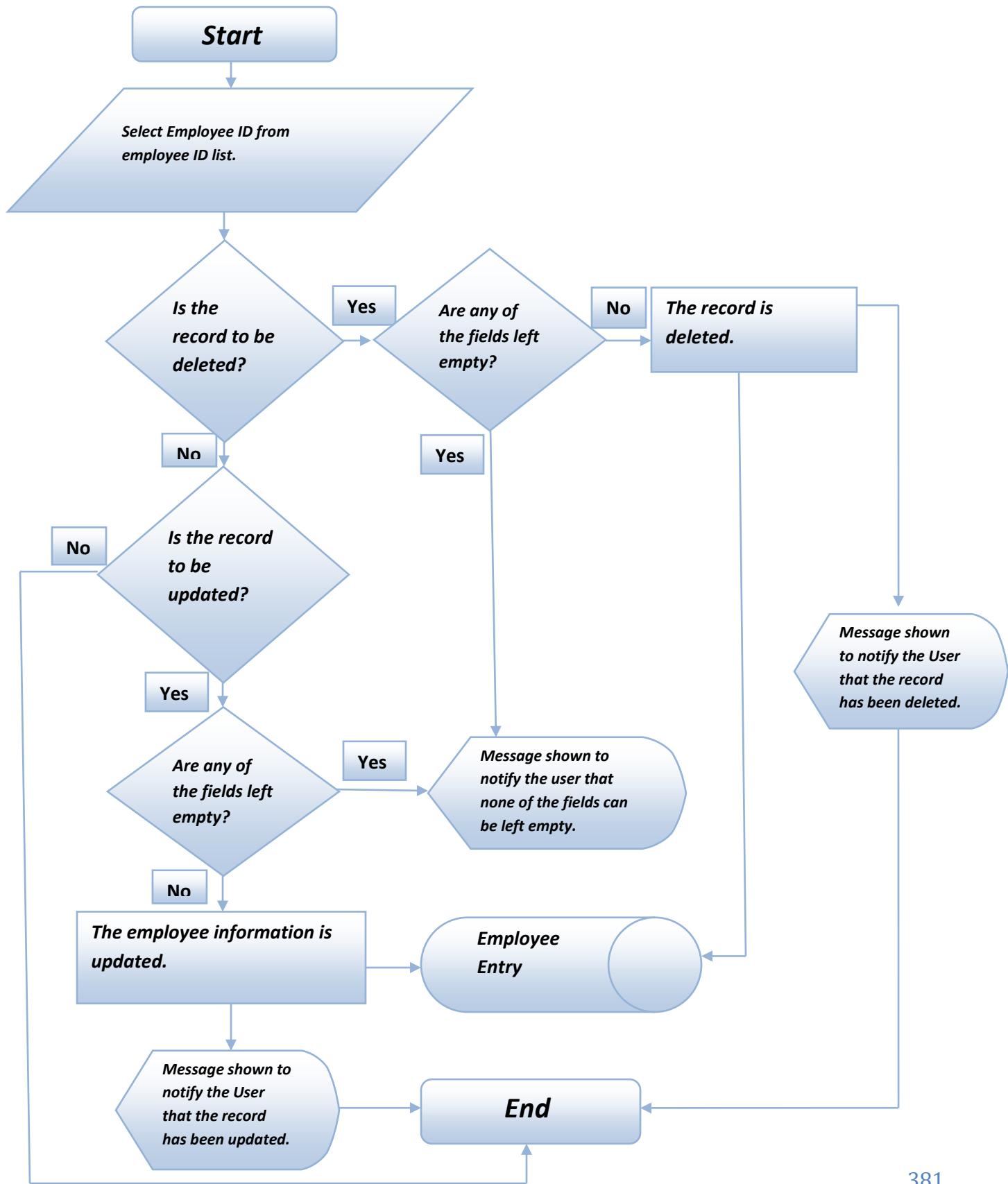
Private Sub txtContactNumber_KeyPress(KeyAscii As Integer)
'Validating that the new vat on picnicspot entered is numeric
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub

```

1. Adding Employee Information to the database:



2. Updating or deleting employee information:



5. Package Entry form

The screenshot below is an actual layout of the Package Entry form.

The screenshot shows the 'Package entry' window with the title 'Package Entry' and date '29-03-2015'. The form is divided into sections:

- Package information:** Contains fields for Package type (dropdown), Name of package (text box), Package ID (text box with value '1123'), Cost per person (text box), Accommodation type (dropdown), Journey Date (date picker with value '29-Mar-15'), Arrival time (time picker with value '12:00:00 AM'), Starting time (time picker with value '12:00:00 AM'), and Duration (text box with dropdown menu).
- Update:** Contains a 'Package type' dropdown and a 'Package Name list' list box.
- Number of reservations available?**: A text box.
- Itinerary**: A large text area.
- Package details**: A large text area.
- Action buttons:** Save, Refresh, Update, Delete, Cancel.

The screenshot below shows the actual coding of the Package Entry form.

The screenshot shows the Microsoft Visual Basic design environment with the 'Project1 - Microsoft Visual Basic [design] - [PackageEntry (Code)]' window open. The code in the 'txtRes' module is as follows:

```

Dim OldPID As String
'Click event Package type combobox of the update section of the form.
Private Sub cboPackageTypeUpdate_Click()
'The Package name list cleared if package type is changed.
    lstPackageNameUpdate.Clear
    Set rs = New Recordset
    'Opening the Package Entry database.
    rs.Open "select * from Package_entry where Package_type ='" & cboPackageTypeUpdate.Text & "'", con, adOpenDynamic, adLockOptimistic
    'Loading the Package names of the particular type selected into the Package name list.
    Do While Not rs.EOF
        lstPackageNameUpdate.AddItem (rs.Fields("Package_name"))
        rs.MoveNext
    Loop
End Sub
'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
    Unload Me
End Sub
'Delete button click event.
Private Sub cmdDelete_Click()
    Set rs = New Recordset
    'Validation check to make sure none of the important fields are left empty
    If cboPackageType.Text = "" Or txtPackageName.Text = "" Or txtCostPerPerson.Text = "" Or txtPackageDetails.Text = "" Then
        'Message shown if any empty fields are found.
        MsgBox "All the information need to be provided."
        Exit Sub
    End If
    'Deletion of selected record with the particular Package name from the Package Entry database.
    rs.Open "delete * from Package_entry where Package_name=''" & lstPackageNameUpdate.Text & "' and Package_type=''" & cboPackageTypeUpdate.Text & "'"
    lstPackageNameUpdate.Clear
    lstPackageNameUpdate.Refresh
    cboPackageTypeUpdate.Clear
    cboPackageTypeUpdate.Refresh
    cboPackageTypeUpdate.Text = ""
End Sub

```

The 'Properties' window on the right shows the form's properties:

- Name: PackageEntry
- Appearance: 1 - 3D
- AutoRedraw: False
- BackColor: #F8E8D8
- BorderStyle: 2 - Sizable
- Caption: Package entry
- ClipControls: True
- ControlBox: True
- DrawMode: 13 - Copy Pen
- DrawStyle: 0 - Solid

The coding of the Package Entry form:

```

Dim OldPID As String
'Click event Package type combobox of the update section of the form.
Private Sub cboPackageTypeUpdate_Click()
'The Package name list cleared if package type is changed.
    lstPackageNameUpdate.Clear
    Set rs = New Recordset
'Opening the Package Entry database.
    rs.Open "select * from Package_entry where Package_type ='" & cboPackageTypeUpdate.Text & "'",
    con, adOpenDynamic, adLockOptimistic
'Loading the Package names of the particular type selected into the Package name list.
    Do While Not rs.EOF
        lstPackageNameUpdate.AddItem (rs.Fields("Package_name"))
        rs.MoveNext
    Loop
End Sub

'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
    Unload Me
End Sub

'Delete button click event
Private Sub cmdDelete_Click()
    Set rs = New Recordset
    'Validation check to make sure none of the important fields are left empty
    If cboPackageType.Text = Empty Or txtPackageName.Text = Empty Or txtPackageDetails.Text = Empty
    Or txtCostPerPerson.Text = Empty Or cboAccommodation = Empty Or dtpJourneyDate.Value = Date Or
    dtpStartingTime = Default Or dtpArrivalTime.Value = Default Or txtDuration.Text = Empty Or
    txtItinerary.Text = Empty Then
        'Message shown if any empty fields are found.
        MsgBox "All the information need to be provided."
        Exit Sub
    End If
'Deletion of selected record with the particular Package name from the Package Entry database.
    rs.Open "delete * from Package_entry where Package_name=''" & lstPackageNameUpdate.Text & "'"
    and Package_type=''" & cboPackageTypeUpdate.Text & "'", con, adOpenDynamic, adLockOptimistic

    lstPackageNameUpdate.Clear
    lstPackageNameUpdate.Refresh
    cboPackageTypeUpdate.Clear
    cboPackageTypeUpdate.Refresh

    cboPackageTypeUpdate.Text = ""
    cboPackageType.Text = ""
    txtPackageName.Text = ""
    txtCostPerPerson.Text = ""

```

```
cboAccommodation.Text = ""  
dtpJourneyDate.Value = Default  
dtpStartingTime.Value = Default  
dtpArrivalTime.Value = Default  
txtDuration.Text = ""  
txtItinerary.Text = ""  
txtPackageDetails.Text = ""  
txtPackageID.Text = ""  
txtNRes.Text = ""  
lstPackageNameUpdate.Clear  
lstPackageNameUpdate.Refresh  
MsgBox "Record has been deleted successfully."
```

```
lstPackageNameUpdate.Clear  
lstPackageNameUpdate.Refresh
```

Call Form_Load

End Sub

'Refresh button click event.

```
Private Sub cmdRefresh_Click()
```

'Emptying all the fields and refreshing the forms if the user requires it.

```
cboPackageTypeUpdate.Text = ""  
cboPackageType.Text = ""  
txtPackageName.Text = ""  
txtCostPerPerson.Text = ""  
cboAccommodation.Text = ""  
dtpJourneyDate.Value = Default  
dtpStartingTime.Value = Default  
dtpArrivalTime.Value = Default  
txtDuration.Text = ""  
txtItinerary.Text = ""  
txtPackageDetails.Text = ""  
txtPackageID.Text = ""  
txtNRes.Text = ""  
lstPackageNameUpdate.Clear  
lstPackageNameUpdate.Refresh
```

Call Form_Load

End Sub

'Save button click event.

```
Private Sub cmdSave_Click()
```

'Validation check to make sure none of the important fields are left empty

```
If cboPackageType.Text = Empty Or txtPackageName.Text = Empty Or txtPackageDetails.Text = Empty  
Or txtCostPerPerson.Text = Empty Or cboAccommodation = Empty Or dtpJourneyDate.Value = Date Or  
dtpStartingTime = Default Or dtpArrivalTime.Value = Default Or txtDuration.Text = Empty Or  
txtItinerary.Text = Empty Then
```

```
'Message shown if any empty fields are found.  
MsgBox "All the information need to be provided."  
Exit Sub  
End If  
'Validation check to make sure same Package ID is not used twice.  
If Not rs.EOF Then  
    Dim IsOldID As Boolean  
    IsOldID = False  
    Do While Not rs.EOF  
        If txtPackageID.Text = rs.Fields("PackageID") Then  
            IsOldID = True  
            Exit Do  
        End If  
        rs.MoveNext  
    Loop  
End If  
  
If IsOldID = True Then  
    MsgBox "This Package ID is already in use. Please provide a differnt Package ID."  
    txtPackageID.Text = ""  
    Exit Sub  
End If  
'Storing Package information into the database.  
Set rs = New Recordset  
rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic  
rs.AddNew  
rs.Fields("Package_type") = cboPackageType.Text  
rs.Fields("Package_name") = txtPackageName.Text  
rs.Fields("Accommodation_type") = cboAccommodation.Text  
rs.Fields("Journey_date") = dtpJourneyDate.Value  
rs.Fields("Starting_time") = dtpStartingTime.Value  
rs.Fields("Duration/days") = txtDuration.Text  
rs.Fields("Arrival_time") = dtpArrivalTime.Value  
rs.Fields("Cost_per_person") = txtCostPerPerson.Text  
rs.Fields("Date_of_entry") = lblDate.Caption  
rs.Fields("Itinerary") = txtItinerary.Text  
rs.Fields("Package_details") = txtPackageDetails.Text  
rs.Fields("PackageID") = txtPackageID.Text  
rs.Fields("NumberOfReservations") = txtNRes.Text  
  
rs.Update  
'Message shown to notify the user that the package information have been added successfully.  
MsgBox "Record has been added successfully"  
'Emptying the fields after the information is added to the database.  
cboPackageTypeUpdate.Text = ""  
cboPackageType.Text = ""  
txtPackageName.Text = ""
```

```

txtCostPerPerson.Text = ""
cboAccommodation.Text = ""
dtpJourneyDate.Value = Default
dtpStartingTime.Value = Default
dtpArrivalTime.Value = Default
txtDuration.Text = ""
txtItinerary.Text = ""
txtPackageDetails.Text = ""
txtPackageID.Text = ""
txtNRes.Text = ""
lstPackageNameUpdate.Clear
lstPackageNameUpdate.Refresh

rs.MoveLast
txtPackageID.Text = CInt(rs!PackageID) + 1

'Reloading the names on the Package Type combobox of the update section to make sure that the new
names appear immediately.
Set rs = New Recordset
rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic

If rs.EOF = False Then
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboPackageTypeUpdate.ListCount - 1
            If cboPackageTypeUpdate.List(i) = rs.Fields("Package_type") Then
                IsNewType = False
            End If
        Next
        If IsNewType = True Then
            cboPackageTypeUpdate.AddItem (rs.Fields("Package_type"))
        End If
        rs.MoveNext
    Loop
End If

End Sub

'Update button click event.
Private Sub cmdUpdate_Click()
'Validation check to make sure none of the fields are left empty.
    If cboPackageType.Text = Empty Or txtPackageName.Text = Empty Or txtPackageDetails.Text = Empty
    Or txtCostPerPerson.Text = Empty Or cboAccommodation = Empty Or dtpJourneyDate.Value = Default
    Or dtpStartingTime = Default Or dtpArrivalTime.Value = Default Or txtDuration.Text = Empty Or
    txtItinerary.Text = Empty Then

```

```
    MsgBox "All the information need to be provided."
    Exit Sub
End If

'Ensuring Package ID is not changed.
If OldPID <> txtPackageID.Text Then
    MsgBox "PackageID cannot be changed."
    txtPackageID.Text = OldPID
    Exit Sub
End If

'Updating information on the database.
rs.Fields("Package_type") = cboPackageType.Text
rs.Fields("Package_name") = txtPackageName.Text
rs.Fields("Accommodation_type") = cboAccommodation.Text
rs.Fields("Journey_date") = dtpJourneyDate.Value
rs.Fields("Starting_time") = dtpStartingTime.Value
rs.Fields("Duration/days") = txtDuration.Text
rs.Fields("Arrival_time") = dtpArrivalTime.Value
rs.Fields("Cost_per_person") = txtCostPerPerson.Text
rs.Fields("Date_of_entry") = lblDate.Caption
rs.Fields("Itinerary") = txtItinerary.Text
rs.Fields("Package_details") = txtPackageDetails.Text
rs.Fields("PackageID") = txtPackageID.Text
rs.Fields("NumberOfReservations") = txtNRes.Text
rs.Fields("NumberReserved") = 0
rs.Update

'Message shown to notify user that the information is updated.
MsgBox "Package information has been updated successfully."
'The fields are updated after the database is updated and the form is refreshed.
cboPackageTypeUpdate.Text = ""
cboPackageType.Text = ""
txtPackageName.Text = ""
txtCostPerPerson.Text = ""
cboAccommodation.Text = ""
dtpJourneyDate.Value = Default
dtpStartingTime.Value = Default
dtpArrivalTime.Value = Default
txtDuration.Text = ""
txtItinerary.Text = ""
txtPackageDetails.Text = ""
txtPackageID.Text = ""
txtNRes.Text = ""
lstPackageNameUpdate.Clear
lstPackageNameUpdate.Refresh

Call Form_Load
```

```

End Sub
'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
    Call dblink
'Showing current date.
    lblDate.Caption = Format(Now, "dd/mm/yyyy")
'Current date assigned to Journey date datepicker.
    dtpJourneyDate.Value = Date
'Loading Package types into the Package type combobox of the update section.
    Set rs = New Recordset
    rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic

    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True
            For i = 0 To cboPackageTypeUpdate.ListCount - 1
                If cboPackageTypeUpdate.List(i) = rs.Fields("Package_type") Then
                    IsNewType = False
                End If
            Next
            If IsNewType = True Then
                cboPackageTypeUpdate.AddItem (rs.Fields("Package_type"))
            End If
            rs.MoveNext
        Loop
    End If

'Updating Package ID textbox.
    rs.MoveLast
    txtPackageID.Text = CInt(rs!PackageID) + 1

End Sub
'Click event of the list of Package names.
Private Sub lstPackageNameUpdate_Click()
'The package information is loaded onto the form at the selection of a Package name.
    Set rs = New Recordset
    rs.Open "select * from Package_entry where Package_name ='" & lstPackageNameUpdate.Text & "'", con, adOpenDynamic, adLockOptimistic
    cboPackageType.Text = rs.Fields("Package_type")
    txtPackageName.Text = rs.Fields("Package_name")
    txtItinerary.Text = rs.Fields("Itinerary")
    txtPackageDetails.Text = rs.Fields("Package_details")
    cboAccommodation.Text = rs.Fields("Accommodation_type")
    dtpStartingTime.Value = rs.Fields("Starting_time")

```

```
txtDuration.Text = rs.Fields("Duration/days")
dtpArrivalTime.Value = rs.Fields("Arrival_time")
dtpJourneyDate.Value = rs.Fields("Journey_date")
txtCostPerPerson.Text = rs.Fields("Cost_per_person")
txtPackageID.Text = rs.Fields("PackageID")
txtNRes.Text = rs.Fields("NumberOfReservations")
rs.Update
```

```
OldPID = txtPackageID.Text
```

```
End Sub
```

```
'Cost per person keypress event.
```

```
Private Sub txtCostPerPerson_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure only numerical data is entered.
```

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

```
'Duration keypress event.
```

```
Private Sub txtDuration_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure only numerical data is entered.
```

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

```
'Number of reservations keypress event.
```

```
Private Sub txtNRes_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure only numerical data is entered.
```

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

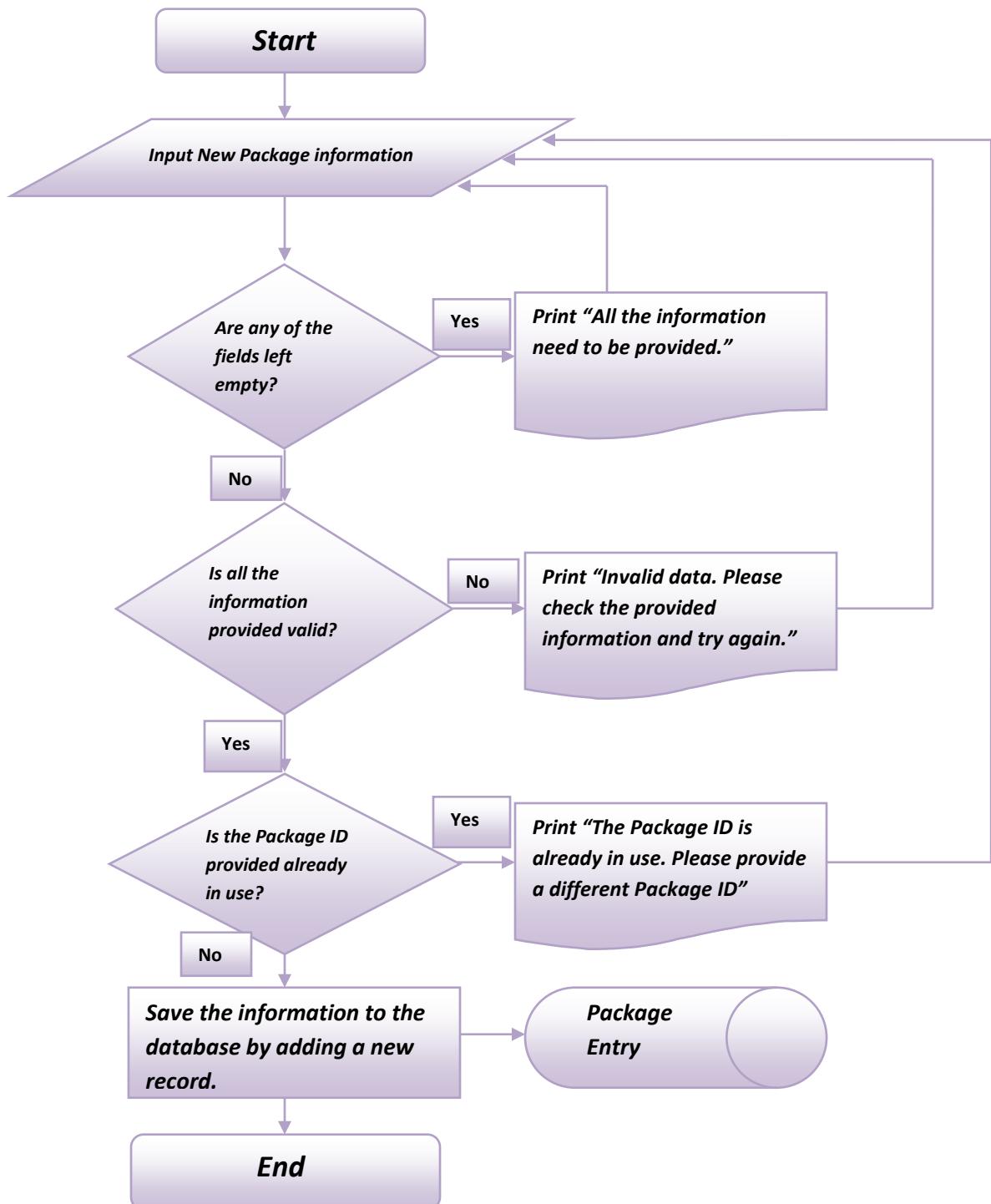
```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

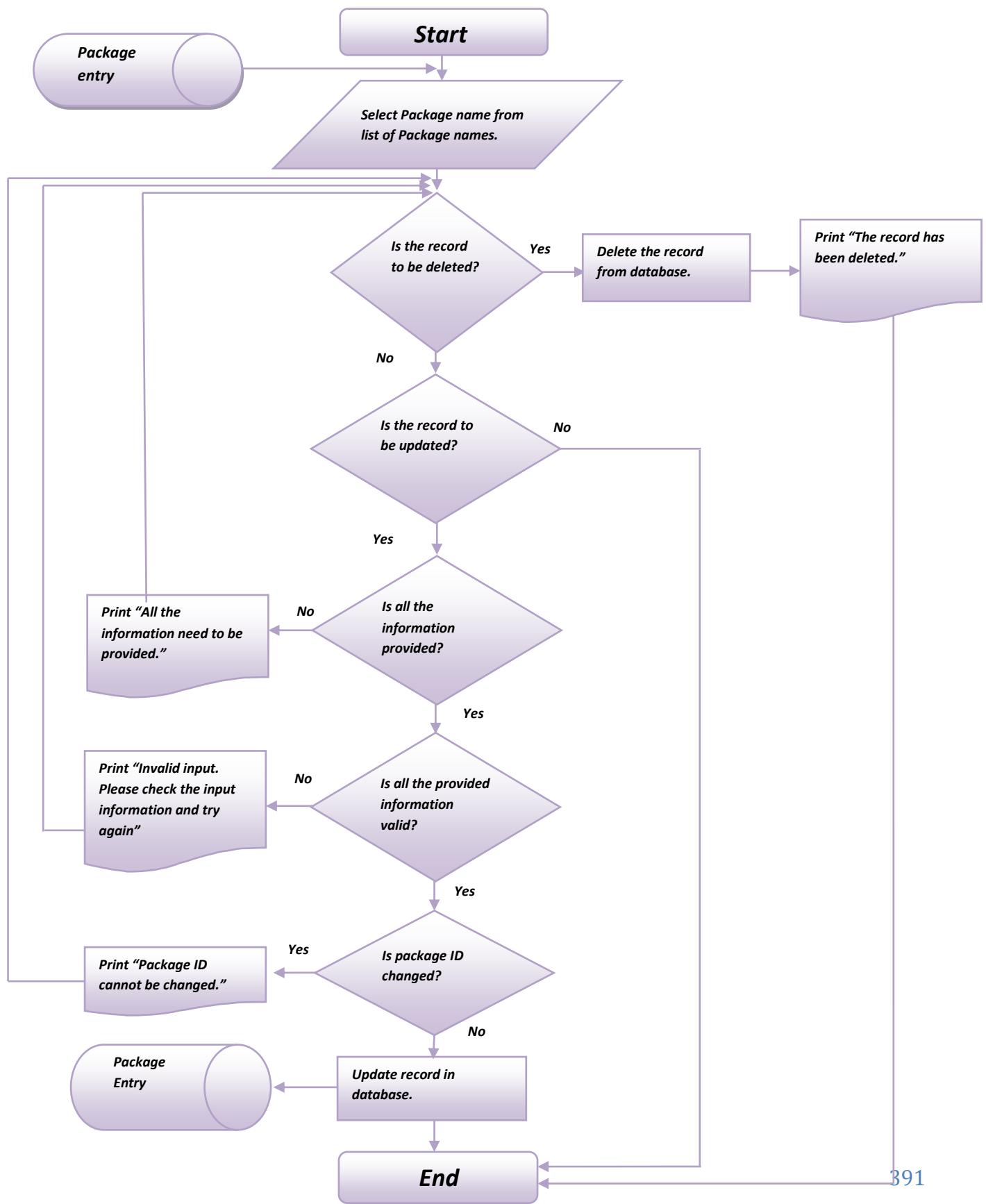
```
End If
```

```
End Sub
```

1. Adding new Packages to the database:



2. Package update or delete:



6. Package Booking form

The screenshot below shows the actual layout of the Package Booking form.

The screenshot below shows the actual coding of the Package Booking form.

```

Project1 - Microsoft Visual Basic [design] - [PackageBooking (Code)]
File Edit View Project Format Debug Run Query Diagram Tools Add-ins Window Help
L1, Col1

Private Sub lstPackageName_Click()
    Set rs = New Recordset
    'Clearing data from text boxes in case they need to be changed.
    txtNPeople.Text = ""
    txtNetTotal.Text = ""
    'Package information loaded automatically from Package_Entry database at the selection of a Package name.
    rs.Open "select * from Package_Entry where Package_name = '" & lstPackageName.Text & "'", con, adOpenDynamic, adLockOptimistic
    'The package information is loaded automatically at the selection of a package.
    txtCostPerPerson.Text = rs.Fields("Cost_per_person")
    lblItinerary.Caption = rs.Fields("Itinerary")
    lblPackageDetails.Caption = rs.Fields("Package_details")
    lblJourneyDate.Caption = rs.Fields("Journey_date")
    lblArrivalTime.Caption = rs.Fields("Arrival_time")
    lblStartingTime.Caption = rs.Fields("Starting_time")
    lblAccommodation.Caption = rs.Fields("Accommodation_type")
    lblDuration.Caption = rs.Fields("Duration/days")
    txtPackageID.Text = rs.Fields("PackageID")
    lblVacancies.Caption = rs.Fields("NumberOfReservations") - rs.Fields("NumberReserved")
End Sub

Private Sub cboPackageType_Click()
    'The Package_name list is always cleared when the user changes the package type in case the old imformation is not overridden.
    lstPackageName.Clear
    txtNPeople.Text = ""
    txtNetTotal.Text = ""
    txtPackageID = ""
    lblItinerary.Caption = ""
    lblPackageDetails.Caption = ""
    lblAccommodation.Caption = ""
    lblJourneyDate.Caption = ""
    lblStartingTime.Caption = ""
    lblArrivalTime.Caption = ""
    lblDuration.Caption = ""
End Sub

```

The coding of the Package Booking form is shown below.

```
'Package name list click event.
Private Sub lstPackageName_Click()
    Set rs = New Recordset
'Clearing data from text boxes in case they need to be changed.
    txtNPeople.Text = ""
    txtNetTotal.Text = ""
'Package information loaded automatically from Package Entry database at the selection of a Package
name.
    rs.Open "select * from Package_entry where Package_name = '" & lstPackageName.Text & "'", con,
adOpenDynamic, adLockOptimistic

'The package information is loaded automatically at the selection of a package.
    txtCostPerPerson.Text = rs.Fields("Cost_per_person")
    lblItinerary.Caption = rs.Fields("Itinerary")
    lblPackageDetails.Caption = rs.Fields("Package_details")
    lblJourneyDate.Caption = rs.Fields("Journey_date")
    lblArrivalTime.Caption = rs.Fields("Arrival_time")
    lblStartingTime.Caption = rs.Fields("Starting_time")
    lblAccommodation.Caption = rs.Fields("Accommodation_type")
    lblDuration.Caption = rs.Fields("Duration/days")
    txtPackageID.Text = rs.Fields("PackageID")
    lblVacancies.Caption = rs.Fields("NumberOfReservations") - rs.Fields("NumberReserved")
End Sub
```

```
'Package type combobox click event.
Private Sub cboPackageType_Click()
'The Package name list is always cleared when the user changes the package type in case the old
information is not overriden.
    lstPackageName.Clear
    txtNPeople.Text = ""
    txtNetTotal.Text = ""
    txtPackageID = ""
    lblItinerary.Caption = ""
    lblPackageDetails = ""
    lblAccommodation.Caption = ""
    lblJourneyDate.Caption = ""
    lblStartingTime.Caption = ""
    lblArrivalTime.Caption = ""
    lblDuration.Caption = ""
```

```
'Package names are loaded onto the Package names list on the selection of a particular package type.
Only packages of this particular type are selected and loaded onto the list.
'Opening the Package Entry database.
    Set rs = New Recordset
'Opening the Package Entry database.
```

```

rs.Open "select * from Package_entry where Package_type ='" & cboPackageType.Text & "'", con,
adOpenDynamic, adLockOptimistic
'Loading the Package names of the particular type selected into the Package name list.
Do While Not rs.EOF
    lstPackageName.AddItem (rs.Fields("Package_name"))
    rs.MoveNext
Loop

End Sub

'Confirm button click event.
Private Sub cmdConfirm_Click()
'Validation check to make sure all the important fields are filled.
    If cboPackageType.Text = "" Or txtPackageID.Text = "" Or txtCostPerPerson.Text = "" Or txtNPeople =
    "" Or txtBookingID.Text = "" Or txtCustomerName.Text = "" Or txtAddress.Text = "" Or
    txtContactNumber.Text = "" Then
        'Message to notify the user that all of the information need to be provided.
        MsgBox "All the Package information and customer information need to provided in order to
proceed with the booking."
        Exit Sub
    End If

'Length check to make sure contact number is of 11 digits.
    If Len(txtContactNumber.Text) <> 11 Then
        MsgBox "Number of digits used for contact number is invalid."
        Exit Sub
    End If

    Set rs = New Recordset
'Updating reservation information in the Package Entry database.
    rs.Open "select * from Package_entry where Package_name ='" & lstPackageName.Text & "'", con,
adOpenDynamic, adLockOptimistic
    p = rs.Fields("NumberReserved")
    p = p + txtNPeople.Text
    If rs.Fields("NumberOfReservations") > p Then
        rs.Fields("NumberReserved") = p
        rs.Update
    Else
        'Validation check to check if there are any vacancies.
        'Message to notify the user that there are no vacancies that are available.
        MsgBox "Sorry, there are no vacancies for this package. All the slots have been booked. Please try
another package."
        Exit Sub
    End If
    rs.Close
'Opening the Package booking databse.
    rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic
'Validation check to make sure that the same ID cannot be used twice.

```

```

Dim IsNewID As Boolean
IsNewID = True
If Not rs.EOF Then
    Do While Not rs.EOF
        If txtBookingID.Text = rs.Fields("PackageBookingID") Then
            IsNewID = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID = False Then
    'Message to notify that the user if booking ID is already in use.
    MsgBox "This Booking ID is already in use. Please try again."
    txtBookingID.Text = ""
    Exit Sub
End If

'Storage of Booking information into the database.
rs.AddNew
    rs.Fields("PackageBookingID") = txtBookingID.Text
    rs.Fields("Customer_name") = txtCustomerName.Text
    rs.Fields("Customer_address") = txtAddress.Text
    rs.Fields("Package_type") = cboPackageType.Text
    rs.Fields("Journey_Date") = lblJourneyDate.Caption
    rs.Fields("Contact") = txtContactNumber.Text
    rs.Fields("Cost_per_person") = txtCostPerPerson.Text
    rs.Fields("Package_name") = lstPackageName.Text
    rs.Fields("Number_of_people") = txtNPeople.Text
    rs.Fields("Net_total") = txtNetTotal.Text
    rs.Fields("Date_of_booking") = lblDate.Caption
    rs.Fields("PackageID") = txtPackageID.Text
    rs.Fields("Accommodation_type") = lblAccommodation.Caption
    rs.Fields("Journey_date") = lblJourneyDate.Caption
    rs.Fields("Starting_time") = lblStartingTime.Caption
    rs.Fields("Arrival_time") = lblArrivalTime.Caption
    rs.Fields("Duration/days") = lblDuration.Caption
    rs.Fields("Itinerary") = lblItinerary.Caption
    rs.Fields("Package_details") = lblPackageDetails.Caption
rs.Update
'Message shown to notify the user that storage was sucessful.
MsgBox "Booking has been made successfully."
rs.Close

'The fields are emptied after the storage of a record.
txtBookingID.Text = ""

```

```
txtCustomerName.Text = ""  
txtAddress.Text = ""  
cboPackageType.Text = ""  
txtContactNumber.Text = ""  
txtCostPerPerson.Text = ""  
txtNPeople.Text = ""  
txtNetTotal.Text = ""  
txtPackageID = ""  
lblItinerary.Caption = ""  
lblPackageDetails = ""  
lblAccommodation.Caption = ""  
lblJourneyDate.Caption = ""  
lblStartingTime.Caption = ""  
lblArrivalTime.Caption = ""  
lblDuration.Caption = ""  
lstPackageName.Clear  
lstPackageName.Refresh
```

```
Call Form_Load  
End Sub
```

```
'Cancel button click event.  
Private Sub cmdCancel_Click()  
'Unloading the form.  
Unload Me  
End Sub
```

```
'Refresh button click event.  
Private Sub cmdRefresh_Click()  
'The fields are emptied and the form is refreshed to allow the user to start anew.  
txtBookingID.Text = ""  
txtCustomerName.Text = ""  
txtAddress.Text = ""  
cboPackageType.Text = ""  
txtContactNumber.Text = ""  
txtCostPerPerson.Text = ""  
txtNPeople.Text = ""  
txtNetTotal.Text = ""  
txtPackageID = ""  
lblItinerary.Caption = ""  
lblPackageDetails = ""  
lblAccommodation.Caption = ""  
lblJourneyDate.Caption = ""  
lblStartingTime.Caption = ""  
lblArrivalTime.Caption = ""  
lblDuration.Caption = ""  
lstPackageName.Clear  
lstPackageName.Refresh
```

```

Call Form_Load
End Sub
'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
    Call dblink
Showing current date on the form.
    lblDate.Caption = Format(Now, "dd/mm/yyyy")
    Set rs = New Recordset
    rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic
    rs.MoveLast
    txtBookingID.Text = CInt(rs!PackageBookingID) + 1
    rs.Close

    Set rs = New Recordset
    rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic
'Loading Package Types from Package Entry Database.
'A check is made to make sure that each type is not loaded twice.
    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True
            For i = 0 To cboPackageType.ListCount - 1
                If cboPackageType.List(i) = rs.Fields("Package_type") Then
                    IsNewType = False
                End If
            Next
            If IsNewType = True Then
                cboPackageType.AddItem (rs.Fields("Package_type"))
            End If
            rs.MoveNext
        Loop
    End If

End Sub
>Contact Number keypress event.
Private Sub txtContactNumber_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
    If KeyAscii = 8 Then Exit Sub
    If Not IsNumeric(Chr(KeyAscii)) Then
        MsgBox "Please input numbers only"
        Exit Sub
    End If
End Sub
'Cost per person keypress event.
Private Sub txtCostPerPerson_KeyPress(KeyAscii As Integer)

```

```
'Character type check to make sure only numerical data is entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only"  
    Exit Sub  
End If  
End Sub
```

```
'Net total keypress event.
```

```
Private Sub txtNetTotal_KeyPress(KeyAscii As Integer)  
'Character type check to make sure only numerical data is entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only"  
    Exit Sub  
End If  
End Sub
```

```
'Number of people keypress event.
```

```
Private Sub txtNPeople_KeyPress(KeyAscii As Integer)  
'Character type check to make sure only numerical data is entered.  
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only"  
    Exit Sub  
End If  
End Sub
```

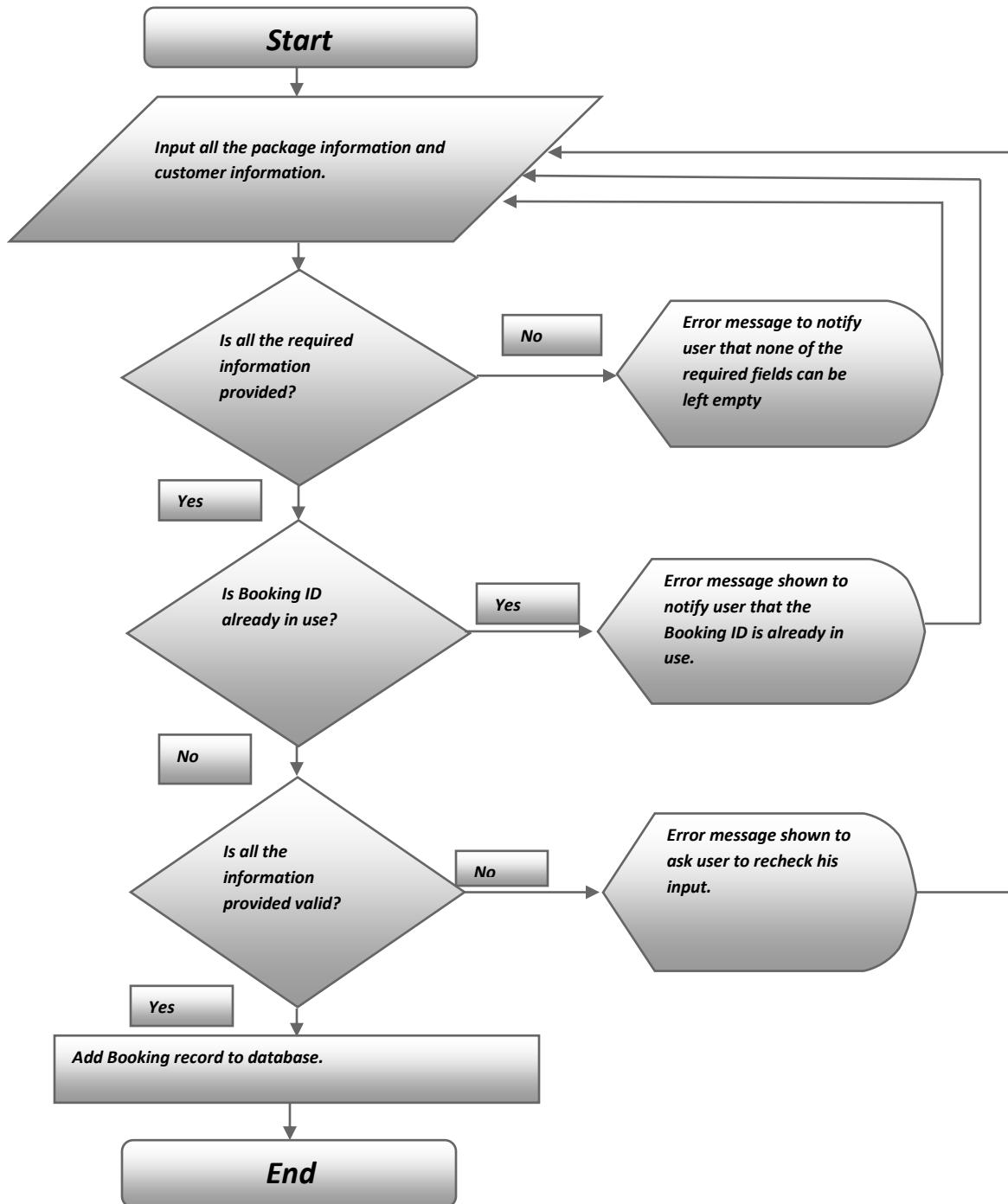
```
'Number of people keyup event.
```

```
Private Sub txtNPeople_KeyUp(KeyCode As Integer, Shift As Integer)  
'The bar moves to booking ID textbox when enter is pressed after the number of people textbox is filled.  
If KeyCode = vbKeyReturn Then  
    txtBookingID.SetFocus  
End If  
End Sub
```

```
'Number of people lostfocus event.
```

```
Private Sub txtNPeople_LostFocus()  
'Calculation of net total by multiplying Number of people with Cost per person.  
    txtNetTotal.Text = txtNPeople.Text * txtCostPerPerson.Text  
End Sub
```

Adding information to the form:



7. Package Booking Update form

The screenshot below shows the actual layout of the Package Booking Update form.

The screenshot below shows the actual coding of the Package Booking Update form

```

' Global declaration of a string data and an integer data.
Dim BID As String
Dim OldNumberPeople As Integer

'Package name combobox click event.
Private Sub cboPackageName_Click()
    Set rs = New Recordset
    'The fields Number of people and Net total are emptied if package name is changed in case they need to be updated as well.
    txtNPeople.Text = ""
    txtNetTotal.Text = ""

    'Loading cost per person from Package entry database.
    rs.Open "select * from Package_entry where Package_name = '" & cboPackageName.Text & "'", con, adOpenDynamic, adLockOptimistic
    txtPackageID.Text = rs.Fields("PackageID")
    txtCostPerPerson.Text = rs.Fields("Cost_per_person")
    lblVacancies.Caption = rs.Fields("NumberOfReservations") - rs.Fields("NumberReserved")
    lblItinerary.Caption = rs.Fields("Itinerary")
    lblPackageDetails.Caption = rs.Fields("Package_details")
    lblJourneyDate.Caption = rs.Fields("Journey_date")
    lblArrivalTime.Caption = rs.Fields("Arrival_time")
    lblStartingTime.Caption = rs.Fields("Starting_time")
    lblAccommodation.Caption = rs.Fields("Accommodation_type")
    lblDuration.Caption = rs.Fields("Duration/days")

End Sub

'Package type combobox click event.
Private Sub cboPackageType_Click()
    'The Package name combobox is cleared when a new package type is selected.
    cboPackageName.Clear
    cboPackageName.Text = ""

    txtCostPerPerson.Text = ""
    txtNPeople.Text = ""
    txtNetTotal.Text = ""
    txtPackageID = ""
    lblItinerary.Caption = ""
    lblPackageDetails = ""
    lblAccommodation.Caption = ""
End Sub

```

The coding of the Package Booking Update form is shown below:

```
'Global declaration of a string data and an integer data.  
Dim BID As String  
Dim OldNumberPeople As Integer  
  
'Package name combobox click event.  
Private Sub cboPackageName_Click()  
    Set rs = New Recordset  
'The fields Number of people and Net total are emptied if package name is changed in case they need to  
be updated as well.  
    txtNPeople.Text = ""  
    txtNetTotal.Text = ""  
'Loading cost per person from Package Entry database.  
    rs.Open "select * from Package_entry where Package_name = '" & cboPackageName.Text & "'", con,  
adOpenDynamic, adLockOptimistic  
    txtPackageID.Text = rs.Fields("PackageID")  
    txtCostPerPerson.Text = rs.Fields("Cost_per_person")  
    lblVacancies.Caption = rs.Fields("NumberOfReservations") - rs.Fields("NumberReserved")  
    lblItinerary.Caption = rs.Fields("Itinerary")  
    lblPackageDetails.Caption = rs.Fields("Package_details")  
    lblJourneyDate.Caption = rs.Fields("Journey_date")  
    lblArrivalTime.Caption = rs.Fields("Arrival_time")  
    lblStartingTime.Caption = rs.Fields("Starting_time")  
    lblAccommodation.Caption = rs.Fields("Accommodation_type")  
    lblDuration.Caption = rs.Fields("Duration/days")  
  
'Package type combobox click event.  
Private Sub cboPackageType_Click()  
'The Package name combobox is cleared when a new package type is selected.  
    cboPackageName.Clear  
    cboPackageName.Text = ""  
  
    txtCostPerPerson.Text = ""  
    txtNPeople.Text = ""  
    txtNetTotal.Text = ""  
    txtPackageID = ""  
    lblItinerary.Caption = ""  
    lblPackageDetails = ""  
    lblAccommodation.Caption = ""  
    lblJourneyDate.Caption = ""  
    lblStartingTime.Caption = ""  
    lblArrivalTime.Caption = ""  
    lblDuration.Caption = ""  
    lblItinerary.Caption = ""  
    lblPackageDetails.Caption = ""
```

'The Package names under the selected Package type are loaded.

```
Set rs = New Recordset
rs.Open "select * from Package_entry where Package_type=''" & cboPackageType.Text & "", con,
adOpenDynamic, adLockOptimistic
Do While Not rs.EOF
    cboPackageName.AddItem (rs.Fields("Package_name"))
    rs.MoveNext
Loop
End Sub
```

'Cancel button click event.

```
Private Sub cmdCancel_Click()
'Unloading the form.
Unload Me
End Sub
```

'Delete button click event.

```
Private Sub cmdDelete_Click()
```

'Validation check to make sure none of the fields are left empty.

```
If txtBookingID.Text = Empty Or lblName.Caption = Empty Or txtAddress.Text = Empty Or
cboPackageType.Text = Empty Or txtContactNumber.Text = Empty Or txtCostPerPerson.Text = Empty Or
cboPackageName.Text = Empty Or txtNPeople.Text = Empty Or txtNetTotal = Empty Or
txtPackageID.Text = Empty Then
```

'Message shown to notify the user that none of the fields can be left empty.

```
MsgBox "None of the fields can be left empty."
```

```
Exit Sub
```

```
End If
```

'The selected record is deleted.

```
rs.Delete
```

'Message shown to notify user that the record has been deleted successfully.

```
MsgBox "The Booking has been deleted."
```

```
rs.Close
```

'The reservation data is updated in Package Entry database after deletion of a booking.

```
rs.Open "select * from Package_entry where Package_name ='" & cboPackageName.Text & "", con,
adOpenDynamic, adLockOptimistic
```

```
Dim NReserve As Integer
```

```
NReserve = rs.Fields("NumberReserved")
```

```
NReserve = NReserve - txtNPeople.Text
```

```
rs.Fields("NumberReserved") = NReserve
```

```
rs.Update
```

'The fields are emptied after deletion of the record.

```
cboPackageName.Text = ""
```

```
txtBookingID.Text = ""
```

```
lblName.Caption = ""
```

```
txtAddress.Text = ""
```

```
cboPackageType.Text = ""
```

```
txtContactNumber.Text = ""
```

```
txtCostPerPerson.Text = ""
```

```
txtNPeople.Text = ""
```

```
txtNetTotal.Text = ""  
txtPackageID = ""  
lblItinerary.Caption = ""  
lblPackageDetails = ""  
lblAccommodation.Caption = ""  
lblJourneyDate.Caption = ""  
lblStartingTime.Caption = ""  
lblArrivalTime.Caption = ""  
lblDuration.Caption = ""  
lblBookingDate.Caption = ""
```

[Call Form_Load](#)

[End Sub](#)

'Refresh button click event.

```
Private Sub cmdRefresh_Click()
```

'The fields are cleared and the form is refreshed on the click of a button.

```
cboPackageName.Text = ""  
txtBookingID.Text = ""  
lblName.Caption = ""  
txtAddress.Text = ""  
cboPackageType.Text = ""  
txtContactNumber.Text = ""  
txtCostPerPerson.Text = ""  
txtNPeople.Text = ""  
txtNetTotal.Text = ""  
txtPackageID = ""  
lblItinerary.Caption = ""  
lblPackageDetails = ""  
lblAccommodation.Caption = ""  
lblJourneyDate.Caption = ""  
lblStartingTime.Caption = ""  
lblArrivalTime.Caption = ""  
lblDuration.Caption = ""  
lblItinerary.Caption = ""  
lblPackageDetails.Caption = ""
```

```
cboPackageType.Clear  
cboPackageName.Clear
```

[Call Form_Load](#)

[End Sub](#)

'Search button click event.

```
Private Sub cmdSearch_Click()
```

'Making sure Booking ID is provided.

```
If txtBookingID.Text = "" Then
```

```
    MsgBox "Please provide Booking ID and try again."
    Exit Sub
End If

'The Package booking database is searched for a particular Booking ID.
Set rs = New Recordset
rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic
Dim found As Boolean
found = False
rs.MoveFirst
Do While Not rs.EOF And Not found
    If rs.Fields("PackageBookingID") = txtBookingID.Text Then
        found = True
    End If
    rs.MoveNext
Loop

If found = False Then
    'Message shown to notify User if Booking ID not found.
    MsgBox "There is no record for this Booking ID."
    Exit Sub
End If
rs.Close

'If booking ID found, the information the record containing this Booking ID holds is loaded.
rs.Open "select * from Package_booking where PackageBookingID ='" & txtBookingID.Text & "'", con,
adOpenDynamic, adLockOptimistic
cboPackageType.Text = rs.Fields("Package_type")
txtPackageID.Text = rs.Fields("PackageID")
lblName.Caption = rs.Fields("Customer_name")
txtAddress.Text = rs.Fields("Customer_address")
lblJourneyDate.Caption = rs.Fields("Journey_date")
txtContactNumber.Text = rs.Fields("Contact")
txtCostPerPerson.Text = rs.Fields("Cost_per_person")
cboPackageName.Text = rs.Fields("Package_name")
txtNPeople.Text = rs.Fields("Number_of_people")
txtNetTotal.Text = rs.Fields("Net_total")
lblBookingDate.Caption = rs.Fields("Date_of_booking")
lblItinerary.Caption = rs.Fields("Itinerary")
lblPackageDetails.Caption = rs.Fields("Package_details")
lblAccommodation.Caption = rs.Fields("Accommodation_type")
lblJourneyDate.Caption = rs.Fields("Journey_date")
lblArrivalTime.Caption = rs.Fields("Arrival_time")
lblStartingTime.Caption = rs.Fields("Starting_time")
lblDuration.Caption = rs.Fields("Duration/days")
lblItinerary.Caption = rs.Fields("Itinerary")
lblPackageDetails.Caption = rs.Fields("Package_details")
```

```
rs.Update

'Values are assigned to the variables BID and OldNumberPeople.
BID = txtBookingID.Text
OldNumberPeople = txtNPeople.Text
End Sub

'Update button click event.
Private Sub cmdUpdate_Click()

'Making sure a record is opened to cause alterations.
If txtBookingID.Text = "" Then
    MsgBox "Please provide Booking ID, press search and try again."
    Exit Sub
End If

'Validation check to make sure Booking ID is not changed.
If txtBookingID.Text <> BID Then
    MsgBox "Booking ID cannot be changed."
    txtBookingID = BID
    Exit Sub
End If

'Presence check to make sure all the required fields are filled by the customer.
If cboPackageType.Text = "" Or cboPackageName.Text = "" Or txtPackageID.Text = "" Or
txtCostPerPerson.Text = "" Or txtNPeople = "" Or txtAddress.Text = "" Or txtContactNumber.Text = ""
Then
    'Message to notify the user that all of the information need to be provided.
    MsgBox "All the Package information and customer information need to provided in order to
proceed with the booking."
    Exit Sub
End If

'Length check to make sure contact number is of 11 digits.
If Len(txtContactNumber.Text) <> 11 Then
    MsgBox "Number of digits used for contact number is invalid."
    Exit Sub
End If

Set rs = New Recordset
'Validation check to ensure the updated reservations doesnot exceed the reservation limit.
rs.Open "select * from Package_entry where Package_name ='" & cboPackageName.Text & "'", con,
adOpenDynamic, adLockOptimistic
Dim NewNumberReserved As Integer
NewNumberReserved = rs.Fields("NumberReserved")

NewNumberReserved = NewNumberReserved - OldNumberPeople + txtNPeople.Text
```

```
If NewNumberReserved > rs.Fields("NumberOfReservations") Then
    MsgBox "There are not enough vacancies to make this reservation."
    Exit Sub
End If

If NewNumberReserved < rs.Fields("NumberOfReservations") Then
    rs.Fields("NumberReserved") = NewNumberReserved
    rs.Update
End If

rs.Close

'Validation check to ensure none of the fields are left empty.
If txtBookingID.Text = Empty Or lblName.Caption = Empty _
Or txtAddress.Text = Empty Or cboPackageType.Text = Empty _
Or txtContactNumber.Text = Empty Or txtCostPerPerson.Text = Empty _
Or cboPackageName.Text = Empty Or txtNPeople.Text = Empty _
Or txtNetTotal = Empty Or txtPackageID.Text = Empty Then
    'Message shown to notify user that none of the fields can be kept empty.
    MsgBox "None of the fields can be left empty."
    Exit Sub
End If

'The fields are added to the database.
rs.Open "select * from Package_booking", con, adOpenDynamic, adLockOptimistic
rs.Fields("Customer_name") = lblName.Caption
rs.Fields("Customer_address") = txtAddress.Text
rs.Fields("Package_type") = cboPackageType.Text
rs.Fields("Journey_Date") = lblJourneyDate.Caption
rs.Fields("Contact") = txtContactNumber.Text
rs.Fields("Cost_per_person") = txtCostPerPerson.Text
rs.Fields("Package_name") = cboPackageName.Text
rs.Fields("Number_of_people") = txtNPeople.Text
rs.Fields("Net_total") = txtNetTotal.Text
rs.Fields("Date_of_booking") = lblDate.Caption
rs.Fields("Accommodation_type") = lblAccommodation.Caption
rs.Fields("Journey_date") = lblJourneyDate.Caption
rs.Fields("Starting_time") = lblStartingTime.Caption
rs.Fields("Arrival_time") = lblArrivalTime.Caption
rs.Fields("Duration/days") = lblDuration.Caption
rs.Update

MsgBox "Booking information has been Updated successfully."

'The fields on the form are emptied.
cboPackageName.Text = ""
txtBookingID.Text = ""
lblName.Caption = ""
txtAddress.Text = ""
cboPackageType.Text = ""
```

```

txtContactNumber.Text = ""
txtCostPerPerson.Text = ""
txtNPeople.Text = ""
txtNetTotal.Text = ""
txtPackageID = ""
lblItinerary.Caption = ""
lblPackageDetails = ""
lblAccommodation.Caption = ""
lblJourneyDate.Caption = ""
lblStartingTime.Caption = ""
lblArrivalTime.Caption = ""
lblDuration.Caption = ""
lblItinerary.Caption = ""
lblPackageDetails.Caption = ""
lblBookingDate.Caption = ""

Call Form_Load

End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
Call dblink
lblDate.Caption = Format(Now, "dd/mm/yyyy")
'Loading Package types from database.
Set rs = New Recordset
rs.Open "select * from Package_entry", con, adOpenDynamic, adLockOptimistic

If rs.EOF = False Then
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboPackageType.ListCount - 1
            If cboPackageType.List(i) = rs.Fields("Package_type") Then
                IsNewType = False
            End If
        Next
        If IsNewType = True Then
            cboPackageType.AddItem (rs.Fields("Package_type"))
        End If
        rs.MoveNext
    Loop
End If

End Sub

'Booking ID keyup event.

```

```
Private Sub txtBookingID_KeyUp(KeyCode As Integer, Shift As Integer)
'Shifts to Search button when "Enter" pressed.
If KeyCode = vbKeyReturn Then
    cmdSearch.SetFocus
End If
End Sub
```

```
'Contact number keypress event.
Private Sub txtContactNumber_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```

```
'Cost per person keypress event.
Private Sub txtCostPerPerson_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```

```
'Net total keypress event.
Private Sub txtNetTotal_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
```

```
'Number of people keypress event.
Private Sub txtNPeople_KeyPress(KeyAscii As Integer)
'Character type check for numerical input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```

```
'Number of people Keyup event.
Private Sub txtNPeople_KeyUp(KeyCode As Integer, Shift As Integer)
'Character type check for numerical input.
If KeyCode = vbKeyReturn Then
    cmdUpdate.SetFocus
End If
```

End Sub

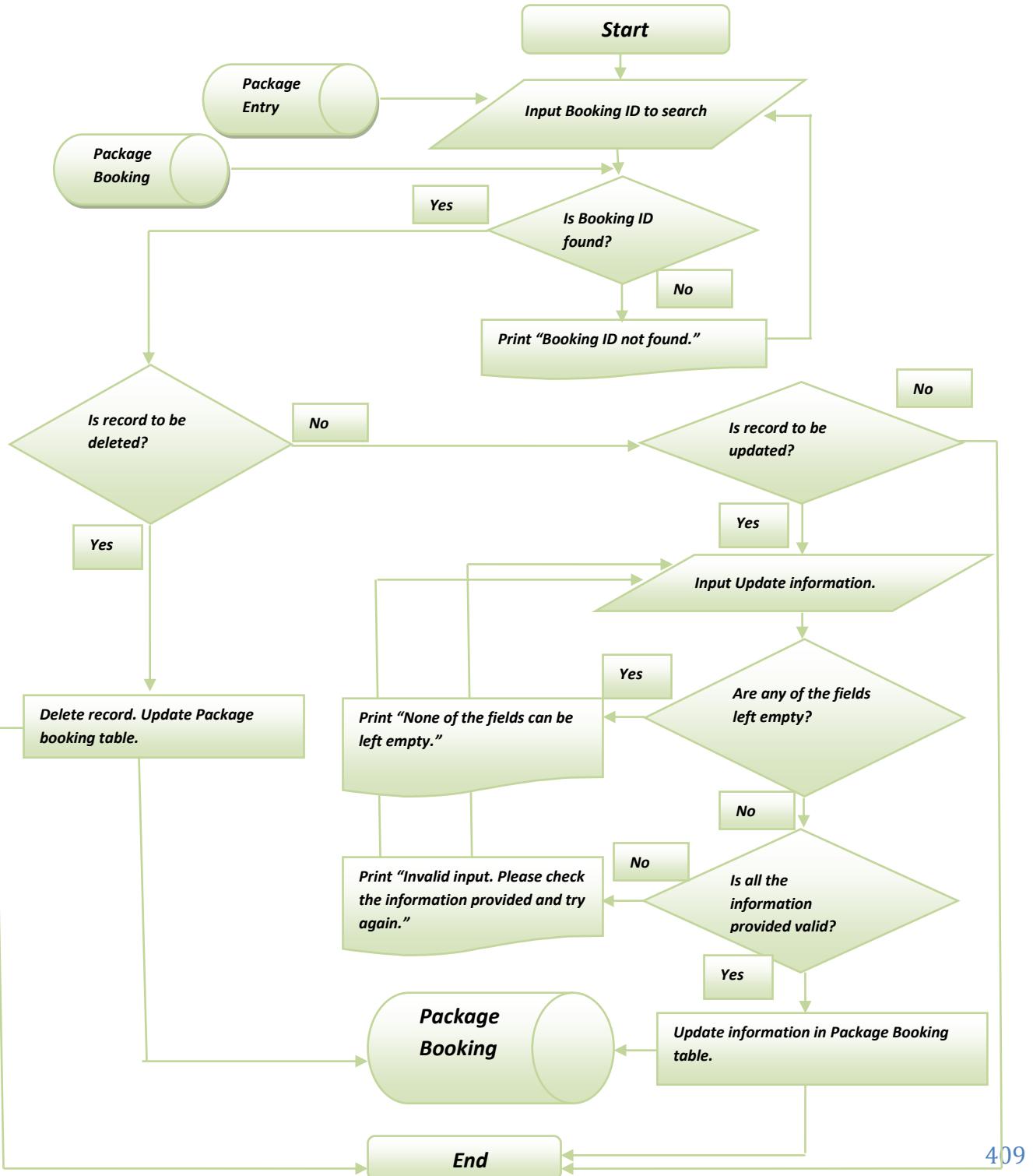
'Number of people Lostfocus event.

Private Sub txtNPeople_LostFocus()

'Calculation of Net total.

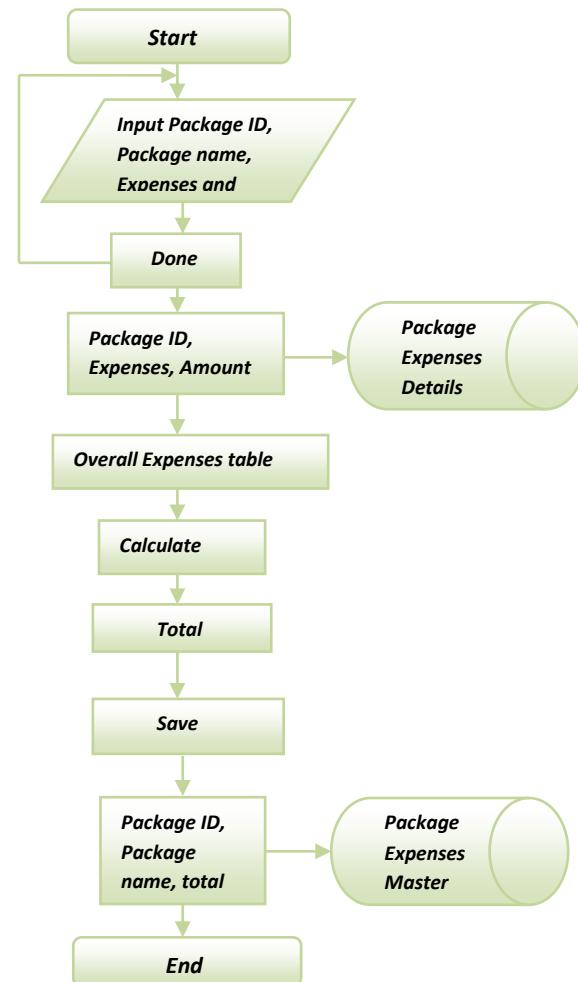
txtNetTotal.Text = txtNPeople.Text * txtCostPerPerson.Text

End Sub



8. Package Expenses Entry form

The screenshot below shows the actual layout of the Package Expenses Entry form



The screenshot below shows the actual coding of the Package Expenses Entry form

```

'Declaration of a global variable.
Dim r As Integer
'Calculation button click event.
Private Sub cmdCalculate_Click()
'Calculation of total expenses
    Dim total As Double
    total = 0
    For i = 1 To r
        total = total + Val(flxgrdExpensesDetails.TextMatrix(i, 2))
    Next i

    txtTotal.Text = total
End Sub
'Done button click event.
Private Sub cmdDone_Click()
'Presence check to make sure all the information is provided.
    If txtPackageID.Text = "" Or txtPackageName.Text = "" Or cboExpenses.Text = "" Or txtAmount.Text = "" Then
        MsgBox "None of the fields in the Package Expenses information section can be left empty. "
        Exit Sub
    End If

    'Adding expenses information into the flexgrid.
    r = r + 1
    flxgrdExpensesDetails.TextMatrix(r, 0) = txtPackageID.Text
    flxgrdExpensesDetails.TextMatrix(r, 1) = cboExpenses.Text
    flxgrdExpensesDetails.TextMatrix(r, 2) = txtAmount.Text

    flxgrdExpensesDetails.Rows = flxgrdExpensesDetails.Rows + 1
'Adding expenses information into the Package expenses details database table.
    Set rs = New Recordset
    rs.Open "select * from Package_expenses_details", con, adOpenDynamic, adLockOptimistic
    rs.AddNew
        rs.Fields("PackageID") = txtPackageID.Text
        rs.Fields("Expenses") = cboExpenses.Text
        rs.Fields("Amount") = Val(txtAmount.Text)
        rs.Update
    rs.MoveNext
End Sub
  
```

```

'Declaration of a global variable.
Dim r As Integer

'Calcualte button click event.
Private Sub cmdCalculate_Click()
'Calculation of total expenses
    Dim total As Double
    total = 0
    For i = 1 To r
        total = total + Val(flxgrdExpensesDetails.TextMatrix(i, 2))
    Next i

    txtTotal.Text = total
End Sub

'Done button click event.
Private Sub cmdDone_Click()
'Presence check to make sure all the information is provided.
    If txtPackageID.Text = "" Or txtPackageName.Text = "" Or cboExpenses.Text = "" Or txtAmount.Text = "" Then
        MsgBox "None of the fields in the Package Expenses information section can be left empty."
        Exit Sub
    End If

'Adding expenses information into the flexgrid.
    r = r + 1
    flxgrdExpensesDetails.TextMatrix(r, 0) = txtPackageID.Text
    flxgrdExpensesDetails.TextMatrix(r, 1) = cboExpenses.Text
    flxgrdExpensesDetails.TextMatrix(r, 2) = txtAmount.Text

    flxgrdExpensesDetails.Rows = flxgrdExpensesDetails.Rows + 1
'Adding expenses information into the Package expenses details database table.
    Set rs = New Recordset
    rs.Open "select * from Package_expenses_details", con, adOpenDynamic, adLockOptimistic
    rs.AddNew
        rs.Fields("PackageID") = txtPackageID.Text
        rs.Fields("Expenses") = cboExpenses.Text
        rs.Fields("Amount") = Val(txtAmount.Text)
    rs.Update
    rs.MoveNext
'These fields are cleared in case they need to be changed.
    cboExpenses.Text = ""
    txtAmount.Text = ""
End Sub

'Cancel button click event.
Private Sub cmdCancel_Click()
'Unloading the form.
    Unload Me
End Sub

```

```
'Refresh button click event.  
Private Sub cmdRefresh_Click()  
'All the fields are emptied and refreshed in case the user wants to start anew.  
    txtPackageID.Text = ""  
    txtPackageName.Text = ""  
    cboExpenses.Text = ""  
    flxgrdExpensesDetails.Clear  
    flxgrdExpensesDetails.Rows = 2  
    Call Form_Load  
    txtTotal.Text = ""  
End Sub  
  
'Save button click event.  
Private Sub cmdSave_Click()  
'Storing final cost and expenses information in the Package expenses master database.  
    Set rs = New Recordset  
    rs.Open "select * from Package_expenses_master", con, adOpenDynamic, adLockOptimistic  
    rs.AddNew  
        rs.Fields("PackageID") = txtPackageID.Text  
        rs.Fields("Package_name") = txtPackageName.Text  
        rs.Fields("Total") = txtTotal.Text  
    rs.Update  
    MsgBox "The record has been saved."  
End Sub  
  
'Form load event.  
Private Sub Form_Load()  
'Calling the function to connect the form to the database.  
    Call dblink  
'Showing the current date on the form.  
    lblDate.Caption = Date  
'Assigning fields to the flexgrid columns.  
    flxgrdExpensesDetails.TextMatrix(0, 0) = "Package ID"  
    flxgrdExpensesDetails.TextMatrix(0, 1) = "Expenses"  
    flxgrdExpensesDetails.TextMatrix(0, 2) = "Amount"  
    r = 0  
  
End Sub  
  
'Amount keypress event.  
Private Sub txtAmount_KeyPress(KeyAscii As Integer)  
'Character type check to make sure only numerical data is entered.  
    If KeyAscii = 8 Then Exit Sub  
    If Not IsNumeric(Chr(KeyAscii)) Then  
        MsgBox "Please input numbers only."  
        Exit Sub  
    End If  
  
End Sub
```

'Amount textbox keyup event.

```
Private Sub txtAmount_KeyUp(KeyCode As Integer, Shift As Integer)
```

'Directing the cursor to the Done button after information is inserted in the amount textbox.

```
If KeyCode = vbKeyReturn Then
```

```
    cmdDone.SetFocus
```

```
End If
```

```
End Sub
```

'Total keypress event.

```
Private Sub txtTotal_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure only numerical data is entered.

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

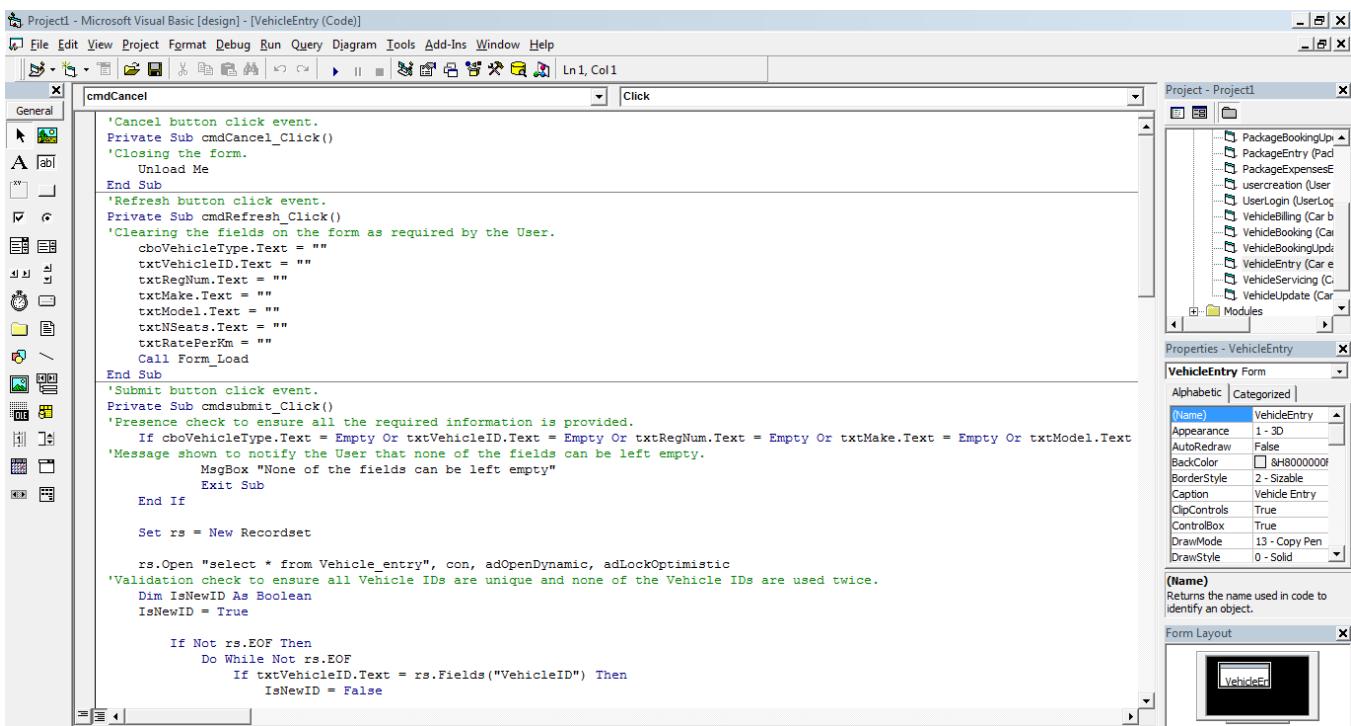
9. Vehicle Entry form

The screenshot shows a Windows application window titled "Vehicle Entry". The window has a standard title bar with minimize, maximize, and close buttons. The main area is labeled "Vehicle Entry" in a large, bold, italicized font. In the top right corner, there is a "Date" label followed by the date "23-03-2015". The form is organized into several groups of input fields:

- Vehicle type:** A dropdown menu.
- Vehicle ID:** A text input field.
- Registration number:** A text input field.
- Make:** A text input field.
- Model:** A text input field.
- Number of seats:** A text input field.
- Rate per kilometer:** A text input field.

At the bottom of the form, there are three buttons arranged horizontally: "Submit", "Refresh", and "Cancel".

The screenshot below shows the actual layout of the Vehicle Entry form



The screenshot below shows the actual coding of the Vehicle Entry form

The coding of the Vehicle Entry form is shown below:

```
'Cancel button click event.
Private Sub cmdCancel_Click()
'Closing the form.
Unload Me
End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
'Clearing the fields on the form as required by the User.
cboVehicleType.Text = ""
txtVehicleID.Text = ""
txtRegNum.Text = ""
txtMake.Text = ""
txtModel.Text = ""
txtNSeats.Text = ""
txtRatePerKm = ""
Call Form_Load
End Sub

'Submit button click event.
Private Sub cmdsubmit_Click()
'Presence check to ensure all the required information is provided.
If cboVehicleType.Text = "" Or txtVehicleID.Text = "" Or txtRegNum.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Then
    MsgBox "None of the fields can be left empty."
    Exit Sub
End If

Set rs = New Recordset

rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Validation check to ensure all Vehicle IDs are unique and none of the Vehicle IDs are used twice.
Dim IsNewID As Boolean
IsNewID = True

If Not rs.EOF Then
    Do While Not rs.EOF
        If txtVehicleID.Text = rs.Fields("VehicleID") Then
            IsNewID = False
        End If
    Loop
End If

If IsNewID = False Then
    MsgBox "Vehicle ID already exists. Please enter a unique vehicle ID."
    Exit Sub
End If

rs.AddNew
rs("VehicleID").Value = txtVehicleID.Text
rs("VehicleType").Value = cboVehicleType.Text
rs("RegNum").Value = txtRegNum.Text
rs("Make").Value = txtMake.Text
rs("Model").Value = txtModel.Text
rs("NSeats").Value = txtNSeats.Text
rs("RatePerKm").Value = txtRatePerKm
rs.Update
MsgBox "Vehicle entry successful."
End Sub
```

```
If cboVehicleType.Text = Empty Or txtVehicleID.Text = Empty Or txtRegNum.Text = Empty Or
txtMake.Text = Empty Or txtModel.Text = Empty Or txtNSeats.Text = Empty Or txtRatePerKm.Text =
Empty Then
'Message shown to notify the User that none of the fields can be left empty.
    MsgBox "None of the fields can be left empty"
    Exit Sub
End If

Set rs = New Recordset

rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Validation check to ensure all Vehicle IDs are unique and none of the Vehicle IDs are used twice.
Dim IsNewID As Boolean
IsNewID = True

If Not rs.EOF Then
    Do While Not rs.EOF
        If txtVehicleID.Text = rs.Fields("VehicleID") Then
            IsNewID = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID = False Then
'Message shown if Vehicle ID is already in use.
    MsgBox "This Vehicle ID is already in use. Please try again."
    txtVehicleID.Text = ""
    Exit Sub
End If
rs.MoveFirst
Dim IsNewID2 As Boolean
IsNewID2 = True

If Not rs.EOF Then

    Do While Not rs.EOF
        If txtRegNum.Text = rs.Fields("RegistrationNo") Then
            IsNewID2 = False
            Exit Do
        End If
        rs.MoveNext
    Loop
End If

If IsNewID2 = False Then
```

```

    MsgBox "This Registration number is already in use. Please try again."
    txtRegNum.Text = ""
    Exit Sub
End If

'Adding new Vehicle information into the database.
rs.AddNew
    rs.Fields("Vehicle_type") = cboVehicleType.Text
    rs.Fields("VehicleID") = txtVehicleID.Text
    rs.Fields("RegistrationNo") = txtRegNum.Text
    rs.Fields("Make") = txtMake.Text
    rs.Fields("Model") = txtModel.Text
    rs.Fields("NSeats") = txtNSeats.Text
    rs.Fields("Rate_per_km") = txtRatePerKm.Text
    rs.Fields("Date_of_entry") = lblDate.Caption
rs.Update

'Message shown to notify the User that new vehicle information is added successfully.
MsgBox "Vehicle information has been added successfully"
'Clearing the fields to ready the form for new transaction.
cboVehicleType.Text = ""
txtVehicleID.Text = ""
txtRegNum.Text = ""
txtMake.Text = ""
txtModel.Text = ""
txtNSeats.Text = ""
txtRatePerKm = ""

Call Form_Load
End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
    Call dblink
'Showing the current date on the form.
    lblDate.Caption = Format(Now, "dd/mm/yyyy")
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
    rs.MoveLast
    txtVehicleID.Text = CInt(rs!VehicleID) + 1

End Sub

'Number of seats Keypress event.
Private Sub txtNSeats_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
    If KeyAscii = 8 Then Exit Sub
    If Not IsNumeric(Chr(KeyAscii)) Then

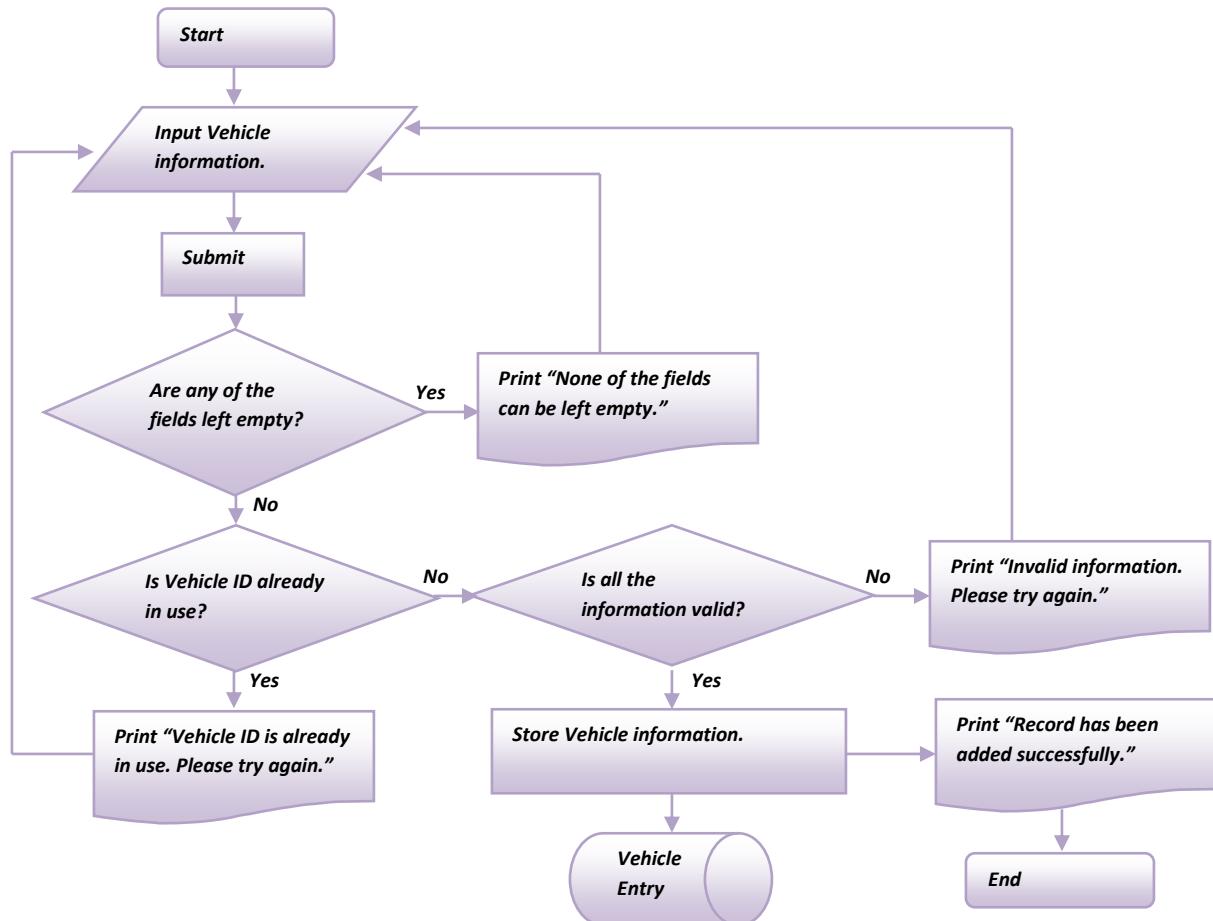
```

```

    MsgBox "Please input numbers only."
    Exit Sub
End If

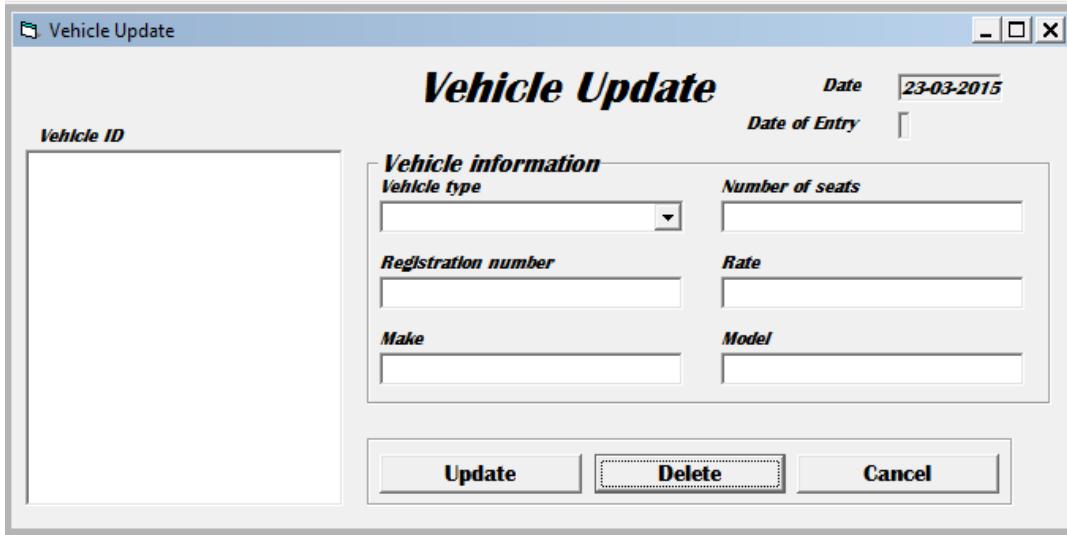
End Sub
'Rate per kilometer Keypress event.
Private Sub txtRatePerKm_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub

```



10. Vehicle Update form

The screenshot below shows the actual layout of the Vehicle Update form.



The screenshot below shows the actual coding of the Vehicle Update form.

```

'Cancel button click event.
Private Sub cmdCancel_Click()
    'Closing the form.
    Unload Me
End Sub

'Delete button click event.
Private Sub cmdDelete_Click()
    'Presence check to make sure the customer provides all the necessary information.
    If lstVehicleID.Text = "" Or cboVehicleType.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or txtRegNum.Text = "" Or txtRate.Text = "" Then
        MsgBox "All the required information not provided. Please fill up all the fields and try again."
        Exit Sub
    End If

    'Deletion of the record from the database.
    rs.Delete
    lstVehicleID.Refresh
    MsgBox "Record has been deleted"

    'The information of the deleted record is cleared from the form.
    cboVehicleType.Text = ""
    txtRegNum = ""
    txtMake.Text = ""
    txtModel.Text = ""
    txtNSeats.Text = ""
    txtRate.Text = ""
    lblEntryDate.Caption = ""
    lstVehicleID.Clear

    Call Form_Load

End Sub

'Update button click event.
Private Sub cmdUpdate_Click()
    'Presence check to make sure the customer provides all the necessary information.
    If lstVehicleID.Text = "" Or cboVehicleType.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or txtRegNum.Text = "" Or txtRate.Text = "" Then
        MsgBox "All the required information not provided. Please fill up all the fields and try again."
    End If

```

The coding of the Vehicle Update form is shown below.

```
'Cancel button click event.  
Private Sub cmdCancel_Click()  
'Closing the form.  
    Unload Me  
End Sub  
'Delete button click event.  
Private Sub cmdDelete_Click()  
  
'Presence check to make sure the customer provides all the necessary information.  
If lstVehicleID.Text = "" Or cboVehicleType.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or  
txtRegNum.Text = "" Or txtRate.Text = "" Or txtNSeats.Text = "" Then  
    MsgBox "All the required information not provided. Please fill up all the fields and try again."  
    Exit Sub  
End If  
  
'Deletion of the record from the database.  
rs.Delete  
lstVehicleID.Refresh  
MsgBox "Record has been deleted"  
  
'The information of the deleted record is cleared from the form.  
cboVehicleType.Text = ""  
txtRegNum = ""  
txtMake.Text = ""  
txtModel.Text = ""  
txtNSeats.Text = ""  
txtRate.Text = ""  
lblEntryDate.Caption = ""  
lstVehicleID.Clear  
  
Call Form_Load  
  
End Sub  
'Update button click event.  
Private Sub cmdUpdate_Click()  
'Presence check to make sure the customer provides all the necessary information.  
If lstVehicleID.Text = "" Or cboVehicleType.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or  
txtRegNum.Text = "" Or txtRate.Text = "" Or txtNSeats.Text = "" Then  
    MsgBox "All the required information not provided. Please fill up all the fields and try again."  
    Exit Sub  
End If  
  
'The Vehicle information in the database are updated.  
rs.Fields("Vehicle_type") = cboVehicleType.Text  
rs.Fields("RegistrationNo") = txtRegNum.Text
```

```

rs.Fields("Make") = txtMake.Text
rs.Fields("Model") = txtModel.Text
rs.Fields("NSeats") = txtNSeats.Text
rs.Fields("Rate_per_km") = txtRate.Text
rs.Update
'Message shown to notify the User that the Update was successful.
MsgBox "Vehicle information has been updated"
'Clearing the fields in the form to ready form for the next job.
cboVehicleType.Text = ""
txtRegNum = ""
txtMake.Text = ""
txtModel.Text = ""
txtNSeats.Text = ""
txtRate.Text = ""
lblEntryDate.Caption = ""
lstVehicleID.Clear

Call Form_Load
End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
Call dblink
'Showing the current date on the form.
lblDate.Caption = Format(Now, "dd/mm/yyyy")
'Loading the Vehicle ID information into the list of Vehicle IDs from the database.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
Do While Not rs.EOF
lstVehicleID.AddItem (rs.Fields("VehicleID"))
rs.MoveNext
Loop

End Sub

'Vehicle ID list click event.
Private Sub lstVehicleID_Click()
'Loading information for the particular Vehicle ID selected into the form from the database.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry where VehicleID = " & lstVehicleID.Text & "", con,
adOpenDynamic, adLockOptimistic

cboVehicleType.Text = rs.Fields("Vehicle_type")
txtRegNum.Text = rs.Fields("RegistrationNo")
txtMake.Text = rs.Fields("Make")
txtModel.Text = rs.Fields("Model")
txtNSeats.Text = rs.Fields("NSeats")
txtRate.Text = rs.Fields("Rate_per_km")

```

```

lblEntryDate.Caption = rs.Fields("Date_of_entry")
rs.Update
End Sub

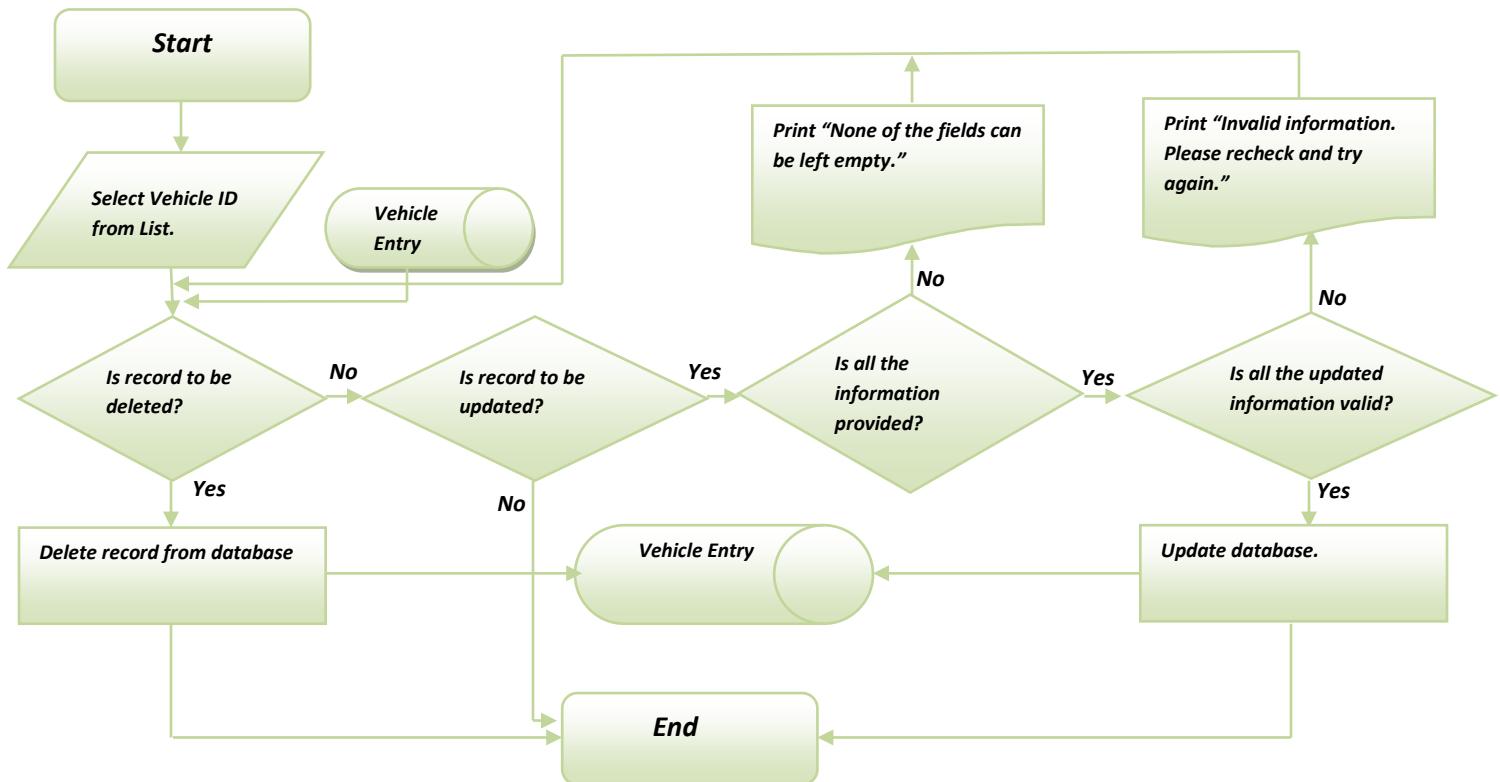
'Number of seats Keypress event.
Private Sub txtNSeats_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub

'Rate per kilometer keypress event.
Private Sub txtRate_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub

```



11. Vehicle Booking form

The screenshot shows a Windows application window titled "Vehicle Booking". At the top right, it displays the date "23 - 03 - 2015". The form is divided into two main sections: "Customer information" and "Vehicle information".

Customer information:

- Booking ID: [Text Box]
- Address: [Text Box]
- Customer Name: [Text Box]
- Contact: [Text Box]
- Journey date: [Text Box] containing "23-Mar-15" with a dropdown arrow.
- Drop off date: [Text Box] containing "23-Mar-15" with a dropdown arrow.

Vehicle information:

- Vehicle type: [ComboBox]
- Registration number: [Text Box]
- Make: [Text Box]
- Model: [Text Box]
- VehicleID: [Text Box]
- Rate per kilometer: [Text Box]
- Advance paid: [Text Box]
- Number of seats: [Text Box]
- Driver name: [Text Box]
- Starting kilometer: [Text Box]

At the bottom of the form are three buttons: "Confirm", "Refresh", and "Cancel".

The screenshot shows the Microsoft Visual Basic IDE with the code editor open for the "VehicleBooking" form. The code handles the "Click" event for the "cboVehicleType" combobox.

```

'Vehicle type combobox click event.
Private Sub cboVehicleType_Click()
    'The fields that depend on the type of the vehicle are cleared when the combobox information are changed to prevent database inform
    lstMake.Clear
    lstModel.Clear
    lstVehicleID.Clear
    txtRegNum.Text = ""
    'The Makes for the selected Vehicle type are loaded onto the Make list.
    'Each particular Make name is selected only once.
    Set rs = New Recordset
    rs.Open "select distinct * from Vehicle_entry where Vehicle_type='" & cboVehicleType.Text & "'", con, adOpenDynamic, adLockOptimistic
    'Each Make name is loaded onto the list only once.
    'Validation check ensure only the vehicles that are not booked are loaded onto the list.
    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True
            For i = 0 To lstMake.ListCount - 1
                If lstMake.List(i) = rs.Fields("Make") Then
                    IsNewType = False
                End If
            Next
            If IsNewType = True Then
                If rs.Fields("IsBooked?") = False Then
                    lstMake.AddItem (rs.Fields("Make"))
                End If
            End If
            rs.MoveNext
        Loop
    End If
End Sub
'Confirm button click event.
Private Sub cmdConfirm_Click()

```

The IDE also shows the project tree on the right, listing various modules and forms, and the properties window for the "VehicleBooking" form.

```

'Vehicle type combobox click event.
Private Sub cboVehicleType_Click()
'The fields that depend on the type of the vehicle are cleared when the combobox information are
changed to prevent database information storage override.
    lstMake.Clear
    lstModel.Clear
    lstVehicleID.Clear
    txtRegNum.Text = ""
'The Makes for the selected Vehicle type are loaded onto the Make list.
'Each particular Make name is selected only once.
    Set rs = New Recordset
    rs.Open "select distinct * from Vehicle_entry where Vehicle_type=" & cboVehicleType.Text & "", con,
adOpenDynamic, adLockOptimistic

'Each Make name is loaded onto the list only once.
'Validation check ensure only the vehicles that are not booked are loaded onto the list.
    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True
            For i = 0 To lstMake.ListCount - 1
                If lstMake.List(i) = rs.Fields("Make") Then
                    IsNewType = False
                End If
            Next
            If IsNewType = True Then
                If rs.Fields("IsBooked?") = False Then
                    lstMake.AddItem (rs.Fields("Make"))
                End If
            End If
            rs.MoveNext
        Loop
    End If

End Sub

'Comfirm button click event.
Private Sub cmbConfirm_Click()
'Presence check to make sure that none of the fields are left empty.
    If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = "" Or
cboVehicleType.Text = "" Or txtRegNum.Text = "" Or txtAdvancePaid.Text = "" Or txtDriverName.Text =
"" Or txtStartingKm.Text = "" Then
        MsgBox "Some of the required fields are not filled. Please fill up all of the fields and try again."
        Exit Sub
    End If

```

```
'Length check to make sure contact number is of 11 digits.  
If Len(txtContact.Text) <> 11 Then  
    MsgBox "Number of digits used for contact number is invalid."  
    Exit Sub  
End If  
  
Set rs = New Recordset  
  
rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic  
'Validation check to make sure the same Booking ID is not used twice.  
If Not rs.EOF Then  
    Dim IsNewID As Boolean  
    IsNewID = True  
    Do While Not rs.EOF  
        If txtBookingID.Text = rs.Fields("VehicleBookingID") Then  
            IsNewID = False  
            Exit Do  
        End If  
        rs.MoveNext  
    Loop  
End If  
'Message shown to notify the user that the Booking ID is in use if the same User ID is found in database.  
If IsNewID = False Then  
    MsgBox "This Booking ID is already in use. Please try again."  
    txtBookingID.Text = ""  
    Exit Sub  
End If  
'Storage of Vehicle Booking information into the database.  
rs.AddNew  
rs.Fields("VehicleBookingID") = txtBookingID.Text  
rs.Fields("Customer_name") = txtName.Text  
rs.Fields("Customer_address") = txtAddress.Text  
rs.Fields("Contact") = txtContact.Text  
rs.Fields("Journey_date") = dtpJourneyDate.Value  
rs.Fields("Drop_off_date") = dtpDropOffDate.Value  
rs.Fields("Vehicle_type") = cboVehicleType.Text  
rs.Fields("Registration_number") = txtRegNum.Text  
rs.Fields("Make") = lstMake.Text  
rs.Fields("Model") = lstModel.Text  
rs.Fields("VehicleID") = lstVehicleID.Text  
rs.Fields("Rate_per_kilometer") = lblRatePerKm.Caption  
rs.Fields("Number_of_seats") = lblNSeats.Caption  
rs.Fields("Advance_paid") = txtAdvancePaid.Text  
rs.Fields("Driver_name") = txtDriverName.Text  
rs.Fields("Date_of_booking") = lblDate.Caption  
rs.Fields("Starting_Km") = txtStartingKm.Text  
rs.Update
```

```
'Message shown notify the User that Booking was successful.  
MsgBox "Booking has been made successfully."  
rs.Close  
'The Vehicle is made unavailable to further bookings.  
rs.Open "select * from Vehicle_entry where VehicleID ='" & lstVehicleID.Text & "'", con,  
adOpenDynamic, adLockOptimistic  
rs.Fields("IsBooked?") = True  
rs.Update  
'The fields are all cleared and the form is refreshed after the booking is made.  
txtBookingID.Text = ""  
txtName.Text = ""  
txtContact.Text = ""  
txtAddress.Text = ""  
dtpJourneyDate.Value = Date  
dtpDropOffDate.Value = Date  
cboVehicleType.Text = ""  
txtRegNum.Text = ""  
lstMake.Clear  
lstModel.Clear  
lstVehicleID.Clear  
lblRatePerKm.Caption = ""  
lblINSeats.Caption = ""  
txtAdvancePaid = ""  
txtDriverName.Text = ""  
txtStartingKm.Text = ""  
  
Call Form_Load  
End Sub  
'Cancel button click event  
Private Sub cmdCancel_Click()  
'Unloading the form.  
Unload Me  
End Sub  
'Refresh button click event.  
Private Sub cmdRefresh_Click()  
'The fields are cleared at the click of the button as the User might require in case he/she needs to start anew.  
txtBookingID.Text = ""  
txtName.Text = ""  
txtContact.Text = ""  
txtAddress.Text = ""  
dtpJourneyDate.Value = Date  
dtpDropOffDate.Value = Date  
cboVehicleType.Text = ""  
txtRegNum.Text = ""  
lstMake.Clear
```

```

IstModel.Clear
IstVehicleID.Clear
lblRatePerKm.Caption = ""
lblNSeats.Caption = ""
txtAdvancePaid = ""
txtDriverName.Text = ""
txtStartingKm.Text = ""

Call Form_Load

End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
    Call dblink
'Shows current date.
    lblDate.Caption = Format(Now, "dd / mm / yyyy")
'The date pickers are assigned current dates as default.
    dtpJourneyDate.Value = Date
    dtpDropOffDate.Value = Date
'Loading the Vehicle type information onto the Vehicle type combobox from Vehicle entry database.

    Set rs = New Recordset
    rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic
        rs.MoveLast
        txtBookingID.Text = CInt(rs!VehicleBookingID) + 1
    rs.Close
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Validity check to make sure each type is loaded only once.
    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True
            For i = 0 To cboVehicleType.ListCount - 1
                If cboVehicleType.List(i) = rs.Fields("Vehicle_type") Then
                    IsNewType = False
                End If
            Next
            If IsNewType = True Then
'Validity check to make sure any booked vehicles are not loaded.
                If rs.Fields("IsBooked?") = False Then
                    cboVehicleType.AddItem (rs.Fields("Vehicle_type"))
                End If
            End If
        End While
    End If

```

```

    rs.MoveNext
    Loop
End If

End Sub
'Make list click event.
Private Sub lstMake_Click()
'The information dependent on the Make of the vehicle are cleared to make sure no override occurs
when storing data into the database.
    lstModel.Clear
    lstVehicleID.Clear
    txtRegNum.Text = ""
'The particular Models are loaded for the selected Make.
    Set rs = New Recordset
    rs.Open "select distinct * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "' and
Make ='" & lstMake.Text & "'", con, adOpenDynamic, adLockOptimistic
'Each Model name is loaded onto the list only once.
'Validation check to make sure only the vehicles that are not booked are loaded onto the list.
    If rs.EOF = False Then
        Do While rs.EOF = False
            Dim i As Integer
            Dim IsNewType As Boolean
            IsNewType = True
            For i = 0 To lstModel.ListCount - 1
                If lstModel.List(i) = rs.Fields("Model") Then
                    IsNewType = False
                End If
            Next
            If IsNewType = True Then
                If rs.Fields("IsBooked?") = False Then
                    lstModel.AddItem (rs.Fields("Model"))
                End If
            End If
        rs.MoveNext
    Loop
End If

End Sub
'Model list click event.
Private Sub lstModel_Click()
'The information that are dependant on the Make and Model are refreshed if Model is changed, to
prevent database storage override.
    lstVehicleID.Clear
    txtRegNum.Text = ""

```

'The Vehicle IDs of the Model of Vehicles of the particular Make is loaded.

```

Set rs = New Recordset
rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType & "' and Make = '" &
IstMake.Text & "' and Model = '" & IstModel.Text & "'", con, adOpenDynamic, adLockOptimistic
rs.MoveFirst

'Each Make name is loaded onto the list only once.
'Validation check ensure only the vehicles that are not booked are loaded onto the list.
If rs.EOF = False Then
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To lstVehicleID.ListCount - 1
            If lstVehicleID.List(i) = rs.Fields("VehicleID") Then
                IsNewType = False
            End If
        Next
        If IsNewType = True Then
            If rs.Fields("IsBooked?") = False Then
                lstVehicleID.AddItem (rs.Fields("VehicleID"))
            End If
        End If
        rs.MoveNext
    Loop
End If
End Sub

```

'Vehicle ID list click event.

```

Private Sub lstVehicleID_Click()
'Further required information for the selected Vehicle is loaded.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry where VehicleID ='" & lstVehicleID.Text & "'", con,
adOpenDynamic, adLockOptimistic
lblRatePerKm.Caption = rs.Fields("Rate_per_km")
lblNSeats.Caption = rs.Fields("NSeats")
txtRegNum.Text = rs.Fields("RegistrationNo")
End Sub

```

'Advance paid keypress event.

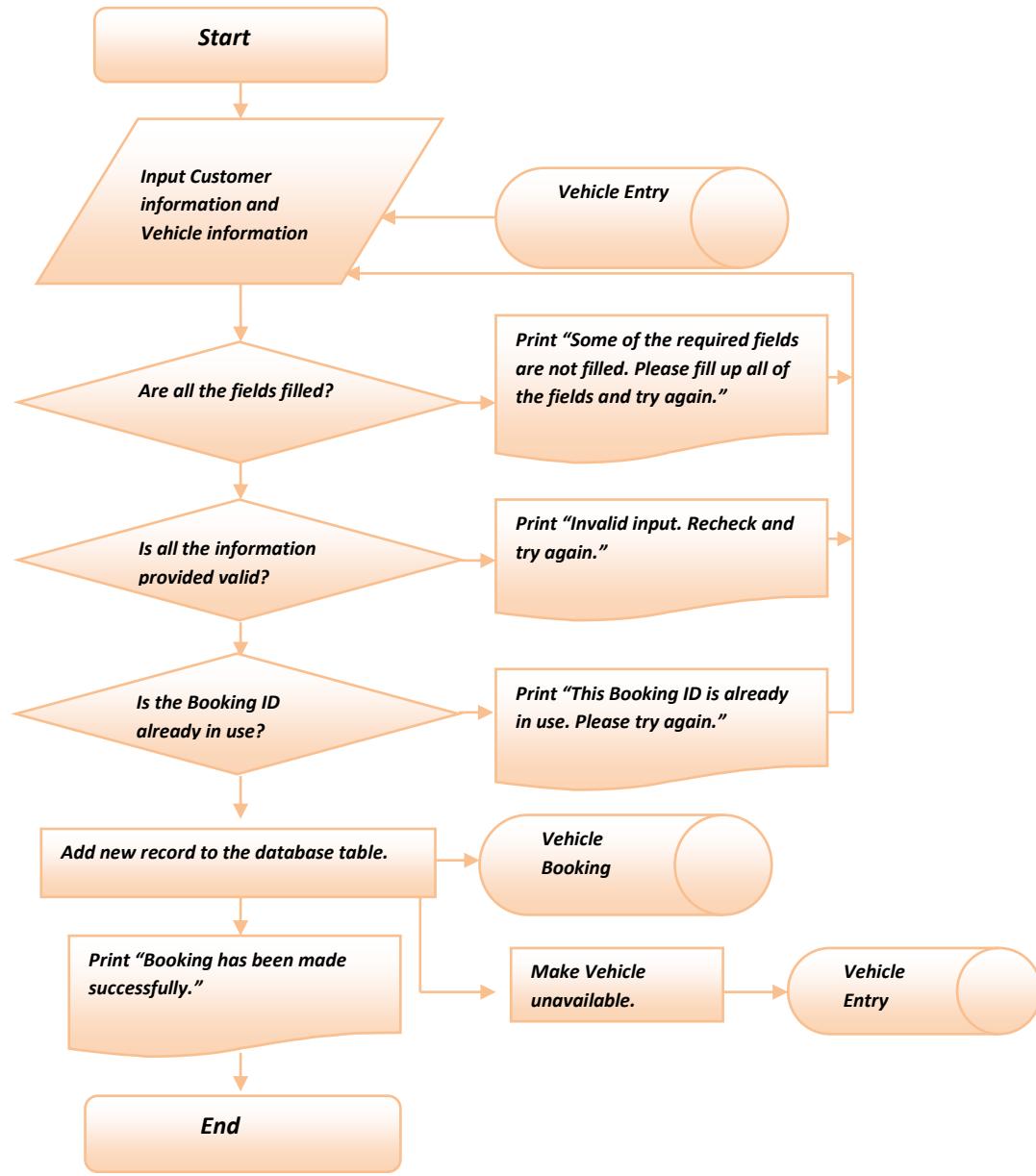
```

Private Sub txtAdvancePaid_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub

```

```
'Contact number keypress event.
Private Sub txtContact_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```

```
'Starting kilometer keypress event.
Private Sub txtStartingKm_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```



12. Vehicle Booking Update

The screenshot shows a Windows application window titled "Vehicle Booking Update". The window is divided into two main sections: "Customer information" on the left and "Vehicle information" on the right. The "Customer information" section contains fields for Booking ID, Customer Name, Address, Contact number, Journey date (set to 23-Mar-15), Drop off date (set to 23-Mar-15), and Booking date (labeled "Label19"). The "Vehicle information" section contains fields for Vehicle type, Make, Model, VehicleID, Registration number, Number of seats, Rate per kilometer, Advance paid, Driver name, and Starting Kilometer. On the right side of the window, there is a vertical column of buttons: Search, Update, Delete, Refresh, and Cancel. The date "23-03-2015" is displayed at the top right.

The screenshot shows the Microsoft Visual Basic IDE in design mode. The main area displays the following VBA code:

```

'Declaring global variables.
Dim BID As String
Dim OldVID As String
'Make combobox click event.
Private Sub cboMake_Click()
'Clearing the fields that are dependant on Make of the vehicle to prevent errors while booking, if Make of the Vehicle is changed.
    cboModel.Clear
    cboVehicleID.Clear
    lblRegNum.Caption = ""
    cboModel.Text = ""
    cboVehicleID.Text = ""
'Loading the Models available for the selected Make onto the list combobox from the database.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "' and Make ='" & cboMake.Text & "'", con,
'Ensuring that every available Model is loaded only once.
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboModel.ListCount - 1
            If cboModel.List(i) = rs.Fields("Model") Then
                IsNewType = False
                Exit For
            End If
        Next
        If IsNewType = True Then
            cboModel.AddItem (rs.Fields("Model"))
        End If
        rs.MoveNext
    Loop
End Sub
'Model combobox click event.
Private Sub cboModel_Click()
'Clearing the fields that are dependant on the Model of the vehicle to prevent errors in booking information if Model chosen by the
    cboVehicleID.Clear
    lblRegNum.Caption = ""
    cboVehicleID.Text = ""

```

The Project Explorer on the right shows the project structure with various modules and forms. The Properties window on the right shows settings for the "VehicleBooking" form, such as Appearance (1 - 3D), BackColor (&H8000000F), BorderStyle (2 - Sizable), and Caption (Vehicle Booking). The Form Layout window shows a diagram of the form with a single control labeled "VehicleBook".

```

'Declaring global variables.
Dim BID As String
Dim OldVID As String

'Make combobox click event.
Private Sub cboMake_Click()
'Clearing the fields that are dependant on Make of the vehicle to prevent errors while booking, if Make
of the Vehicle is changed.
    cboModel.Clear
    cboVehicleID.Clear
    lblRegNum.Caption = ""
    cboModel.Text = ""
    cboVehicleID.Text = ""

'Loading the Models available for the selected Make onto the list combobox from the database.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "' and Make
    ='" & cboMake.Text & "'", con, adOpenDynamic, adLockOptimistic
'Ensuring that every available Model is loaded only once.
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboModel.ListCount - 1
            If cboModel.List(i) = rs.Fields("Model") Then
                IsNewType = False
                Exit For
            End If
        Next
        If IsNewType = True Then
            cboModel.AddItem (rs.Fields("Model"))
        End If
        rs.MoveNext
    Loop
End Sub

'Model combobox click event.
Private Sub cboModel_Click()
'Clearing the fields that are dependant on the Model of the vehicle to prevent errors in booking
information if Model chosen by the user is changed.
    cboVehicleID.Clear
    lblRegNum.Caption = ""
    cboVehicleID.Text = ""

'Loading the Vehicle IDs for the Vehicle types of the selected Make and Model.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType & "' and Make = '" &
cboMake.Text & "' and Model = '" & cboModel.Text & "'", con, adOpenDynamic, adLockOptimistic
    rs.MoveFirst
    Do While Not rs.EOF
        cboVehicleID.AddItem (rs.Fields("VehicleID"))
    Loop
End Sub

```

```

    rs.MoveNext
    Loop
End Sub

'Vehicle ID combox click event.
Private Sub cboVehicleID_Click()
'Loading necessary booking information onto the form from the database for the selected Vehicle ID.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where VehicleID ='" & cboVehicleID.Text & "'", con,
adOpenDynamic, adLockOptimistic
    lblRatePerKm.Caption = rs.Fields("Rate_per_km")
    lblNSeats.Caption = rs.Fields("NSeats")
    lblRegNum.Caption = rs.Fields("RegistrationNo")
End Sub

'Vehicle type combobox click event.
Private Sub cboVehicleType_Click()
'Clearing the fields that are depemdamt on the Vehicle type to prevent errors in booking information if
the Vehicle type is changed.
    cboMake.Clear
    cboModel.Clear
    cboVehicleID.Clear
    lblRegNum.Caption = ""

    cboMake.Text = ""
    cboModel.Text = ""
    cboVehicleID.Text = ""

'The Makes that are available for the selected Vehicle type are loaded onto the Make combobox from
the database.
    Set rs = New Recordset
    rs.Open "select distinct * from Vehicle_entry where Vehicle_type ='" & cboVehicleType.Text & "'", 
con, adOpenDynamic, adLockOptimistic
'Validation check to ensure that the vehicles that are booked are not loaded.
    If rs.EOF = False Then
        Do While rs.EOF = False
            If rs.Fields("IsBooked?") = False Then
                cboMake.AddItem (rs.Fields("Make"))
            End If
            rs.MoveNext
        Loop
    End If

'Message shown to notify the user that there are no Makes available for the chosen Vehicle type if
necessary.
    If cboMake.ListCount = 0 Then
        MsgBox "No cars of that type available"
        cboVehicleType.Text = ""
        Exit Sub
    End If

```

```
End Sub
```

```
'Update button click event.
```

```
Private Sub cmbUpdate_Click()
```

```
'Validation check to make sure the User provides the Booking ID before update.
```

```
If txtBookingID.Text = Empty Then
```

```
    MsgBox "Please enter the Booking ID."
```

```
    Exit Sub
```

```
End If
```

```
'Presence check to make sure all the necessary information is provided.
```

```
If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = "" Or  
cboVehicleType.Text = "" Or lblRegNum.Caption = "" Or cboMake.Text = "" Or cboModel.Text = "" Or  
cboVehicleID.Text = "" Or lblRatePerKm.Caption = "" Or lblNSeats.Caption = "" Or txtAdvancePaid = "" Or  
txtDriverName.Text = "" Then
```

```
    MsgBox "None of the fields can be left empty."
```

```
    Exit Sub
```

```
End If
```

```
'Length check to make sure contact number is of 11 digits.
```

```
If Len(txtContact.Text) <> 11 Then
```

```
    MsgBox "Number of digits used for contact number is invalid."
```

```
    Exit Sub
```

```
End If
```

```
'Checking if Booking ID is changed.
```

```
If txtBookingID.Text <> BID Then
```

```
'Message shown to notify the User that Booking ID cannot be changed.
```

```
    MsgBox "Booking ID cannot be changed."
```

```
    txtBookingID = BID
```

```
    Exit Sub
```

```
End If
```

```
'Checking if Vehicle ID is changed, which means that the customer changed his choice of vehicle.
```

```
If cboVehicleID.Text <> OldVID Then
```

```
    Set rs = New Recordset
```

```
'Updating availability information if Vehicle ID is changed.
```

```
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
```

```
Do While Not rs.EOF
```

```
    If rs.Fields("VehicleID") = OldVID Then
```

```
        rs.Fields("IsBooked?") = False
```

```
    End If
```

```
    If rs.Fields("VehicleID") = cboVehicleID.Text Then
```

```
        rs.Fields("IsBooked?") = True
```

```
    End If
```

```
    rs.Update
```

```
    rs.MoveNext
```

```
Loop
```

```
rs.Close
End If

'Updating database information for the selected Booking ID
Set rs = New Recordset
rs.Open "select * from Vehicle_booking where VehicleBookingID = '" & txtBookingID.Text & "'", con,
adOpenDynamic, adLockOptimistic

    rs.Fields("Customer_name") = txtName.Text
    rs.Fields("Customer_address") = txtAddress.Text
    rs.Fields("Contact") = txtContact.Text
    rs.Fields("Journey_date") = dtpJourneyDate.Value
    rs.Fields("Drop_off_date") = dtpDropOffDate.Value
    rs.Fields("Vehicle_type") = cboVehicleType.Text
    rs.Fields("Registration_number") = lblRegNum.Caption
    rs.Fields("Make") = cboMake.Text
    rs.Fields("Model") = cboModel.Text
    rs.Fields("VehicleID") = cboVehicleID.Text
    rs.Fields("Rate_per_kilometer") = lblRatePerKm.Caption
    rs.Fields("Number_of_seats") = lblNSeats.Caption
    rs.Fields("Advance_paid") = txtAdvancePaid.Text
    rs.Fields("Driver_name") = txtDriverName.Text
    rs.Fields("Date_of_booking") = lblDate.Caption
    rs.Fields("Starting_Km") = txtStartingKm.Text
    rs.Update
'Message shown to notify the User that the update was successful.
MsgBox "Booking has been updated successfully."
'All the fields are cleared after the information is updated to ready form for next job.
txtBookingID.Text = ""
txtName.Text = ""
txtContact.Text = ""
txtAddress.Text = ""
dtpJourneyDate.Value = Date
dtpDropOffDate.Value = Date
cboVehicleType.Text = ""
lblRegNum.Caption = ""
cboMake.Clear
cboModel.Clear
cboVehicleID.Clear
lblRatePerKm.Caption = ""
lblNSeats.Caption = ""
txtAdvancePaid = ""
txtStartingKm = ""
txtDriverName.Text = ""

Call Form_Load
```

```
End Sub
```

'Cancel button click event.

```
Private Sub cmdCancel_Click()
```

'Closing the form.

Unload Me

```
End Sub
```

'Delete button click event.

```
Private Sub cmdDelete_Click()
```

'Presence check to make sure all the required information is provided.

```
If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" And txtAddress.Text = "" Or  
cboVehicleType.Text = "" Or lblRegNum.Caption = "" Or cboMake.Text = "" Or cboModel.Text = "" Or  
cboVehicleID.Text = "" Or lblRatePerKm.Caption = "" Or lblNSeats.Caption = "" Or txtAdvancePaid = "" Or  
txtDriverName.Text = "" Then
```

'Message shown to notify the User that all the information needs to be provided.

```
MsgBox "None of the fields can be left empty."
```

```
Exit Sub
```

```
End If
```

'Deletion of the selected record from the database.

```
rs.Delete
```

```
MsgBox "Booking has been deleted."
```

```
rs.Close
```

'Updating the availability information for the vehicle in the Vehicle Entry database.

```
rs.Open " select * from Vehicle_entry where VehicleID ='" & cboVehicleID.Text & "'", con,  
adOpenDynamic, adLockOptimistic
```

```
rs.Fields("IsBooked?") = False
```

```
rs.Update
```

'Clearing the unnecessary information on the form after deletion of the record.

```
txtBookingID.Text = ""
```

```
txtName.Text = ""
```

```
txtContact.Text = ""
```

```
txtAddress.Text = ""
```

```
dtpJourneyDate.Value = Date
```

```
dtpDropOffDate.Value = Date
```

```
cboVehicleType.Text = ""
```

```
lblRegNum.Caption = ""
```

```
cboMake.Clear
```

```
cboModel.Clear
```

```
cboVehicleID.Clear
```

```
lblRatePerKm.Caption = ""
```

```
lblNSeats.Caption = ""
```

```
txtAdvancePaid = ""
```

```
txtStartingKm = ""
```

```
txtDriverName.Text = ""
```

```
rs.Close
```

```
Call Form_Load
```

```
End Sub
```

```
'Refresh button click event.
```

```
Private Sub cmdRefresh_Click()
```

```
'Clearing all the fields on the form as required by the User in case the User wants to start anew.
```

```
txtBookingID.Text = ""
```

```
txtName.Text = ""
```

```
txtContact.Text = ""
```

```
txtAddress.Text = ""
```

```
dtpJourneyDate.Value = Date
```

```
dtpDropOffDate.Value = Date
```

```
cboVehicleType.Text = ""
```

```
lblRegNum.Caption = ""
```

```
cboMake.Clear
```

```
cboModel.Clear
```

```
cboVehicleID.Clear
```

```
lblRatePerKm.Caption = ""
```

```
lblNSeats.Caption = ""
```

```
txtAdvancePaid = ""
```

```
txtStartingKm.Text = ""
```

```
txtDriverName.Text = ""
```

```
Call Form_Load
```

```
End Sub
```

```
'Search button click event.
```

```
Private Sub cmdSearch_Click()
```

```
'Presence check to ensure the User provides the Booking ID to be searched.
```

```
If txtBookingID.Text = Empty Then
```

```
'Message shown to ask for the Vehicle ID to be searched from the User.
```

```
MsgBox "Please enter the Booking ID."
```

```
Exit Sub
```

```
End If
```

```
'The Vehicle booking database is searched for a particular Booking ID.
```

```
Set rs = New Recordset
```

```
rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic
```

```
Dim found As Boolean
```

```
found = False
```

```
rs.MoveFirst
```

```
Do While Not rs.EOF And Not found
```

```
If rs.Fields("VehicleBookingID") = txtBookingID.Text Then
```

```
found = True
```

```
End If
```

```
rs.MoveNext
```

```
Loop
```

```
If found = False Then
```

```

'Message shown to notify User if Booking ID not found.
MsgBox "Booking ID being searched is not found. Please enter correct Booking ID and try again."
Exit Sub
End If
rs.Close

'The Booking information for the particular Booking ID provided is loaded onto the form from the
database.
Set rs = New Recordset
rs.Open "select * from Vehicle_booking where VehicleBookingID = '" & txtBookingID.Text & "'", con,
adOpenDynamic, adLockOptimistic

txtName.Text = rs.Fields("Customer_name")
txtContact.Text = rs.Fields("Contact")
txtAddress.Text = rs.Fields("Customer_address")
dtpJourneyDate.Value = rs.Fields("Journey_date")
dtpDropOffDate.Value = rs.Fields("Drop_off_date")
lblBookingDate.Caption = rs.Fields("Date_of_booking")
cboVehicleType.Text = rs.Fields("Vehicle_type")
cboMake.Text = rs.Fields("Make")
cboModel.Text = rs.Fields("Model")
cboVehicleID.Text = rs.Fields("VehicleID")
lblRegNum.Caption = rs.Fields("Registration_number")
lblNSeats.Caption = rs.Fields("Number_of_seats")
lblRatePerKm.Caption = rs.Fields("Rate_per_kilometer")
txtAdvancePaid.Text = rs.Fields("Advance_paid")
txtDriverName.Text = rs.Fields("Driver_name")
txtStartingKm.Text = rs.Fields("Starting_Km")

rs.Update

OldVID = cboVehicleID.Text
BID = txtBookingID.Text
End Sub

'Form load event.
Private Sub Form_Load()
'Calling function to connect to database.
Call dblink
Showing the current date on the form.
lblDate.Caption = Format(Now, "dd/mm/yyyy")
dtpJourneyDate.Value = Date
dtpDropOffDate.Value = Date
'Loading Vehicle type information onto the Vehicle type combobox from the database.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Check made to make sure each type is loaded only once from the database.
If rs.EOF = False Then
Do While rs.EOF = False

```

```
Dim i As Integer
Dim IsNewType As Boolean
IsNewType = True
For i = 0 To cboVehicleType.ListCount - 1
    If cboVehicleType.List(i) = rs.Fields("Vehicle_type") Then
        IsNewType = False
    End If
Next
If IsNewType = True Then
    cboVehicleType.AddItem(rs.Fields("Vehicle_type"))
End If
rs.MoveNext
Loop
End If
End Sub
```

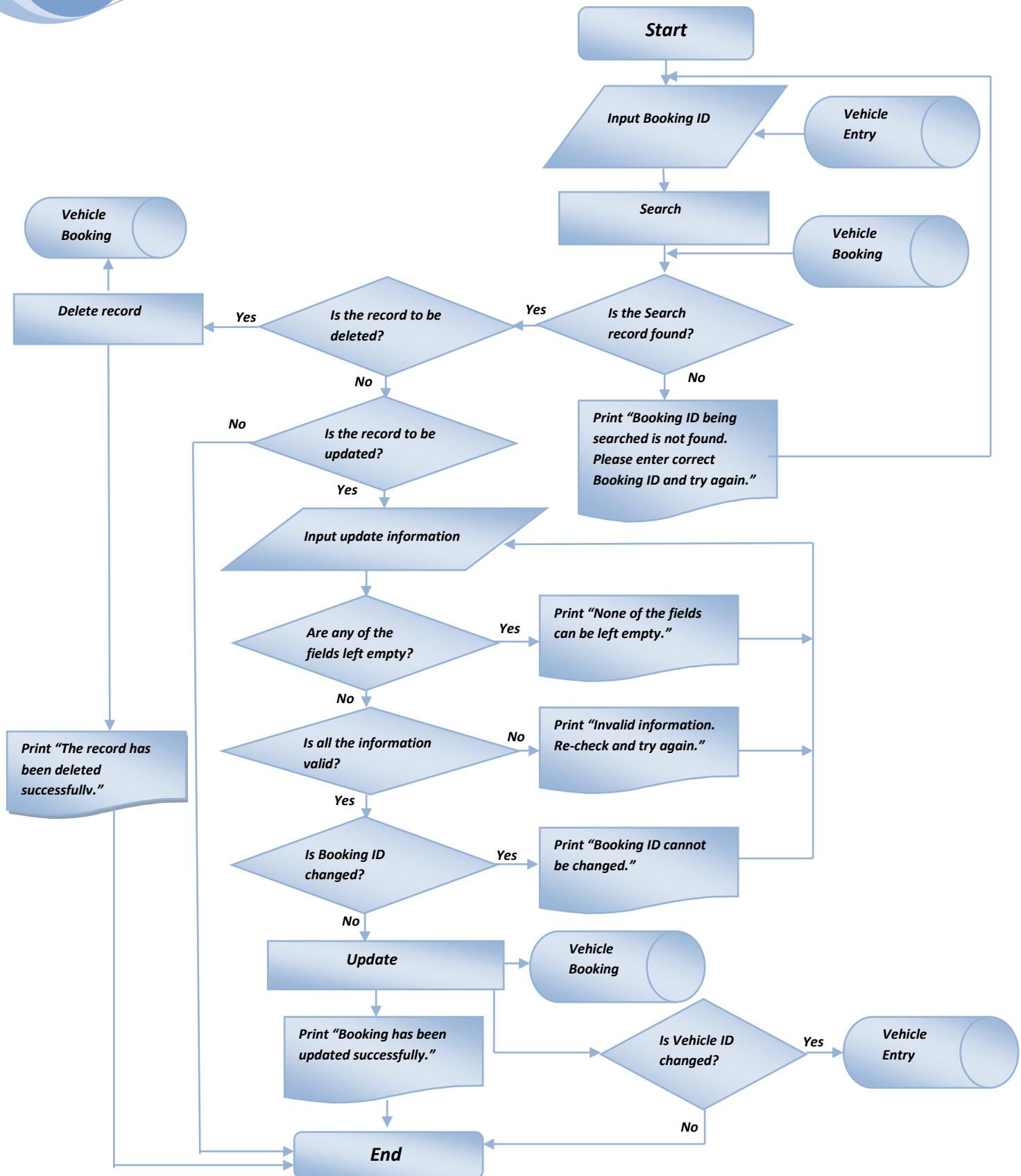
```
'Advance paid keypress event.
Private Sub txtAdvancePaid_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```

```
Private Sub txtContact_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```

```
Private Sub txtStartingKm_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
End Sub
```

Jan. 1

Travel Agency Management System



13. Vehicle Billing form

Customer information

Vehicle information

Billing Information

Cost of travel

Dates

Billing

```

'Declaring global variables.
Dim BID As String
Dim OldVID As String
'Make combobox click event.
Private Sub cboMake_Click()
'Clearing the fields that are dependant on Make of the vehicle to prevent errors while booking, if Make of the Vehicle is changed.
    cboModel.Clear
    cboVehicleID.Clear
    lblRegNum.Caption = ""
    cboModel.Text = ""
    cboVehicleID.Text = ""
'Loading the Models available for the selected Make onto the list combobox from the database.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "' and Make ='" & cboMake.Text & "'", con,
'Ensuring that every available Model is loaded only once.
    Do While rs.EOF = False
        Dim i As Integer
        Dim IsNewType As Boolean
        IsNewType = True
        For i = 0 To cboModel.ListCount - 1
            If cboModel.List(i) = rs.Fields("Model") Then
                IsNewType = False
                Exit For
            End If
        Next
        If IsNewType = True Then
            cboModel.AddItem (rs.Fields("Model"))
        End If
        rs.MoveNext
    Loop
End Sub
'Model combobox click event.
Private Sub cboModel_Click()
'Clearing the fields that are dependant on the Model of the vehicle to prevent errors in booking information if Model chosen by the
    cboVehicleID.Clear
    lblRegNum.Caption = ""
    cboVehicleID.Text = ""

```

```
'Search button click event.  
Private Sub cmbSearch_Click()  
'Validation check to make sure Booking ID is provided.  
If txtBookingID.Text = Empty Then  
    MsgBox "Please enter the Booking ID and try again."  
    Exit Sub  
End If  
  
'The Vehicle booking database is searched for a paticular Booking ID.  
Set rs = New Recordset  
rs.Open "select * from Vehicle_booking", con, adOpenDynamic, adLockOptimistic  
Dim found As Boolean  
found = False  
rs.MoveFirst  
Do While Not rs.EOF And Not found  
    If rs.Fields("VehicleBookingID") = txtBookingID.Text Then  
        found = True  
    End If  
    rs.MoveNext  
Loop  
  
If found = False Then  
'Message shown to notify User if Booking ID not found.  
    MsgBox "There is no record for this Booking ID."  
    Exit Sub  
End If  
rs.Close  
'The Billing information is loaded from the Vehicle Booking database.  
Set rs = New Recordset  
rs.Open "select * from Vehicle_booking where VehicleBookingID=""" & txtBookingID.Text & """", con,  
adOpenDynamic, adLockOptimistic  
txtName.Text = rs.Fields("Customer_name")  
txtContact.Text = rs.Fields("Contact")  
txtAddress.Text = rs.Fields("Customer_address")  
txtVehicleType.Text = rs.Fields("Vehicle_type")  
txtMake.Text = rs.Fields("Make")  
txtModel.Text = rs.Fields("Model")  
txtRegNum.Text = rs.Fields("Registration_number")  
txtVehicleID.Text = rs.Fields("VehicleID")  
txtDriverName.Text = rs.Fields("Driver_name")  
lblNSeats.Caption = rs.Fields("Number_of_seats")  
txtStartingKm.Text = rs.Fields("Starting_km")  
lblRatePerKm.Caption = rs.Fields("Rate_per_kilometer")  
dtpBookingDate.Value = rs.Fields("Date_of_booking")  
dtpJourneyDate.Value = rs.Fields("Journey_date")  
dtpDropOffDate.Value = rs.Fields("Drop_off_date")  
txtAdvancePaid.Text = rs.Fields("Advance_paid")
```

```
txtBookingID.Locked = True
```

'The difference between the dates is calculated.

```
Dim days As Integer  
days = DateDiff("d", dtpDropOffDate.Value, dtpBillingDate.Value)  
If dtpBillingDate.Value > dtpDropOffDate.Value Then  
'Calculation of late fine.  
    txtLateFine.Text = days * 50  
Else  
    txtLateFine.Text = 0  
End If
```

```
txtLateFine2.Text = txtLateFine.Text  
txtBookingCharge.Text = 2000
```

```
End Sub
```

'Cancel button click event.

```
Private Sub cmdCancel_Click()  
'Unloading the form.  
    Unload Me  
End Sub
```

'Generate Bill button click event.

```
Private Sub cmdGenerateBill_Click()
```

'Validation check to make sure none of the fields are left empty.

```
If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = Empty Or  
txtVehicleType.Text = "" Or txtVehicleID.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or  
txtRegNum.Text = "" Or txtDriverName.Text = "" Or lblINSeats.Caption = "" Or txtStartingKm.Text = "" Or  
txtEndingKm.Text = "" Or txtKmTravelled.Text = "" Or txtCostOfTravel.Text = "" Or txtLateFine.Text = ""  
Or txtBookingCharge.Text = "" Or txtDriverCharge.Text = "" Or txtAdditionalExp.Text = "" Or  
txtDiscount.Text = "" Or txtAdvancePaid.Text = "" Then
```

```
    MsgBox "Please make sure none of the fields are left empty and try again."
```

```
    Exit Sub
```

```
End If
```

'Length check to make sure contact number is of 11 digits.

```
If Len(txtContact.Text) <> 11 Then  
    MsgBox "Number of digits used for contact number is invalid."  
    Exit Sub  
End If
```

```
If txtDiscount.Text = "" Then  
    txtDiscount.Text = 0  
End If
```

```
txtGross.Text = Val(txtBookingCharge.Text) + Val(txtDriverCharge.Text) + Val(txtCostOfTravel2.Text) +  
Val(txtLateFine2.Text) + Val(txtAdditionalExp.Text)  
txtNet.Text = Val(txtGross.Text) - Val(txtDiscount.Text) - Val(txtAdvancePaid.Text)
```

```
End Sub
```

'Refresh button click event.

```
Private Sub cmdRefresh_Click()
```

'All the fields on the form are cleared on the click of the button as required by the User.

```
txtBookingID.Text = ""  
txtName.Text = ""  
txtAddress.Text = ""  
txtContact.Text = ""  
txtVehicleType.Text = ""  
txtMake.Text = ""  
txtModel.Text = ""  
txtVehicleID = ""  
txtRegNum.Text = ""  
txtDriverName.Text = ""  
lblNSeats.Caption = ""  
txtStartingKm.Text = ""  
txtEndingKm.Text = ""  
txtKmTravelled.Text = ""  
lblRatePerKm.Caption = ""  
txtCostOfTravel.Text = ""  
dtpBookingDate.Value = Date  
dtpJourneyDate.Value = Date  
dtpDropOffDate.Value = Date  
dtpBillingDate.Value = Date  
txtLateFine.Text = ""  
txtBookingCharge.Text = ""  
txtDriverCharge.Text = ""  
txtCostOfTravel2.Text = ""  
txtLateFine2.Text = ""  
txtAdditionalExp.Text = ""  
txtGross.Text = ""  
txtDiscount.Text = ""  
txtAdvancePaid.Text = ""  
txtNet.Text = ""
```

```
txtBookingID.Locked = False
```

```
End Sub
```

'Save button click event.

```
Private Sub cmdSave_Click()
```

'Validation check to make sure none of the fields are left empty.

```
If txtBookingID.Text = "" Or txtName.Text = "" Or txtContact.Text = "" Or txtAddress.Text = Empty Or  
txtVehicleType.Text = "" Or txtVehicleID.Text = "" Or txtMake.Text = "" Or txtModel.Text = "" Or  
txtRegNum.Text = "" Or txtDriverName.Text = "" Or lblNSeats.Caption = "" Or txtStartingKm.Text = "" Or  
txtEndingKm.Text = "" Or txtKmTravelled.Text = "" Or txtCostOfTravel.Text = "" Or txtLateFine.Text = ""  
Or txtBookingCharge.Text = "" Or txtDriverCharge.Text = "" Or txtAdditionalExp.Text = "" Or  
txtGross.Text = "" Or txtDiscount.Text = "" Or txtAdvancePaid.Text = "" Or txtNet.Text = "" Then  
'Message shown to notify the User that none of the fields can be left empty.
```

```

    MsgBox "Please make sure none of the fields are left empty and try again."
    Exit Sub
End If

'Length check to make sure contact number is of 11 digits.
If Len(txtContact.Text) <> 11 Then
    MsgBox "Number of digits used for contact number is invalid."
    Exit Sub
End If

'Storage of Billing information in the Vehicle Billing database.
Set rs = New Recordset
rs.Open "select * from Vehicle_billing", con, adOpenDynamic, adLockOptimistic
rs.AddNew
    rs.Fields("BookingID") = txtBookingID.Text
    rs.Fields("Customer_name") = txtName.Text
    rs.Fields("Customer_address") = txtAddress.Text
    rs.Fields("VehicleID") = txtVehicleID.Text
    rs.Fields("Kilometer_travelled") = txtKmTravelled.Text
    rs.Fields("Rate_per_kilometer") = lblRatePerKm.Caption
    rs.Fields("Cost_of_travel") = txtCostOfTravel.Text
    rs.Fields("Journey_date") = dtpJourneyDate.Value
    rs.Fields("Drop_off_date") = dtpDropOffDate.Value
    rs.Fields("Billing_date") = dtpBillingDate.Value
    rs.Fields("Booking_charge") = txtBookingCharge.Text
    rs.Fields("Driver_charge") = txtDriverCharge.Text
    rs.Fields("Cost_of_travel") = txtCostOfTravel.Text
    rs.Fields("Late_fine") = txtLateFine.Text
    rs.Fields("Additional_expenditures") = txtAdditionalExp.Text
    rs.Fields("Gross_total") = txtGross.Text
    rs.Fields("Discount") = txtDiscount.Text
    rs.Fields("Advance_paid") = txtAdvancePaid.Text
    rs.Fields("Net_total") = txtNet.Text
rs.Update

MsgBox "The Billing information is saved."

End Sub

'Form load click event.
Private Sub Form_Load()
'Calling the function to connect to database.
    Call dblink
    lblDate.Caption = Date
    dtpBillingDate.Value = Date

End Sub

Private Sub txtAdditionalExp_KeyPress(KeyAscii As Integer)

```

```
'Character type check to make sure non-numerical data is not allowed to be entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtBookingCharge_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure non-numerical data is not allowed to be entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtAdvancePaid_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure non-numerical data is not allowed to be entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
End Sub
```

```
Private Sub txtContact_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure non-numerical data is not allowed to be entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtCostOfTravel_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure non-numerical data is not allowed to be entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then  
    MsgBox "Please input numbers only."  
    Exit Sub  
End If  
  
End Sub
```

```
Private Sub txtCostOfTravel2_KeyPress(KeyAscii As Integer)
```

```
'Character type check to make sure non-numerical data is not allowed to be entered.
```

```
If KeyAscii = 8 Then Exit Sub  
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub
Private Sub txtDiscount_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub
Private Sub txtDriverCharge_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub
Private Sub txtEndingKm_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub
Private Sub txtGross_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub
Private Sub txtKmTravelled_KeyPress(KeyAscii As Integer)
'Character type check to make sure non-numerical data is not allowed to be entered.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If
```

```
End If
```

```
End Sub
```

'Kilometers travelled textbox lostfocus event.

```
Private Sub txtKmTravelled_LostFocus()
```

'Calculation of total amount charged to the customer for the total distance travelled.

```
txtKmTravelled.Text = Val(txtEndingKm.Text) - Val(txtStartingKm.Text)
```

```
txtCostOfTravel.Text = Val(txtKmTravelled.Text) * Val(lblRatePerKm.Caption)
```

```
txtCostOfTravel2.Text = txtCostOfTravel.Text
```

```
End Sub
```

```
Private Sub txtLateFine_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure non-numerical data is not allowed to be entered.

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

```
Private Sub txtLateFine2_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure non-numerical data is not allowed to be entered.

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

```
Private Sub txtNet_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure non-numerical data is not allowed to be entered.

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

```
Private Sub txtStartingKm_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure non-numerical data is not allowed to be entered.

```
If KeyAscii = 8 Then Exit Sub
```

```
If Not IsNumeric(Chr(KeyAscii)) Then
```

```
    MsgBox "Please input numbers only."
```

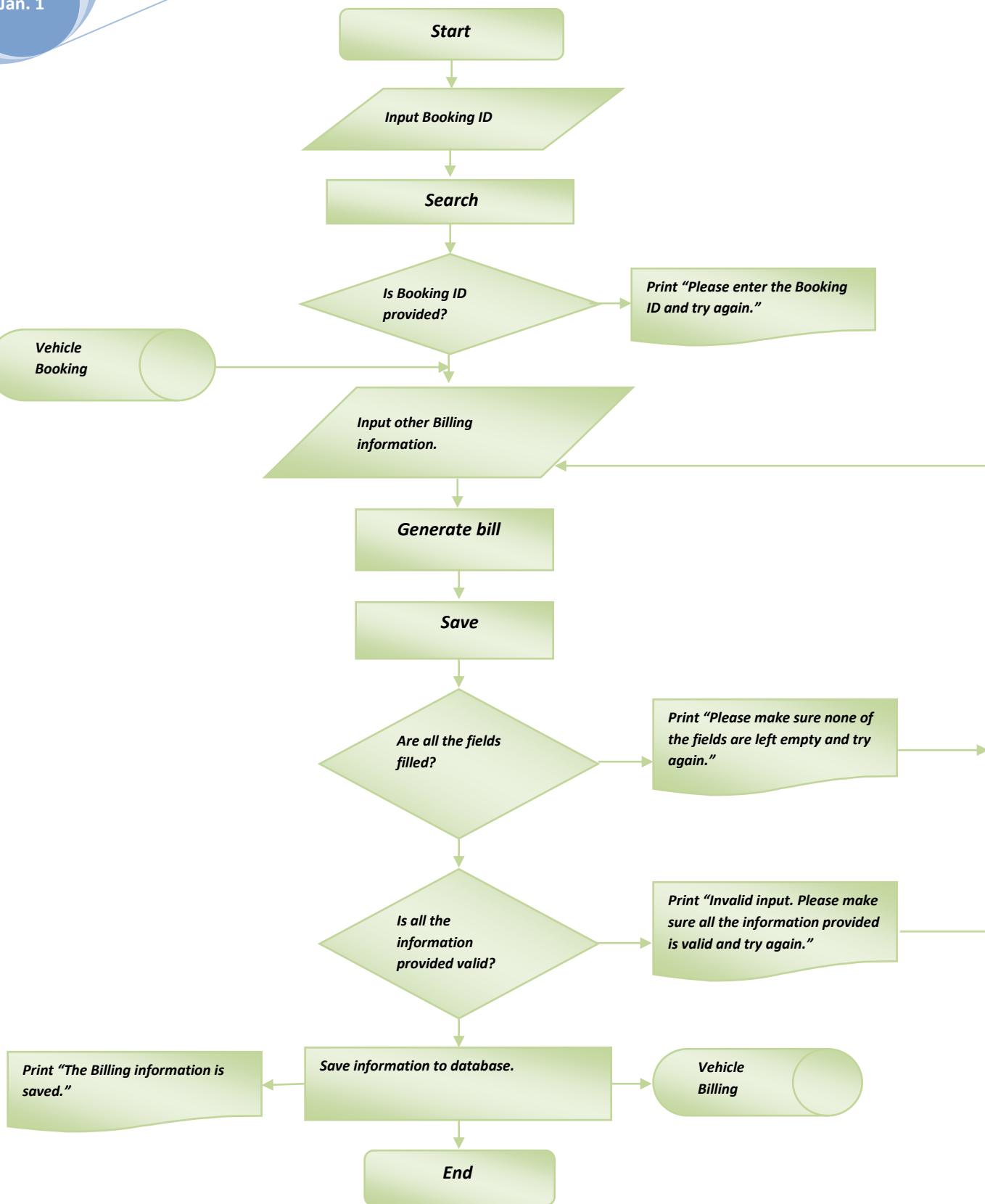
```
    Exit Sub
```

```
End If
```

```
End Sub
```

Jan. 1

Travel Agency Management System



14. Vehicle Servicing

Vehicle Servicing

Vehicle type: **Date**: **23-Mar-15**

Vehicle ID:

Servicing details:

Amount:

Save | **Refresh** | **Cancel**

Servicing History Preview

Date	Servicing Details	Amount

Project1 - Microsoft Visual Basic [design] - [VehicleServicing (Code)]

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

General

VehicleServicing

```

Private Sub cboVehicleType_Click()
    'Clearing Vehicle ID list before entering new data.
    lstVehicleID.Clear
    txtAmount.Text = ""
    txtServicingDetails.Text = ""
    'Loading Vehicle IDs onto the Vehicle ID list.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "'", con, adOpenDynamic, adLockOptimistic
    Do While rs.EOF = False
        lstVehicleID.AddItem (rs.Fields("VehicleID"))
        rs.MoveNext
    Loop
    rs.Close
End Sub

'Cancel button click event.
Private Sub cmdCancel_Click()
    'Clearing the fields as the User requires in case the User wants to start anew.
    cboVehicleType.Text = ""
    lstVehicleID.Clear
    txtAmount.Text = ""
    txtServicingDetails.Text = ""

End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
    'Clearing the fields as the User requires in case the User wants to start anew.
    cboVehicleType.Text = ""
    lstVehicleID.Clear
    txtAmount.Text = ""
    txtServicingDetails.Text = ""

End Sub

'Save button click event.
Private Sub cmdSave_Click()
    'Presence check to make sure the User provides all the necessary information.
    If cboVehicleType.Text = "" Or lstVehicleID.Text = "" Or txtAmount.Text = "" Or txtServicingDetails.Text = "" Then
        MsgBox "Some of the required fields are left empty. Please fill them up and try again."
    End If
End Sub

```

```
'Vehicle type combobox click event.  
Private Sub cboVehicleType_Click()  
'Clearing Vehicle ID list before entering new data.  
    lstVehicleID.Clear  
    txtAmount.Text = ""  
    txtServicingDetails.Text = ""  
'Loading Vehicle IDs onto the Vehicle ID list.  
    Set rs = New Recordset  
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "'", con,  
adOpenDynamic, adLockOptimistic  
    Do While rs.EOF = False  
        lstVehicleID.AddItem (rs.Fields("VehicleID"))  
        rs.MoveNext  
    Loop  
    rs.Close  
End Sub  
  
'Cancel button click event.  
Private Sub cmdCancel_Click()  
'Closing the form.  
    Unload Me  
End Sub  
  
'Refresh button click event.  
Private Sub cmdRefresh_Click()  
'Clearing the fields as the User requires in case the User wants to start anew.  
    cboVehicleType.Text = ""  
    lstVehicleID.Clear  
    txtAmount.Text = ""  
    txtServicingDetails.Text = ""  
  
End Sub  
  
'Save button click event.  
Private Sub cmdSave_Click()  
'Presence check to make sure the User provides all the necessary information.  
    If cboVehicleType.Text = "" Or lstVehicleID.Text = "" Or txtAmount.Text = "" Or txtServicingDetails.Text  
= "" Then  
        MsgBox "Some of the required fields are left empty. Please fill them up and try again."  
        Exit Sub  
    End If  
'Storing information into the database.  
    Set rs = New Recordset  
    rs.Open "select * from Vehicle_servicing", con, adOpenDynamic, adLockOptimistic  
    rs.AddNew  
        rs.Fields("Vehicle_type") = cboVehicleType.Text  
        rs.Fields("VehicleID") = lstVehicleID.Text  
        rs.Fields("Servicing_details") = txtServicingDetails.Text  
        rs.Fields("Amount") = txtAmount.Text  
        rs.Fields("Date") = Date
```

```

rs.Update
'Clearing the fields to input new data.
txtAmount.Text = ""
txtServicingDetails.Text = ""
flxgrdHistory.Rows = 2
Call lstVehicleID_Click
End Sub

'Form load event.
Private Sub Form_Load()
'Calling funtion to connect to database.
Call dblink
'Showing current date on the form.
lblDate.Caption = Date
'Loading the Vehicle type information onto the Vehicle type combobox from Vehicle entry database.
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
'Validity check to make sure each type is loaded only once.
If rs.EOF = False Then
  Do While rs.EOF = False
    Dim i As Integer
    Dim IsNewType As Boolean
    IsNewType = True
    For i = 0 To cboVehicleType.ListCount - 1
      If cboVehicleType.List(i) = rs.Fields("Vehicle_type") Then
        IsNewType = False
      End If
    Next
    If IsNewType = True Then
      cboVehicleType.AddItem (rs.Fields("Vehicle_type"))
    End If
    rs.MoveNext
  Loop
End If
flxgrdHistory.TextMatrix(0, 0) = "Date"
flxgrdHistory.TextMatrix(0, 1) = "Servicing Details"
flxgrdHistory.TextMatrix(0, 2) = "Amount"
flxgrdHistory.ColWidth(1) = 3000
flxgrdHistory.RowHeight(1) = 500
End Sub

'Vehicle ID list click event.
Private Sub lstVehicleID_Click()
Dim r As Integer
r = 2
txtAmount.Text = ""
txtServicingDetails.Text = ""
Set rs = New Recordset
rs.Open "select * from Vehicle_servicing where VehicleID ='" & lstVehicleID.Text & "'", con,

```

```

adOpenDynamic, adLockOptimistic
While Not rs.EOF
    r = r + 1

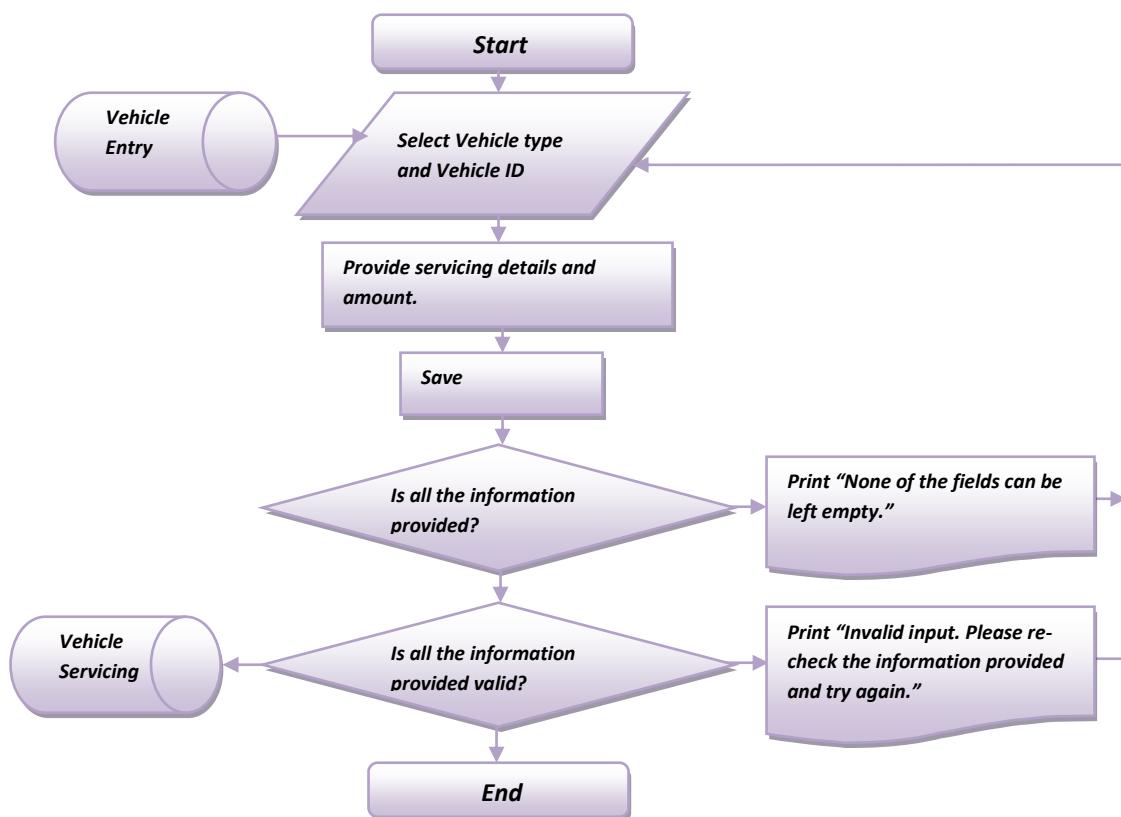
    flxgrdHistory.TextMatrix(r, 0) = rs!Date
    flxgrdHistory.TextMatrix(r, 1) = rs!Servicing_details
    flxgrdHistory.TextMatrix(r, 2) = rs!Amount

    rs.MoveNext
    flxgrdHistory.Rows = flxgrdHistory.Rows + 1
Wend
End Sub

'Amount keypress event.
Private Sub txtAmount_KeyPress(KeyAscii As Integer)
'Character type check to make sure only numerical data is input.
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only."
    Exit Sub
End If

End Sub

```



15. Fuel Record

Fuel Record

Date 23-03-2015

VehicleID Quantity litres

Fuel type Cost

Save Refresh Cancel

Project1 - Microsoft Visual Basic [design] - [VehicleServicing (Code)]

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

General cboVehicleType Click

```

'Vehicle type combobox click event.
Private Sub cboVehicleType_Click()
'Clearing Vehicle ID list before entering new data.
    lstVehicleID.Clear
    txtAmount.Text = ""
    txtServicingDetails.Text = ""
'Loading Vehicle IDs onto the Vehicle ID list.
    Set rs = New Recordset
    rs.Open "select * from Vehicle_entry where Vehicle_type = '" & cboVehicleType.Text & "'", con, adOpenDynamic, adLockOptimistic
    Do While rs.EOF = False
        lstVehicleID.AddItem (rs.Fields("VehicleID"))
        rs.MoveNext
    Loop
    rs.Close
End Sub

'Cancel button click event.
Private Sub cmdCancel_Click()
'Closing the form.
    Unload Me
End Sub

'Refresh button click event.
Private Sub cmdRefresh_Click()
'Clearing the fields as the User requires in case the User wants to start anew.
    cboVehicleType.Text = ""
    lstVehicleID.Clear
    txtAmount.Text = ""
    txtServicingDetails.Text = ""

End Sub

'Save button click event.
Private Sub cmdSave_Click()
'Presence check to make sure the User provides all the necessary information.
    If cboVehicleType.Text = "" Or lstVehicleID.Text = "" Or txtAmount.Text = "" Or txtServicingDetails.Text = "" Then
        MsgBox "Some of the required fields are left empty. Please fill them up and try again."
    End If
End Sub

```

Project - Project1

Properties - VehicleServicing

VehicleServicing Form

(Name)	VehicleServicing
Appearance	1 - 3D
AutoRedraw	False
BackColor	0x48000000
BorderStyle	2 - Sizable
Caption	Vehicle servicing
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid

Form Layout

```
'Cancel button click event.
```

```
Private Sub cmdCancel_Click()
```

```
'Unloading the form.
```

```
    Unload Me
```

```
End Sub
```

```
'Refresh button click event.
```

```
Private Sub cmdRefresh_Click()
```

```
'All the fields on the form are cleared.
```

```
    cboVehicleID.Text = ""
```

```
    cboFuelType.Text = ""
```

```
    txtQuantity.Text = ""
```

```
    txtCost.Text = ""
```

```
End Sub
```

```
'Save button click event.
```

```
Private Sub cmdSave_Click()
```

```
'Validation check to make sure that none of the important fields are kept empty.
```

```
    If cboVehicleID.Text = Empty Or cboFuelType.Text = Empty Or txtQuantity.Text = Empty Or
```

```
    txtCost.Text = Empty Then
```

```
        MsgBox "None of the fields can be left empty."
```

```
        Exit Sub
```

```
    End If
```

```
'Storing information into the database.
```

```
    Set rs = New Recordset
```

```
    rs.Open "select * from Fuel_record", con, adOpenDynamic, adLockOptimistic
```

```
    rs.AddNew
```

```
        rs.Fields("VehicleID") = cboVehicleID.Text
```

```
        rs.Fields("Date_of_entry") = lblDate.Caption
```

```
        rs.Fields("FuelType") = cboFuelType.Text
```

```
        rs.Fields("FuelQuantity/litres") = txtQuantity.Text
```

```
        rs.Fields("FuelCost") = txtCost.Text
```

```
    rs.Update
```

```
    MsgBox "Record has been added successfully"
```

```
'The fields are cleared after the information are stored into the database.
```

```
    cboVehicleID.Text = ""
```

```
    cboFuelType.Text = ""
```

```
    txtQuantity.Text = ""
```

```
    txtCost.Text = ""
```

```
End Sub
```

```
'Form load click event.
```

```
Private Sub Form_Load()
```

```
'Calling function to connect to the database.
```

```
    Call dblink
```

```
'Shows current date.
```

```
    lblDate.Caption = Format(Now, "dd/mm/yyyy")
```

'The vehicle types available on the Vehicle Entry database are loaded.

```
Set rs = New Recordset
rs.Open "select * from Vehicle_entry", con, adOpenDynamic, adLockOptimistic
Do While Not rs.EOF
    cboVehicleID.AddItem (rs.Fields("VehicleID"))
    rs.MoveNext
Loop
End Sub
```

'Cost of fuel keypress event.

```
Private Sub txtCost_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure only numerical data is entered.

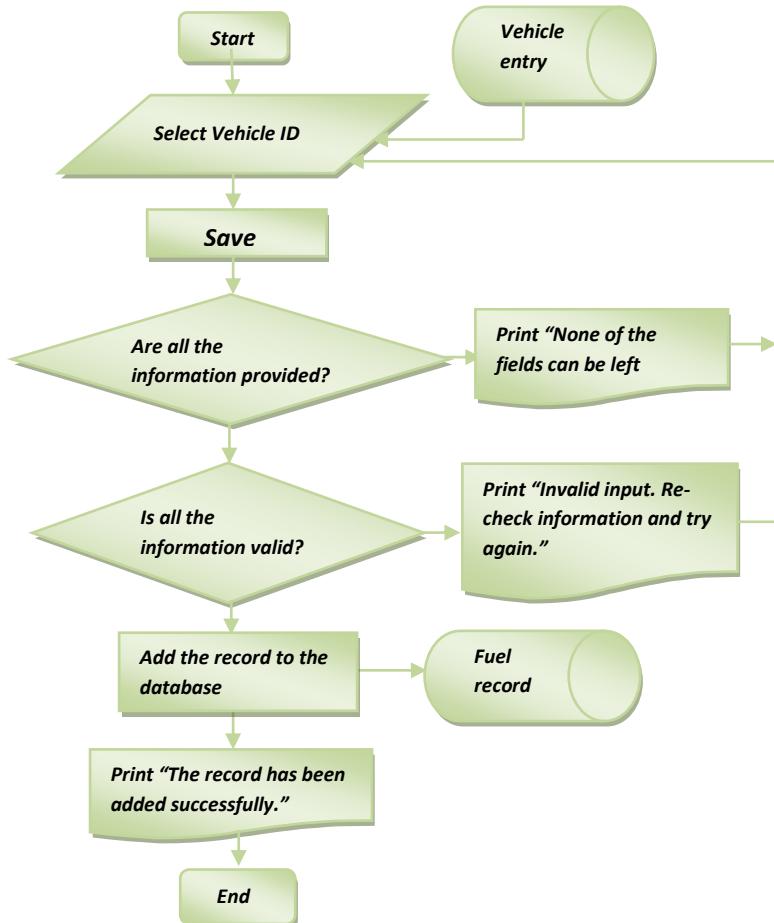
```
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```

'Quantity of fuel keypress event.

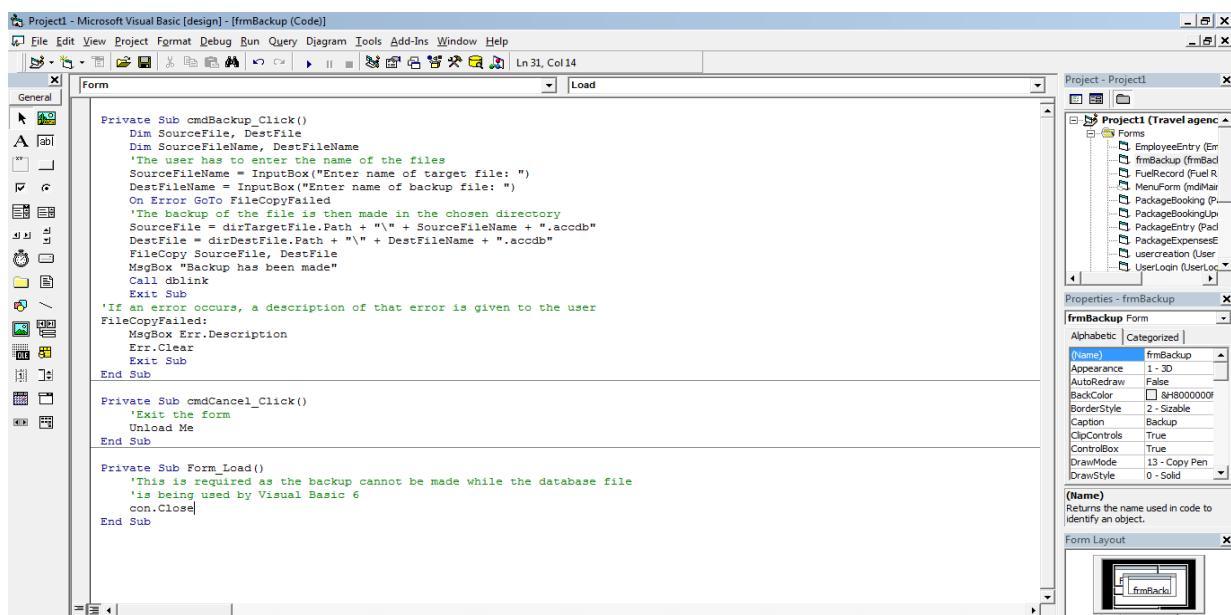
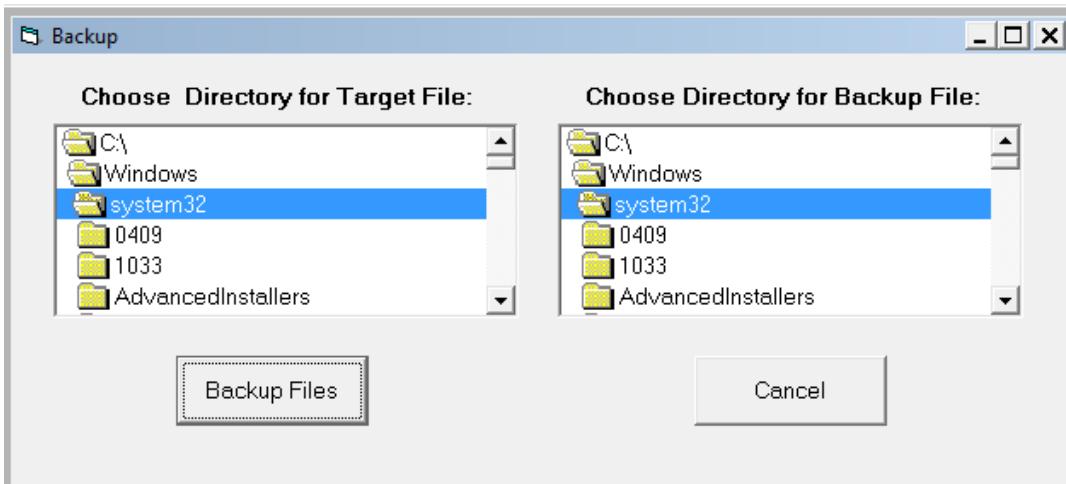
```
Private Sub txtQuantity_KeyPress(KeyAscii As Integer)
```

'Character type check to make sure only numerical data is entered.

```
If KeyAscii = 8 Then Exit Sub
If Not IsNumeric(Chr(KeyAscii)) Then
    MsgBox "Please input numbers only"
    Exit Sub
End If
End Sub
```



16. Backup form



```

Private Sub cmdBackup_Click()
    Dim SourceFile, DestFile
    Dim SourceFileName, DestFileName
    'The user has to enter the name of the files
    SourceFileName = InputBox("Enter name of target file: ")
    DestFileName = InputBox("Enter name of backup file: ")
    On Error GoTo FileCopyFailed
    'The backup of the file is then made in the chosen directory
    SourceFile = dirTargetFile.Path + "\\" + SourceFileName + ".accdb"
    DestFile = dirDestFile.Path + "\\" + DestFileName + ".accdb"
    FileCopy SourceFile, DestFile
    MsgBox "Backup has been made"
    Call dblink
    Exit Sub
    'If an error occurs, a description of that error is given to the user
    FileCopyFailed:
    MsgBox Err.Description
    Err.Clear
    Exit Sub
End Sub

Private Sub cmdCancel_Click()
    'Exit the form
    Unload Me
    End Sub

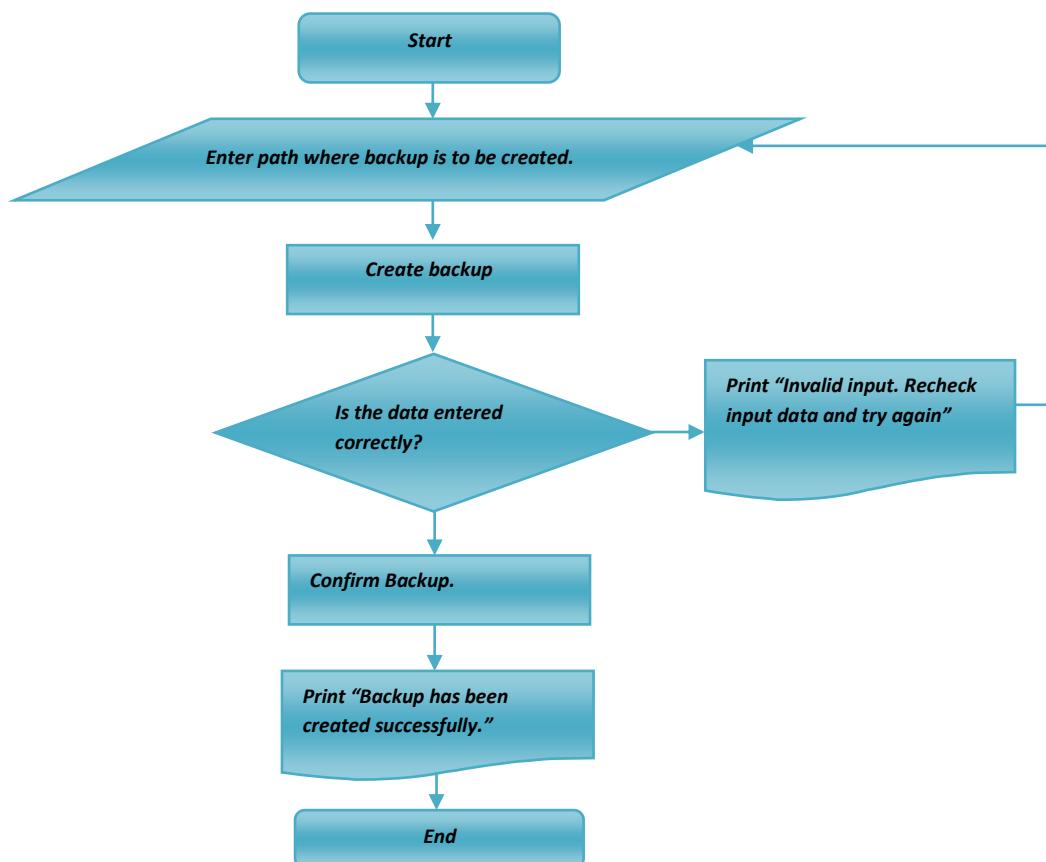
Private Sub Form_Load()
    'This is required as the backup cannot be made while the database file
    'is being used by Visual Basic 6
    con.Close
End Sub

```

```

Call dblink
Exit Sub
'If an error occurs, a description of that error is given to the user
FileCopyFailed:
    MsgBox Err.Description
    Err.Clear
    Exit Sub
End Sub
Private Sub cmdCancel_Click()
    'Exit the form
    Unload Me
End Sub
Private Sub Form_Load()
    'This is required as the backup cannot be made while the database file
    'is being used by Visual Basic 6
    con.Close
End Sub

```



❖ Report Designs

Crystal report software design wizard is used to design all the reports

a. Employee details

Employee Details

Employee ID	Employee Name	Address	Contact	Designation	Basic salary	Date of entry
EmployeeID	EName	Address	Contact	Designation	Basic_salary	Date_of_entry

2. Fuel Record

Report Title

Vehicle ID	Fuel Type	Quantity/litres	Cost	Date of entry
VehicleID	FuelType	FuelQuantity_litres	FuelCost	Date_of_entry

3. List of Packages

The screenshot shows the Crystal Reports interface with the following details:

- Report Header:** Contains the "MAAS Travels and Tours" logo.
- Page Header:** Contains the "Report Title".
- Details:** A repeating group containing fields: Package type, Package name, Package ID, Package details, Accommodation type, Cost per person, and Journey date.
- Report Footer:** Contains the "Page Number" field.

The report is currently in Design mode, as indicated by the tabs at the top. The toolbar and various report structures are visible on the left side of the main workspace.

4. List of Vehicles

The screenshot shows the Crystal Reports interface with the title bar "Crystal Reports - [List of Vehicles]". The menu bar includes File, Edit, View, Insert, Format, Database, Report, Analyzer, Window, Help. The toolbar has various icons for report creation and modification. The main workspace is divided into sections: Report Header, Page Header, Body, Details, Report Footer, and Page Footer. In the Body section, there is a logo for "MAAS Travels and Tours" and a table with the following columns:

Vehicle type	Vehicle ID	Registration number	Make	Model	Number of Seats	Rate per kilometer	Date of entry
--------------	------------	---------------------	------	-------	-----------------	--------------------	---------------

The "Details" section shows field mappings for these columns:

Field	Source
Vehicle_type	VehicleID
VehicleID	RegistrationNo
RegistrationNo	Make
Make	Model
Model	NSeats
NSeats	Rate_per_km
Rate_per_km	Date_of_entry

The "Page Footer" section contains a placeholder for "Page Number".

5. Customer list of Package Booking

Crystal Reports - [CustomerListofPackageBooking]

Design | Preview |

Report Header

Page Header

Page Header

Customer List of Package Booking

Package ID	Booking ID	Customer name	Address	Contact number	Number of people	Net total	Date of booking
PackageBookingID	Customer_name	Customer_address	Contact	number_of_people	Net_total	Date_of_booking	

Details

Report Footer

Page Footer

Page Footer

Print Date

Page Numbers

Records: 3

6. Customer list of Vehicle Booking

Crystal Reports - [CustomerListofVehicleBooking]

File Edit View Insert Format Database Report Analyzer Window Help

Design | Preview |

Report Header

Page Header

Page Header

Customer List of Vehicle Booking

Vehicle ID	Booking ID	Customer name	Address	Contact	Vehicle type	Advance paid	Date of booking	Journey date	Drop off date	Driver name
VehicleBookingID	Customer_name	Customer_address	Contact	Vehicle_type	Advance_paid	Date_of_booking	Journey_date	Drop_off_date	Driver_name	

Details

Report Footer

Page Footer

Page Number

For Help, press F1

Records: 1

7. List of Takings from Vehicle Rentals

Crystal Reports - [ListofTakingsFromVehicleRentals]

File Edit View Insert Format Database Report Analyzer Window Help

Design | Preview |

Report Header:

- Print Date
- 

Page Header:

Report Footer:

Page Footer:

For Help, press F1

Records: 0

List of Takings from Vehicle Rentals

Booking ID	Customer name	Vehicle ID	Net total
BookingID	Customer_name	VehicleID	Net_total

Page Number

8. Package Expenses list

Crystal Reports - [Package Expenses List]

File Edit View Insert Format Database Report Analyzer Window Help

Design | Preview |

Report Header:

- 

Page Header:

Report Footer:

Page Footer:

For Help, press F1

Records: 10

Report Title

PackageID	Expenses	Amount
PackageID	Expenses	Amount

Page Number

Jan. 1

Travel Agency Management System

9. List of Total Package Expenses

Crystal Reports - [ListOfTotalPackageExpenses]

File Edit View Insert Format Database Report Analyzer Window Help

04-Apr-15 2:46 1 of 1

Design Preview

PH

06-Apr-15

MAAS Travels and Tours

List of Total Package Expenses

Package ID	Package name	Total
1121	Summer Isles	\$22,650.00
1122	Wind of Brisbane	\$53,000.00

For Help, press F1

Records: 2 100%

10. Vehicle Servicing details

Crystal Reports - [Vehicle Servicing Details List]

File Edit View Insert Format Database Report Analyzer Window Help

04-Apr-15 2:46 1 of 1

Design Preview

Report Header

Page Header

MAAS Travels and Tours

Report Title

Vehicle type	Vehicle ID	Servicing details	Amount	Date of entry
Vehicle type	VehicleID	Servicing details	Amount	Date

Details

Report Footer

Page Footer

Page Number

For Help, press F1

Records: 4

Travel Agency Management System

11. Package Booking Receipt

Crystal Reports - [Package Booking Receipt]

Design | Preview |

Report Header

Page Header

Report Title



Date : Print Date

Booking ID : PackageBookingID

Date of booking : Date_of_bg

Customer name : Customer_name
Package ID : PackageID
Cost per person : per_person
Number of people : Number_of_p
Net total : Net_total

Details

Report Footer

Page Footer

For Help, press F1

Page Number

Records: 3

12. Vehicle Rental Token

Crystal Reports - [Token]

Design | Preview |

Page Header


Vehicle Rental Token

Date of booking : Date_of_booking

BookingID : VehicleBookingID Print date : Print Date

Customer name : Customer_name
Vehicle ID : VehicleID
Rate per kilometer : per_kilometer
Starting km : Starting_km
Journey date : Journey
Drop off date : Drop_of
Driver name : Driver_name

Advance paid : Advance_paid

Details

Page Footer

Page Number

For Help, press F1

Records: 1

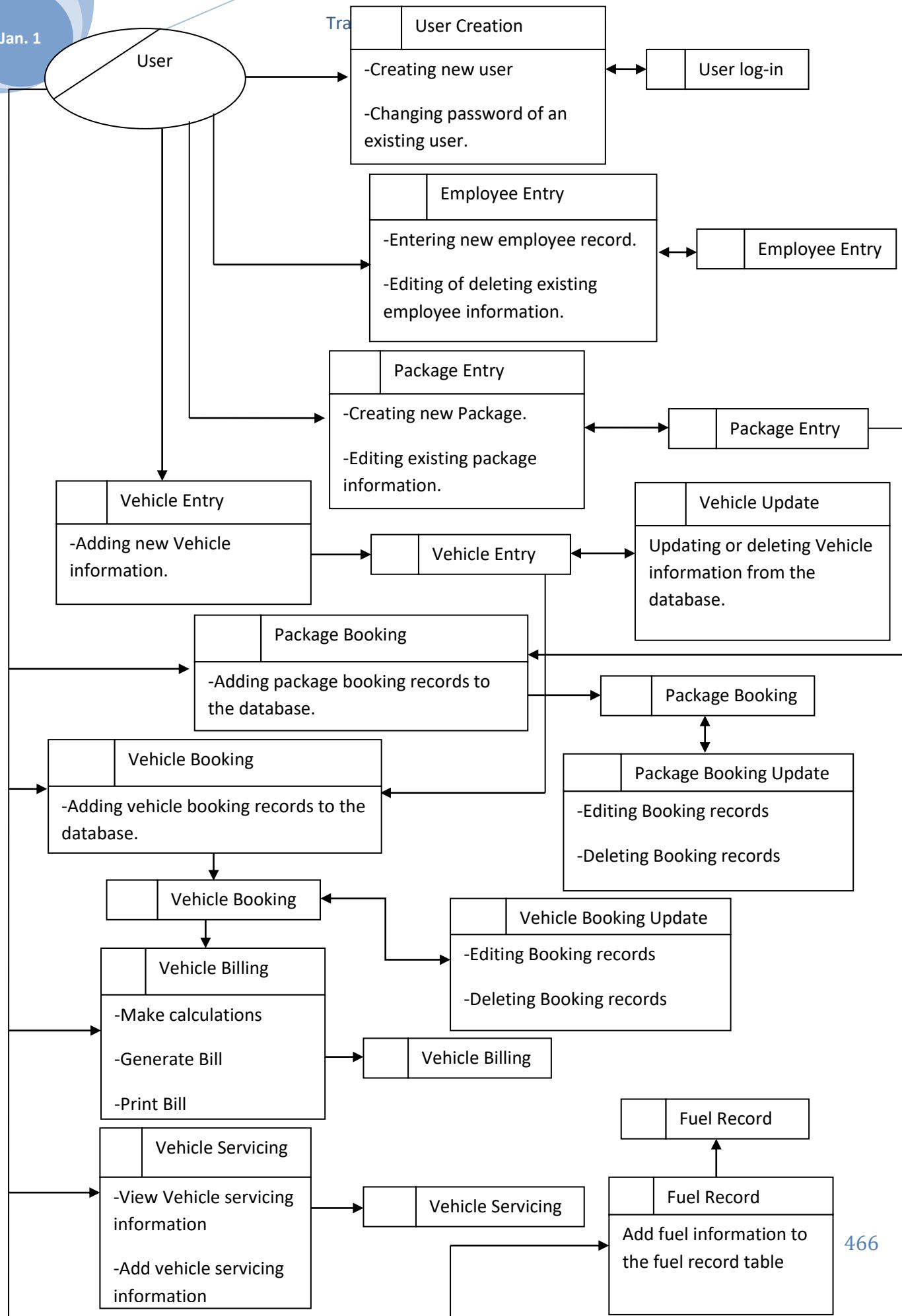
13. Vehicle Rental Bill

The screenshot shows the Crystal Reports interface with the title bar "Crystal Reports - [VehicleRentalBill]". The menu bar includes File, Edit, View, Insert, Format, Database, Report, Analyzer, Window, and Help. The toolbar contains various icons for report creation and modification. The left pane shows the report structure with sections: Report Header, Page Header, and Details. The Page Header section contains a logo for "MAAS Travels and Tours" and a title "Vehicle Rental Bill". The Details section contains a table with two columns of data. The left column includes fields like BookingID, Customer_name, Journey_date, Drop_off_date, Kilometers travelled, Rate per kilometer, Booking_charge, Driver_charge, Cost of travel, Late_fine, Additional_expenditures, Gross_total, Less, Discount, Advance_paid, and Net_total. The right column includes fields like Booking_charge, Driver_charge, Cost_of_travel, Late_fine, Additional_expenditures, Gross_total, Discount, Advance_paid, and Net_total. The bottom status bar indicates "Records: 1".

(ii) User documentation:

Data flow diagram:

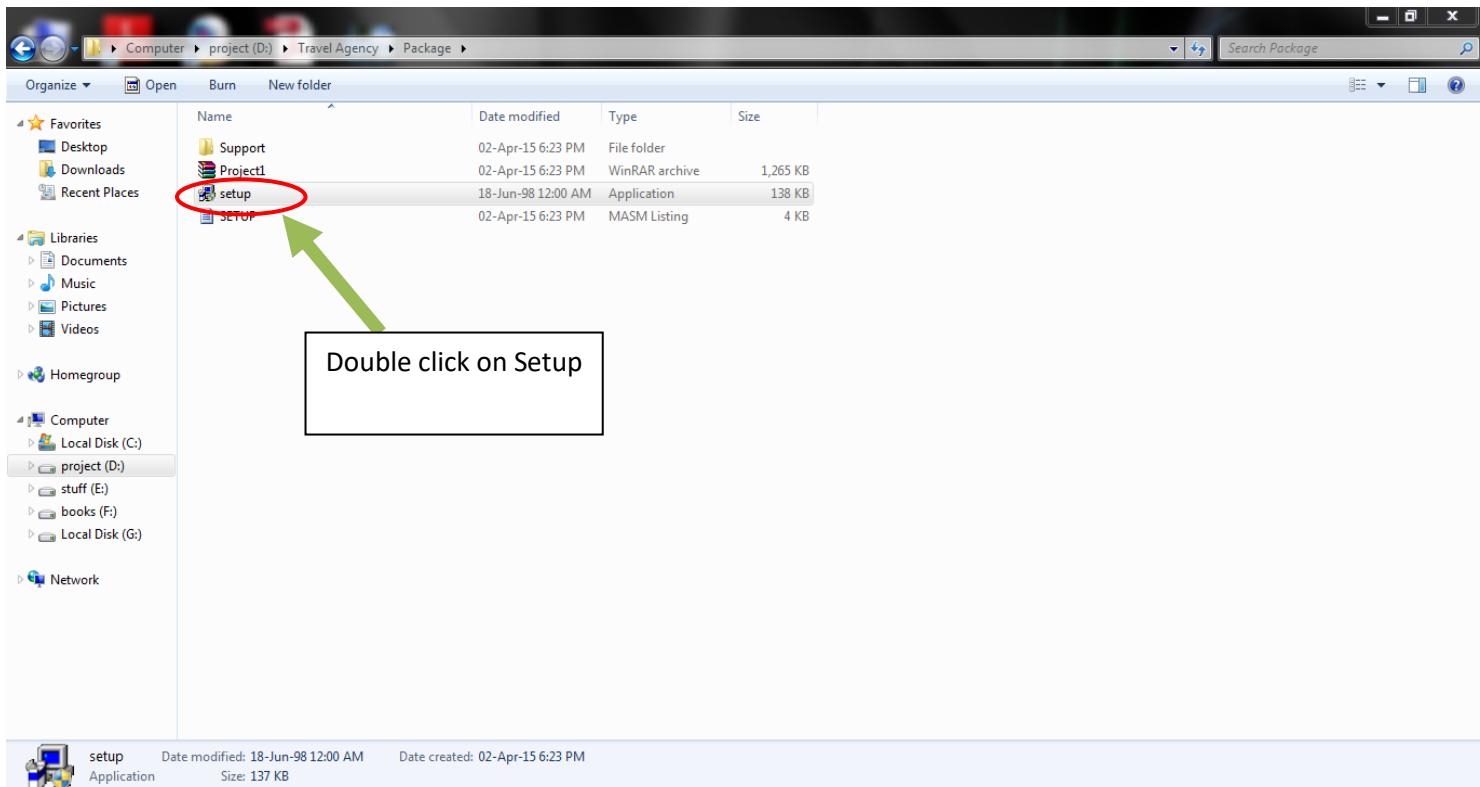
[Please turn over]



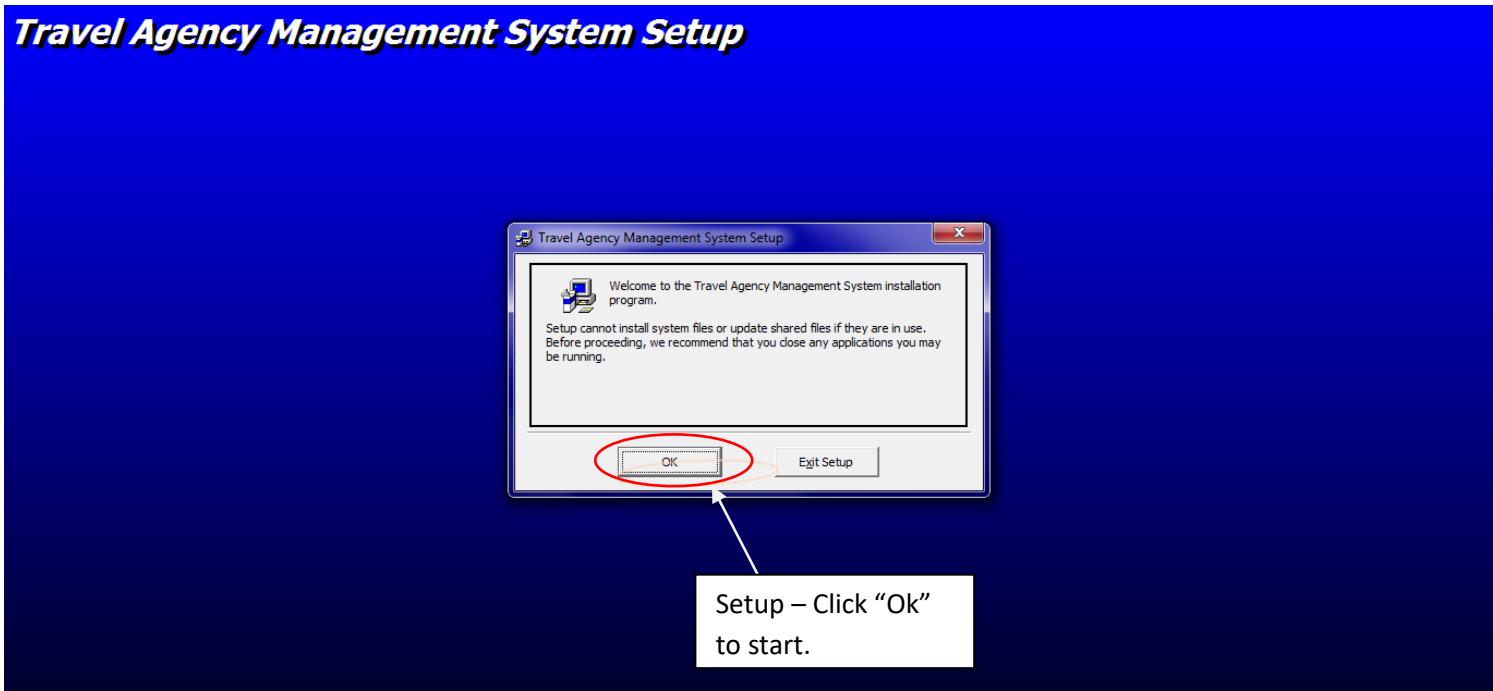
Installation instructions

The steps that need to be taken for the installation of the software are:

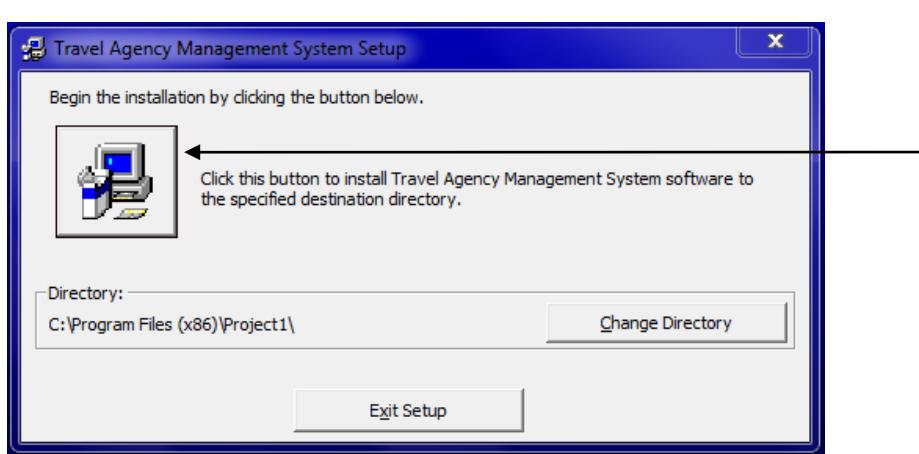
8. i. Open “Travel Agency” folder from Project (D:) drive.
- ii. Open folder named “Package”.
- iii. Double click on Setup.exe file.



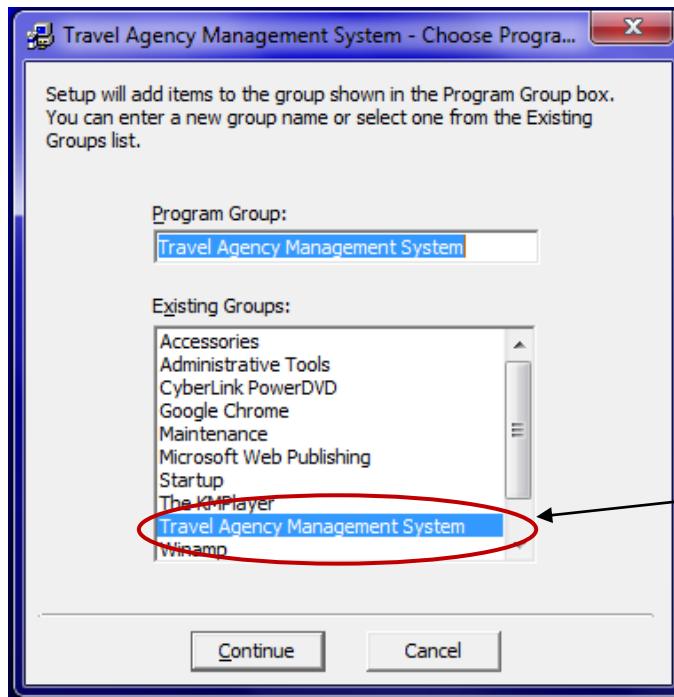
9. Click “Ok”



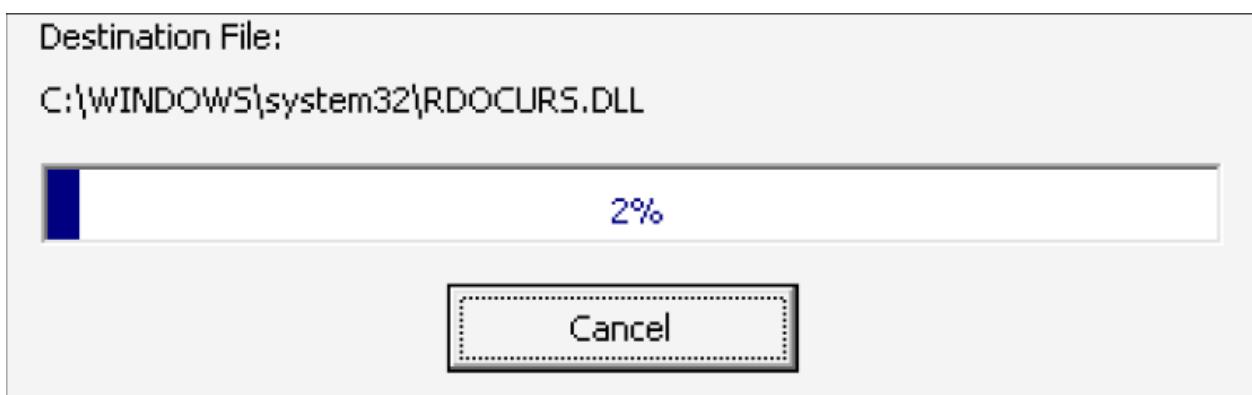
10. Click the button shown by the arrow.



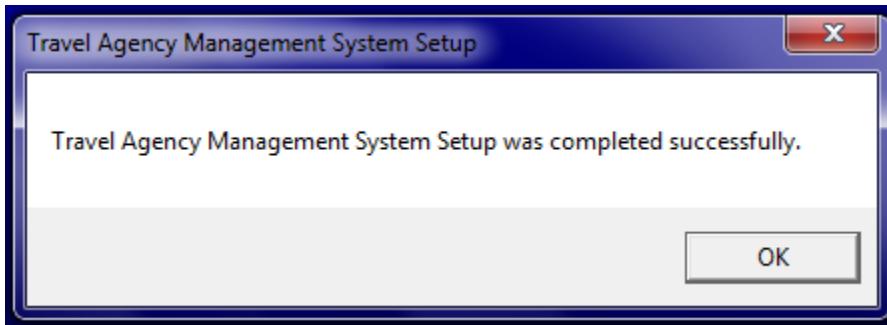
11. Click “Continue”.



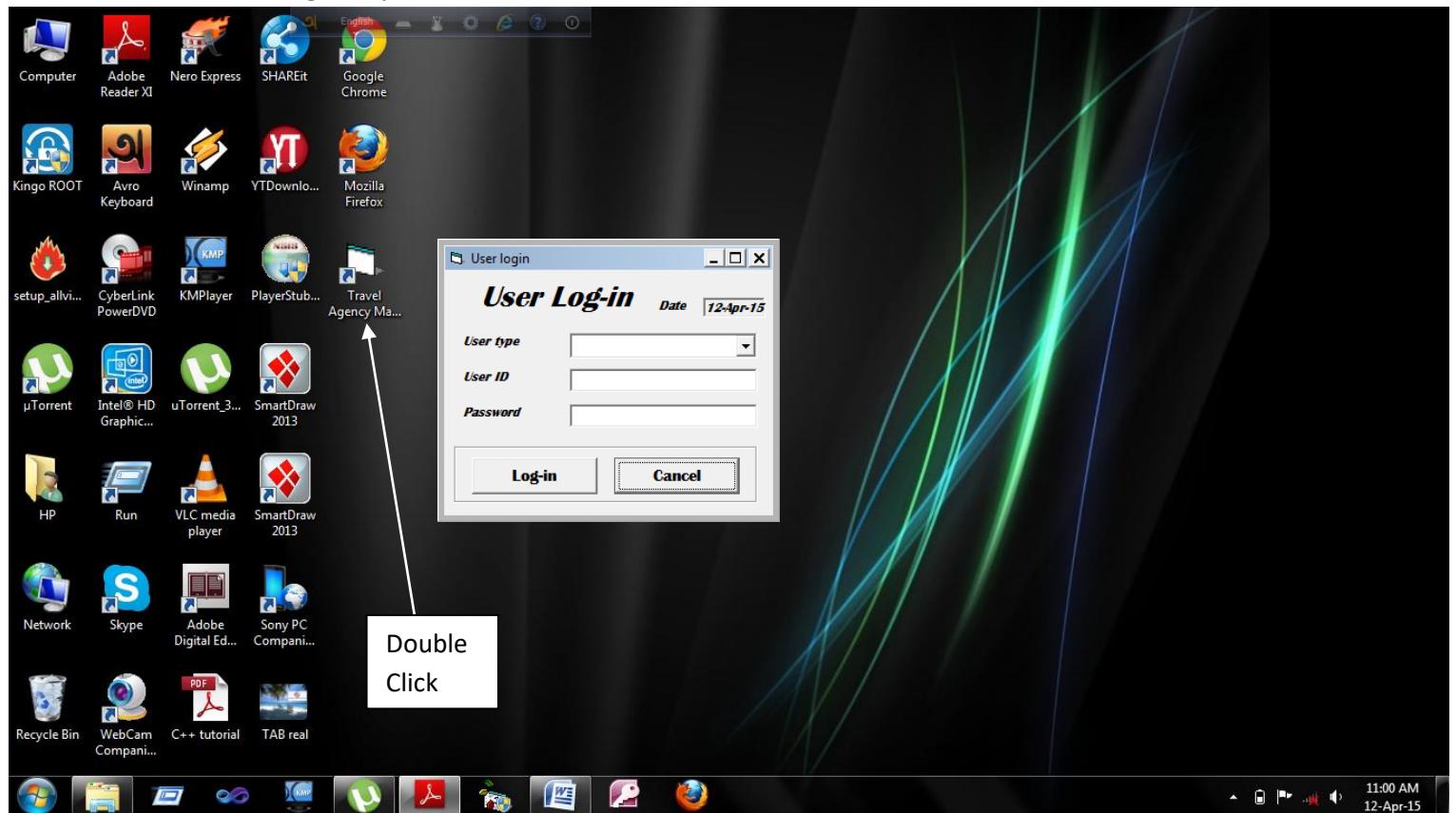
12.A progress bar is shown



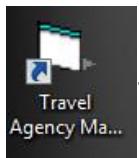
13. The installation is finished when the message shown below appears.



14. Starting the system



The Log-in form appears when the following icon is clicked from the desktop window. The access to Main Menu form can only be gained if the correct User type, User ID and Password are provided.



"Travel Agency Management System.exe" will be shown in the desktop when a shortcut of setup file is created. Double click on the icon to begin program as shown above.

The system basically has two parts. A part that is designed to interact with us and the other part is used to store the data we provide to the system and allow the computer system to make calculations and monitor the things happening. The part that was made to interact with the user is designed using Visual Basic 6.0 and the other part is made using Ms Access. Programming techniques have been used to link the two parts to function efficiently i.e. a connection has been set up between the database and the data environment of Visual Basic 6.0.

Logging into the system

1. Enter the correct User type, User ID and Password.
2. Click Log-in
3. When the button is pressed, the Main Menu form will appear.

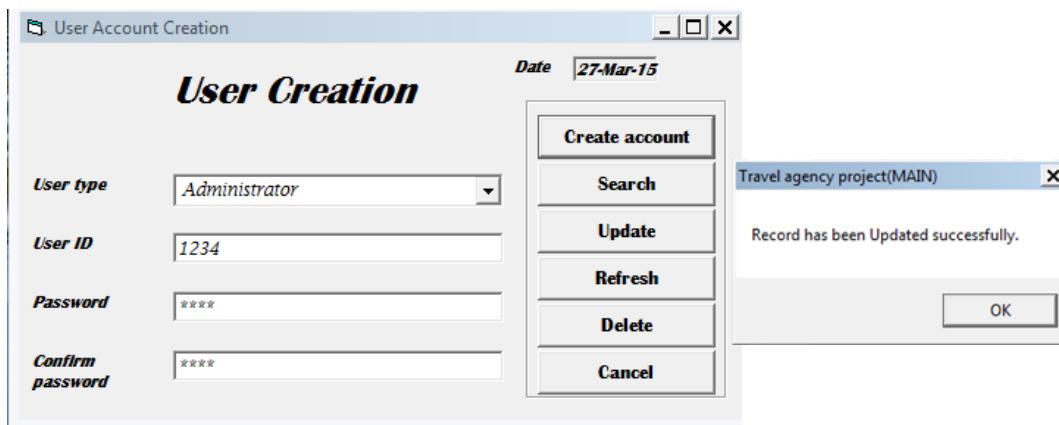


Adding a new User to the system

1. Select “User Creation” form from the “User information” section of the Main Menu.
2. Enter User type, User ID, Password and Confirm Password.
3. The Password and Confirm Password must match.
4. Click “Create Account” button.
5. A message will appear that the account has been created successfully.
6. Click “Ok”

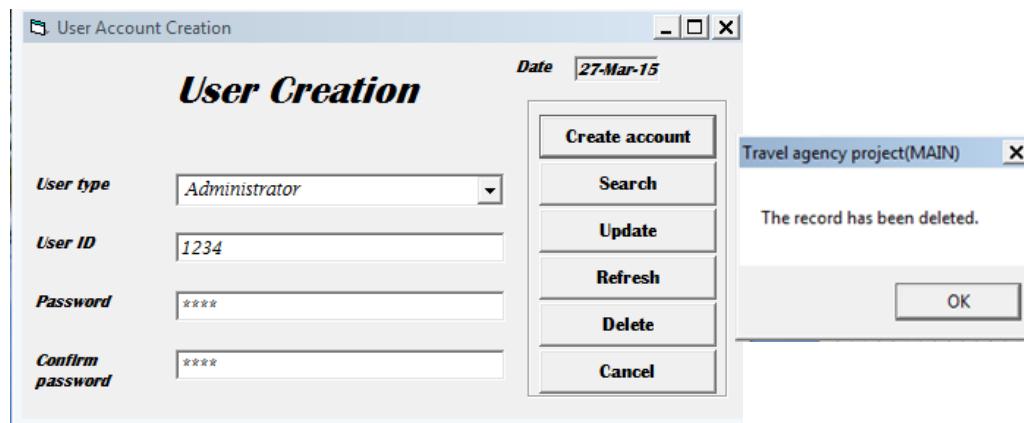
Changing the Password of an existing User

1. Select “User Creation” form from the “User information” section of the Main Menu.
2. Provide User type and User ID in the User creation form.
3. Click “Search”.
4. The previous Password and Confirm password will appear.
5. Change the Password and Confirm Password both. It should be made sure that both of them match.
6. Click “Update”.
7. Confirmation message shown on successful update of data.
8. Click “Ok”.



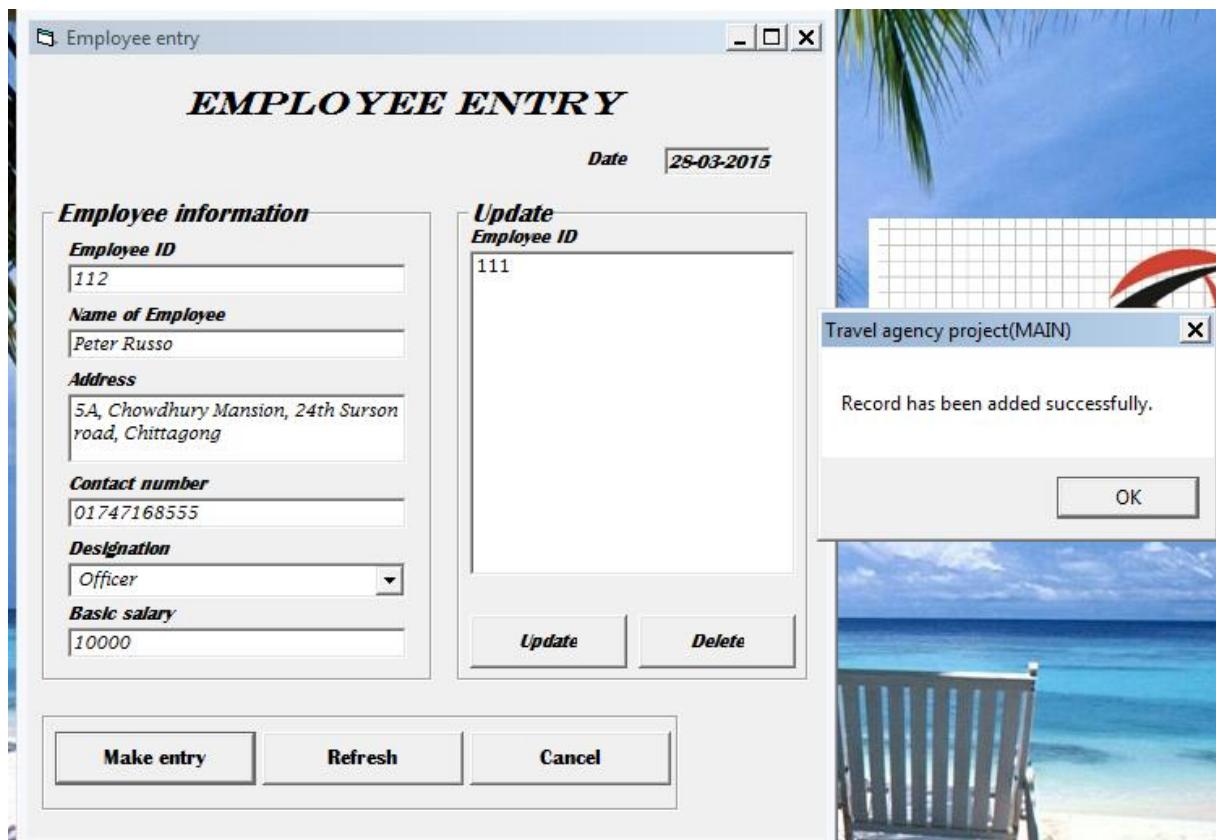
Deleting an existing User Account

1. Select “User Creation” from the “User information” section of the Main Menu.
2. Provide User type and User ID.
3. Click “Search”.
4. Click “Delete”.
5. A confirmation message will appear for the deletion of account.
6. Click “Ok”.



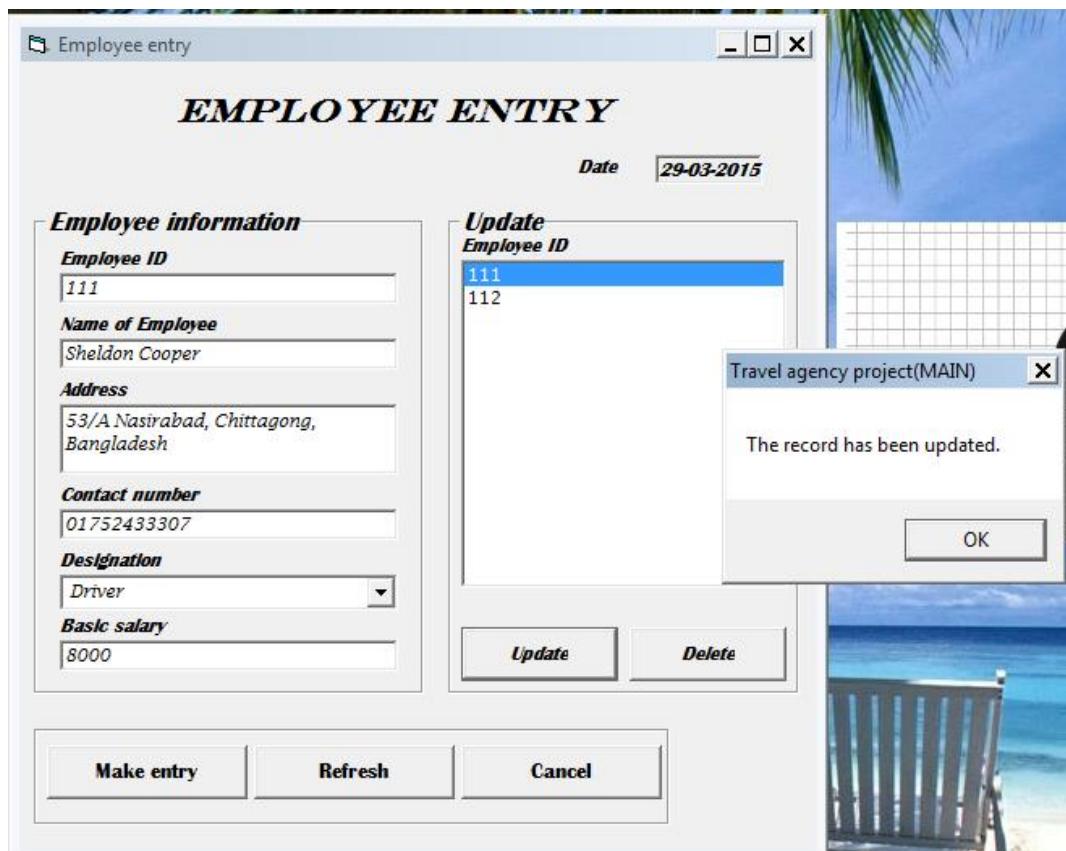
Adding a new Employee record

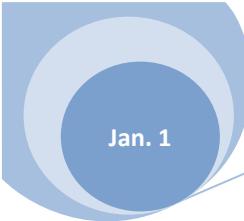
1. Select “Employee Entry” form from “Employee information” section of the Main Menu.
2. Enter all the Employee information.
3. Click “Make Entry”
4. Confirmation message appears for the successful addition of data.
5. Click “OK”.



Updating an existing Employee record

1. Select “Employee Entry” form from the “Employee information” section of the Main Menu.
2. Select an Employee ID from the “Employee ID list” of the Update section of the form.
3. Make changes to the information.
4. Click “Update”.
5. Confirmation message appears on successful update. Click “Ok”.





Jan. 1

Travel Agency Management System

Deleting an existing Employee record

1. Select an Employee ID from the Employee ID list of the Update section of the form.
2. Click “Delete”.
3. Confirmation message appears on successful deletion of the record.
4. Click “Ok”.

Adding a new Package

1. Select “Package Entry” form from the “Package information” section of the Main Menu.
2. Provide Package information, Number of reservations, itinerary and Package Details.
3. Click “Save”.
4. Confirmation message appears on successful addition of the record.
5. Click “Ok”.

Package entry

Package Entry Date 29-03-2015

Package information

Package type	Multi-day
Name of package	River run in Venice
Package ID	1124
Cost per person	25000
Accommodation type	Normal
Journey Date	12-Feb-16
Arrival time	10:00:00 AM
Starting time	11:00:00 AM
Duration	3 Days

Number of reservations available? 300

Itinerary

12/2/16 Depart Zia Airport - Emirates
13/2/16 Meet and Greet on arrival and transfer to hotels
14/2/16 Sail through Venice

Package details

2 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options

Update

Package type	
Package Name	List

Travel agency project(MAIN) Record has been added successfully OK

Buttons

Save Refresh Update Delete Cancel

Updating a Package

1. Select “Package Entry” form from the “Package information” section of the Main Menu.
2. Select the “Package type” from the “Update” section of the form.
3. List of Package names is loaded.
4. Select the Package name.
5. The fields on the form are filled.
6. Make required changes.
7. Click “Update”.
8. Confirmation message appears on update of information.
9. Click “Ok”

Package entry

Package Entry Date 29-03-2015

Package information

Package type	Multi-day
Name of package	Nights in Madagascar
Package ID	1120
Cost per person	20000
Accommodation type	Deluxe
Journey Date	12-Feb-16
Arrival time	10:00:00 AM
Starting time	11:00:00 PM
Duration	3 Days

Number of reservations available? 500

Itinerary

12/2/16 Depart Zia Airport - Emirates
 13/2/16 Meet and Greet on arrival and transfer to hotels
 14/2/16 African Zoo - Outback Africa Dinner show

Package details

4 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options

Update

Package type Multi-day

Package Name list

- Wind of Brisbane
- Nights in Madagascar
- Sailing in Paris
- River run in Venice

Travel agency project(MAIN)

Package information has been updated successfully.

OK

Save Refresh Update Delete Cancel

Deleting a Package

1. Select “Package Entry” form from the “Package information” section of the Main Menu.
2. Select a “Package type” from the “Update” section.
3. List of Package names is loaded. Select a Package name.
4. Package information for the Package is loaded.
5. Click “Delete”.
6. Confirmation message appears on successful deletion of the record.
7. Click “Ok”.

Package entry

Package Entry Date **29-03-2015**

Package information		Update
Package type	Multi-day	Package type
Name of package	Nights in Madagascar	Package Name list
Package ID	1120	Wind of Brisbane
Cost per person	15000	Nights in Madagascar
Accommodation type	Deluxe	Sailing in Paris
Journey Date	12-Feb-16	River run in Venice
Arrival time	10:00:00 AM	
Starting time	11:00:00 PM	
Duration	3 Days	
Number of reservations available? 500		
Itinerary		
12/2/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/2/16 African Zoo - Outback Africa Dinner show		
Package details		
4 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options		
<input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/>		

Travel agency project(MAIN)

Record has been deleted successfully.

OK

Booking a Package

1. Select “Package Booking” form from the “Package information” section Main Menu form.
2. Select a Package type. List of Package names is loaded. Select a Package name.
3. Check if there are any vacancies for the Package. Enter Number of People. Net total is automatically calculated.
4. Enter Customer information.
5. Click “Confirm” button.
6. Confirmation message appears on successful booking made.
7. Click “Ok”.

Package booking

Package Booking Date: 30-03-2015

Package Specifications		Information on Package	Customer Details
Package Type <input type="button" value="Multi-day"/> Package name <input type="button" value="Wind of Brisbane"/> <i>Sailing in Paris</i> <i>River run in Venice</i>		Itinerary <i>12/2/16 Depart Zia Airport - Emirates</i> <i>13/2/16 Meet and Greet on arrival and transfer to hotels</i> <i>14/2/16 Australia Zoo - Outback Australia Dinner show</i> <i>15/2/16 Day of leisure</i>	Booking ID <input type="text" value="1240"/> Customer Name <input type="text" value="Chris Gellar"/> Address <input type="text" value="25th Wolwin Heights Chittagong"/>
Package Details <input type="text" value="3 star hotel for 3 night Breakfast, Internatio Flights, Transfers, To and all options"/>		Travel agency project(MAIN) <input type="button" value="X"/> <i>Booking has been made successfully.</i> <input type="button" value="OK"/>	
PackageID <input type="text" value="1122"/> Cost per person <input type="text" value="5000"/> Number of people <input type="text" value="3"/> Net total <input type="text" value="15000"/>		Accommodation type <input type="text" value="Normal"/> Journey Date <input type="text" value="12-Feb-16"/> Arrival time <input type="text" value="10:00:00 AM"/> Starting time <input type="text" value="11:00:00 PM"/> Duration Number of Vacancies <input type="text" value="3"/> Days <input type="text" value="491"/>	<input type="button" value="Confirm"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>

Updating a Package

1. Select “Package Booking Update” form from the “Package information” section of the Main Menu form.
2. Provide Booking ID and click “Search” button.
3. The fields on the form are filled.
4. Make required changes.
5. Click “Update” button.
6. Confirmation message appears on successful update of record.
7. Click “Ok”.

Package Booking Update

Date

Customer Details		Information on Package	Action Buttons
Booking ID <input type="text" value="1238"/> Customer Name <input type="text" value="Tyler Crowing"/> Address <input type="text" value="25th, Bashundhara street, Chittagong"/> Contact number <input type="text" value="01819625123"/>		Itinerary 12/3/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/3/16 Eiffel tower - Outback Paris Dinner show 15/3/16 Day of leisure	<input type="button" value="Search"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Refresh"/>
Package Specifications		Package Details 3 star Travel agency project(MAIN) Booking information has been Updated successfully. <input type="button" value="OK"/>	
Date Booking made <input type="text" value="30-Mar-15"/> PackageType <input type="text" value="Multi-day"/> Package name <input type="text" value="Sailing in Paris"/> Package ID <input type="text" value="1123"/> Cost per person <input type="text" value="30000"/> Number of people <input type="text" value="5"/> Net total <input type="text" value="150000"/>		Accommodation type <input type="text" value="Deluxe"/> Journey Date <input type="text" value="12-Mar-16"/> Arrival time <input type="text" value="10:00:00 AM"/> Starting time <input type="text" value="11:00:00 PM"/> Duration Number of vacances <input type="text" value="3"/> Days <input type="text" value="495"/>	

Deleting a Booking record

1. Select “Package Booking Update” form from the “Package information” section of the Main Menu.
2. Provide Booking ID of the record to be deleted.
3. Click “Search” button.
4. All the fields on the form are filled.
5. Click “Delete” button.
6. Confirmation message appears on successful deletion of record.
7. Click “Ok”.

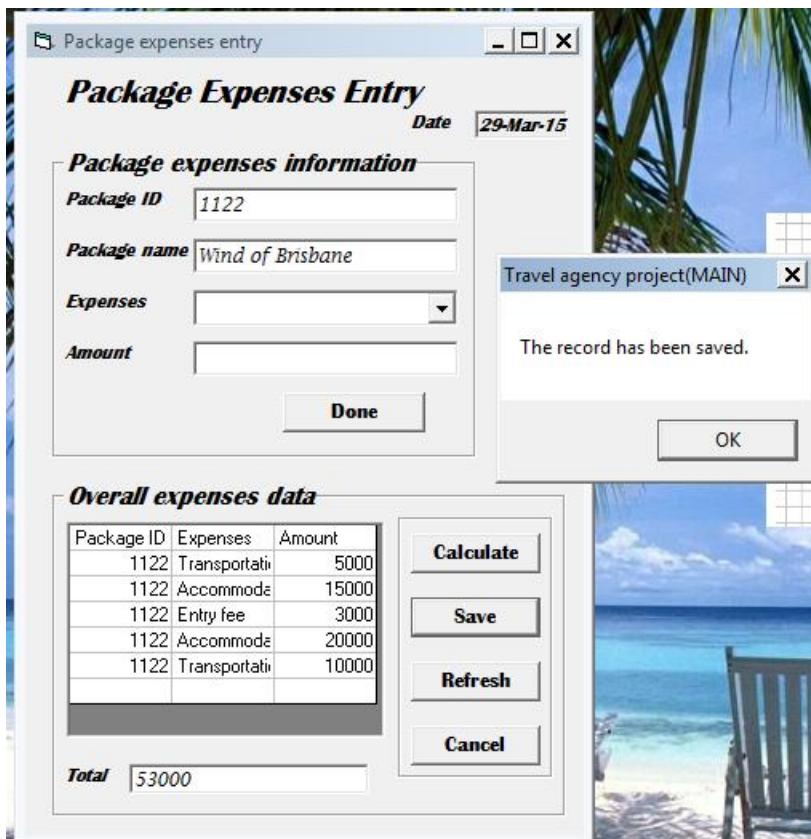
Package Booking Update

Date 30-03-2015

Customer Details		Information on Package	Action Buttons
Booking ID <input type="text" value="1238"/> Customer Name <input type="text" value="Tyler Crowing"/> Address <input type="text" value="24th, Bashundhara street, Chittagong"/> Contact number <input type="text" value="01819625123"/>		Itinerary 12/2/16 Depart Zia Airport - Emirates 13/2/16 Meet and Greet on arrival and transfer to hotels 14/2/16 Australia Zoo - Outback Australia Dinner show 15/2/16 Day of leisure	<input type="button" value="Search"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> (highlighted) <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>
Package Specifications		Package Details 3 star hotel for 3 nights with Breakfast, International Flights, Transfers, Tour host and all options	
Date Booking made <input type="text" value="30-Mar-15"/> PackageType <input type="text" value="Multi-day"/> Package name <input type="text" value="Sailing in Paris"/> Package ID <input type="text" value="1123"/> Cost per person <input type="text" value="30000"/> Number of people <input type="text" value="5"/> Net total <input type="text" value="150000"/>		Accommodation type Normal Journey Date <input type="text" value="12-Feb-16"/> Arrival time <input type="text" value="10:00:00 AM"/> Starting time <input type="text" value="11:00:00 PM"/> Duration 3 Days Number of vacancies <input type="text"/>	<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> Travel agency project(MAIN) X </div> <p>The Booking has been deleted.</p> <input type="button" value="OK"/>

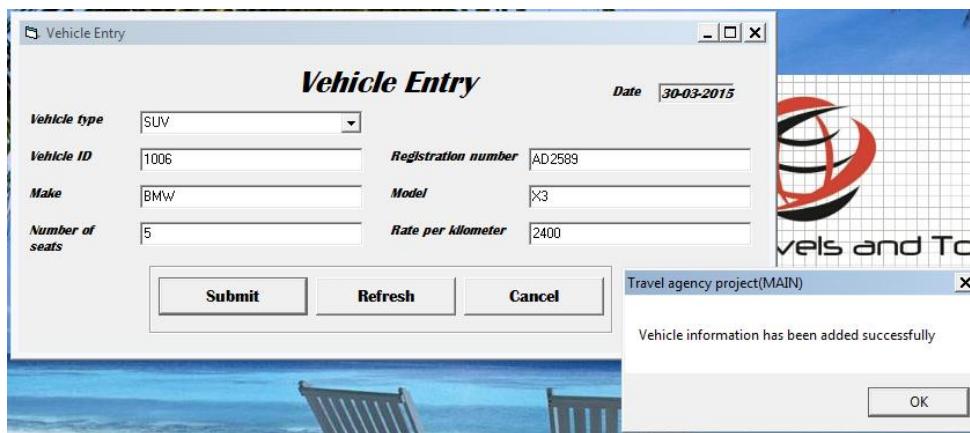
Recording Package Expenses information

1. Select “Package Expenses Entry” form from the “Package information” section of the Main Menu.
2. Provide Package ID, Package name, Expenses and Amount.
3. Click “Done”. The record is added to the table in “Overall Expenses data” section of the form.
4. Repeat the process for the number of expenses present.
5. After all the expenses data have been provided, click “Calculate” button.
6. The total amount is calculated and total generated in the textbox.
7. Click “Save”.
8. Confirmation message appears for successful saving of data.
9. Click “Ok”



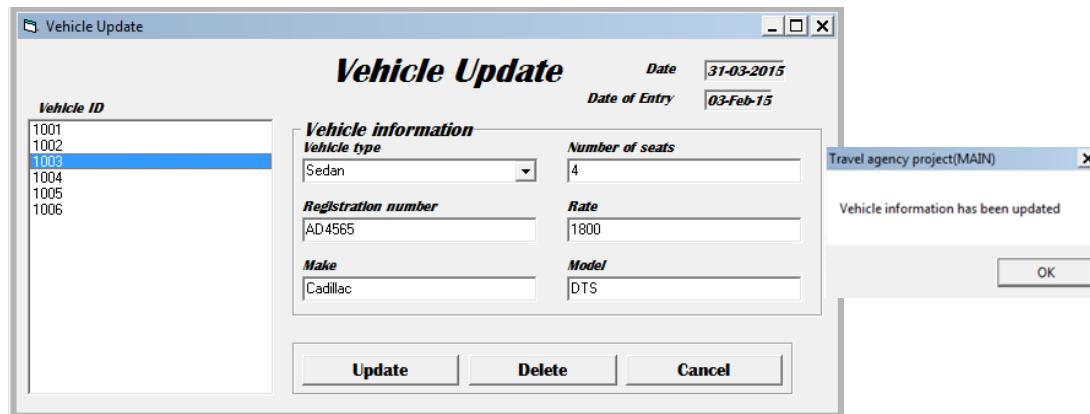
Adding a record for the information of a new Vehicle

1. Select “Vehicle Entry” form from the “Vehicle information” section of the Main Menu.
2. A Vehicle ID is generated.
3. Fill the form with all the Vehicle information.
4. Press “Submit” button
5. Confirmation message appears on addition of record to the database.
6. Click “Ok”.



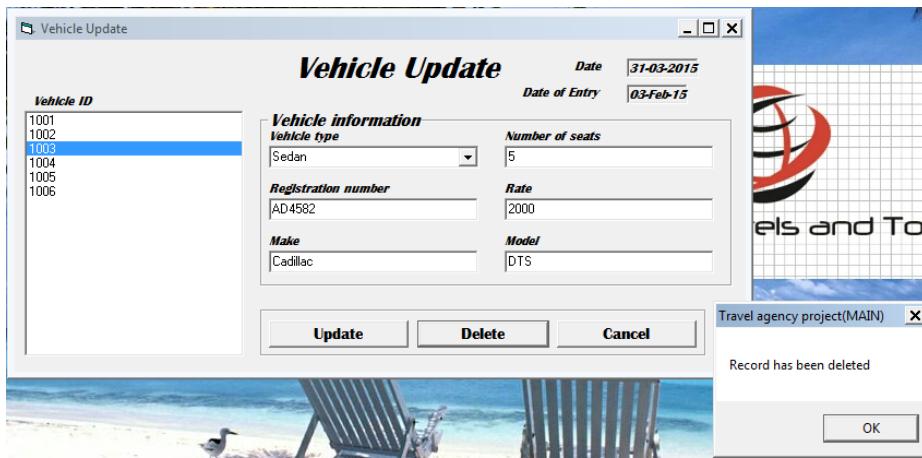
Updating the information of a record of a Vehicle

1. Select “Vehicle Update” form from the “Vehicle information” section of the Main Menu.
2. Select Vehicle ID from the Vehicle ID list. The form is filled.
3. Make required changes.
4. Click “Update” button.
5. Confirmation message appears on successful update.
6. Click “Ok”.



Deleting a record of information on a Vehicle

1. Select “Vehicle Update” form from the “Vehicle information” section of the Main Menu form.
2. Select the Vehicle ID of the record to be deleted from the Vehicle ID list.
3. Click Delete.
4. Confirmation message appears on successful deletion of the record.
5. Click “OK”.



Booking a Vehicle

1. Select “Vehicle Booking” form from the “Vehicle information” section of the Main Menu form.
2. “Vehicle Booking” form appears and a Booking ID is generated.
3. Fill up the Customer information section of the form and provide Journey date and Drop-off date.
4. Select Vehicle ID from the Vehicle information section. A list of Makes appears.
5. On selection of a Make, a list of Models appears. A list of Vehicle IDs appears on selection of a Model. Finally, Registration number, Rate per kilometer and Number of seats appears on selection of a Vehicle ID
6. Enter Starting kilometer, Advance paid and Driver name.
7. Click “Confirm” button.
8. Confirmation message appears on successful booking.
9. Click “Ok”

Vehicle Booking Date 31 - 03 - 2015

Customer information

Booking ID	1424	Address	23rd, Golpahar, Chittagong
Customer Name	Helen Greene		
Contact	01815585664		
Journey date	31-Mar-15	Drop off date	09-Apr-15

Vehicle information

Vehicle type	SUV	Registration number	AD2589
Make	BMW	VehicleID	1006
Model	X1 X3		
Rate per kilometer	2400	Advance paid	2500
Number of seats	5	Driver name	Dylan Walker
Starting kilometer	321354		

Confirmation Message: Booking has been made successfully.

Buttons: Confirm, Refresh, Cancel, OK

Updating information of an existing booking

1. Select “Vehicle Booking Update” form from the “Vehicle information” section of the Main Menu form.
2. Provide Booking ID. Click “Search” button.
3. The rest of the fields on the form are filled.
4. Make changes to the information.
5. Click “Update” button.
6. Confirmation message appears on successful update of the record.
7. Click “Ok”.

Vehicle Booking Update

Date **31-03-2015**

Customer information		Vehicle information		<input type="button" value="Search"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/>
Booking ID	1423	Vehicle type	Sedan	
Customer Name	Michelle Winger	Make	Toyota	
Address	23rd, Sholoshohor, Chittagong	Model	Vista	
Contact number	0182554558	VehicleID	1002	
Journey date	21-Feb-15	Registration number	AD8646	
Drop off date	04-Aug-15	Number of seats	5	
Booking date	23-Jan-15	Rate per kilometer	1400	
		Advance paid	2500	
		Driver name	Tywell Bittersman	
		Starting Kilometer	13213	

Travel agency project(MAIN)

Booking has been updated successfully.

Deleting an existing booking

1. Select “Vehicle Booking Update” form from the Vehicle information section of the Main Menu form.
2. Provide Booking ID. Click “Search” button.
3. Click “Delete” button.
4. Confirmation message appears on successful deletion.
5. Click “Ok”.

The screenshot shows the 'Vehicle Booking Update' window. On the left, under 'Customer information', fields include: Booking ID (1424), Customer Name (Helen Greene), Address (23rd, Golpahar, Chittagong), Contact number (01815585664), Journey date (31-Mar-15), Drop off date (09-Apr-15), and Booking date (31-Mar-15). On the right, under 'Vehicle information', fields include: Vehicle type (SUV), Make (BMW), Model (X3), VehicleID (1006), Registration number (AD2589), Number of seats (5), Rate per kilometer (2400), Advance paid (2500), Driver name (Dylan Walker), and Starting Kilometer (321354). A vertical toolbar on the right contains buttons for Search, Update, Delete, Refresh, and Cancel. The 'Delete' button is highlighted. A confirmation dialog box titled 'Travel agency project(MAIN)' says 'Booking has been deleted.' with an 'OK' button. The background features a blue sky and palm trees.

Billing a Vehicle Rental

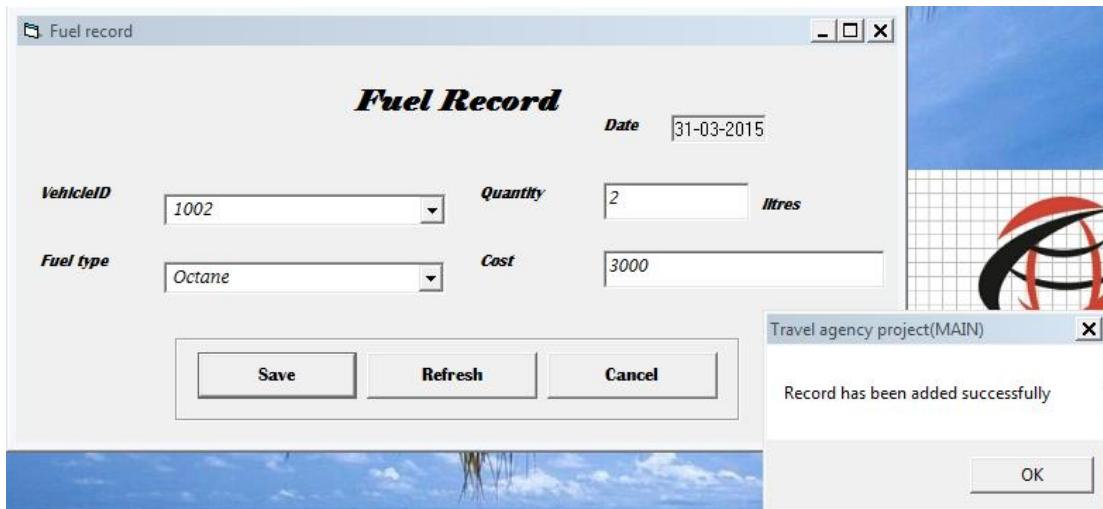
1. Select the “Vehicle Billing” form from the “Vehicle information” section of the Main Menu form.
2. Provide Booking ID. Click “Search” button.
3. Provide “Ending kilometer”. The “Cost of travel” is calculated automatically.
4. Provide Booking charge, Driver charge, Additional Expenditures and Discount.
5. Click “Generate Bill” button.
6. Click “Save” button.
7. Click “Print” button.
8. Click “Refresh” button.

Vehicle Billing

Customer information	Billing Information	Billing
Booking ID: <input type="text" value="1423"/> Customer Name: <input type="text" value="Michelle Winger"/> Contact number: <input type="text" value="01825545581"/> Address: <input type="text" value="23rd, Sholoshohor, Chittagong"/>	Cost of travel Starting kilometer: <input type="text" value="13213"/> Ending kilometer: <input type="text" value="13220"/> Kilometer travelled: <input type="text" value="7"/> Rate per kilometer: <input type="text" value="1400"/> Cost of travel: <input type="text" value="9800"/>	Date: <input type="text" value="31-Mar-15"/> <input type="button" value="Search"/> <input type="button" value="Refresh"/> <input type="button" value="Cancel"/> Dates Booking Date: <input type="text" value="31-Mar-15"/> Journey date: <input type="text" value="21-Feb-15"/> Drop off date: <input type="text" value="29-Mar-15"/> Billing date: <input type="text" value="31-Mar-15"/> Late fine(if any): <input type="text" value="100"/>
Vehicle information		
Vehicle type: <input type="text" value="Sedan"/> Make: <input type="text" value="Toyota"/> Model: <input type="text" value="Vista"/> VehicleID: <input type="text" value="1002"/> Registration number: <input type="text" value="AD8646"/> Driver name: <input type="text" value="Tywell Bittersman"/> Number of seats: <input type="text" value="5"/>		Booking Charge: <input type="text" value="2000"/> Driver charge: <input type="text" value="4000"/> Cost of travel: <input type="text" value="9800"/> Late fine(if any): <input type="text" value="100"/> Additional Expenditures: <input type="text" value="2600"/> Gross total: <input type="text" value="18500"/> (Less) Discount(if any): <input type="text" value="500"/> Advance paid: <input type="text" value="2500"/> Net total: <input type="text" value="15500"/>
		<input type="button" value="Generate bill"/> <input type="button" value="Save"/> <input type="button" value="Print"/>

Addition of Fuel Record data

1. Select “Fuel Record” form from the “Vehicle information” section of the Main Menu form.
2. Enter Vehicle ID, Fuel type, Quantity and Amount.
3. Click “Save” button.
4. Confirmation message shown on successful addition of data to the database.
5. Click “Ok”.



Recording data on servicing of the Vehicles

1. Select “Vehicle Servicing” form from the “Vehicle information” section of the Main Menu form.
2. Select a Vehicle type. List of Vehicle IDs is loaded.
3. Select a Vehicle ID. Table of “Servicing History Preview” is loaded.
4. Enter “Servicing Details” and “Amount”.
5. Click “Save”.
6. The record is added to the table and the textboxes are emptied.

Vehicle Servicing

Vehicle type	SUV	Date	06-Apr-15												
Vehicle ID		Servicing details													
1004		Engine repairs													
1005															
1006															
		Amount	3000												
Save Refresh Cancel															
Servicing History Preview															
<table border="1"> <thead> <tr> <th>Date</th> <th>Servicing Details</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>06-Apr-15</td> <td>Painting</td> <td>500</td> </tr> <tr> <td>06-Apr-15</td> <td>New exhausts</td> <td>1500</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>				Date	Servicing Details	Amount	06-Apr-15	Painting	500	06-Apr-15	New exhausts	1500			
Date	Servicing Details	Amount													
06-Apr-15	Painting	500													
06-Apr-15	New exhausts	1500													

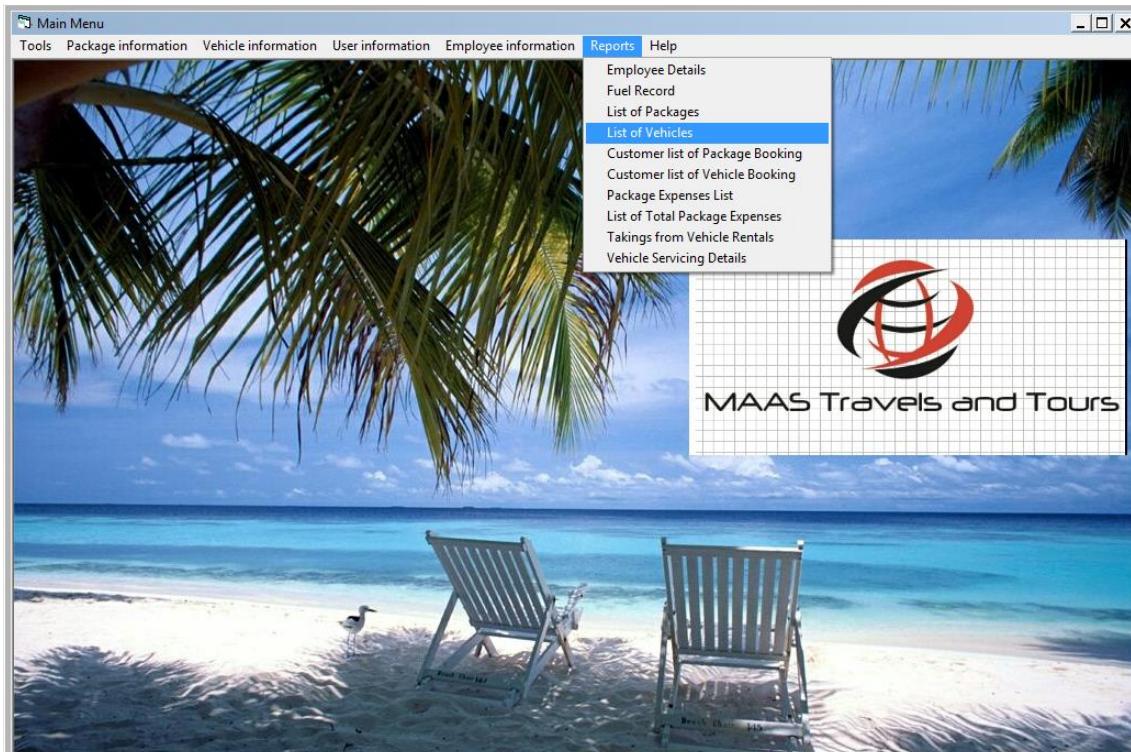
Vehicle Servicing

Vehicle type	SUV	Date	06-Apr-15															
Vehicle ID		Servicing details																
1004																		
1005																		
1006																		
		Amount																
Save Refresh Cancel																		
Servicing History Preview																		
<table border="1"> <thead> <tr> <th>Date</th> <th>Servicing Details</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>06-Apr-15</td> <td>Painting</td> <td>5</td> </tr> <tr> <td>06-Apr-15</td> <td>New exhausts</td> <td>15</td> </tr> <tr> <td>06-Apr-15</td> <td>Engine repairs</td> <td>30</td> </tr> <tr> <td><</td> <td></td> <td>></td> </tr> </tbody> </table>				Date	Servicing Details	Amount	06-Apr-15	Painting	5	06-Apr-15	New exhausts	15	06-Apr-15	Engine repairs	30	<		>
Date	Servicing Details	Amount																
06-Apr-15	Painting	5																
06-Apr-15	New exhausts	15																
06-Apr-15	Engine repairs	30																
<		>																

New record added to the table.

Generating Reports

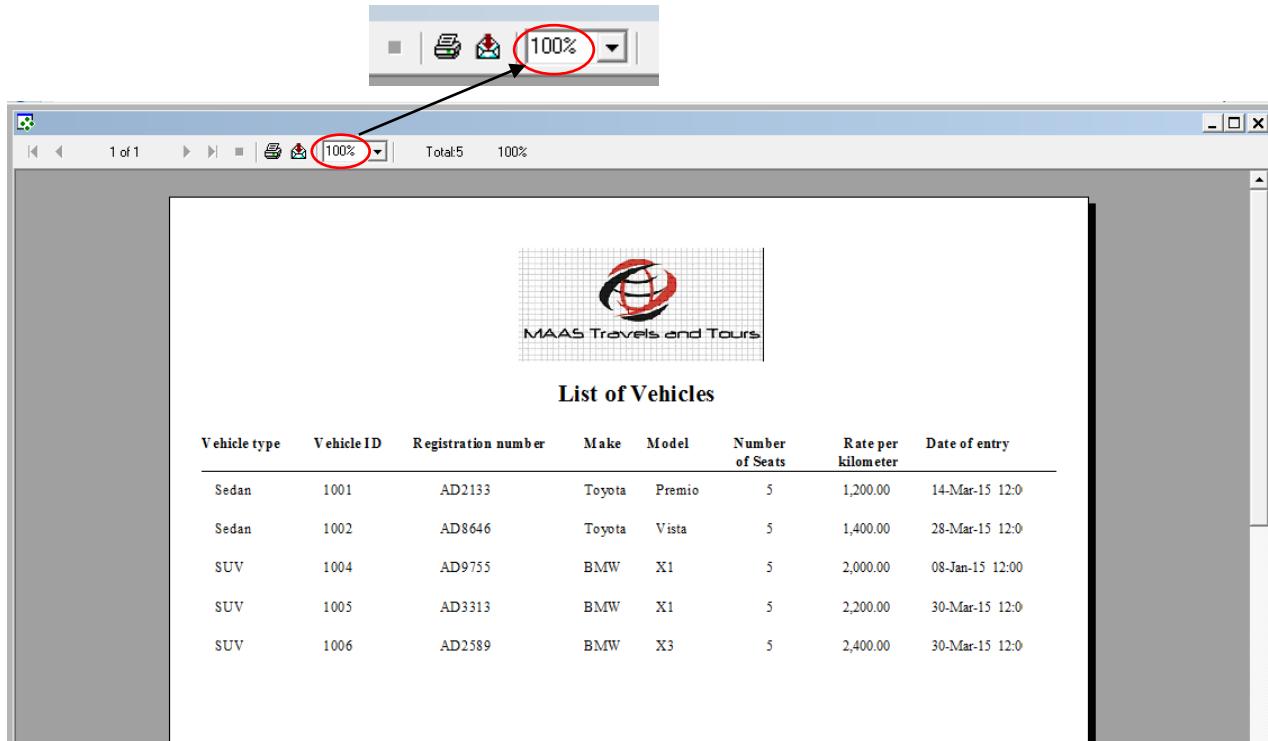
1. Select the name of the reports to be viewed from the “Reports” section of Main Menu form.
2. The report is generated.

A screenshot of a report window titled "List of Vehicles". The window has a header section with the "MAAS Travels and Tours" logo. Below the header is a table with the following data:

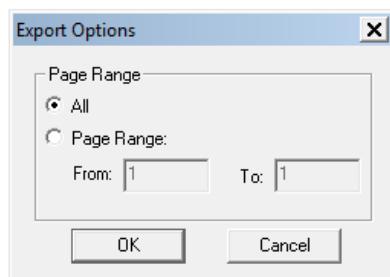
Vehicle type	Vehicle ID	Registration number	Make	Model	Number of Seats	Rate per kilometer	Date of entry
Sedan	1001	AD2133	Toyota	Premio	5	1,200.00	14-Mar-15 12:0
Sedan	1002	AD8646	Toyota	Vista	5	1,400.00	28-Mar-15 12:0
SUV	1004	AD9755	BMW	X1	5	2,000.00	08-Jan-15 12:0
SUV	1005	AD3313	BMW	X1	5	2,200.00	30-Mar-15 12:0
SUV	1006	AD2589	BMW	X3	5	2,400.00	30-Mar-15 12:0

Printing Reports

1. Select the report to be printed from the “Reports” section of the Main Menu form.
2. Click on the button highlighted below.



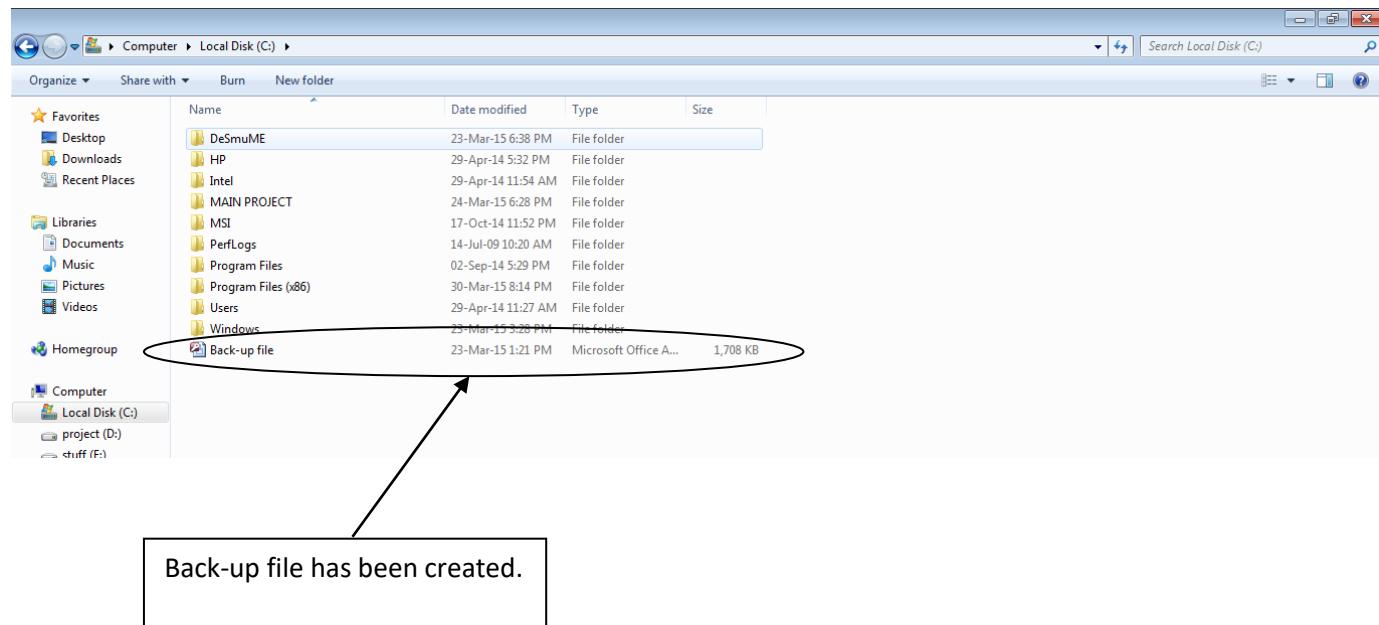
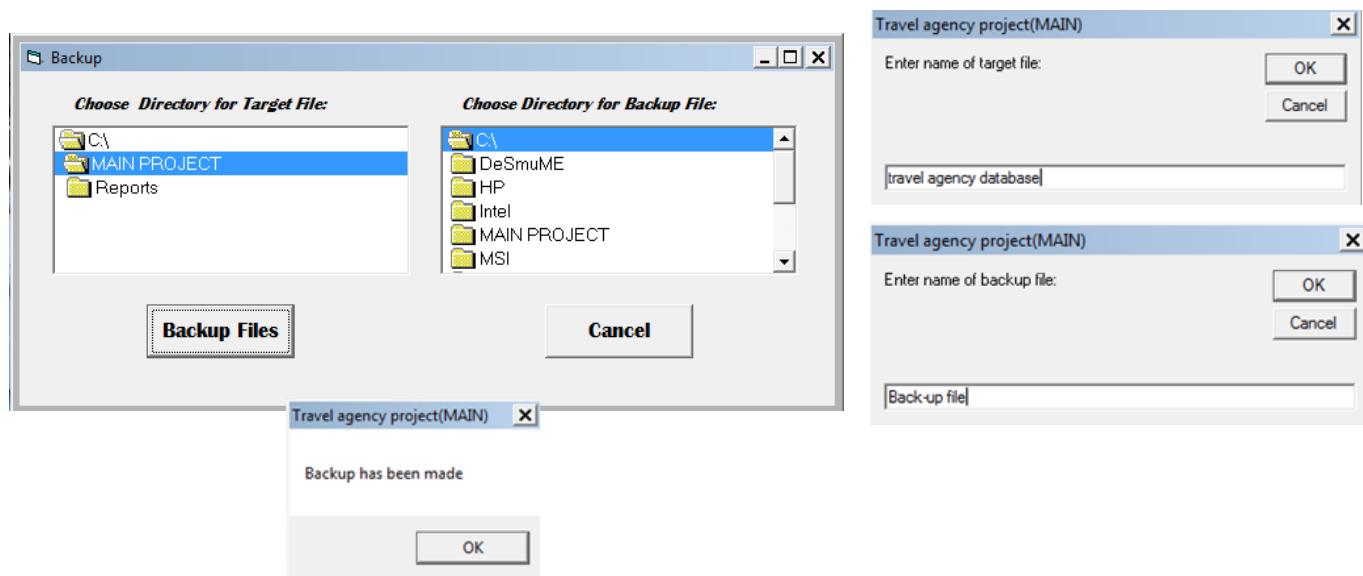
3. The following window appears. Enter number of pages that need to be printed.



4. Click "Ok" to print the report.

Creating Backup

1. Select “Backup” form from the “Tools” section of the Main Menu.
2. Select directory of target file and the directory of backup file.
3. Click “Backup files” button.
4. A window appears asking for name of target file. Enter name of target folder and click “Ok”.
5. Another window appears asking for name of backup file. Enter name of Backup file and click “Ok”.
6. Confirmation message appears on successful creation of backup file.



The whole system is linked with Ms Access. All kinds of processes (such as adding, searching, editing, deleting etc.) require Ms Access to work on the background. Therefore, it is an essential part for the functioning of the whole system.

➤ Security Measures

Since the program will include important data, it becomes crucial to take measures to keep the data safe. Special care must be taken to make sure anyone who is not authorized does not get access to the system.

Physical protection: The computer must be shut down and locked away when it is not being used. When data is carried through flash drives they should be kept safe and measures should be taken so that they are not lost. In case portable computers (such as laptops) are being used, they should be kept safe and out of reach of other people as it is easy to snatch them and run away.

Backing up: The database information should be backed up periodically and frequently (as transactions are made on an hourly basis) to prevent loss of data and to compensate for accidental loss of data due to unforeseen reasons. It is also advisable to move the backed up information to a new storage medium regularly in case a computer system crashes. Furthermore, it is also advisable to keep a different storage media for the storage of files which could be kept away from the agency venue to allow retrieval of data in case of any unforeseen hazard (such as a fire).

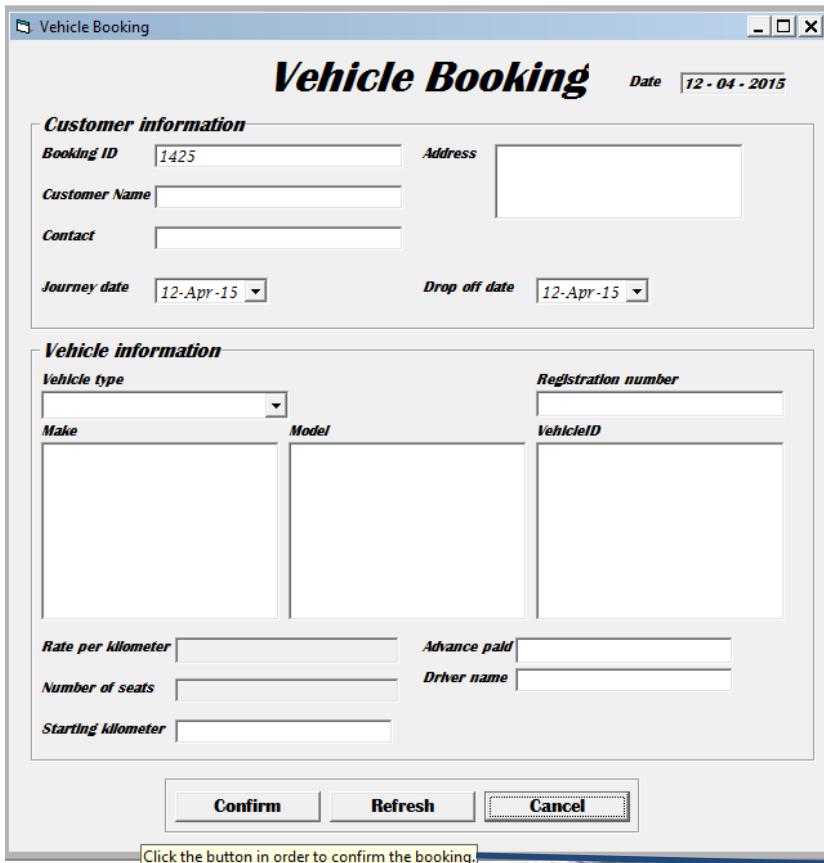
Virus protection: A licensed antivirus should be installed to the system which needs to be updated periodically in order to protect files from getting infected by a virus. An infected file may be corrupted which will result in loss of data. With an antivirus comes a strong firewall that will prevent illegal access to the system and protect the files in the system.

Password and encryption: Depending on the version being used, it is possible to set a windows password (for further information, see Windows “Help”). This is highly recommended as it prevents unknown people from accessing user account of the operating system. My system is already password protected for access to only users of the system. No one can use my system without a proper User type, User ID and Password.

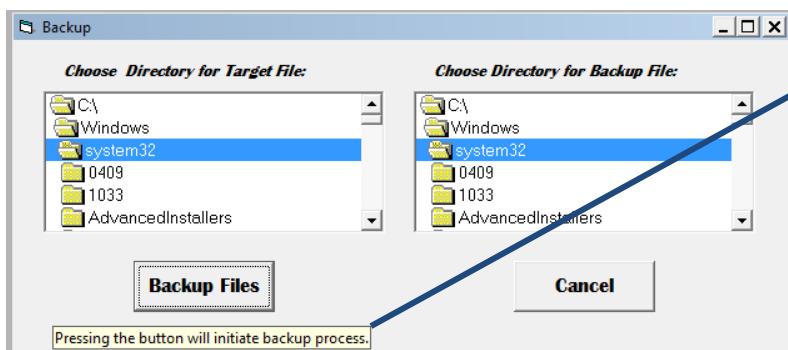
The data can also be encrypted so that no one can read the data even at gaining illegal access.

✓ **On-screen help examples**

Throughout the system, on-screen help is available. Whenever the mouse pointer hovers over a control, a message appears notifying the user of the function of the control.



When the mouse pointer is paused over the buttons, a text appears to explain the function of button.



Tool-tip texts are provided throughout the system. Moreover, the customer can refer to the "Help" option in the Main Menu form in case they forget the next step to be taken in the middle of a process.

- **Troubleshooting**

This section is a guide to what should be done in case any part of the system is not working as well as it should. It should help encounter small problems. However, any further problem faced requires the council of the programmer.

What should I do if my report does not print?

- ✓ *Thoroughly check if the printer is properly installed to the computer properly.*
- ✓ *If the printer is installed, check if the printer is turned on and if there are papers in the printer. Ensure proper connection of the printer to the computer.*
- ✓ *If the above does not work, reinstall the printer and try again.*

What should I do if there is no action even when I press a button on the form?

- ✓ *It is possible that the system has crashed. It is advisable to restart the computer and try again.*

What should I do if I am not able to gain access to the system due to wrong password?

- ✓ *It is advisable to check the password provided thoroughly. Sometimes access may not be granted due to case sensitivity, or leaving “Caps Lock” activated by accident.*

I have installed the system but clicking on the icon on the screen does not start the system. What should I do?

- ✓ *Probably the system was not installed properly. It is advisable to uninstall the system (from the “Programs and features” option of the control panel) and reinstall it again.*

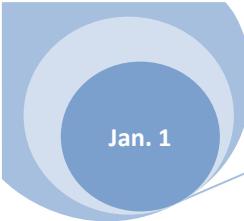
Pressing the power button is not turning on my computer. What should I do?

- ✓ *Check if the computer is connected to the power supply properly. If it is connected properly and still the computer does not start, the user should consult with a technician to examine the power box of the System Unit.*

➤ Glossary of terms

- Antivirus: Software designed to detect and deal with virus malware.
- ASCII: American Standard Code of Information Interchange.
- ASCII value: these are standard recognized ASCII values of different characters. For example, 'A' has an ASCII value of 65, which is taken as a standard identification of 'A' by computers. That is, when the computer receives an ASCII value of 65, it deems it as 'A'.
- Backup: Copy of data stored for security reasons and used in the event of loss or corruption of live data.
- Database: Collection of related data. A complete set of data is called a record, while the different types of data present in the tables are called fields.
- Encryption: Process whereby a message when transmitted can only be understood by sender and receiver. The coded format cannot be understood by anyone else who reads it. For example, the number "1121" can be written as "3343", where each number is incremented by 1. The latter is the encrypted version of the former code.
- Flash drive: A type of portable storage media (also known as pen drive or memory stick).
- Malware: Software that has been designed for mischievous or criminal purposes; it might slow down the system or cause deletion of files etc.
- Microsoft Access: database software that has been used to develop the system. It has a wide range of features that makes it efficient database software. It is compatible with Visual basic 6.0. It was made by the Microsoft Company.
- Reports: Hard or soft copy outputs-usually providing summary information.
- Tables: One of the many collections of data in form of fields and records which might be present in a database.
- Tool tip text: A function in Visual Basic that allows a text to be displayed when the mouse is paused over a button.
- Validation check: The data input to the system are checked for validity through a number of tests. These include length checks, presence checks and character type checks.
- Virus: Malicious self-replicating software that can harm the computer system.
- Visual Basic 6.0: An event driven programming language which has been used for the development of this program.

A	Page
Adding a new user to the system	7



Jan. 1

Travel Agency Management System

(e)

Evaluation

*(i) Discussion of the degree of success in meeting the original objectives***❖ Objectives and description of solution**

1. System should be password protected and allow access to the user only when correct User type, User ID and Password are provided. (Evidence test number-1,2,3,5)
2. The system should allow the user to change his/her password as required, only on provision of correct User type and User ID. (Evidence test number-21)
3. Validation checks to make sure incorrect data cannot be entered into the database and key fields (such as Booking ID, Employee ID, Package ID etc.) cannot be changed. (Evidence test number-12,18,19,34,72,102)
4. The system must allow new User accounts to be added, edited and deleted easily. (Evidence test number-13,21,24)
5. The system should allow Packages to be added, edited and deleted easily. (Evidence test number-43,47,49)
6. The system should allow Vehicle details to be added, edited and deleted easily. (Evidence test number-83,89,90)
7. The system should allow Employee records to be added, edited and deleted easily. (Evidence test number-31,35,36)
8. Packages and vehicles can be booked easily. (Evidence test number-65,96)
9. The system must provide a way to change booking information when required. (Evidence test number-77,103)
10. The system must provide a way to cancel a booking if required. (Evidence test number-78,104)
11. All arithmetical calculations should be done automatically in the forms to reduce mistakes. (Evidence test number-54,62,76,106,107)
12. The entry and booking forms must generate IDs automatically to prevent overwrite. (Evidence test number-26,39,80,83,96,)
13. ID checks should be provided for all kinds of IDs in all forms so that the same ID cannot be used twice. (Evidence test number-11,30,)
14. A list of the names or packages should be present in certain forms (such as Package Entry, Vehicle Entry, Vehicle Update, Vehicle Servicing and Employee Entry etc.) which should be updated after every booking. This allows the system administrator to keep the options provided by the agency in supervision. Also, the he can choose directly from the list when he/she wants to edit or delete a particular record. All the information on the form appears immediately on selection. (Evidence test number-26,32,35,36,39,43,45,47,49,67,85,110)
15. Easy to use search system that only requires the user to provide the Booking ID only when it comes to searching for bookings. The system can be used when the customer wants to edit or delete any booking information or just wants to check the information. (Evidence test number-16,70,101,105)
16. User interactive forms that show options according to customer choice and react to customer's choices to maintain security. (Evidence test number-4, 27,32, 41,45, 51,58,59, 60, 71,76, 82,86, 87, 92,106,110)
17. If a vehicle is already booked, it is automatically removed from booking the forms. It automatically reappears when job is done or the booking is deleted. Therefore the system automatically keeps track of the availability of vehicles. The vehicles are made unavailable on booking, and made available again on cancellation of a booking, when a customer changes his choice of vehicle (where one vehicle is made available while another one is made unavailable) or after the job is done on billing. The availability options of vehicles are updated after every transaction. (Evidence test number-94,96,103,104,108)

18. The system should automatically check for the number of vacancies available for a package before every booking. The customer should be informed immediately if there are no vacancies for a particular package. The vacancy check information should be updated automatically after every booking transaction made (such as a new booking or cancellation of a booking or update of a booking). (Evidence test number-64,65,75,77,78)
19. A form for Vehicle Billing that will automatically load all required information from Vehicle Booking database on the customer and the services provided to him/her on provision of their Booking ID. The rest of the information needs to be provided manually and then all the calculations should be done automatically on bill generation. (Evidence test number-105,106,107,108,109)
20. A Package Expenses Entry form needs to be made to keep track of the expenses on the package. The package expenses details are stored in two different tables, one containing individual expenses details and another containing total expense on a particular package. (Evidence test number-50,53,55)
21. A Vehicle servicing form needs to be made to keep track of information on servicing and status of vehicle. The form will servicing information for a particular vehicle on selection from the Vehicle ID list and any new information will be added to the list and to the database on addition of the record. (Evidence test number-110,111)
22. The system contains a “Fuel Records” form that stores data to a Fuel record table to provide a way to store fuel records data. (Evidence test number-112)
23. Presence checks to make sure all the information are provided before information is being saved, searched, updated or deleted on all firms. (Evidence test number-6,7,8,10,14,17,22,23,28,33,40,48,52,61,68,73,81,88,89,93,99)
24. Character type checks to make sure non-numerical data cannot be input in fields such as Cost per person, Net total and Contact number and so on. (Evidence test number-27,41,51,60,71,82,87,92)
25. Length checks to make sure there are 11 digits provided for contact number. (Evidence test number-29,63,74,95)
26. A “Refresh” button present in all forms to allow the user to clear the forms immediately as they wish. This allows users to start anew. (Evidence test number-10,22,37,44,46,56,66,79,84,88,97)
27. Generate receipt for Package booking. (Evidence test number-65)
28. Generate token for Vehicle booking. (Evidence test number-96)
29. Generate bill for Vehicle Billing. (Evidence test number-109)
30. Generate reports. (Evidence test number-114)
31. Provide User manual and option to back-up. (Evidence test number-113,115)

*(ii) Evaluation of the client's and user's response to the system***Letter from the Administrative Head of MAAS Travels and Tours:**

MAAS Travels and Tours
852, Green Line Road,
GEC, Chittagong



April 3, 2015

Priontu Chowdhury

53, Nasirabad I/A, Chittagong, Bangladesh

Dear Priontu Chowdhury,

We are writing to inform you that MAAS Travels and Tours has been using the system that you have developed and is very pleased with it. The system has really changed the way things were here, for the better. The officers are quite comfortable with the way everything is working. The calculations are swift and the information stored are accurate and up-to-date.

The system is running smoothly here at MAAS Travels and Tours. The increased customer satisfaction is noticeable in the everyday transactions, and we are deeply thankful to you for your contribution in it.

This is a thank you letter from the company. We have not yet found any major errors in the system; it quite fully meets the objectives that it was meant to accomplish.

Thanking you,

A handwritten signature in black ink, appearing to read "Abdullah Sheikh".

Mr. Abdullah Sheikh
Administrative Head
MAAS Travels and Tours

Merits and demerits of the new system:

Merits:

- It is labor intensive as less monitoring is required for every single detail or record. Therefore workers can be assigned to do more productive activities.
- It is cost-efficient. It is less costly as daily expenses for required materials, as well as cost of paper are reduced.
- Productive efficiency is enhanced by using this system as all jobs are speeded up and each worker can finish more work for the same amount of time used for the manual system.
- Higher customer and staff satisfaction as bookings, billings and other entries can be made much faster.
- Addition, deletion, updates and searching of data is much faster and easier with the new system.
- Saves up valuable space (such as space for piling of paper, drawer, boxes, cartons for transferring paper etc.) which can be used more productively.
- Allows backing up of data, which was too difficult to implement by using the manual system.
- Calculation of data is done automatically and the computer system does not make any arithmetical mistakes which the manual system is frequently prone to.
- Human errors are reduced (such as forgetting to record the availability of a vehicle or checking the limit of vacancy), so no booking can be done twice or exceed limits.
- Navigation through forms is easier and reports can be generated easily and automatically without employee intervention, which reduces human error a great deal.
- Bills, receipts and tokens can be printed automatically.
- Validation checks further reduce the human errors possible by the staff, such as entering non-numerical data for a numerical data type field, entering wrong number of digits for contact number etc.
- The implementation of the new system is more profitable as more work can be done with the same number of workers for the same time in a day and therefore, more bookings and billings can be made.

- The new system will allow the agency to keep up the pace with the other businesses and gain a respectable position in the competitive market.
- Allows the user to transfer data from one computer to another, which was not possible using the old system as most of the information was contained in papers, and it is hard to transfer piles of paper. Even though most of the job, such as making the reports, supervision of availability of vehicles and limits of vacancies, is done by the computer automatically on installation of the system, the documents can be transferred to home computer easily and safely by using memory sticks, portable hard drives etc. in case the data need to be revised for human errors.
- Allows the business to keep up with the modern world technology, which provides many perks, other than the ones already described.

Demerits:

- Most of the staff is computer illiterate, so it might take some time for the system to reach its full potential and work at peak efficiency.

(iii) Desirable Extensions:

- Online Booking system
- Hotline booking system
- Inclusion of credit card payment method
- Payroll system of staff and customers
- Automated Graphical representation of data.