**Cab Online Booking**

A Project Report

Submitted in partial fulfilment of the requirements for the Award of degree (BCA)

**2015–2018**

**Submitted by: Guided by:**

Mr. Mukul Khanna Mrs. Ruchika Sharma

Roll No - 65



# Directorate of Distance Education

**JaganNath University, Jaipur**

1

# Project Certificate

This is to certify that the project report entitled Cab Online Booking Submitted to **JaganNath University, Jaipur** in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF COMPUTER APPLICATIONS (BCA)**, is an original work carried out by Mr. Mukul Khanna, Enrolment No.: 65/GDIT/JIMS/2015, under the guidance of Mrs. Ruchika Sharma

The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfilment of the requirement of any course of study.

**Name of the student: Name of the Guide**

**MUKUL KHANNA Mrs. RUCHIKA SHARMA**

**Signature of the student Signature of The Guide**

**Enrolment No.: Date:**  **65/GDIT/JIMS/2015**

# ACKNOWLEDGEMENT

I would like to thanks to all my guides who really acted as pillars to help my way throughout this project that has led to successful and satisfactory completion of this study of this project.

Firstly, I would like to thanks my project in‐charge **Mrs.Ruchika Sharma** under whose guidance and motivation this work has been performed.

The inspiration of the faculty members of the Information Technology Department of enabled me to make a thorough study of this subject.

Last but not the least, no acknowledgement will be complete without mentioning my family and friends. They have also supported me throughout the development projects.

**INDEX**

## INTRODUCTION

This document will propose all features and procedures to develop the system.

**Cab Online Booking** is very useful for Educational Institute to prepare an exam, safe the time that will take to check the paper and prepare mark sheets. It will help the Institute to testing of students and develop their skills. But the disadvantages for this system, it takes a lot of times when you prepare the exam at the first time for usage. And we are needs number of computers with the same number of students.

The effective use of "Cab Online Booking", any Educational Institute or training centers can be use it to develop their strategy for putting the exams, and for getting better results in less time

**Advantages**

* Examiners get tired checking huge number of answer sheets, hence the system reduces their workload by automating the manual checking process accurately.
* The system calculates the distance and show the amount automatically.

**Disadvantages**

* The system must be given proper inputs otherwise system can produce wrong results.

## OBJECTIVES

* To provide an interface through which student can appear for examination online for objective type questions
* Objective answers will be checked automatically by the system from the database
* To provide an interface from where controller will Register to Admin for Exam Scenerio
* The Exam Department will manage the questions sent by Teachers/experts.
* To allow faculty to create tests and answer key.
* The Main Purpose is to Reduce human errors that commonly occur during manual checking.
* The system provides an unbiased result.

.

## TOOLS

**OPERATING ENVIRONMENT**

**Operating environment for the Sport Management System is as listed below**

Client/Server system

Operating System : Windows 8.1

Database: phpMySql

Platform: Dot Net

**EXTERNAL INTERFACE REQUIREMENTS**

**USER INTERFACES**

Front-end software: VB Dot Net

Back-end software: phpMySql

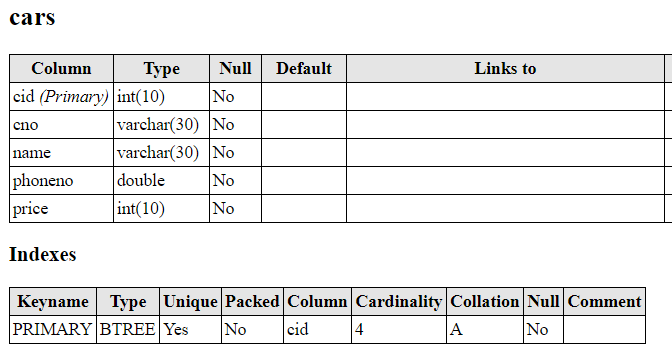
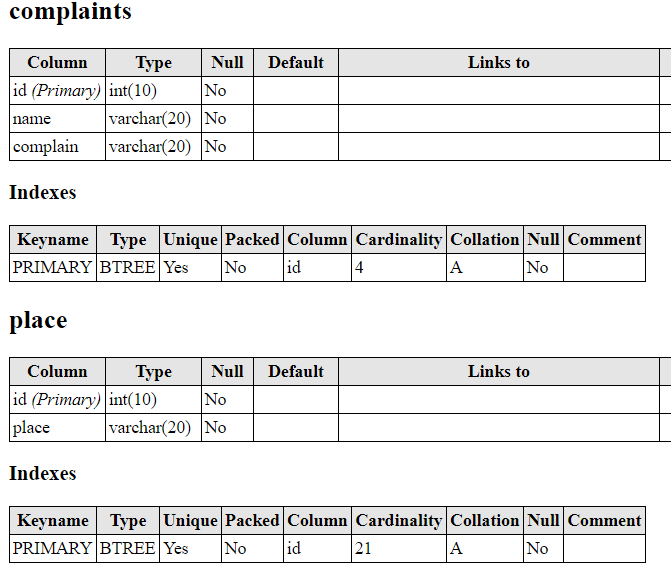
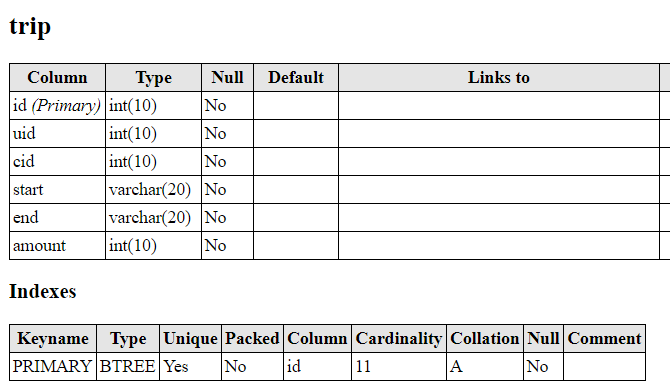
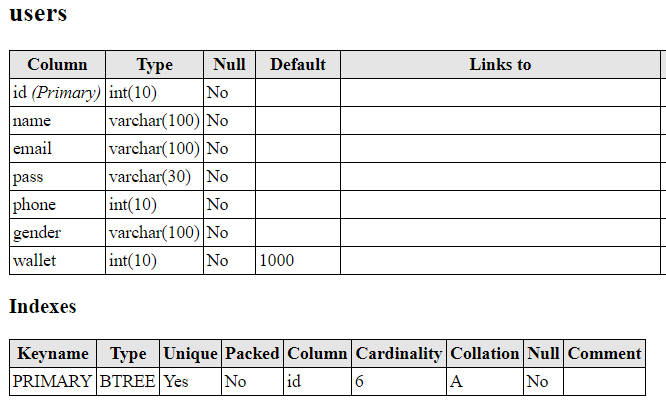
**HARDWARE INTERFACES**

Windows 8.1

**Software Interface:-**

|  |  |
| --- | --- |
| **SOFTWARES USED** | **DESCRIPTION** |
| Operating System | Windows 8.1 is chosen for its features |
| Database | phpMySql for saving the records of Student |
| Dot Net | To implement the project we have chosen Dot Net language for its more interactive support. |

**Data Dictionary**

****

**SRS (Software Requirement Specification)**

**Introduction:**

The Online Examination &Result Management System is an online Exam & Result management system to effectively manage Exam And Generate Result According To The input

Purpose:

The purpose of this SRS document is to write the functional and non functional user or system requirements that represent the characteristics of ***Cab Online Booking***.

The scope and limitation of this system is:

* The on-line exam system design to educational institutes.
* Hold all operation and generate reports to Student  Support multiple choices questions.
* Allow the User to prochoice the answer and to see his mark.

**Scope**:

Methodology of the system is clear that result in the complete and proper working of the system from the requirements achieved. it contains the working model for the development of system

###  Module

**User Login**

* User Name
* Password

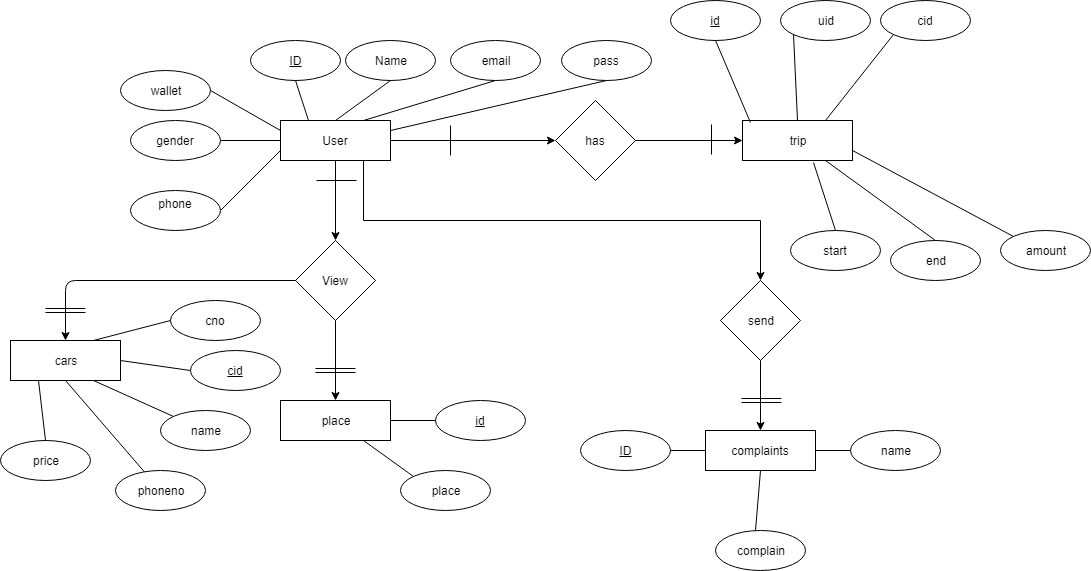
###  Module

**User Registraion**

* Name
* Email
* Gender
* Phone No.
* Password

**E-R DIAGRAM OF ONLINE EXAMINATION & RESULT**

**MANAGEMENT**

****

9

### Functional and Behavioral Modeling

Data Flow Diagram (level 0):

RReereree

**Online**

**Examination**

**&**

**Result**

**Management**

**Student**

tt

R

1

.Request

To

Register

Admin

2.

Registered

4.

Exam

Started

3.

Request

To

StartExam

.Request

5

Result

6

.Result

Confirmed

#### Data Flow Diagram (level 1): For Registration

User

For online

registeration

Request send

to admin

Admin

Registered

Student informed registration

stud

request

1

2

3

#### Data Modularization Details

Modularization is a technique to divide a software system into multiple discrete and independent modules, which are expected to be capable of carrying out tasks independently. These modules may work as basic constructs for the entire software. Designers tend to design modules such that they can be executed and/or compiled separately and independently.

Modular design unintentionally follows the rules of ‘divide and conquer’ problem solving strategy this is because there are many other benefits attached with the modular design of a software.

-Advantage of modularization:

-Smaller components are easier to maintain

-Program can be divided based on functional aspects

-Desired level of abstraction can be brought in the program

-Components with high cohesion can be re-used again

-Concurrent execution can be made possible

**Cab Online Booking: -** This project is divided into certain modules depanding upon the functionality at each process. It provides ease to work with the system as each function is separately available and user friendly.

Hierarchical Structure of project

* User Login form
* Register for User
* Select the destination
* Check the Information of Cab
* Add Cash in Virtual Wallet
* Get Past Trip Details

#### Data Integrity and Constraints including Database Design

**Data integrity** is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life‐cycle and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data.

#### Types of Data Integrity

This section describes the rules that can be applied to table columns to enforce different types of data integrity.

1. Null Rule

A null rule is a rule defined on a single column that allows or disallows inserts or updates of rows containing a null (the absence of a value) in that column.

2. Unique Column Values

A unique value rule defined on a column (or set of columns) allows the insert or update of a row only if it contains a unique value in that column (or set of columns).

3. Primary Key Values

A primary key value rule defined on a key (a column or set of columns) specifies that each row in the table can be uniquely identified by the values in the key.

4. Referential Integrity Rules

A referential integrity rule is a rule defined on a key (a column or set of columns) in one table that guarantees that the values in that key match the values in a key in a related table (the referenced value).

### User Interface Design

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human computer interaction.

UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

* Attractive
* Simple to use
* Responsive in short time
* Clear to understand
* Consistent on all interfacing screens

#### Program code

**Cab Online Booking (Source Code)**

**Login Form**

Imports MySql.Data.MySqlClient

Public Class Form1

Dim MysqlConn As MySqlConnection

Dim COMMAND As MySqlCommand

Public id As Double

Public wallet As Double

Private Sub B1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles B1.Click

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

Try

MysqlConn.Open()

Dim Query As String

Query = "Select \* from cab.users where email='" & t1.Text & "' and pass='" & t2.Text & "' "

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = COMMAND.ExecuteReader

Dim count As Integer

count = 0

While READER.Read

id = READER.GetString("id")

wallet = READER.GetString("wallet")

count = count + 1

End While

If count = 1 Then

MessageBox.Show("Successfully logged in!")

Form3.Show()

Me.Hide()

Else

MessageBox.Show("Username and password does not match")

End If

MysqlConn.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

MysqlConn.Dispose()

End Try

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Form2.Show()

Me.Hide()

End Sub

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

Private Sub Label1\_Click(sender As Object, e As EventArgs) Handles Label1.Click

End Sub

End Class

**Registration Form**

Imports System.Text.RegularExpressions

Imports MySql.Data.MySqlClient

Imports System.Data

Public Class Form2

Dim MysqlConn As MySqlConnection

Dim COMMAND As MySqlCommand

Dim COMMAND2 As MySqlCommand

Public id As Integer

Private Sub B1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles B1.Click

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

Dim email\_err As Boolean

Dim err As Integer

Dim phone As Double

err = 0

Dim email, pass, name, gender As String

name = t1.Text()

pass = pass1.Text()

phone = t3.Text()

If rb1.Checked = True Then

gender = "male"

Else

gender = "female"

End If

email = t4.Text()

Try

MysqlConn.Open()

Dim Query As String

Dim Query2 As String

'all-validations

'gender

If rb1.Checked = True Then

err = 0

Else

err = err + 1

End If

'phone no

If IsNumeric(t3.Text) And t3.Text.Length = 10 Then

err = 0

Else MsgBox("Please Enter valid no.")

err = err + 1

End If

'name

If Not Regex.IsMatch(t1.Text, "^[a-z ]\*$", RegexOptions.IgnoreCase) Then

err = err + 1

MessageBox.Show("Please Enter Alphabetic Characters Only!")

t1.Focus()

t1.Clear()

End If

'email

email\_err = Regex.IsMatch(t4.Text, "([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\." +

")|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})", RegexOptions.IgnoreCase)

If Not email\_err Then

err = err + 1

MsgBox("Invalid Email")

End If

If err = 0 Then

Query = "INSERT INTO cab.users (name,email,pass,phone,gender) VALUES ('" & name & "', '" & email & "','" & pass & "', '" & phone & "', '" & gender & "') "

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = COMMAND.ExecuteReader

Dim count As Integer

count = 0

While READER.Read

count = count + 1

End While

READER.Close()

Query2 = "Select \* from cab.users where email='" & email & "' and pass='" & pass & "' "

COMMAND2 = New MySqlCommand(Query2, MysqlConn)

READER = COMMAND2.ExecuteReader

While READER.Read

id = READER.GetString("id")

End While

READER.Close()

Form1.Show()

Me.Hide()

MysqlConn.Close()

ElseIf err > 0 Then

MsgBox("There are some errors!")

End If

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

MysqlConn.Dispose()

End Try

End Sub

Private Sub b2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles b2.Click

Form1.Show()

End Sub

End Class

**Main Page**

Public Class Form3

Public correct As Integer

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Form4.Show()

Me.Hide()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Form6.Show()

Me.Hide()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Me.Hide()

Form7.Show()

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

Me.Hide()

Form8.Show()

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

Me.Hide()

Form1.Show()

End Sub

End Class

**Select Path**

Imports MySql.Data.MySqlClient

Public Class Form4

Public MysqlConn As MySqlConnection

Public COMMAND As MySqlCommand

Public selected1 As String

Public selected2 As String

Public amount As Integer

Private Sub b2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles b2.Click

Me.Hide()

Form3.Show()

End Sub

Private Sub Form4\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

Try

MysqlConn.Open()

Dim Query As String

Query = "Select \* from cab.place"

Command = New MySqlCommand(Query, MysqlConn)

READER = Command.ExecuteReader

Dim count As Integer

count = 0

cb\_test.Items.Clear()

While READER.Read

Dim sname As String

sname = READER.GetString("place")

cb\_test.Items.Add(sname)

cb\_test2.Items.Add(sname)

End While

cb\_test.Text = "Select from..."

cb\_test2.Text = "Select from..."

'MsgBox(Form1.id)

MysqlConn.Close()

Catch ex As MySqlException

MessageBox.Show(ex.Message)

Finally

MysqlConn.Dispose()

End Try

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

selected1 = cb\_test.SelectedItem.ToString

selected2 = cb\_test2.SelectedItem.ToString

Dim READER2 As MySqlDataReader

MysqlConn.Open()

Dim Query As String

Query = "Select \* from cab.place where place='" & selected1 & "'"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER2 = COMMAND.ExecuteReader

Dim id1 As Integer

Dim id2 As Integer

While READER2.Read

id1 = READER2.GetString("id")

End While

READER2.Close()

Query = "Select \* from cab.place where place='" & selected2 & "'"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER2 = COMMAND.ExecuteReader

While READER2.Read

id2 = READER2.GetString("id")

End While

MysqlConn.Close()

Dim temp As Integer

Dim Diff As Integer

If id1 < id2 Then

temp = id1

id1 = id2

id2 = temp

End If

Diff = id1 - id2

amount = 0

If Diff >= 0 And Diff <= 5 Then

amount = 100

End If

If Diff > 5 And Diff <= 10 Then

amount = 200

End If

If Diff > 10 And Diff <= 15 Then

amount = 300

End If

If Diff > 15 And Diff <= 20 Then

amount = 400

End If

Form5.Show()

Me.Hide()

End Sub

Private Sub cb\_test\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cb\_test.SelectedIndexChanged

End Sub

End Class

**Show Car Details**

Imports MySql.Data.MySqlClient

Public Class Form5

Dim MysqlConn As MySqlConnection

Dim COMMAND As MySqlCommand

Dim cno As String

Dim phno As String

Dim cid As String

Dim dname As String

Private Sub Form5\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

MysqlConn.Open()

Dim query As String

query = "Select \* from cab.cars where price = '" & Form4.amount & "'"

COMMAND = New MySqlCommand(query, MysqlConn)

READER = COMMAND.ExecuteReader

While READER.Read

cno = READER.GetString("cno")

phno = READER.GetString("phoneno")

cid = READER.GetString("cid")

Dname = READER.GetString("name")

End While

READER.Close()

Label6.Text = Form4.amount

Label4.Text = cno

Label5.Text = phno

Label9.Text = Dname

MysqlConn.Close()

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

MysqlConn.Open()

Dim READER As MySqlDataReader

Dim uid As Integer

Dim start As String

Dim drop As String

Dim amount As String

start = Form4.selected1

amount = Form4.amount

drop = Form4.selected2

Dim query As String

Dim wallet As Integer

uid = Form1.id

wallet = 0

query = "select \* from cab.users where id = '" & Form1.id & "'"

COMMAND = New MySqlCommand(query, MysqlConn)

READER = COMMAND.ExecuteReader

While READER.Read

wallet = READER.GetString("wallet")

End While

READER.Close()

If wallet < (amount) Then

MsgBox("Insufficient Funds ")

Else

Dim subb As Integer

subb = wallet - amount

query = "UPDATE cab.users SET wallet = '" & subb & "' where id='" & Form1.id & "'"

COMMAND = New MySqlCommand(query, MysqlConn)

READER = COMMAND.ExecuteReader

While READER.Read

End While

READER.Close()

query = "INSERT INTO cab.trip (uid,cid,start,end,amount) VALUES ('" & uid & "', '" & cid & "','" & start & "','" & drop & "','" & amount & "')"

COMMAND = New MySqlCommand(query, MysqlConn)

READER = COMMAND.ExecuteReader

While READER.Read

End While

READER.Close()

MsgBox("Successfully Cab Book")

Form3.Show()

Me.Hide()

End If

MysqlConn.Close()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Form4.Show()

Me.Hide()

End Sub

End Class

**Add Money**

Imports MySql.Data.MySqlClient

Public Class Form6

Public MysqlConn As MySqlConnection

Public COMMAND As MySqlCommand

Dim wallet As Integer

Dim temp As Integer

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Dim Value As Integer

Dim final As Integer

Value = ComboBox1.SelectedItem.ToString

final = wallet + Value

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

MysqlConn.Open()

Dim Query As String

Query = "Update cab.users set wallet = '" & final & "' where id = '" & Form1.id & "'"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = COMMAND.ExecuteReader

MsgBox("Added")

Label3.Text = final

End Sub

Private Sub Form6\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

MysqlConn.Open()

Dim Query As String

Query = "Select \* from cab.users where id ='" & Form1.id & "'"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = COMMAND.ExecuteReader

While READER.Read

wallet = READER.GetString("wallet")

End While

temp = wallet

Label3.Text = wallet

ComboBox1.Items.Clear()

ComboBox1.Items.Add("100")

ComboBox1.Items.Add("200")

ComboBox1.Items.Add("300")

ComboBox1.Items.Add("400")

ComboBox1.Items.Add("500")

ComboBox1.Text = "Select from..."

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Me.Hide()

Form3.Show()

End Sub

Private Sub Label2\_Click(sender As Object, e As EventArgs) Handles Label2.Click

End Sub

End Class

**Information of past trips**

Imports MySql.Data.MySqlClient

Public Class Form7

Public MysqlConn As MySqlConnection

Public COMMAND As MySqlCommand

Private Sub Form7\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

MysqlConn.Open()

Dim Query As String

Query = "Select \* from cab.trip where uid='" & Form1.id & "'"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = Command.ExecuteReader

ListView1.View = View.Details

ListView1.Columns.Add("S no.", 100)

ListView1.Columns.Add("FROM", 150)

ListView1.Columns.Add("To", 150)

ListView1.Columns.Add("amount", 150)

Dim i As Integer

i = 1

Dim j As Integer

j = 0

While READER.Read

ListView1.Items.Add(i)

ListView1.Items(j).SubItems.Add(READER.GetString("start"))

ListView1.Items(j).SubItems.Add(READER.GetString("end"))

ListView1.Items(j).SubItems.Add(READER.GetString("amount"))

i = i + 1

j = j + 1

End While

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Form3.Show()

Me.Hide()

End Sub

End Class

**Complaint Page**

Imports MySql.Data.MySqlClient

Public Class Form8

Dim MysqlConn As MySqlConnection

Dim COMMAND As MySqlCommand

Dim cno As String

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost;userid=root;password=;database=cab"

Dim READER As MySqlDataReader

MysqlConn.Open()

Dim query As String

Dim complain As String

Dim dname As String

dname = ComboBox1.SelectedItem.ToString

If RadioButton1.Checked = True Then

complain = "Breaking Rules"

ElseIf RadioButton2.Checked = True Then

complain = "Fast Driving"

ElseIf RadioButton3.Checked = True Then

complain = "Wrong Dressing Sense"

ElseIf RadioButton5.Checked = True Then

complain = "Rude Behaviour"

Else

complain = "Dirty Car Environment"

End If

query = "insert into cab.complaints(name,complain) Values ('" & dname & "','" & complain & "')"

COMMAND = New MySqlCommand(query, MysqlConn)

READER = COMMAND.ExecuteReader

MsgBox("DONE")

READER.Close()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

Form3.Show()

Me.Hide()

End Sub

End Class

## TESTING AND VALIDATION

### Testing Plan

Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals; that is, conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

**Need of Testing**

A Successful test is one that finds an undiscovered error. If the testing is conducted successfully, it will uncover errors in the software.

Testing demonstrate that software function appear to be working according to specification, that behavioral and performance requirements appear to have been met.

Testing is conducted provide a good indication of the software reliability and software quality. Testing cannot show the absent of errors and defects, it can show only that software errors and defects are present.

With the project nearing completion, we have taken up the activity of testing the individuals’ forms and making sure that the interaction among the various forms is smooth and without any glitches. Before actually beginning to test a few things should be born in mind. Among others these include:

1. All tests should be traceable to customer requirements
2. Test should be planned before the testing begins
3. To be effective, testing should be conducted by an independent third party

## TESTING STRATEGY

### 1) SYSTEM TEST

The System tests will focus on the behavior of the system. User scenarios will be executed against the system as well as screen mapping and error message testing. Overall, the system tests will test the integrated system and verify that it meets the requirements defined in the requirements document

### 2) PERFORMANCE TEST

Performance test will be conducted to ensure that the system’s response times meet the user expectation and do not exceed the specified performance criteria. During these tests, response times will be measured under heavy stress and/or volume.

### 3) SECURITY TEST

Security tests will determine how secure the system is. The tests will verify that unauthorized user access to confidential data is prevented.

### 4) AUTOMATED TEST

A suite of automated tests will be developed to test the basic functionality of the system and perform regression testing on areas of the systems that previously had critical/major defects. The tool will also assist us by executing user scenarios thereby emulating several users.

### 4) RECOVERY TEST

Recovery tests will force the system to fail in a various ways and verify the recovery is properly performed. It is vitally important that all payroll data is recovered after a system failure & no corruption of the data occurred.

### 6) DOCUMENTATION TEST

Tests will be conducted to check the accuracy of the user documentation. These tests will ensure that no features are missing, and the contents can be easily understood.

### 7) USER ACCEPTANCE TEST

Once the hotel management system is ready for implementation, the Payroll department will perform User Acceptance Testing. The purpose of these tests is to confirm that the system is developed according to the specified user requirements and is ready for operational use.

## TESTING METHODS

* **BLACK-BOX TESTING**

In using this strategy, the tester views the program as a black – box, tester doesn’t see the code of the program: Equivalence partitioning, Boundary – value analysis, Error guessing.

* **WHITE-BOX TESTING**

In using this strategy, the tester examines the internal structure of the program:

Statement coverage, Decision coverage, condition coverage,

Decision/Conditional coverage, Multiple – condition coverage.

* **GRAY-BOX TESTING**

In using this strategy Black box testing can be combine with knowledge of database validation, such as SQL for database query and adding/loading data sets to confirm functions, as well as query the database to confirm expected result.

* **TEST SCRIPT**

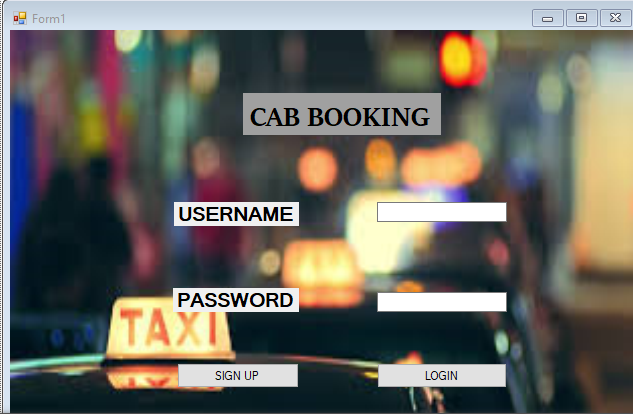
It is type of test file. It is a set of instructions run automatically by a software or hardware test tool.

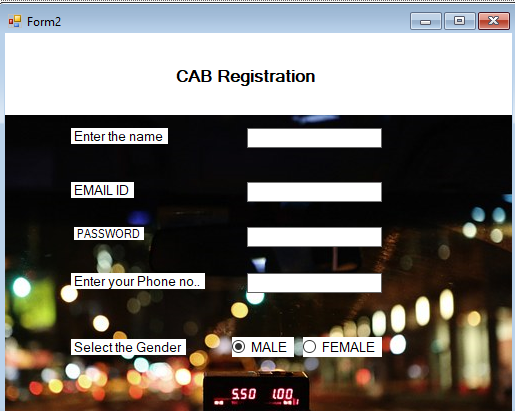
**TESTING**

# Test cases

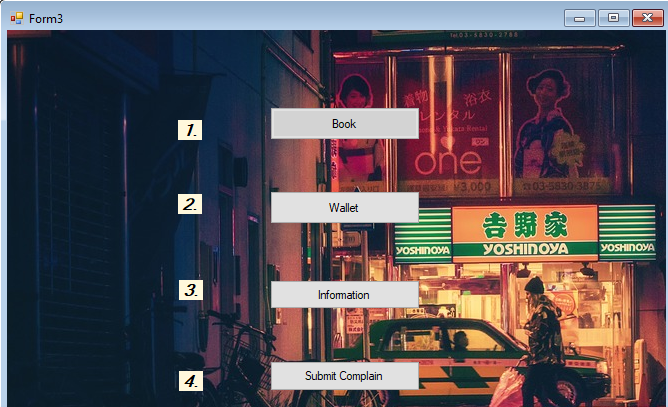
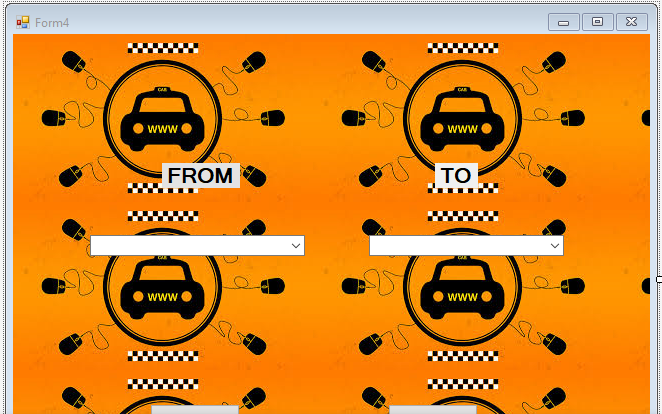
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MODULE | TEST CASE | INPUT | EXPECTED  OUTPUT | ACTUAL  OUTPUT | STATUS |
| SIGN UP | 1 | NAME=ayush  Email=ayush  Pass=ayush  Phone=1234567890 | SIGN IN | Invalid Email | PASS |
|  | 2 | NAME=mukul  Email=mukul@mail.com  Pass=mukul  Phone=1234 | Sign in | Please Enter valid no. | PASS |
|  | 3 | NAME=rohit  [Email=rohit@mail.com](mailto:Email=rohit@mail.com)  Pass=rohitN  Phone=1234567890 | Login In | Login In | PASS |
| LOGIN IN | 1 | Username= rohit@mail.com  Password = | Login | Username and password does not match | PASS |
|  | 2 | [username=](mailto:Email=)mukul@mail.com  Password = | Login | Username and password does not match | PASS |
|  | 3 | [username=mukul](mailto:username=mukul)@mail.com  Password = mukul1234 | Successfully Logged in | Successfully Logged in | PASS |

**INPUT AND OUTPUT SCREEN**

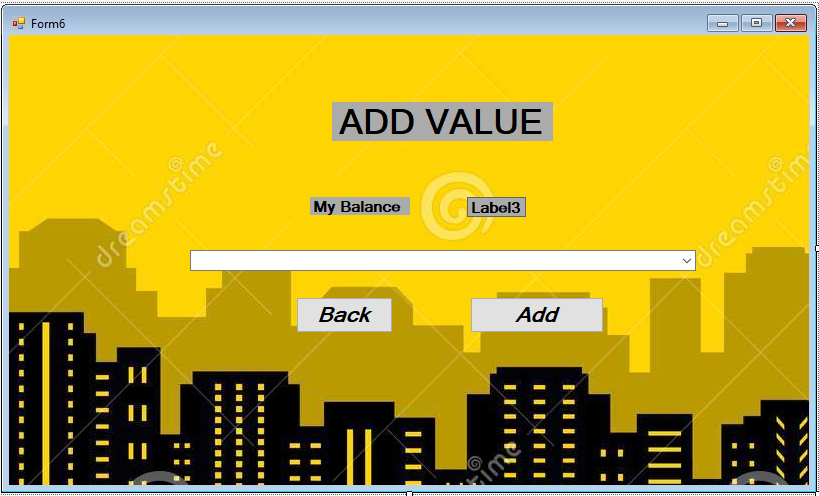
Login Page 

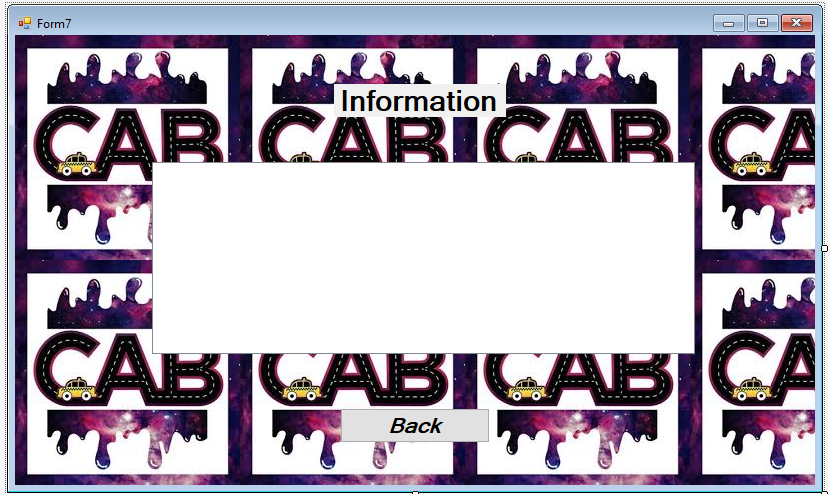
Registration Page 

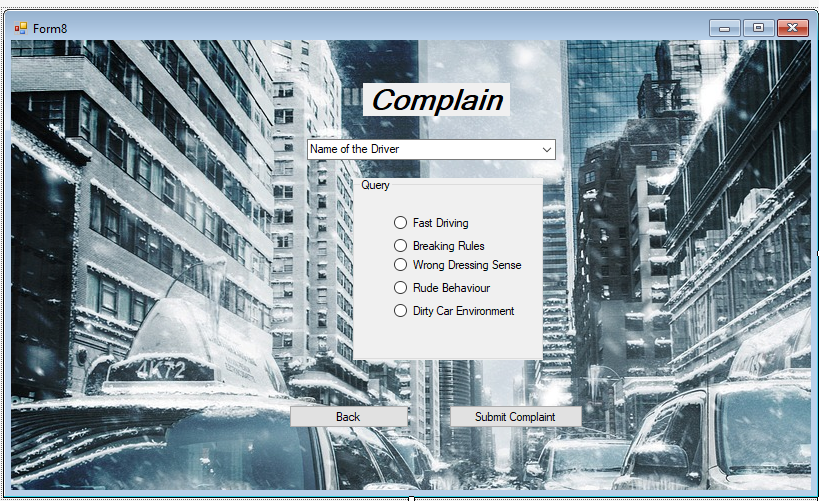
Main page

Select Path

Car Details Show

Add Money In Wallet

Past Trip Show

Complaint Page

## LIMITATIONS OF PROJECT

Everything comes with some positive and some negative faces. Our project is also facing some limitations which is, in this project admin have less power. As he can perform only limited number of task that is view, add, update and delete products.

* Cab Online Booking is not secure so much.
* Cab is selected by pre-defined programming not a user.
* Limited Cars

### FUTURE APPLICATION OF PROJECT

Due to the specific time for the project, the current project has limited information and can be developed and enhanced for the better future use. The various enhancements that can be made in this software are as below:

We will add more cabs for user.

We will also give some more powers to user for change a profile details

We will Improve the user interface.

We can Make It More Secure In Terms Of Back-end Validation.

### BIBLIOGRAPHY

* W3schools.com
* Youtube.com
* Stackoverflow.com
* Tutorialspoint.com