

# ACKNOLEDGEMENT

We are extremely grateful and highly indebted to our advisor **…………………………** for supporting us all along and always being there for consultation. He has been a guide in the true sense of the word and has directed the project in a very efficient and organized manner.

We thank our examiners **………………………………** for their suggestions and guidelines to make some improvements in the project.

Finally we thank **…………………………….**, for her continued drive for better quality in everything that happens at **TIHE**. This report is a small contribution towards the greater goal.

**STUDENT NAME:-**

**Himanshi Khowal**

**Saveen Yadav**

**INDEX**

[TITLE 1](#_Toc434866948)

[CERTIFICATE 2](#_Toc434866953)

[ACKNOLEDGEMENT 3](#_Toc434866954)

**CHAPTER-1 PROBLEM OF DEFINITION…………………………….…………..…..…6-12**

[1.1[ BRIEF DECRIPTION OF COMPANY] 6](#_Toc434866955)

[1.2 GENERAL DECRIPTION OF SYTEM UNDER STUDY 10](#_Toc434866957)

[1.3 [USER REQUIREMENTS] 11](#_Toc434866958)

[1.4 [ ESTABLISH THE NEED OF THE NEW SYSTEM] 11](#_Toc434866959)

[1.5 [ OBJECTIVES OF THE PROJECT] 12](#_Toc434866960)

**CHAPTER-2 SYSTEM ANALYSIS OF EXISTING SYSTEM……………………......…14-43**

[2.1 [Modules/Processes] ..14](#_Toc434866961)

[2.2 [Input and Output] 30](#_Toc434866964)

[2.3 [PROCEDURE FOR CONVERTING INPUT INTO OUTPUT] 36](#_Toc434866965)

[2.4 [Controls] 38](#_Toc434866967)

**CHAPTER-3 REQUIREMENT OF ANALYSIS………………………………………..…45-65**

[3.1[Physical Design] 45](#_Toc434866971)

[3.2[Interface Design] 50](#_Toc434866972)

**CHAPTER-4 SYSTEM DEVELOPMENT………………………………………….……..67-70**

[4.1[PROGRAMME DEVELOPMENT] 67](#_Toc434866973)

[4.2[VALIDATION] 68](#_Toc434866974)

**CHAPTER-5** **SYSTEM TESTING & DEBBUGING……………………………72-82**

[5.1[TESTING & DEBUGGING] 72](#_Toc434866975)

[5.2[TEST CASES] 81](#_Toc434866976)

**CHAPTER-6 SYSTEM IMPLEMENTATION………………………………...……….…84-91**

[6.1[HARDWARE REQUIREMENT] 84](#_Toc434866977)

[6.2[SOFTWARE REQUIREMENT] 87](#_Toc434866978)

[6.3[PEOPLE AND RESOURCE REQUIRED] 88](#_Toc434866979)

[6.4[COST OF THE SYSTEM] 89](#_Toc434866980)

[6.5[TRAINING NEEDS] 90](#_Toc434866981)

**CHAPTER-7 SUMMARY**……………………………………………………….**…..93-97**

[7.1[CONCLUSION] 93](#_Toc434866982)

[7.2[ADVANTAGE] 94](#_Toc434866983)

[7.3[DISADVANTAGE] 96](#_Toc434866984)

[7.4 [Reference] 97](#_Toc434866985)

[**Table of Figures** 98](#_Toc434866986)

**CHAPTER – 1**

INTRODUCTION

1.1 About Organisation

## **EASY CAB SYSTEM**

## ADVANTAGES OF EASY CAB SYSTEM

Now one can easily plan the journey comfortably as the process is efficient and fast with being easy to access. Bookings can be made through the Easy Cab site or by the phone call. This being a big step in terms of improvement in the Taxi system it is widely accepted across the country.

* A route-based booking system that facilitates the issue of journey-cum-booking Taxi, which can be issued from any station to any station.
* Passenger journey to multiple laps of booking can be handled from a single terminal window.
* The booking facility is offered round-the-clock (24 hours uninterrupted).
* Changes in Taxi profiles (Taxi addition, replacement, de-allocation), route structures, etc., can be made effective immediately with the appropriate contingency handling.
* Dynamic definition of the advance booking period is possible. This feature facilitates defining different advance booking periods for different Taxis.
* Any Taxi running schedule can be accommodated.
* Provides on-line aggregation of EIS figures such as revenue, Taxi utilization, etc, and presentation of the summarized data in the form of visual analytics from the operational system's information store. The data aggregation is done incrementally, to inflict minimal impact.
* Provides automatic database recovery against all kinds of hardware and software failures.
* Complete audit trails for transactions and data access.
* The application software is parametric, and standard Taxi business rules are incorporated in the form of data instead of being part of the logic.

1.2 **Introduction about project**

# EASY CAB SYSTEM” aims are to book the taxis at all the fare charges. Manual system that is employed is extremely laborious and quite inadequate. It only makes the process more difficult and hard. The aim of our project is to develop a system that is meant to partially computerize the work performed in the prepaid taxi management system like generating monthly daily bookings, record of routes available , fare charges of every route; store record of the customer. We used Microsoft Visual Basic 6.0 as front end and MS-Access 2000as back end for developing our project. Visual Basic is primarily a visual design environment. We can create a VB application by designing the form and that make up the user interface. Adding visual basic application code to the form and the objects such as buttons and text boxes on them and adding any required support code in additional modular.

# MY SQL is a powerful relational database application with which a desktop user can efficiently create and manipulate database systems. Access targets the desktop category and works best for individuals and workgroup Managing megabytes of data for multi-user access to the same database, Access uses file-server architecture, rather than client-server architecture. Access is included in the professional and developer editions of Microsoft office. The overall project report is divided into some parts.

# The project ‘Easy Cab’ is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses PHP as the front-end software which is an Object Oriented Programming and has connectivity with MY SQL.

# **Need for the system**

Easy Cab service is a major transport service provided by the various transport operators in a particular city. Mostly peoples use Taxi service for their daily transportations need. The company must be a registered and fulfils all the requirements and security standards set by the transport department.Easy Cab Software is a web based platform that allows your customers to book their taxis and executive taxi's (such as Limousines) all online from the comfort of their own home or office. The platform should offer an administration interface where the taxi company can manage the content, and access all bookings and customer information. Usually the platform will include all the required functionality such as hosting, email accounts, updates, a domain name (the web address) and, most importantly, backups!

In a nutshell your Easy Cab Software should be able to:

* Provide the functionality to make your own bookings
* Give your customers the facility to make payments and deposits online with their credit / debit card.
* Generate Invoices
* Update your web site without the need to get a web designer involved.
* Provide the customer with taxi availability
* Track your customers.
* Engage your customers through interaction such as feedback forms

More and more Taxi companies are looking for integrated Easy Cab systems as it makes life much easier for a) the customer - this is highly important and in today's internet age people should be able to book taxis online without having to pick up the phone and b) the taxi company - as all their bookings are now managed via an automated system which means they have an electronic record of future and historic bookings. From the historic data the taxi firm will be able to look at booking trends of set periods of time, and future bookings will allow them to budget their staff and taxi routes accordingly.

Easy Cab Software can be built on top of a great looking web site and third party payment providers can be used to provide secure transactions. One factor that may be a stumbling block for Taxi firms is the cost of the Easy Cab software. Some vendors offer a shared pricing model that allows certain taxi firms to share the cost of the system. This type of model is known as multi-vendor. As the Easy Cab systems are web based this doesn't cause any issues as the software is located on a central web server and the web site front end can be branded in any way required by the taxi company. In today's technological world can your taxi company afford to be without booking software?

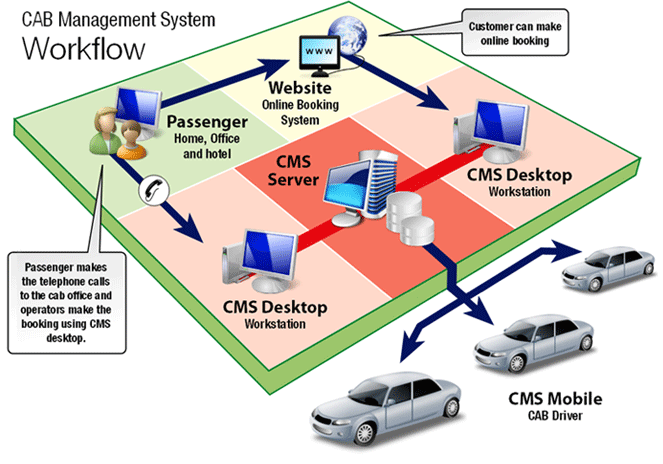
* The fare must be economical so that it must be in reach/budget of every person.
* Taxi must be reach on time on the defined destination.
* There must be a large fleet of Taxis (AC/Non AC).
* Provide the functionality to make your own bookings
* Update your web site without the need to get a web designer involved.
* Provide the customer with taxi availability.
* Track your customers.
* Engage your customers through interaction such as feedback forms
* Easy payment facility must be provided in Taxi i.e. by cash or by card.
* Payment bill must be provided by Taxi driver.
* Driver’s identification data must be given i.e. driver’s name, id & photograph at the time of booking of Taxi.
* Estimated time for a particular journey must be provided.
* Details of the route must be provided to the customer. Customers can my take the Taxi by his/her own route.
* Customer satisfaction is necessary.
* The user interface must be friendly so that the user can easily book a Taxi in few minutes by doing few clicks.
* Payment modes can be also of prepaid or post-paid.
* If the payment mode is prepaid then the customer have to provide its full name, address, type of card(visa, master, electron-visa etc.), account number, bank name, and branch.

In both prepaid and post-paid customer’s email id must be provided.

* At the time of booking the web page must have the interface for the starting point, destination, type of Taxi (AC/Non Ac), charge per kilometre, Taxi driver details, time, payment options, service area etc.
* Night charges are also added to the total fare on the basis of per kilometre.
* In the end the report must be generated which shows the customer name, address, source and destination, total fare, driver details and timings i.e. the invoice must be generated

If the payment is made at the time of booking i.e. prepaid then it must be shown in the report.

* The information must be provided to the customer on its email id and to driver on its job sheet.

****

* 1. **Objective of the project**

1. Easy Cab system is very helpful for user; user can search the taxi according to his/her needs and book the taxi.
2. But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people booking taxi. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper
3. Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as “Easy Cab”.
4. The main aim of our project is to provide a paper-less Easy Cab up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

## 1**.5 METHODOLOGY**

The project ‘Easy Cab’ is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses PHP as the front-end software which is an Object Oriented Programming and has connectivity with MY SQL. It is a web based application in which

number of clients can also access with a server.

**CHAPTER 2**

**SYSTEM REQUIREMENT**

# **2.1 SOFTWARE REQUIREMENTS:**

PHP is a [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) language created in [1995](https://en.wikipedia.org/wiki/1995) and designed for [web development](https://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). As of January 2013, PHP was installed on more than 240 million [websites](https://en.wikipedia.org/wiki/Website) (39% of those sampled) and 2.1 million [web servers](https://en.wikipedia.org/wiki/Web_server). Originally created by [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf" \o "Rasmus Lerdorf) in 1994, the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of PHP (powered by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine" \o "Zend Engine)) is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor, which is a [recursive](https://en.wikipedia.org/wiki/Recursive_acronym) [backronym](https://en.wikipedia.org/wiki/Backronym" \o "Backronym).

PHP code can be simply mixed with [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [templating engines](https://en.wikipedia.org/wiki/Web_template_system" \o "Web template system) and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)), which is usually implemented as a web server's native [module](https://en.wikipedia.org/wiki/Plugin_(computing)) or a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) capability and can be used in[standalone](https://en.wikipedia.org/wiki/Computer_software) [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface).

The canonical PHP interpreter, powered by the Zend Engine, is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](https://en.wikipedia.org/wiki/Operating_system) and [platform](https://en.wikipedia.org/wiki/Computing_platform), free of charge.

Despite its popularity, no written [specification](https://en.wikipedia.org/wiki/Formal_specification) or standard existed for the PHP language until 2014, leaving the canonical PHP interpreter as a [de facto](https://en.wikipedia.org/wiki/De_facto) standard. Since 2014, there is ongoing work on creating a formal PHP specification.

PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions <?php and ?> that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

Although PHP's development is focused on server-side scripting, you can do much more with it. Read on, and see more in the can PHP do? Section, or go right to the introductory tutorial if you are only interested in web programming.

The aim of the study to fully related with Easy Cab.

* The Software is for the automation of Easy Cab.
* It maintains two levels of users:-

\_ Administrator Level

\_ User Level

* The Software includes:-
* Maintaining user details.
* Maintaining driver details
* Maintaining taxi details
* Providing driver according to the cab.
* Providing and maintaining all kinds of services.
* Billing and Report generation.

*2.1Introduction*

#### The introduction states the goals and objectives of the software describing it in the context of the computer -based system. Actually the introduction may be nothing more than the software people of the planning document.

#### 2.2Functional Description

#### The project also provides the facility to contact the registered user easily without any admin interaction. If you find an existing entry meeting your requirement, then you can contact the concerned party directly. If there is no entry, which meets your requirement, you can register your details on the site, so that others can find you. The project improves the efficiency and effectiveness of the whole system.

#### 2.3 Behavioral Description

#### IT Examine the operations of the software as a consequence of external events and internally generated control characteristics. Here the Admin checks the classified entered by the registered users. Admin is the only person who can activate or deactivate the classified entered by the registered users.

#### 2.4 Project Planning

#### I was assigned the duty for developing a computerized system known as “Online Placement”. Working in team reinstates the team for some common guidelines and standard to be followed by all the team members across all team. For the optimum use of practical time it is necessary that every session is planned. Planning of this project will include the following things:

**2.5Input/ Output Design**

I was assigned the duty for developing a computerized system known as “Online Placement”. Working in team reinstates the team for some common guidelines and standard to be followed by all the team members across all team. For the optimum use of practical time it is necessary that every session is planned.

Planning of this project will include the following things:

* Topic Understanding.
* Modular Break –Up of the System.
* Processor Logic for Each Module.
* Database Requirements.

**2.4.1 Topic Understanding:**

It is vital that the field of application as introduced in the project may be totally a new field. I carefully went through the project to identify the requirements of the project

**2.4.2 Description Of The Modules**:

* **Module 1: Password Module**

In this module, registered user enters a password and the software checks its validity. If the password is valid then he is allowed to enter, otherwise “Invalid

#### Module 2: User Registration Module

#### In this module user can submit his details. After submitting his details, he/she can

#### his details. After submitting his details, he/she can easily ready for the interview.

#### Module 3: Booking Report Module

#### In this module, registered user can easily see his/her details by login into their account and book the cab.

#### Module 4:AckonwlgementModule

#### In this module,the acknowledgement of the registration is to be seen by the customer.

* **Module 5: Login Module**

In this module user login with registered user id or password.

**2.4.3Database Requirement**

* Tables Required.
* Fields For These Tables.
* The Various Key Fields (for example Primary key and Foreign key).
* Identify The Various Constraints like Not Null, Unique etc.

#### The Database here used is Ms-Sql. The database tables are shown in DATABASE DESIGN.

#### 2.4.3.1 List of Tables:

#### user table

#### registration Table

#### ccno table

#### 2.4.3.2 Software/Compilers Required To Develop This System

#### Operating System : Windows 7/ 8

#### Front-end : PHP

#### Back-End : SQL

#### 2.4.3.3 Hardware Required To Develop This System

**HARDWARE**

Processor : Pentium 2.4 GHz or above

Memory : 256 MB RAM or above

Cache Memory : 128 KB or above

Hard Disk : 3 GB or above [at least 3 MB free space required]

Pen Drive : 5 GB

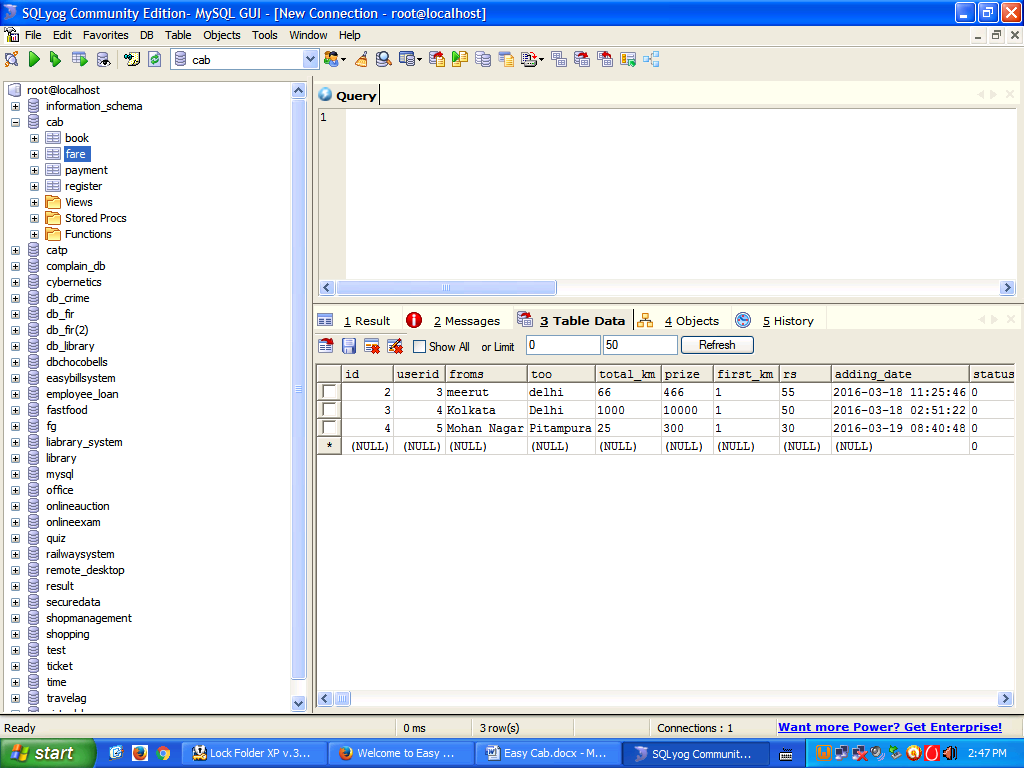
Printer : Laser Printer

.

MY SQL:

Introduction

My SQL is an application used to create computer databases for the Microsoft Windows family of server operating systems. It provides an environment used to generate databases that can be accessed from workstations, the web, or other media such as a personal digital assistant (PDA). MY SQL is probably the most accessible and the most documented enterprise database environment right now. This also means that you can learn it a little quicker than most other database environments on the market



To start, you must have a computer that runs an appropriate operating system like Microsoft Windows >= XP Home Edition: that includes Windows XP Home Edition, Windows XP Professional, Windows 2000 Professional, or any version of Windows Server 2003. In this case, you must install MY SQL Yog.

**What is SQL Used for:**

Using SQL one can create and maintain data manipulation objects such as table, views, sequence etc. These data manipulation objects will be created and stored on the server's hard disk drive, in a table space, to which the user has been assigned.

Once these data manipulation objects are created, they are used extensively in commercial applications.

**DML, DCL, DDL:**

In addition to the creation of data manipulation objects, the actual manipulation of data within these objects is done using SQL.

The SQL sentences that are used to create these objects are called DDL's or Data Definition Language. The SQL sentences used to manipulate data within these objects are called DML's or Data Manipulation Language. The SQL sentences, which are used to control the behavior of these objects, are called DCL's or Data Control Language.

**CHAPTER – 3**

**SYSTEM STUDY**

#### 3.1Feasibility Study

#### A feasibility study is carried out to select the best system that meets performance requirements.

#### Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken.

#### Since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental errors of judgment are made.

#### Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources.

#### The objective of the feasibility study is not to solve the problem but to acquire a sense of its scope . During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined.

#### Consequently, costs and benefits are described with greater accuracy at this stage.

#### It consists of the following:

#### 3.1.1 Statement of the problem: A carefully worded statement of the problem that led to analysis.

#### 3.1.2 Summary of finding and recommendations: A list of the major findings and recommendations of the study. It is ideal for the user who requires quick access to the results of the analysis of the system under study. Conclusion are stated , followed by a list of the recommendation and a justification for them .

#### 3.1.3 Details of findings : An outline of the methods and procedures under-taken by the existing system, followed by coverage of the objectives and procedures of the candidate system. Included are also discussions of output reports, file structures, and costs and benefits of the candidate system.

#### 3.1.4 Recommendations and conclusions: Specific recommendations regarding the candidate system, including personnel assignments, costs, project schedules, and target dates.

#### 3.2Three key consideration are involved in the feasibility analysis these are:

#### Economical Feasibility

#### Technical Feasibility

#### Behavioural Feasibility

Operational Feasibility

**3.2.1 Economical Feasibility**

Economic analysis is the most frequently used method effectiveness of a system. More commonly known as cost/ benefit analysis, the procedure is to determine the benefits and savings that are expected from a system and compare them with cost**.**

Earlier in Computer Craft the work has been done manually which takes lot of time as well as man power which is more economical. Now the same work is computerized which is more effective and efficient, less time consuming, reduces man power which in turn proves to be less economical

**3.2.2 Technical Feasibility**

Technical Feasibility centersaround the existing computer system (hardware/ software) and also it can support the modification. In manual processing there are more chance of errors are there, creating lot of complications, less technical or logical. Through proposed system we can set this process in a very systematic pattern, which is more technical, full proof, authentic, safe and reliable**.**

#### 3.2.3 Behavior Feasibility

#### Our proposed system works to minimize the human errors, take less time, easy interaction with user, bug free. This project/software is further expanded by connecting various interrelated departments and by installing an extension part of this software.

**3.2.4 Operational Feasibility**

If a system meets the controller of programmers and producers requirement then the system can be regarded as operationally feasible.It is mainly related to human organizational and political aspects. The points to be considered are:

* What changes will be bought with the system?
* What organizational structures are distributed?
* What new skills will be required? Do the existing staff members have these skills?
* If not, can they be trained in due course of time?

Generally project will not be rejected simply because of operational feasibility but such considerations are likely to critically effect the nature and scope of the eventual recommendations.

For operational feasibility study we appointed a small group of people who are familiar with information system techniques, who understands the part of the business that are relevant to the project and are skilled in the system analysis and design process.

**3.2.5 Social feasibility**

Social feasibility is a determination of whether a proposed project will be acceptable to the people of not. This determination typically examines the portability of the project being accepted by the group directly affected by the proposed system change.

**3.2.6 Management feasibility**

It is a determination of whether a proposed project will be acceptable to management. If management does not accept a project or gives a negligible support to it than the analyst will tend to view the project as a non-feasibility one.

**3.2.7 Legal feasibility**

Legal feasibility is a determination of whether a proposed project infringes on known acts, statutes, as well as any pending legislation. Although in some instances the project might appear sound, closer investigation it may be found to infringe on several legal areas.

**3.2.8 Time feasibility**

Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame. If a project takes too much time it is likely to be rejected.

**3.3 System Development Life Cycle**

The **systems development life cycle** (**SDLC**), also referred to as the application **development life**-**cycle**, is a term used in **systems** engineering, information **systems** and software engineering to describe a process for planning, creating, testing, and deploying an information **system**.

3.4.1 Development Process

A systems development life cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineers and systems developers to plan for, design, build, test, and deliver information systems. Like anything that is manufactured on an assembly line, an SDLC aims to produce high quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time-frames and cost estimates. Computer systems are complex and often (especially with the recent rise of service-oriented architecture) link multiple traditional systems potentially supplied by different software vendors. To manage this level of complexity, a number of SDLC models or methodologies have been created, such as "waterfall"; "spiral"; "Agile software development"; "rapid prototyping"; "incremental"; and "synchronize and stabilize".

SDLC can be described along a spectrum of agile to iterative to sequential. Agile methodologies, such as XP and Scrum, focus on lightweight processes which allow for rapid changes (without necessarily following the pattern of SDLC approach) along the development cycle. Iterative methodologies, such as Rational Unified Process and dynamic systems development method, focus on limited project scope and expanding or improving products by multiple iterations. Sequential or big-design-up-front (BDUF) models, such as waterfall, focus on complete and correct planning to guide large projects and risks to successful and predictable resultsOther models, such as anamorphic development, tend to focus on a form of development that is guided by project scope and adaptive iterations of feature development.

In project management a project can be defined both with a project life cycle (PLC) and an SDLC, during which slightly different activities occur. According to Taylor (2004) "the project life cycle encompasses all the activities of the project, while the systems development life cycle focuses on realizing the product requirements".

SDLC is used during the development of an IT project; it describes the different stages involved in the project from the drawing board, through the completion of the project.

**3.4.2 Software Lifecycle Model**

The spiral model, originally proposed by Boehm, is an evolutionary Software Engineering Paradigm that couples the iterative nature of prototyping with the controlled and systematic aspects of linear sequential model. It provides for rapid development of incremental version of the software. In the spiral model software, software is developed in a series of incremental releases. During later iteration, increasingly more complete version of engineered system is produced

The spiral model is divided into a four phases. They are:-

**3.4.2.1 User communication:**

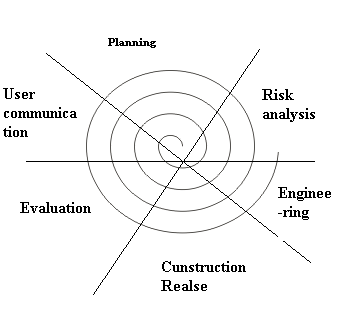
Tasks required establishing effective communication between the developer and user. In the case of this project it is communicate between the developer and application handling system staff.

**3.4.2.2 Planning:** Tasks required defining resources, timelines, and other project related information such that for this project when the appraisal survey/feedback is to be done, check in advance for availability/schedule of HR and BOSS, what are the prerequisite required for survey, and what are the resources are available.

# 3.4.2.3 **Risk Analysis**: Tasks required accessing both technical management risks.Engineering: Task required building one or more representation.Construction & Release: Tasks required constructing, testing installing and providing support.

# 3.4.2.4 **User Evaluation:** Tasks required obtaining user feedback based on evaluation of the software representation created during the engineering stage and implemented during the installation stage**.**

# Spiral Model



**Fig:3.2**

Each of the regions is populated by a series of work tasks that are adapted to the characteristics of the project to be undertake. For small projects, the number of works tasks and their region contains more work tasks that are defined to achieve higher level of formality.

Unlike classical process models that ends when software is delivered, that’s why I will choose the Spiral Model for E-shopping because it can be adapted to apply throughout the life of the computer software.

**Advantages of Spiral model:**

* High amount of risk analysis hence, avoidance of Risk is enhanced.
* Good for large and mission-critical projects.
* Strong approval and documentation control.
* Additional Functionality can be added at a later date.
* Software is produced early in the [software life cycle](http://istqbexamcertification.com/what-are-the-software-development-life-cycle-phases/).

**Disadvantages of Spiral model:**

* Can be a costly model to use.
* Risk analysis requires highly specific expertise.
* Project’s success is highly dependent on the risk analysis phase.
* Doesn’t work well for smaller projects.

**When to use Spiral model:**

* When costs and risk evaluation is important
* For medium to high-risk projects
* Long-term project commitment unwise because of potential changes to economic priorities
* Users are unsure of their needs
* Requirements are complex
* New product line
* Significant changes are expected (research and exploration)

CHAPTER 4

SYSTEM DESIGN

4.1 DataBase Design

4.2 E-R diagram

4.3 data flow diagram (DFD)

#### 4.1Database Design

#### 4.1.1 Data Dictionary (Definition):

#### A data dictionary is a catalog- a repository- of the elements in a system. These elements center on the data and the way they are structured to meet user requirements and organization needs. A data dictionary consists of a list of all the elements composing the data flowing through a system. The major elements are data flows, data stores, and processes. The data dictionary stores details and descriptions of these elements.

#### 4.1.2 Describing Data Elements:

#### Each entry in the data dictionary consists of a set of details describing the data used or produced in the system. Each item is identified by a data name, description, alias, and length and has specific values that are permissible for it in the system being studied.

**4.2 list of table**

Table Structure

* Registration table
* Contact table

**1. sign up table**

|  |  |
| --- | --- |
| Field name | Data type |
| First name | Nvarchar |
| Last name | Nvarchar |
| Address | nvarchar |
| City | Nvarchar |
| State | Nvarchar |
| Pin | Numeric |
| Contact | Numeric |
| Email | Nvarchar |
| Password | Nvarchar |
|  |  |

Table 4.1

**2. login table**

|  |  |
| --- | --- |
| **Field name** | **Data type** |
| **UserId** | **Nvarchar** |
| **Password** | **Nvarchar** |
|  |  |

LOGIN

Transfer

Do/initiate welcome screen

Login

If correct user id,passwd,guest id

Do/login checker

Not Correct

Welcome Screen

Cancelled

#### 4.3 E-R Diagram

#### 4.3.1Entity

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

Enter id & psswd, guest login

If not

exist

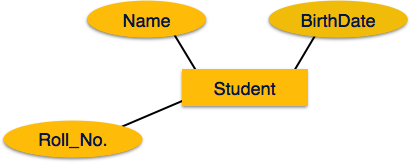
Verification

If exist

Successfully Login

**4.3.2Attributes**

## Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).



**Fig:4.1**

## 4.3.3Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

**4.3.4 Binary Relationship and Cardinality**

A relationship where two entities are participating is called a **binary relationship**. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

* **One-to-one** − When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.

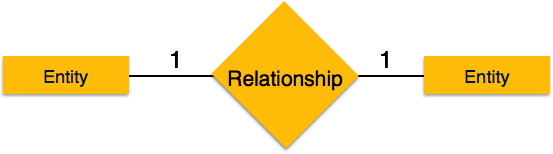


Fig:4.2

* **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.

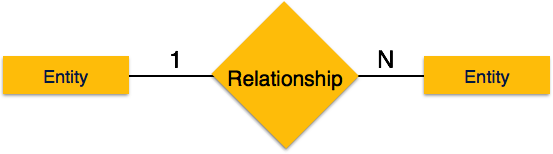


Fig:4.3

* **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.

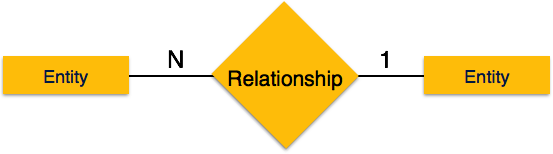
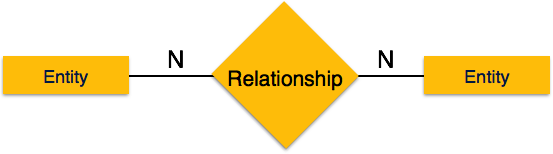


Fig:4.4

* **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.



### Fig:4.5

### 4.3.5Participation Constraints

* **Total Participation** − Each entity is involved in the relationship. Total participation is represented by double lines.
* **Partial participation** − Not all entities are involved in the relationship. Partial participation is represented by single lines.

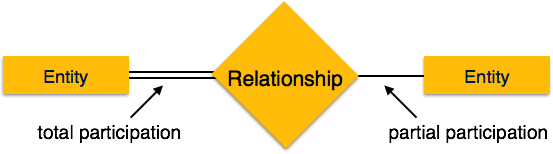


Fig: 4.6

CAB BOOKING SYSTEM

SOURCE/DESTINATION

DATE/TIME

TYPE OF CAB

FARE/KM & TOTAL FARE

PAYMENT OPTION

USER

ADMINISTRATOR

CUSTOMER NAME,EMAIL-ID NEME,ADDRESS AND MODE OF PAYMENT

REPORT

**4.4 Data Flow Diagram**

#### Data flow diagram is a graphical representation of the flow of data through business functions or processes. More generally, a data flow diagram is use for visualization of data processing. It illustrates the processes, data store and external entity, data flow in a business or other system and the relationship between these things. Physical DFD’s represent physical files and transactions, while logical or concept

#### DFD’s can be used to represent functions or processes.

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

0‘ level DFD

Reserve Taxi

Customer

Info

Booking a Taxi Fare Enquiry

Cancel Booking

ADMINISTRATOR Customer List

Types of Taxis Report

**1‘ level DFD**

Operator

Taxi Info Type of Payment Customer Info

# Admin

# User Entry Operator Customer Choice Taxi Driver Info

# Guest user id

# View Reports,Bills etc

#### 4.4.1Definition of Analysis and Design:

#### Analysis and design refers to the process of examining a business situation with the intent of improving it through better procedures and methods.

#### 4.4.2Overview of Analysis and Design:

#### System development can generally be thought of as having two major components - analysis and design

#### Introduction of User:

#### The term user is widely used in the system analysis and design. The term end-user is widely used by the analysts to refer to people who are not professional information systems specialists but who can use computers to perform their Jobs. We can group end-user into four categories.

#### Hands-on Users actually interact with the system. They feed in data or receive output, perhaps using a terminal.

#### Indirect Users benefit from the results or reports produced by these systems but do not directly interact with the hardware or software.

#### These users may be managers of business functions using the system.

#### End-Users are not alike. Some are intermittent users. The end-user can also be a competitor, not an employee of the firm. User managers have management responsibilities for application systems.

#### Definition of System

#### In the broad sense, a system is simply a set of components that interact to accomplish some purpose. Systems are all around us. As computers are used more and more by persons who are not computer professionals , the face of systems development is taking on an additional dimension. Users themselves are undertaking development of some of the systems they use , as the executive in the vignette emphasized.

#### These different situations are represented by three distinct approaches to the development of computer information systems:

#### Systems Development Life Cycle.

#### Structured Analysis Development Method.

#### Systems Prototype Method.

#### Systems Development Life Cycle:

#### Systems development, a process consisting of two major steps of systems analysis and design, starts when management or sometimes systems development personnel realize that a particular business system needs improvement.

#### SDLC is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system.

#### Different parts of the project can be in various phases at the same time, with some components undergoing analysis while others advanced stages.

#### SDLC consists of following activities:

#### Preliminary investigation.

#### Determination of system requirements.

#### Design of system.

#### Development of software.

#### System testing.

#### Implementation and evaluation.

#### Two main steps of development are:

#### 1.Analysis

#### 2.Design

#### 1.Analysis:

#### Objectives:

#### System analysis is conducted with the following objectives in mind:

#### Identify the user’s need.

#### Evaluate the system concept for feasibility.

#### Perform economic and technical analysis.

#### Allocate functions to hardware, software, people, database, and other system elements.

#### Establish cost and schedule constraints.

#### Create a system definition that forms the foundation for all subsequent engineering work. Both hardware and software expertise are required to successfully attain the objectives listed above.

**CHAPTER 5**

**SYSTEM DEVELPOMENT**

* **5.1introductionof PHP**
* **PHP** is a [scripting](https://en.wikipedia.org/wiki/Scripting_programming_language) language designed to fill the gap between [SSI](https://en.wikipedia.org/wiki/Server_Side_Includes) (Server Side Includes) and [Perl](https://en.wikibooks.org/wiki/Perl), intended for the Web environment. Its principal application is the implementation of Web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C-like syntax, and its simplicity. The newest version of PHP is 5.6 and it is heavily recommended to always use the newest version for better security, performance and of course features.
* If you've been to a website that prompts you to login, you've probably encountered a server-side scripting language. Due to its market saturation, this means you've probably come across PHP. [PHP](https://en.wikipedia.org/wiki/PHP) was designed by[RasmusLerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) to display his resume online and to collect data from his visitors.
* Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "**P**HP: **H**ypertext**P**reprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications that can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL.
* Before you embark on the wonderful journey of Server Side Processing, it is recommended that you have a basic understanding of the [HyperTextMarkup Language (HTML)](https://en.wikibooks.org/wiki/HTML" \o "HTML). But [PHP](https://en.wikipedia.org/wiki/PHP) can also be used to build [GUI](https://en.wikipedia.org/wiki/GUI)-driven applications

#### 5.2 PHP Features

* As you may have realized, the PHP language revolves around the central theme of practicality. PHP is about providing the programmer with the necessary tools to get the job done in a quick and efficient fashion. Five important characteristics make PHP’s practical nature possible:
* Familiarity
* • Simplicity
* • Efficiency
* • Security
* • Flexibility
* One final characteristic makes PHP particularly interesting: it’s free!
* **5.2.1Familiarity**
* Programmers from many backgrounds will find themselves already accustomed to the PHP language. Many of the language’s constructs are borrowed from C and Perl, and in many cases PHP code is almost indistinguishable from that found in the typical C or Pascal program. This minimizes the learning curve considerably.
* **5.2.2Simplicity**
* A PHP script can consist of 10,000 lines or one line: whatever you need to get the job done. There is no need to include libraries, special compilation directives, or anything of the sort. The PHP engine simply begins executing the code after the first escape sequence (). If the code is syntactically correct, it will be executed exactly as it is displayed.
* **5.2.3Efficiency**
* Efficiency is an extremely important consideration for working in a multiuser environment such as the WWW. PHP 4.0 introduced resource allocation mechanisms and more pronounced support for object-oriented programming, in addition to session management features. Reference counting has also been introduced in the latest version, eliminating unnecessary memory allocation.
* **5.2.4Security**
* PHP provides developers and administrators with a flexible and efficient set of security safeguards. These safeguards can be divided into two frames of reference: system level and application level.
* *- System-Level Security Safeguards :*
* PHP furnishes a number of security mechanisms that administrators can manipulate, providing for the maximum amount of freedom and security when PHP is properly configured. PHP can be run in what is known as safe mode, which can limit users’ attempts to exploit the PHP implementation in many important ways.
* Limits can also be placed on maximum execution time and memory usage, which if not controlled can have adverse affects on server performance. Much as with a cgi-bin folder, administrators can also place restrictions on the locations in which users can view and execute PHP scripts and use PHP scripts to view guarded server information, such as the passwd file.
* *- Application-Level Security Safeguards :*
* Several trusted data encryption options are supported in PHP’s predefined function set. PHP is also compatible with many third-party applications, allowing for easy-integration with secure ecommerce technologies. Another advantage is that the PHP source code is not viewable through the browser because the script is completely parsed before it is sent back to the requesting user. This benefit of PHP’s server-side architecture prevents the loss of creative scripts to users at least knowledgeable enough to execute a ‘View Source’.
* Security is such an important issue that this book contains an entire chapter on the subject.
* **5.2.5Flexibility**
* Because PHP is an embedded language, it is extremely flexible towards meeting the needs of the developer. Although PHP is generally touted as being used in conjunction solely with HTML, it can also be integrated alongside languages like JavaScript, WML, XML, and many others. Additionally, as with most other mainstream languages, isely planned PHP applications can be easily expanded as needed.
* Browser dependency is not an issue because PHP scripts are compiled entirely on the server side before being sent to the user. In fact, PHP scripts can be sent to just about any kind of device containing a browser, including cell phones, personal digital assistant (PDA) devices, pagers, laptops, not to mention the traditional PC. People who want to develop shell-based applications can also execute PHP from the command line.
* **5.2.6 Free**
* The open source development strategy has gained considerable notoriety in the software industry. The prospect of releasing source code to the masses has resulted in undeniably positive outcomes for many projects, perhaps most notably Linux, although the success of the Apache project has certainly been a major contributor in proving the validity of the open source ideal. The same holds true for the developmental history of PHP, as users worldwide have been a huge factor in the advancement of the PHP project.
* PHP’s embracing of this open source strategy result in great performance gains for users, and the code is available free of charge. Additionally, an extremely receptive user community numbering in the thousands acts as “customer support,” providing answers to even the most arcane questions in popular online discussion groups.
* 5.3INTRODUCTION OF SQL SERVER
* SQL (Structured Query Language) is a database sublanguage for querying and modifying relational databases. It was developed by IBM Research in the mid 70's and standardized by ANSI in 1986. The Relational Model defines two root languages for accessing a relational database -- Relational Algebra and Relational Calculus. Relational Algebra is a low-level, operator-oriented language. Creating a query in Relational Algebra involves combining relational operators using algebraic notation. Relational Calculus is a high-level, declarative language. Creating a query in Relational Calculus involves describing what results are desired.
* SQL is a version of Relational Calculus. The basic structure in SQL is the statement. Semicolons separate multiple SQL statements.

There are 3 basic categories of SQL Statements:

#### SELECT \* is inefficient, particularly when you are only using a few of the columns in the table. This is because it actually makes TWO queries to the database: before it runs your query, it has to query the system tables to determine the name and datatypes of the columns. It is much more efficient to NAME your columns in the SQL query, and this will also help in having your column names right there… so you don’t have to keep flipping back and forth between ASP page and database. In addition, this will prevent ambiguous column names in your resultset (in the case where both or all tables in the JOIN statement have columns with the same name). Further still, your code can break if it relies on ordinal position, and then someone inserts a column at the top of the table (which you can do in Enterprise Manager, or by dropping / re-creating the table). This is also a strong case for always defining columns in the INSERT list.

#### And finally, here’s another reason to avoid SELECT \* : Memo/Text columns, as well as columns containing BLOB data. Microsoft recommends to put BLOB/text columns at the end of the SELECT statement, and if there is more than one, in order of appearance in the table. This is also applicable to VARCHAR columns in SQL Server with a length greater than 255 characters.

#### Microsoft SQL Server 2000 extends the performance, reliability, quality, and ease-of-use of Microsoft SQL Server version 7.0. Microsoft SQL Server 2000 includes several new features that make it an excellent database platform for large-scale online transactional processing (OLTP), data warehousing, and e-commerce applications.

#### The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2012 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. For more information,

#### The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services. For more information,

#### The What's New topics contain brief overviews of the new features and links to relevant conceptual topics that provide more detailed information. These conceptual topics provide links to topics that describe the commands or statements you use to work with these features.

#### 5.3.1[XML Support](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### The relational database engine can return data as Extensible Markup Language (XML) documents. Additionally, XML can also be used to insert, update, and delete values in the database. For more information,

#### 5.3.2[Federated Database Servers](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### SQL Server 2000 supports enhancements to distributed partitioned views that allow you to partition tables horizontally across multiple servers. This allows you to scale out one database server to a group of database servers that cooperate to provide the same performance levels as a cluster of database servers. This group, or federation, of database servers can support the data storage requirements of the largest Web sites and enterprise data processing systems.

#### SQL Server 2000 introduces Net-Library support for Virtual Interface Architecture (VIA) system-area networks that provide high-speed connectivity between servers, such as between application servers and database servers. For more information,

#### [5.3.3 User-Defined Functions](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### The programmability of Transact-SQL can be extended by creating your own Transact-SQL functions. A user-defined function can return either a scalar value or a table. For more information, .

#### [5.3.4 Indexed Views](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### Indexed views can significantly improve the performance of an application where queries frequently perform certain joins or aggregations. An indexed view allows indexes to be created on views, where the result set of the view is stored and indexed in the database. Existing applications do not need to be modified to take advantage of the performance improvements with indexed views.

#### [5.3.5 New Data Types](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### SQL Server 2000 introduces three new data types. bigint is an 8-byte integer type. sql\_variant is a type that allows the storage of data values of different data types. table is a type that allows applications to store results temporarily for later use. It is supported for variables, and as the return type for user-defined functions. For more information

#### 5.3.6[Cascading Referential Integrity Constraints](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### You can control the actions SQL Server 2000 takes when you attempt to update or delete a key to which existing foreign keys point. This is controlled by the new ON DELETE and ON UPDATE clauses in the REFERENCES clause of the CREATE TABLE and ALTER TABLE statements. For more information, .

#### [5.3.7 Full-Text Search Enhancements](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### Full-text search now includes change tracking and image filtering. Change tracking maintains a log of all changes to the full-text indexed data. You can update the full-text index with these changes by flushing the log manually, on a schedule, or as they occur, using the background update index option. Image filtering allows you to index and query documents stored in image columns. The user provides the document type in a column that contains the file name extension that the document would have had if it were stored as a file in the file system. Using this information, full-text search is able to load the appropriate document filter to extract textual information for indexing. For more information .

#### [5.3.8 Multiple Instances of SQL Server](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### SQL Server 2012 supports running multiple instances of the relational database engine on the same computer. Each computer can run one instance of the relational database engine from SQL Server version 6.5 or 7.0, along with one or more instances of the database engine from SQL Server 2000. Each instance has its own set of system and user databases. Applications can connect to each instance on a computer similar to the way they connect to instances of SQL Servers running on different computers. The SQL Server 2000 utilities and administration tools have been enhanced to work with multiple instances. For more information, .

#### [5.3.9 Index Enhancements](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### You can now create indexes on computed columns. You can specify whether indexes are built in ascending or descending order, and if the database engine should use parallel scanning and sorting during index creation. For more information,

#### 5.3.10[Distributed Query Enhancements](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### SQL Server 2012 introduces a new OPENDATASOURCE function, which you can use to specify ad hoc connection information in a distributed query. SQL Server 2000 also specifies methods that OLE DB providers can use to report the level of SQL syntax supported by the provider and statistics on the distribution of key values in the data source. The distributed query optimizer can then use this information to reduce the amount of data that has to be sent from the OLE DB data source. SQL Server 2012 delegates more SQL operations to OLE DB data sources than earlier versions of SQL Server. Distributed queries also support the other functions introduced in SQL Server 2012, such as multiple instances, mixing columns with different collations in result sets, and the new bigint and sql\_variant data types.

#### SQL Server 2012 distributed queries add support for the OLE DB Provider for Exchange and the Microsoft OLE DB Provider for Microsoft Directory Services.

#### 5.3.11[Backup and Restore Enhancements](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### SQL Server 2000 introduces a new, more easily understood model for specifying backup and restore options. The new model makes it clearer that you are balancing increased or decreased exposure to losing work against the performance and log space requirements of different plans. SQL Server 2000 introduces support for recovery to specific points of work using named log marks in the transaction log, and the ability to do partial database restores

#### Users can define passwords for backup sets and media sets that prevent unauthorized users from accessing SQL Server backups.

#### 5.3.12[Text in Row Data](mk:@MSITStore:D:\Program%20Files\Microsoft%20SQL%20Server\80\Tools\Books\whatsnew.chm::/wn_whatnew_6a2b.htm##)

#### SQL Server 2000 supports a new text in row table option that specifies that small text, ntext, and image values be placed directly in the data row instead of in a separate page. This reduces the amount of space used to store small text, ntext, and image data values, and reduces the amount of disk I/O needed to process these values. For more information,

#### 5.4SYSTEM ANALYSIS

#### The analysis must achieve three primary objectivs:

#### 1 .To describe the requirement of the customer

#### 2 To establish the basis for the creation of the software design

#### 3. To define set of requirements that can be validated once the software is built.

#### 5.4.1 An Overview to system analysis

#### The system analysis phase is considered to be one of the most important phases in the system development life cycle. It is immensely important that the software developer make through study of the existing system. Thorough study of the system is made and need i.e. features that are critical to system success and users wants (i.e. features that would be good but not essential) are brought out. The study will enable the developer to know the intricacies of the existing system.

#### Requirement analysis is done in order to understand the problem which the S/W system is to solve e.g., the problem could be automating the existing manual system or developing a completely new automated system or a combination of the two. For large systems having a large number of features and the need to perform many different tasks, understanding the requirement of the system is a major task. The emphasis in requirement analysis is on identifying what is needed from the system, and not how the system achieves its goal.

#### The main objective behind any business organization is to maximize its profit besides maintaining quality and strategic norms. This can be achieved by improving the efficiency of the system by providing more facilities using automation, by adopting faster data access, proper communication.

#### Since Computer Craft is an educational institute, whereas its main objective behind automation is not only to maximize profit but also to take care of students’ interest by providing coaching of latest courses benefiting students, in turn the country.

#### The most important objective behind automation is to minimize Paper Work. Paper Work/Registers are replaced by a Centralized Data Bank, which is well equipped to store / provide information as and when required. Data Bank also helps speed up the communication between various depts. / outside agencies, as there is no need of making request against different departments for a specific data and to wait for it for a long period. This also improves the efficiency as it saves time and human resources.

#### By making the manual system computerized, we can ensure complete utilization of our existing resources. Automation helps in generating the reports / information in a consistent way, which saves time and labour if done manually .

#### 5.5Software Engineering Paradigm

#### The basic objective of software engineering is to: develop methods and procedures for software development that can scale up for large systems and that can be used to consistently produce high quality software at low cost and with a small cycle time. That is, the key objectives are consistency, low cost, high quality, small cycle time, and scalability.

#### The basic approach that software engineering takes is to separate the development process from the software. The premise is that the development process controls the quality, scalability, consistency, and productivity. Hence to satisfy the objectives, one must focus on the development process. Design of proper development process and their control is the primary goal of the software engineering. It is this focus on the process that distinguishes it from most other computing disciplines.

#### Most other computing disciplines focus on some type of the product-algorithms, operating systems, databases etc. while software engineering focuses on the process for producing products. To better manage the development process and to achieve consistency, it is essential that the software development be done in phases.

#### 

#### Fig: 5.1

#### 5.6Different Phases of The Development Process

#### 5.6.1 Requirement Analysis:

#### Requirement analysis is done in order to understand the problem the software system is to solve. The problem could be automating an existing manual process, developing a new automated system, or a combination of the two. The emphasis in requirements analysis is on identifying what is needed from the system, not how the system will achieve its goals. There are atleast two parties involved in the software development-a client and a developer. The developer has to develop the system to satisfy the client’s needs. The developer does not understand the client’s problem domain, and the client does not understand the issues involved in the software systems. This causes a communication gap, which has to be adequately bridged during requirements analysis.

#### 5.6.2 Software Design:

#### The purpose of the design phase is to plan a solution of the problem specified by the requirements documents. This phase is the first step in moving from the problem domain to the solution domain. Starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software. It has a major impact on the later phases, particularly testing and maintenance.

#### The design activity is divided into two phases: System Design and Detailed Design. In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules.

#### 5.6.3 Coding

#### The goal of the coding phase is to translate the design of the system into code in a given programming language. Hence during coding, the focus should be on developing programs that are easy to read and understand, and not simply on developing programs that are easy to write.

#### 5.6.4 Testing

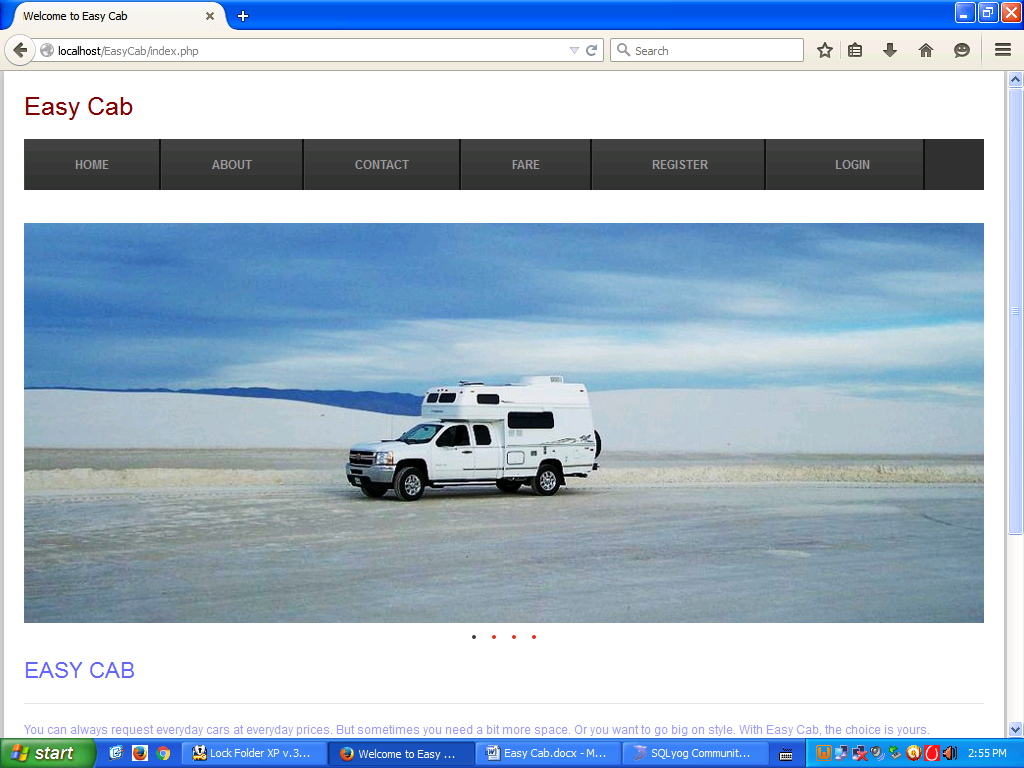
#### Testing is the major quality control measure used during software development. Its basic function is to detect errors in the software. Testing not only uncover errors introduced during coding, but also errors introduced during the previous phases. Thus, the goal of the testing is to uncover requirement, design and coding errors in the programs. Therefore, different levels of testing are used. Testing is an extremely critical and time consuming activity. It requires proper planning of the overall testing process. The output of the testing phase is the test report and the error report. Test report contains the set of test cases and the result of executing the code with these test cases. The error report describes the errors encountered and the action taken to remove the errors.

#### 5.6.5Operation & maintenance phase:

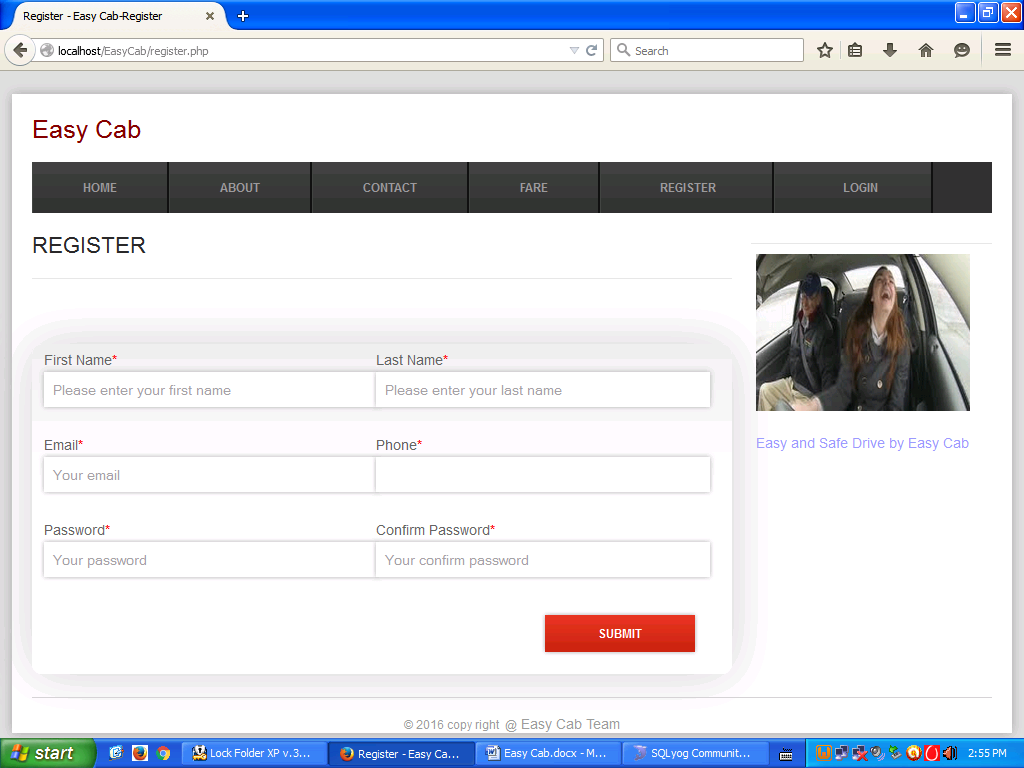
#### Software maintenance is a task that every development group has to face, when the software is delivered to the customer‘s site, installed and is operational. Software maintenance is a very broad activity that includes error correction, enhancement of capabilities, deletion of obsolete capabilities and optimization.

**CHAPTER 6**

**6.1 home page**



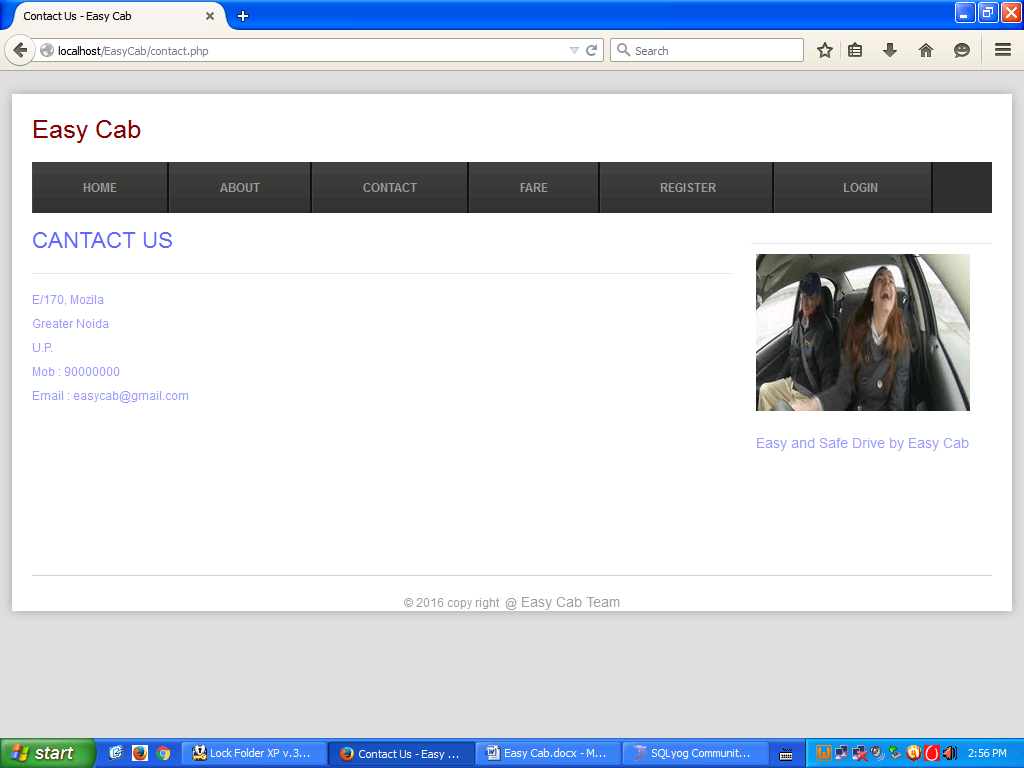
**6.2 new user**

****

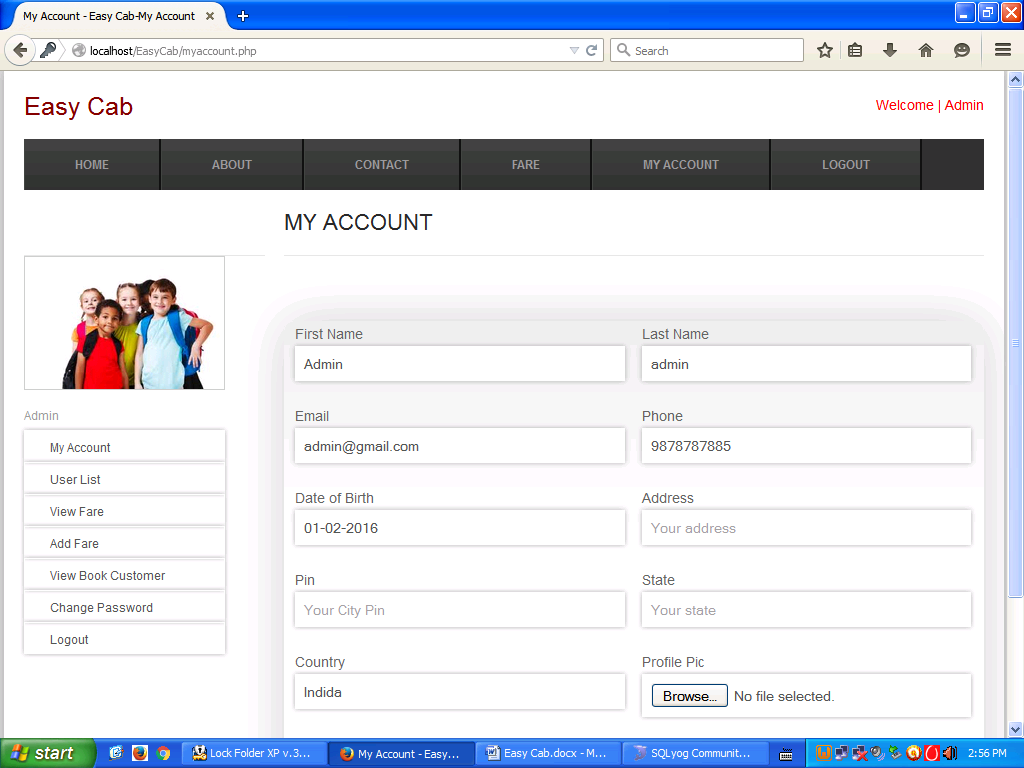
**6.3 about us**

****

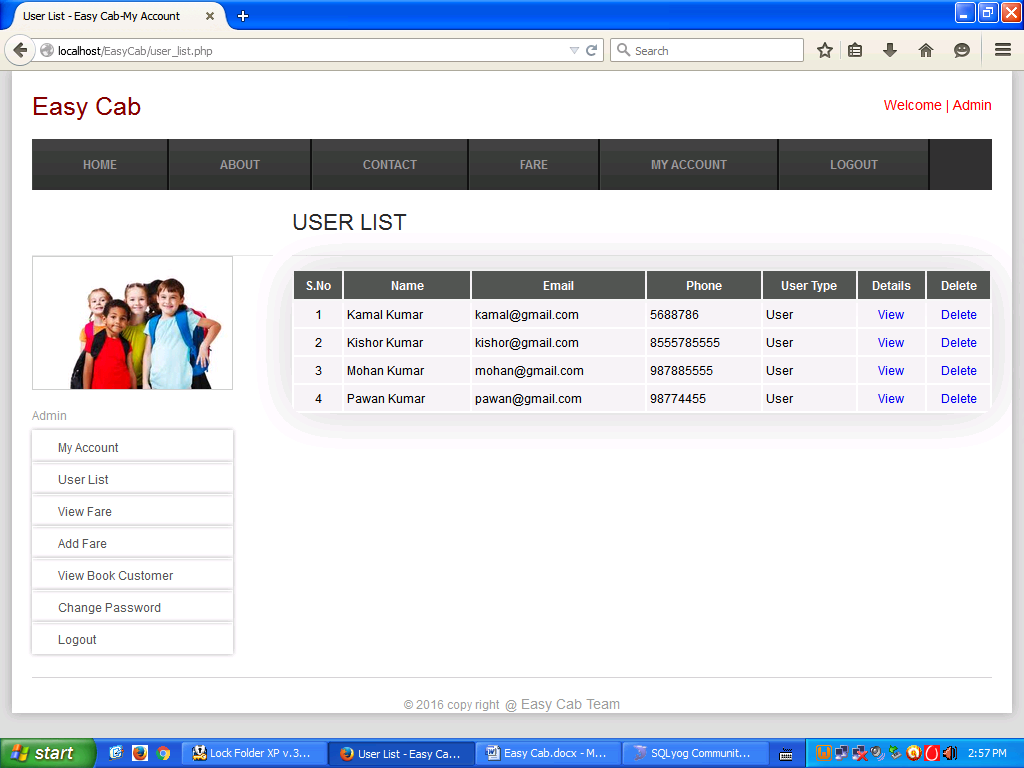
**6.3 Contact us**

****

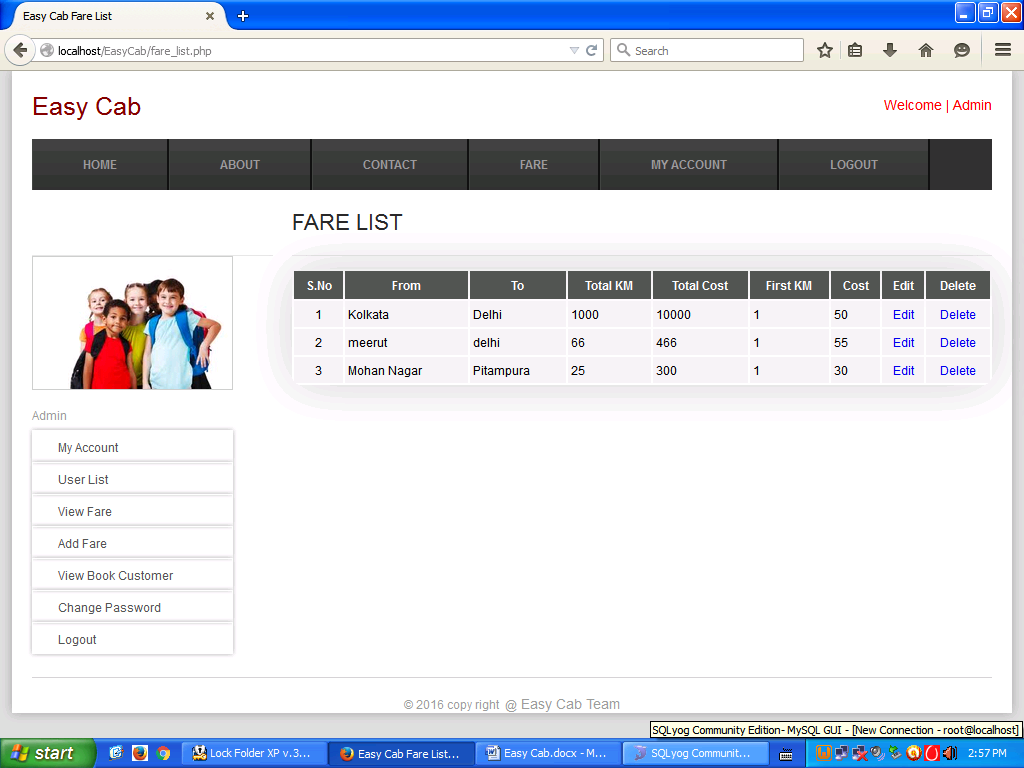
**Admin Home Page**

****

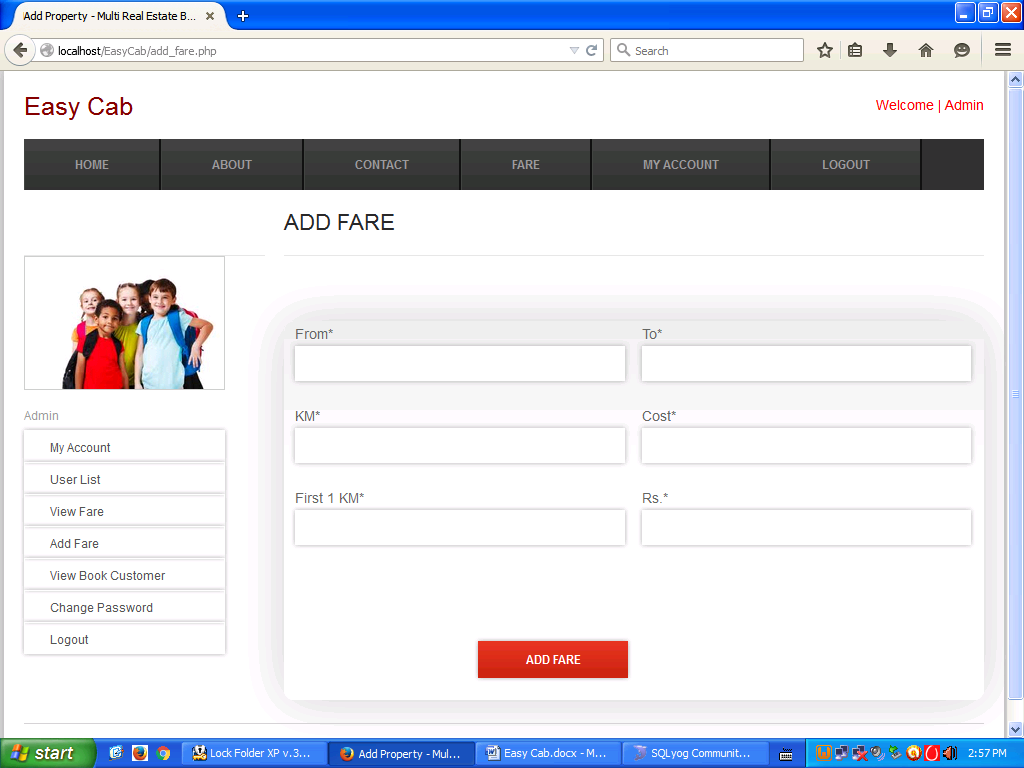
**User List**

****

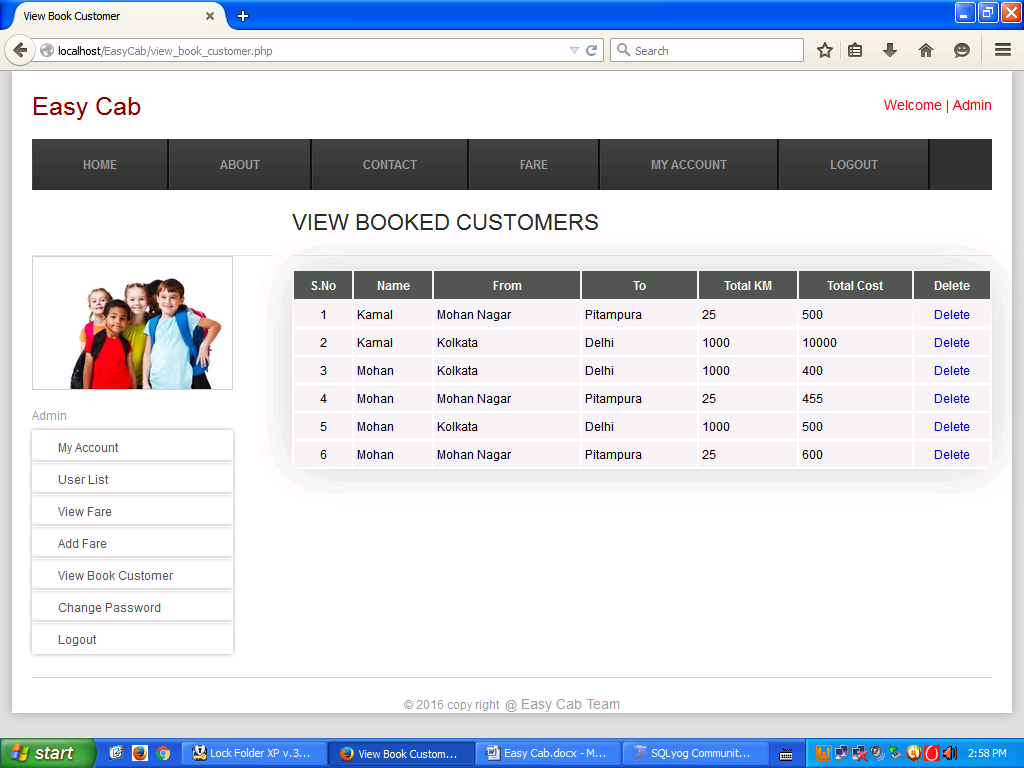
**View fare**

****

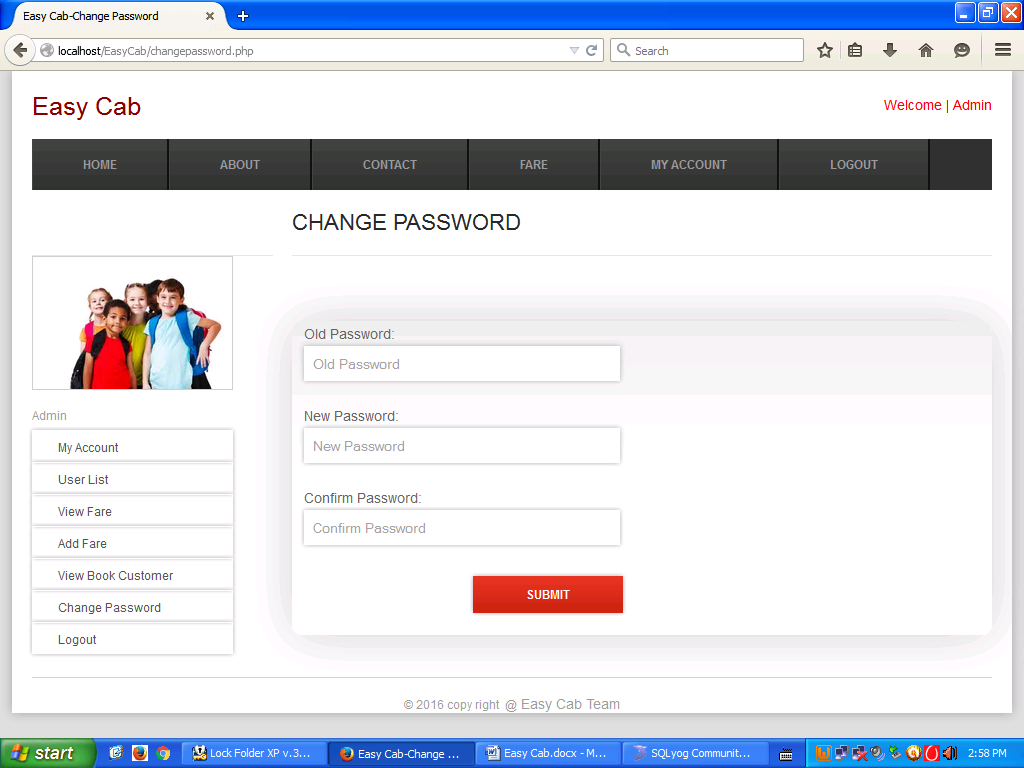
**Add Fare**

****

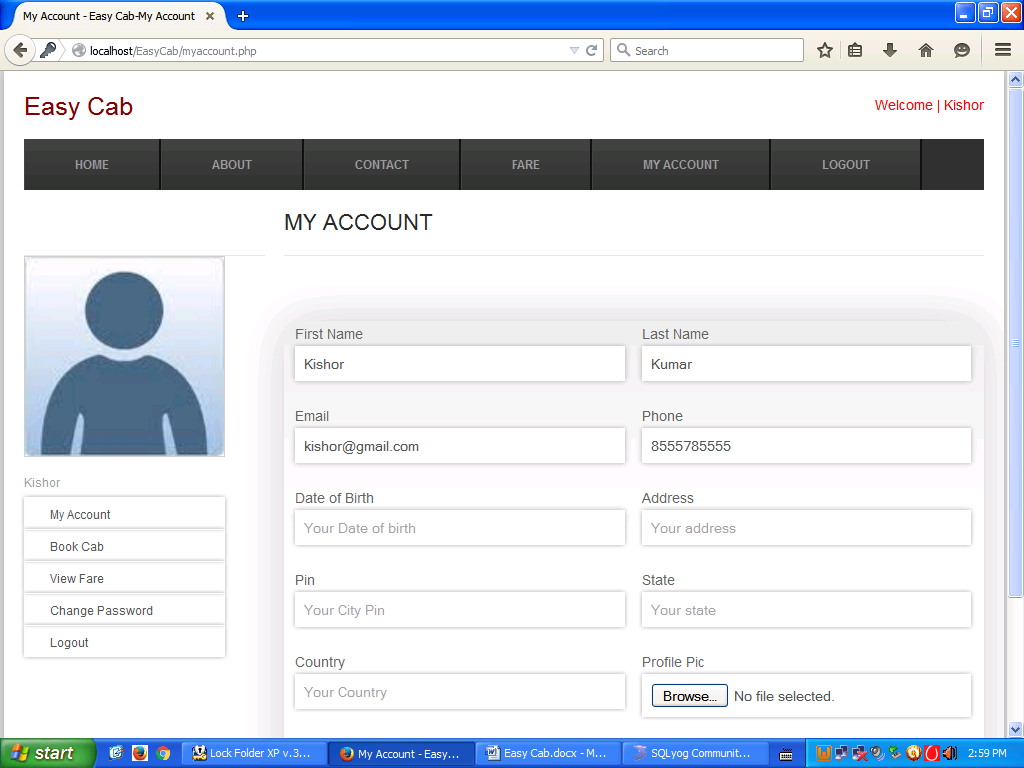
**View Booked Customer**

****

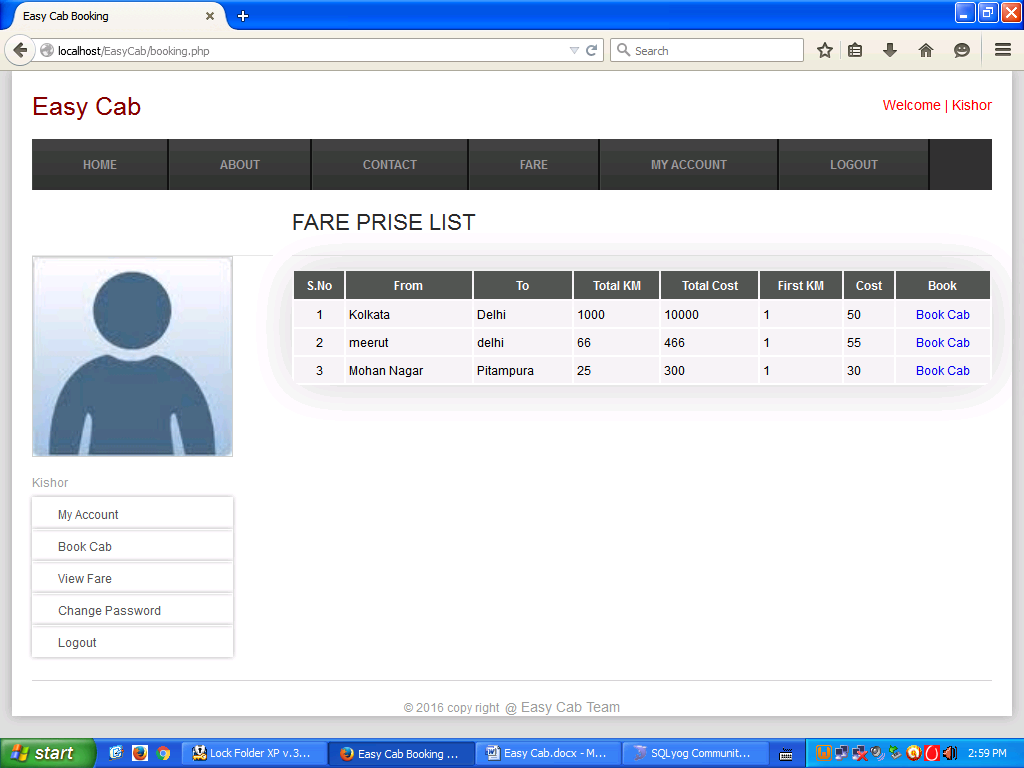
**Change Password**

****

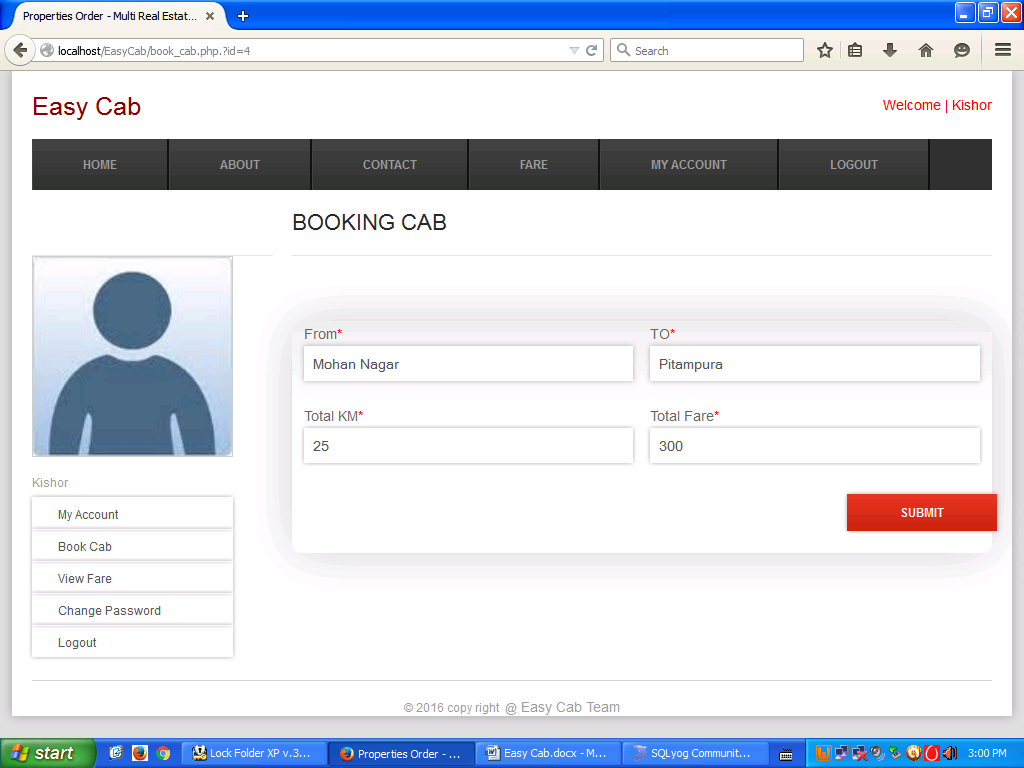
**Customer Home Page**

****

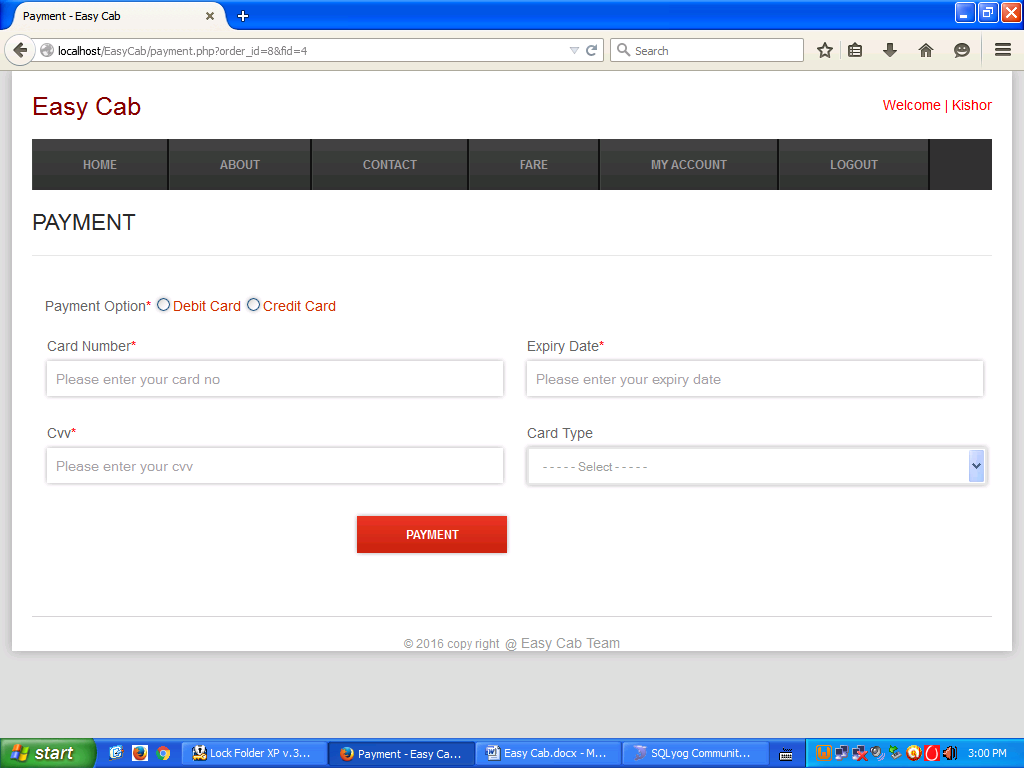
**Fare price list**

****

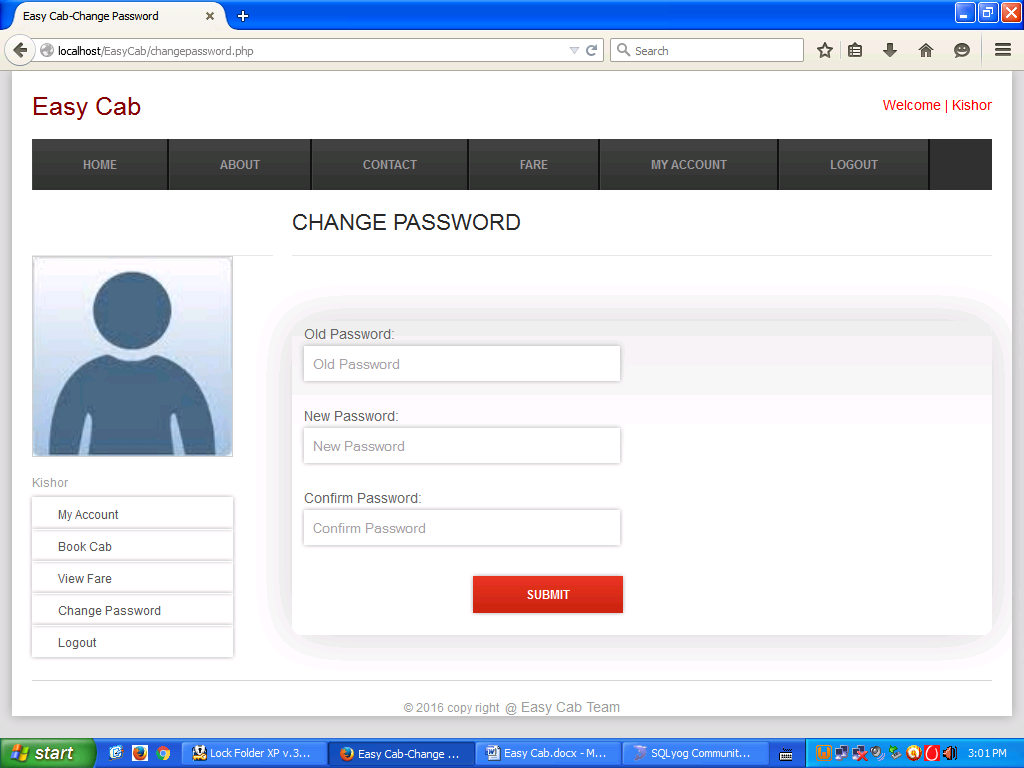
**Book Cab**

****

**Payment**

****

**Change Password**

****

#### 

#### Chapter 7

#### System Implementation

#### 7.1 Hardware Requirement

#### 7.2 Software Requirement

#### 7.1 Hardware

#### Hardware is the term given to machinery itself and to various individual pieces of equipment.

#### It refers to the physical devices of a computer system. Thus the input, storage, processing control and output devices are hardware.

#### Minimum Hardware Requirement of Client Side:

#### Processor : Any Pentium IV or Equivalent Machine

#### RAM : 256 MB

#### HDD : 40 GB

#### FDD : 1.44 MB

#### CD-ROM : 52X

#### 15 inches Color Monitor

#### 104 Keys Keyboards

#### Printer : DeskJet 670 C

#### 7.2 Software Requirement

#### Software means a collection of program where the objective is to enhance the capabilities of the hardware machine.

#### Minimum Software Requirement of Client Side:

#### Operating System : Windows 2007/2008

#### Database : MS – Sql Server 2012

**CHAPTER 8**

*8.1 System Testing*

***It should be clear in mind that the philosophy behind testing is to find errors. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as normal input. However, the data are created with the express intent of determining whether the system will process them correctly. For example, test cases for inventory handling should include situations in which the quantifies to be withdrawn from inventory exceed, equal and are less than the actual quantities on hand. Each test case is designed with the intent of finding errors in the way the system will process it. There are two general strategies for testing software: Code testing and Specification testing. In code testing, the analyst develops that cases to execute every instructions and path in a program. Under specification testing, the analyst examines the program specifications and then writes test data to determine how the program operates under specific conditions. Regardless of which strategy the analyst follows, there are preferred practices to ensure that the testing is useful. The levels of tests and types of test data, combined with testing libraries, are important aspects of the actual test process.***

*Levels of Testing*

***Systems are not designed as entire systems nor are they tested as single systems. The analyst must perform both unit and system testing.***

*8.1 Unit Testing:*

***In unit testing the analyst tests the programs making up a system. For this reason, unit testing is sometimes called program testing. Unit testing gives stress on the modules independently of one another, to find errors. This helps the tester in detecting errors in coding and logic that are contained within that module alone. The errors resulting from the interaction between modules are initially avoided. For example, a hotel information system consists of modules to handle reservations; guest checking and checkout; restaurant, room service and miscellaneous charges; convention activities; and accounts receivable billing. For each, it provides the ability to enter, modify or retrieve data and respond to different types of inquiries or print reports. The test cases needed for unit testing should exercise each condition and option.***

***Unit testing can be performed from the bottom up, starting with smallest and lowest-level modules and proceeding one at a time. For each module in bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embedded within the larger system.***

*8.2 System Testing*

***The important and essential part of the system development phase, after designing and developing the software is system testing. We cannot say that every program or system design is perfect and because of lack of communication between the user and the designer, some error is there in the software development. The number and nature of errors in a newly designed system depend on some usual factors like communication between the user and the designer; the programmer's ability to generate a code that reflects exactly the systems specifications and the time frame for the design.***

***Theoretically, a newly designed system should have all the parts or sub-systems are in working order, but in reality, each sub-system works independently. This is the time to gather all the subsystem into one pool and test the whole system to determine whether it meets the user requirements. This is the last change to detect and correct errors before the system is installed for user acceptance testing. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits.***

***Testing is an important function to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully activated. Another reason for system testing is its utility as a user-oriented vehicle before implementation.***

***System testing consists of the following five steps:***

***Program testing***

***String testing***

***System testing***

#### System documentation

#### User acceptance testing

#### 8.1.1 Program Testing:

#### A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. it is the responsibility of a programmer to have an error free program. At the time of testing the system, there exists two types of errors that should be checked. These errors are syntax and logic. A syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted key words are common syntax errors. These errors are shown through error messages generated by the computer. A logic error, on the other hand, deals with incorrect data fields out of range items, and invalid combinations. Since the logical errors are not detected by compiler, the programmer must examine the output carefully to detect them.

#### When a program is tested, the actual output is compared with the expected output. When there is a discrepancy, the sequence of the instructions, must be traced to determine the problem. The process is facilitated by breaking the program down into self-contained portions, each of which can be checked at certain key points.

#### 8.1.2 String Testing

#### Programs are invariably related to one another and interact in a total system. Each program is tested to see whether it conforms to related programs in the system. Each part of the system is tested against the entire module with both test and live data before the whole system is ready to be tested.

#### 8.1.3 System Testing

#### System testing is designed to uncover weaknesses that were not found in earlier tests. This includes forced system failure and validation of total system as it will be implemented by its user in the operational environment. Under this testing, generally we take low volumes of transactions based on live data. This volume is increased until the maximum level for each transaction type is reached. The total system is also tested for recovery and fallback after various major failures to ensure that no data are lost during the emergency. All this is done with the old system still in operation. When we see that the proposed system is successful in the test, the old system is discontinued.

#### 8.1.4 System Documentation:

#### All design and test documentation should be well prepared and kept in the library for future reference. The library is the central location for maintenance of the new system.

#### 8.1.5 User Acceptance Testing

#### An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and that the integrity of important data is maintained. Performance of an acceptance test is actually the user's show. User motivation is very important for the successful performance of the system. After that a comprehensive test report is prepared. This report shows the system's tolerance, performance range, error rate and accuracy.

#### 8.1.6 Special Systems Tests

#### There are other six tests which fall under special category. They are described below:

#### Peak Load Test: It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.

#### 8.1.7 Storage Testing

#### It determines the capacity of the system to store transaction data on a disk or in other files. For example, verify documentation statements that the system will store 10,000 records of 400 bytes length on a single flexible disk.

#### 8.1.8 Performance Time Testing

#### It determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.

#### 8.1.9 Recovery Testing

#### This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.

#### 8.1.10 Procedure Testing

#### It determines the clarity of documentation on operation and use of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.

#### 8.1.11 Human Factors Testing

#### It determines how users will use the system when processing data or preparing reports.

#### Test Case Specification for new user:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNo. | Test Case | Input | Expected Output | Actual Output |
| 1 | TS1: Successfullogin | User ID=userPassword=passwordConfirm password=password | New user account be successful | User ID=userPassword=passwordNew user account Successful |
| 2 | TS2: Unsuccessful login | User ID=admPassword=admjfConfirm pasword=admjf | New user account should fail with an error “Invalid ID or Password” | User ID=admPassword=admjfInvalid User ID or Password |
| 3 | TS3 | Password<10 | New user account should fail with an error “password” is invalid | New user account fail with an invalid password |
| 4 | TS2 | Password<=10 | Password should be valid | Password is correct |
| 5 | TS2 | Contact no.<10 | Contact no. should be valid | Contact no. is correct |

#### Test Case Specification for Login:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNo. | Test Case | Input | Expected Output | Actual Output |
| 1 | TS1: Successfullogin | User ID=userPassword=password | Login should be successful | User ID=userPassword=passwordLogin Successful |
| 2 | TS2: Unsuccessful login | User ID=admPassword=admjf | Login should fail with an error “Invalid ID or Password” | User ID=admPassword=admjfInvalid User ID or Password |
| 3 | TS2 | User ID=blankPassword=blank | Login should fail with an error “ID and Password can’t be blank” | User ID=blankPassword=blankUser ID and Password can’t be blank |
| 4 | TS2 | User ID=blankPassword=password | Login should fail with an error “ID can’t be blank” | User ID=blankPassword=passwordUser ID can’t be blank |
| 5 | TS2 | User ID=200909Password=blank | Login should fail with an error “Password can’t be blank” | User ID=200909Password=blankPassword can’t be blank |

**8.2 Debugging**

Debugging is a necessary process in almost any new [software](http://searchsoa.techtarget.com/definition/software) or hardware development process, whether a commercial product or an enterprise or personal application program. For complex products, debugging is done as the result of the unit test for the smallest unit of a system, again at component test when parts are brought together, again at system test when the product is used with other existing products, and again during customer [beta test](http://searchcio-midmarket.techtarget.com/definition/beta-test), when users try the product out in a real world situation. Because most computer programs and many programmed hardware devices contain thousands of lines of code, almost any new product is likely to contain a few bugs. Invariably, the bugs in the functions that get most use are found and fixed first. An early version of a program that has lots of bugs is referred to as "buggy."

Debugging tools (called *debugger*s) help identify coding errors at various development stages. Some programming language packages include a facility for checking the code for errors as it is being written.

#### 8.2.1Debugging by Induction

Many errors can be found by using a disciplined thought process without ever going near the computer. One such thought process is induction, where one proceeds from the particulars to the whole. By starting with the symptoms of the error, possibly in the result of one or more test cases, and looking for relationships among the symptoms, the error is often uncovered.

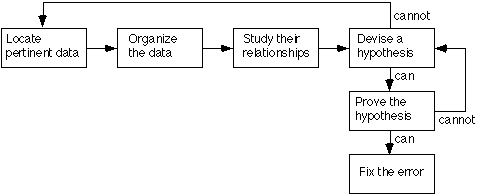
The induction process is illustrated in Figure 1 and described by Myers as follows:

* **Locate the pertinent data.** A major mistake made when debugging a program is failing to take account of all available data or symptoms about the problems. The first step is the enumeration of all that is known about what the program did correctly, and what it did incorrectly (i.e., the symptoms that led one to believe that an error exists). Additional valuable clues are provided by similar, but different, test cases that do not cause the symptoms to appear.
* **Organize the data.** Remembering that induction implies that one is progressing from the particulars to the general, the second step is the structuring of the pertinent data to allow one to observe patterns, of particular importance is the search for contradictions (i.e., "the errors occurs only when the pilot perform a left turn while climbing"). A particularly useful organizational technique that can be used to structure the available data is shown in the following table. The "What" boxes list the general symptoms, the "Where" boxes describe where the symptoms were observed, the "When" boxes list anything that is known about the times that the symptoms occur, and the "To What Extent" boxes describes the scope and magnitude of the symptoms. Notice the "Is" and "Is Not" columns. They describe the contradictions that may eventually lead to a hypothesis about the error.
* **Devise a hypothesis.**

The next steps are to study the relationships among the clues and devise, using the patterns that might be visible in the structure of the clues, one or more hypotheses about the cause of the error. If one cannot devise a theory, more data are necessary, possibly obtained by devising and executing additional test cases. If multiple theories seem possible, the most probable one is selected first.

* **Prove the hypothesis.**

 A major mistake at this point, given the pressures under which debugging is usually performed, is skipping this step by jumping to conclusions and attempting to fix the problem. However, it is vital to prove the reasonableness of the hypothesis before proceeding. A failure to do this often results in the fixing of only a symptom of the problem, or only a portion of the problem. The hypothesis is proved by comparing it to the original clues or data, making sure that this hypothesis completely explains the existence of the clues. If it does not, either the hypothesis is invalid, the hypothesis is incomplete, or multiple errors are present.

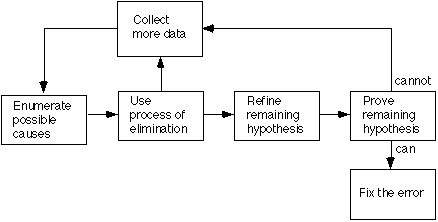


**Figure 8.1 Inductive Debugging Process**

#### 8.2.2Debugging By Deduction

#### An alternate thought process, that of deduction, is a process of proceeding from some general theories or premises, using the processes of elimination and refinement, to arrive at a conclusion. This process is illustrated in Figure 2 and also described by Myers as follows:

* **Enumerate the possible causes or hypotheses.** The first step is to develop a list of all conceivable causes of the error. They need not be complete explanations; they are merely theories through which one can structure and analyze the available data.
* **Use the data to eliminate possible causes.** By a careful analysis of the data, particularly by looking for contradictions (the previous table could be used here), one attempts to eliminate all but one of the possible causes. If all are eliminated, additional data are needed (e.g., by devising additional test cases) to devise new theories. If more than one possible cause remains, the most probable cause (the prime hypothesis) is selected first
* **Refine the remaining hypothesis.** The possible cause at this point might be correct, but it is unlikely to he specific enough to pinpoint the error. Hence, the next step is to use the available clues to refine the theory to something more specific.
* **Prove the remaining hypothesis.** This vital step is identical to the fourth step in the induction method.



**Figure 8.2 Deductive Debugging Process**

#### 8.2.3Debugging by Backtracking

For small programs, the method of backtracking is often used effectively in locating errors. To use this method, start at the place in the program where an incorrect result was produced and go backwards in the program one step at a time, mentally executing the program in reverse order, to derive the state (or values of all variables) of the program at the previous step. Continuing in this fashion, the error is localized between the point where the state of the program was what was expected and the first point where the state was not what was expected.

#### 8.2.4Debugging by Testing

The use of additional test cases is another very powerful debugging method which is often used in conjunction with the induction method to obtain information needed to generate a hypothesis and/or to prove a hypothesis and with the deduction method to eliminate suspected causes, refine the remaining hypothesis, and/or prove a hypothesis.

The test cases for debugging differ from those used for integration and testing in that they are more specific and are designed to explore a particular input domain or internal state of the program. Test cases for integration and testing tend to cover many conditions in one test, whereas test cases for debugging tend to cover only one or a very few conditions. The former are designed to detect the error in the most efficient manner whereas the latter are designed to isolate the error most efficiently.

#### CHAPTER : 9

#### SYSTEM MAINTENANCE

#### Chapter 9: System Maintenance

#### System Maintenance

#### Once the software is delivered and developed, it enters the maintenance phase. After implementation systems need maintenance. Beyond monkey testing during Software development some errors may not appear. During its usage by the end-user with actual data certain errors may disclose.

#### Though Maintenance is not a part of software development, it is an extremely important activity in the life of a software product. Maintenance involves understanding the existing software (code and related documents), understanding the effects of change, making the changes-to both the code and documents-testing the new parts and retesting the old part.

#### For successful and smooth running of the system, maintenance is the prominent part of the project. Any error, which hinders the functioning of any part of the project, may lead to bad impression of the developer.

#### There are majorly two types of errors: Compilation error and Runtime errors. Compilation errors are errors during coding and are to be taken care by the developer during development process.

#### Runtime errors are those which occurs during running of the program. Whenever there is an occurrence of error an ‘Error Window’ opens in the middle of the screen displaying the the type of error, Error Number and the Nearest Possible reason as to why the error has occurred. With the occurrence of this Error Window the operator (End-user) should note the type of error, the error number and the description of the error and should immediately report to the concerned Developer or Administrator.

#### Now comes the role of the Maintenance Personals. After knowing the entire details from the end-user like where or at which screen does this error occurred or what type of data was feeded by the user or the point of malfunctioning. Considering this error as the main reason for the malfunctioning the programmer now re-examines all the possible factors, which act behind the particular screen where error has occurred.

MAINTAIN TAXIS INFORMATION

Cab Driver

Administrator

Cab Menu

Database

Display Menu

Enter Taxi no &distance

Entered

Enter Source & Destn

Entered

Enter Dept & Arr TimeSubmit Details

Entered Get Details

Verify detail

Add/Delete/Modify Error / success

Taxi Message

Add/Delete/Modify

Update Taxi Detail

Maintain customer services

Enter Details

Not exist

Verification

If exist

ADD Customer

Delete Customer

Modify customer

Display Details

#### 9.1 Code Efficiency

#### For Code efficiency, the following conditions must be fulfilled:

#### 9.1.1 Option Explicit:

#### All variables must be declared with proper data type. Option Explicit keyword must be used, it forces each variable or expression in an application to be defined with a specific data type through the use of the Dim, Private, Public, Redim or Static keywords. If Option Explicit is not used, undefined variables are automatically defined as Variant. Variant data type must be avoided because it takes 16 bytes of memory.

#### 9.1.2 Early Binding Vs Late Binding:

#### Early binding is faster than Late Binding because the application doesn’t have to interrogate the object at runtime to determine the objects properties and methods. In Late Binding the objects are defined as objects. Late biding is slower than Early Binding because the application must interrogate the object to determine its properties and methods.

#### 9.1.3 Reclaiming Space:

#### Reclaim space from strings by setting a string variable to the empty string (“”) space can also be reclaim from object by setting an object variable to Nothing. It frees the space associated with the object if it is the last reference to the object. It also has the advantage of reducing the amount of cleanup work needed when the program terminates.

#### 9.1.4 Control Array:

#### Complex forms needs more memory. The more controls put on a form, the more memory it will require when it becomes loaded in to memory. Therefore, reducing the number of controls reduces the memory requirements. For this a control array must be used. A control array counts as only one name. Control arrays consumes fewer resources than the equivalent number of independent controls.

#### 9.1.5 Reducing Memory Requirement:

#### By using Dynamic Arrays we can reduce memory requirments by triming the array to just the elements we need. Use the Redim Statement to change the dimensions of the array. Erase statement can be used to free the memory associated with the dynamic array.

#### Use fewer graphics. Graphic images consumes memory. The more graphics we have, the more memory we are going to use.

#### Destroy forms when we finished with them. If we are finished with a form we can use the unload statement to remove the form from memory. This will free some memory resources. To free all of the form resources we must set forms object name to nothing.

#### In an effort to maintain computational efficiency and to allow the eventual adaptation of the algorithm to face tracking applications, intense optimization of the code has been performed. Although further development is in progress, the algorithm is currently fast and compact enough to run interactively on most generic platforms.

#### First, the sequential hierarchical search which proceeds from large scales to small scales. This allows a rapid convergence if the face is dominant in the image. Furthermore, the algorithm does not always flow through the complete loop. It stops as soon as one of the modules reports a failure and loops back to an earlier stage. For example, we do not search for a mouth if no eyes are found. In this case, no time is wasted in the mouth module.

#### Additionally, we utilized special programming techniques to reduce the run-time. For instance, wave propagation is used to generate the symmetry maps. This provides a computational efficiency that makes the symmetry operator a practical tool. The 3D normalization algorithm is also extremely efficient and uses look-up tables and minimal calculations for increased speed.

#### The execution times are measure on an SGI Indy machine which has a rating equivalent to that of a 1996 home personal computer. The efficiency of the code allows a face to be found in an image in less than 1 second if it is the dominant structure. However, we loop through all objects in the scene in an attempt to find all possible faces. Thus, the algorithm's loop is traversed multiple times even though a face could have been detected in an earlier iteration of the loop.

#### Future Application of The Project

The future of the Project looks more promising. The main advantage of this project is toIn addition, the online advertisements for jobs have also proved effective in grabbing the viewer’s attention. With online job search, gaining popularity among the masses, the Curriculum Vitae, better known as CV, has become extremely important. To catch the eye of the employer among thousands of other candidates, one needs to put in concerted efforts to make an attractive and job specific CV. The biggest advantage of online job search lies in the fact that the CV can be uploaded on a job portal and the interested employers can browse through numerous CVs and contact the best-suited candidate.

The job portals do not charge any fee from the prospective job seekers and allow them to upload the resumes for free. On the other hand, employers and other recruitment agencies are charged for accessing their database of resumes of suitable employees. Thus, the online job search portals have become the most preferred platform for both the employers and the employees to interact in a 2-way process for mutual benefit.

**Activity Diagram: REPORT AND TICKET GENERATION**

Generate report

Print Ticket

Closed

**CONCLUSION**

The aim of this project is to develop one Easy Cab Booking processes for all taxi services. Improve client experience with ordering of placement services. The main aim of this is to maintain the Cab Booking system using web forms to reduce the paper and printing and distribution work. It also includes manage your cab company requirement, monitor your booking, system. It provides a very friendly environment for both customer and admin which improve efficiency of order processing.

In future, we can have the SMS facility for the employees and employee search engine that can provide the result on the basis of different criteria to search. We can also have one more module of implementing this project on web.

**BIBLIOGRAPHY**

* [www.google.com](http://www.google.com)
* <https://en.wikipedia.org/wiki/PHP>
* <http://www.w3schools.com/php/php_intro.asp>
* <https://en.wikipedia.org/wiki/Feasibility_study>
* <https://www.extension.iastate.edu/agdm/wholefarm/html/c5-65.html>