**A**

**PROJECT REPORT**

**ON**

**“HOTEL MANAGEMENT SYSTEM”**

**BY**

**ASHLESHA PRAKASH VAKIL**

**UNDER GUIDENCE OF**

**Mrs. SUNANDA MULGUND**

**MRS. SANDHYA PANDEY.**

**UNIVERSITY OF MUMBAI**

**BACHELOR OF COMPUTER SCIENCE**

**G. R. PATIL COLLEGE OF ARTS, COMMERCE, SCIENCE**

**DOMBIVALI (E) 421201**

**ACADAMIC YEAR**

**2010-2011**



**UNIVERSITY OF MUMBAI**

G. R. Patil College Of Arts, Science & Commerce

(Affiliated to University Of Mumbai)

2010 – 2011

**Project Report On**

**HOTEL MANAGEMENT SYSTEM**

Project work submitted in partial fulfillment of the requirements for the award of the degree of

B.Sc.CS (Computer Science)

**SUBMITTED BY : ASHLESHA P. VAKIL**

**PROJECT GUIDE : MRS. SUNANDA MULGUND,**

**MRS. SANDHYA PANDEY.**

G. R. Patil College Of Arts, Science & Commerce

(Affiliated to University Of Mumbai)

2010 – 2011

**CERTIFICATE**

This is certify that this project entitled HOTEL MANAGEMENT SYSTEM is a Bonafied work done by ASHLESHA P. VAKIL in partial fulfillment of the requirements for the B.SC. Computer Science degree during the academic year 2010-2011 and is the original work of the candidate.

**PROJECT GUIDE:** MRS. SUNANDA MULGUND,

MRS. SANDHYA PANDEY.

Project Guide: EXTERNAL EXAMINAR:

­­­­­­­­­­­­­­­­­­­­­­­­­

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

H.O.D: PRINCIPAL:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgement**

One of the great pleasures of writing this project book is acknowledging the efforts of many people whose names may not appear on the cover but whose hard work, co-operation, friendship and understanding were crucial to the preparation of this report on “HOTEL MANAGEMENT SYSTEM”.

It gives us pleasure in presenting this project book whose justification will never sound good if we do not express our vote of thanks to all staff members of Computer Science department of our college for allowing us to work in the lab, without whose help our project and its thesis would have neither begin nicely nor would have reached the best ending.

We are much thankful to our project guide Mrs. Sunanda Madam and all other teaching staff for their suggestion, reviews, encouraging, criticism and time to time help & support which led us to achieve our goals and overcome the difficulties.

Last but not the least, we are thankful to our parents , friends & seniors who gave us the much needed moral support and encouragement which made developing this project a work of sheer joy and satisfaction . Thank you all once again.

**Preface**

This project has been accomplished through the help of advance programming language. The project will allow the users to keep record of customer, employee, and rooms.

The aim of our project is to computerize almost all transactions. It different from standard looking systems. The report we are presenting is for the user who want to know about the function of the project and how to operate the system.

It has combined the power of VisualStudio.Net2008 with Microsoft SQL-Server 2005 desired results. We have strived to make this project as error free as possible.

**ASHLESHA P. VAKIL**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Title** | **Page No.** |
| **I.** | **Preliminary Investigation** | **8** |
| **1.1** | Organization Overview | 9 |
| **1.2** | Description of system | 10 |
| **1.3** | Limitation of present system | 11 |
| **1.4** | Proposed System | 12 |
| **1.5** | Feasibility Study | 14 |
| **1.6** | Stack Holder | 19 |
| **1.7** | Gantt Chart | 23 |
| **II.** | **System Analysis** | **25** |
| **2.1** | Fact-finding Techniques | 26 |
| **2.2** | Prototypes | 30 |
| **2.3** | Event Table | 32 |
| **2.4** | Use case Diagram | 34 |
| **2.5** | Entity Relationship Diagram | 37 |
| **2.6** | Activity Diagram | 41 |
| **2.7** | Class Diagram | 43 |
| **2.8** | Sequence Diagram | 45 |
| **2.9** | State Diagram | 48 |
| **III.** | **System Design** | **50** |
| **3.1** | Converting ERD into tables | 51 |
| **3.2** | Package Diagram | 53 |
| **3.3** | Program Flow Chart | 55 |
| **3.4** | System Flow Chart | 56 |
| **3.5** | Structure Chart | 57 |
| **IV.** | **System Coding** | **58** |
| **4.1** | Menu Tree | 59 |
| **4.2** | List Of Tables with attributes and constraints | 60 |
| **4.3** | Program Description with Naming Conventions | 63 |
| **4.4** | Screen Layout | 66 |
| **4.5** | Report Layouts | 78 |
| **4.6** | Validations | 81 |
| **4.7** | Test Data | 86 |
| **V.** | Source Code | 89 |
| **VI.** | System Implementation | 107 |
| **VII.** | References and Bibliography | 111 |

Preliminary Investigation

**Organizational Overview**

The Existing System is manual. All day to day work is maintained in different registers. For booking of rooms different registers are maintained in which they had to maintain the customer records, different booking dates. They had to maintain separate registers for lodging and wedding and parties along with the rate and their details.

To find a record of particular customer they had to search in different registers. If the customer had visited again then it is very difficult to find the details of the customer.

There is no maintenance of purchase and sales of menu items all the records are saved temporarily. Also many errors are occurred during maintaining the records manually.

It is also very much time consuming and not a user Friendly activity.

Thus the existing system needs to be computerized which would result in fast telephonic or personal inquiries about availability of rooms in the Saavli Hotel.

**Description Of System**

By using the automated proposed system over time will not consume and also mistakes, errors are get decreases. The proposed system is more benifitable than existing present system.

The HOTEL SAVALI is a hotel and the star category hotel located in the heart of Dombivali city. Designed to effectively meet the demand of the modern customers. The hotel is professionally designed and managed to offer utmost comfort and value to our discerning clientele.

**Limitations of Present System**

The present system is the manual system, since the manual work is always overshadowed by the automatic or mechanical work following are the major drawbacks:-

* The information regarding the of payment of rooms is done in registers.
* The information regarding the of payment of rooms is done in registers
* Errors can be encountered in manual system while maintaining the records manually.
* It is time consuming
* To point our particular record is difficult in manual system.
* It is not user friendly.
* The existing system does not provide integrated information.

**Proposed system And Its Advantages**

The proposed is completely automated. It is more reliable and comfort than existing or manual system.

Designed to effectively meet the demand of the modern customers. The hotel is professionally designed and managed to offer utmost comfort and value to our discerning clientele.

People who just want to get away from the crowd and pollution of the city. People on a leisure trip who are looking for a quiet place to relax themselves.

All rooms are air conditioned and well appointed with running hot and cold water. TV with cable connection and 24 hours. Room service, other facilities available on request are Laundry, Ticketing, Car on hire etc.

**MAIN ADVANTAGES OF PRESENT SYSTEM**

* It is quite easy and flexible to use .
* It is quite user-friendly for Manager and staff.
* The main Advantage of this System is that this System is

it saves much of the time which is time consuming in

existing System.

* All the records of sales of menu items ,customer details are well maintained.
* User can see the details of Customer, room Tariff, any of

the menu items easily only by just clicking on the name.

**Feasibility Study**

The next step in analysis is to verify the feasibility of the proposed system. All project are feasible given unlimited resource and time are scare project should be optimal in three consumption of resources. This places a constant on approval any project.

Feasibility Study is important in project as it gives idea about which types of hardware, software ,technology &

other resources can be used by the system to all the people related to project like requirement gathering people

,analysts ,designers and also end users.

A feasibility study is the study of positive possibilities of the project. It is also measure of how beneficial or practical development of information system would be to an organization.

Feasibility as applied to our system pertain the following area:-

1. Economic Feasibility

2. Operational Feasibility

3. Technical Feasibility

1. **Economic Feasibility:-**

Higher level of automation most often requires more funds. Hence based on the hardware and software specification a desirable alternative costs and benefits to see if the investment made in creating / developing a new system is costlier or more beneficial.

Financial benefits must equal or exceed the costs. To assure this one must estimate the

following:

1. If the Organization has adequate cash flow for funding the development
2. The cost to conduct a full system investigation.
3. The cost of hardware and software for the class of application being considered.
4. The benefits in the form of reduced costs or fewer costly errors.
5. The cost if nothing changes (i.e. the proposed system is not developed) for a project to be judged feasible, it must pass all these tests.
6. If any one of these issues appears infeasible the decision must be reconsidered.

My system is economically feasible as the costs involved in implementing the automated system is in lieu with the cash flow of the shop. Thus the system developed will be beneficial to the users.

However there will be an additional cost for implementing the

technology on which the system is to be implemented.

**2. Operational Feasibility:-**

The operational feasibility is obtained by consulting the system user whether it satisfies the user’s requirements.

A system with an easy interface will always help the user to use

the system.

The new system has completely user friendly interface. It has been designed to be pretty intuitive, so that even an inexperienced person can easily handle the system.

Business functions are reengineered to achieve broader scope and higher level of automation.

Manual processes too are modified. Every company has its own culture

and new system should fit the company culture.

The issues to be taken into concern are**:**

* Corporate Culture.
* Level of computer competency.
* Loss of control on employee by staff/management.
* Change of job responsibility.
* Loss of employment due to increased automation.
* The nature and level of user involvement in the development and implementation of system.
* Revisal of old, longstanding work procedures.
* It is usually a practice to include people trained in organizational

behavior to assist in managing these changes.

As the system will be automated there is a sense of insecurity among the employees working for an event. As the system provides a user friendly simple GUI it is operationally feasible to implement it in the Hotel.

This automation will save time as well as reduce errors.

**3. Technical Feasibility:-**

To determine whether the proposed system is technically feasible we should take consideration the technical issues involved behind the system. The proposed system uses the Microsoft Visual Studio.Net 2008 which is rampantly employed these days worldwide. The world without the web is incomprehensible today. That goes to prove that the system is founded on existing and widely popular technologies. Hence the proposed system is technically feasible.

It is essential to check whether the proposed system is technically feasible and to determine the technology and skills necessary to carry out the successful implementation of the project.

The necessary software required for the development of system is

* Microsoft Visual Studio 2008
* SQL Server 2005

There is requirement for the mentioned software and an expertise for handling system.

Thus in the presence of required hardware, software the proposed system is technically feasible.

Hardware and Software Requirement-

**Hardware Requirements** :-

Pentium4 (P4) or higher version

1 GB RAM or more.

500MB free space in Hard Disk.

**Software Requirements:-**

Back end: - SQL Server 2005.

Front End: - VB.NET

Operating System - Windows XP or higher version.

**Technology Used:-**

**Visual Basic .NET 2008**

Visual Basic .NET provides the easiest, most productive language and tool for rapidly building Windows and Web applications. Visual Basic .NET comes with enhanced visual designers, increased application performance, and a powerful integrated development environment (IDE). It also supports creation of applications for wireless, Internet-enabled hand-held devices. The following are the features of Visual Basic .NET with .NET Framework 1.0 and Visual Basic .NET 2003 with .NET Framework 1.1. This also answers why should I use Visual Basic .NET, what can I do with it?

**Powerful Windows-based Applications**

**Building Web-based Applications**

**Simplified Deployment**

**Powerful, Flexible, Simplified Data Access**

**Improved Coding**

**Direct Access to the Platform**

**Full Object-Oriented Constructs**

**SQL SERVER 2005:-**

1. Database Mirroring
2. Data Partitioning
3. Online Indexing Operations
4. Fast Recovery
5. Standards-based Information Access
6. Dedicated Administrator Connection

**Stakeholders**

Your primary source of information for system requirements is the various stake holders of the new system. The stake holders of are all people who and interest in the successful implementation of the system.

Generally we categorized stake holders into one of three groups:

1. The user, those who actually use the system on the daily basis.
2. The customer, those who pay for and own the system operators in the computing environment of the organization.

The various kind stake holders who have an interest in a new system. During analysis, the analyst also needs to consider the technical staff as well. One of the most important first steps in determining g system requirements is to identify these various system stake holders. In the past, problems have arisen with new system because only some of the stake holders where included in the project and the system was built exclusively for them. One of an analysts first task is to identify every type of stake holder category are available to the project as business expert.

**Users as stake holders:**

User roles-that is type of system users should be identified in two dimensions: Horizontally and vertically.

By horizontally we mean that the analyst must look for information flow across business department or functions.

For example:-

A new inventory system may affect receiving garage, sales and manufacturing. So individuals from each of these departments must describe their requirements. The sales department may need to determine when and how to update inventory quantities or to commit inventory at the time of the same, but before it is shipped. Manufacturing may need certain information from the inventory system to assist in scheduling production. So, remembering to include the horizontal dimension in the definition of requirements will ensure that May different departments, even those that may appear unrelated to the new system are included.

By vertical dimension, we mean the information needs of clerical staff of middle management and of senior executives. Each of these stake holders has different information request for the system that must be included in the design. The following section described the characteristics and information needs of the various users on the vertical dimensions. These same characteristics also applied to each development across the horizontal language.

Business User:-

Business users are the people who use system to perform day to day operation of an organization like clerk of showroom. Business uses are providing information about daily operation of sales of car and system must support them.

Information User:-

An information user is a person who provides information to customer. This person might be an operational user or someone else. In some cases a business might want to make information directly available to customer information user may not be permission on sales transition of products.

Management User:-

Managers are responsible for seeing that the company it’s performing for its daily procedure efficiently and effectively

\* What kind of reports must the system produce?

\* What kind of sales performance?

\* What kind of product information and specification must the system keep?

Executive Users:-

The tap executive of an organization interested in strategic issue, as well as the daily issues just described. The typically want information from a system so that they can compare overall improvements in resource utilization.

External User:-

More and more quantity allow external user to have direct access to the system customer may access the system directly through internet with help of website of company. The external user not allows checking transaction. These users more difficult to identify and access because they are not a regular member of the organization.

**GANTT CHART:-**

Gantt charts are a project-planning tool that can be used to represent the timing of tasks required to complete a project.

Because Gantt charts are simple to understand and easy to construct, they are used by most project managers for all but the most complex projects.

In a Gantt chart, each task takes up one row. Dates run along the top in increments of days, weeks or months, depending on the total length of the project.

The expected time for each task is represented by a horizontal bar whose left end marks the expected beginning of the task and whose right end marks the expected completion date. Tasks may run sequentially, in parallel or overlapping.

As the project progresses, the chart is updated by filling in the bars to a length proportional to the fraction of work that has been accomplished on the task. This way, one can get a quick reading of project progress by drawing a vertical line through the chart at the current date.

Completed tasks lie to the left of the line and are completely filled in. Current tasks cross the line and are behind schedule if their filled-in section is to the left of the line and ahead of schedule if the filled-in section stops to the right of the line. Future tasks lie completely to the right of the line.

In constructing a Gantt chart, keep the tasks to a manageable number (no more than 15 or 20) so that the chart fits on a single page. More complex projects may require subordinate charts which detail the timing of all the subtasks which make up one of the main tasks.

For team projects, it often helps to have an additional column containing numbers or initials, which identify that on the team is responsible for the task.

Often the project has important events, which you would like to appear on the project timeline, but which are not tasks. For example, you may wish to highlight when a prototype is complete or the date of a design review.

We planned our project using according to the Gantt chart as shown as follows

**SYSTEM ANALYSIS**

**Fact finding techniques**

After obtaining the background knowledge, the information on the existing system, its inputs, outputs, costs and other important requirement and features have to be collected and analyzed. The following tools were used for knowing more about the system and gathering more information for developing a new system.

For developing a system for an organization one need to acquire necessary information regarding the system. Information plays vital role in an organization. So collection of information before designing a system solves many problems. Following are the fact finding techniques used.

1. Documentation
2. On Site Observation
3. Interviews
4. Questionnaires

It is called information gathering.

**Documentation:-**

This involves review of written documents, literature, procedures, forms, manuals govt. publications etc . The preliminary drawback is search time. Sometimes it may be difficult to get certain reports and documentation or may be outdated or expensive and may not be updated. Close look at printed forms are necessary since printed forms are widely used for capturing and providing information. The objective of this is to understand how forms are used.

The following questions are useful in using this methods for fact finding

1. Who uses forms? How important they are to user?
2. Does the forms include all necessary information ?
3. What items should be added or deleted?
4. How many departments receive existing forms? Why?
5. How easy and readable it is to follow?
6. What are the other used of the form?
7. How the forming helps user in decision making?

**On-Site Observation:-**

The major objective of On site observation is to get close to the “real”

System being studied. This is the most difficult fact finding method. In this the role of the analysis is of observer or listener. He should avoid advice, passing moral judgments on show hostility towards one person and unique friendliness towards other.

The following questions can serve as guideline for on site observation.

i) what kind of system is it? What it is?

ii) What its function and other contribution to organization?

iii) Who are the peoples involved in the system?

iv) What is the history of system?

v) what is the assurance that observation presses is not seriously affecting the system or behavior being observed.

**Interview:-**

The interview is face to face interpersonal meeting for information gathering information. It is a widely used tool. It provides greater flexibility for gathering information.

In on site observation, we cannot learn people’s perceptions, feeling and motivation etc. Much of the information can be acquiring by direct questions. But there is information of more difficult nature that user staff may be redundant to give directly. For E.g. information on company policies, or specification with senior etc. If asked directly interviewer may give invalid information. But if properly handled correct information can be successfully obtained with interviews and questionnaires. More freely given information is more valid.

Interview offers better opportunity than the questionnaires to evaluate validity of information gathered. Interviewing is an art and requires experience.

Following are the guidelines for successful interview:-

1. Set the stage for interview and open it.
2. Create friendly atmosphere and put the interviewer at ease.
3. The location, time and order of interviewing should assurance privacy and minimum interruption.
4. Phrase questions clearly.
5. Be a good listener, avoid arguments and records responses.
6. Evaluate the outcome of the interview.

**Questionnaires**

The questionnaire is self-administered tools that is more economical and require less skill to administrate than interview.

In examiners a large number of respondents. At the same provides standardizes wording and instruction places less pressures on subjects for immediate response.

Following are the procedures for the questionnaires construction which consist of six steps.

1. Decide the data to be collected i.e. define the problem to be investigated.
2. Decide what type of questionnaires should be used :-fill in the blanks type or multiple type questions.
3. Outline the topics for the questionnaire and then write the questions.
4. Edit the questionnaires for technical defect or basis that reflect personal value.
5. Pretest (try out) the questionnaires to see how well it works.
6. Do a final editing to ensure that the questionnaires are ready for administration.

**Prototype**

**Agile software development**

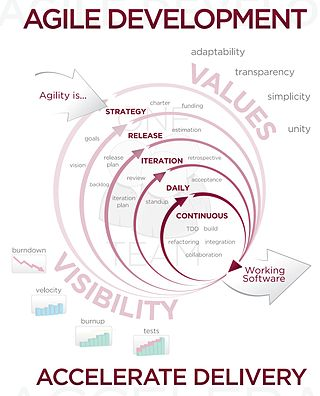
**Agile software development** is a group of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies) based on [iterative and incremental development](http://en.wikipedia.org/wiki/Iterative_and_incremental_development), where requirements and solutions evolve through collaboration between [self-organization](http://en.wikipedia.org/wiki/Self-organization#Self-organization_in_agile_software_development), [cross-functional teams](http://en.wikipedia.org/wiki/Cross-functional_team).

There are many specific agile development methods. Most promote development, teamwork, collaboration, and process adaptability throughout the life-cycle of the project.

Agile methods break tasks into small increments with minimal planning, and do not directly involve long-term planning. Iterations are short time frames ([timeboxes](http://en.wikipedia.org/wiki/Timeboxing" \o "Timeboxing)) that typically last from one to four weeks. Each iteration involves a team working through a full software development cycle including planning, [requirements analysis](http://en.wikipedia.org/wiki/Requirements_analysis), [design](http://en.wikipedia.org/wiki/Software_Design), [coding](http://en.wikipedia.org/wiki/Computer_programming), [unit testing](http://en.wikipedia.org/wiki/Unit_test), and [acceptance testing](http://en.wikipedia.org/wiki/Acceptance_test) when a working product is demonstrated to stakeholders. This minimizes overall risk and allows the project to adapt to changes quickly. Stakeholders produce documentation as required. An iteration may not add enough functionality to warrant a market release, but the goal is to have an available release (with minimal [bugs](http://en.wikipedia.org/wiki/Software_bug)) at the end of each iteration.[[8]](http://en.wikipedia.org/wiki/Agile_software_development#cite_note-embracing_change-7) Multiple iterations may be required to release a product or new features.

Team composition in an agile project is usually cross-functional and self-organizing without consideration for any existing corporate hierarchy or the corporate roles of team members. Team members normally take responsibility for tasks that deliver the functionality an iteration requires. They decide individually how to meet an iteration's requirements.

Agile methods emphasize face-to-face communication over written documents when the team is all in the same location. Most agile teams work in a single open office (called a [bullpen](http://en.wiktionary.org/wiki/bullpen)), which facilitates such communication. Team size is typically small (5-9 people) to simplify team communication and team collaboration. Larger development efforts may be delivered by multiple teams working toward a common goal or on different parts of an effort. This may require a co-ordination of priorities across teams. When a team works in different locations, they maintain daily contact through [videoconferencing](http://en.wikipedia.org/wiki/Videoconferencing), voice, e-mail, etc.

****

**Event Table**

• Event table: A Table that lists events in rows and key pieces of information about each event in columns.

• Event: An Event occurs at a specific time and place, can be described and should be remembered by the system.

• Trigger: An occurrence that tells the system that a event has occurred, either arrival of data needing processing or of a point in time.

• Source: An external agent or an actor that supplies data to the system.

• Activity: Behavior that the system performs when an event occurs.

• Response: An output produced by the system that goes to the destination.

• Destination: An external agent or an actor that receives data from the system.

**Description:**

• Events are the transactions.

• Events are the cause of transactions.

• Events will generate transactions.

• While developing the list of events the analyst should note the additional information about the events for later use.

• The list events with trigger, source, activity, response(s), and destination for each event can be placed in the event table to keep track of them for later use.

• Event table is a convenient way to record about the requirements for information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***EVENT*** | ***TRIGGER*** | ***SOURCE*** | ***ACTIVITY*** | ***RESPONCE*** | ***DESTINA-TION*** |
| **Providing Menu** | **Customer enter in restaurant** | **Staff** | **Providing menu** | **Menu is Provided** | **Customer** |
| **Providing Item** | **Request of item from customer** | **Staff** | **Providing item** | **Item is provided** | **Customer** |
| **Booking of room** | **Request of room from customer** | **Staff** | **Booking of room** | **Room provided to the customer** | **Customer** |
| **Bill Generation** | **Leaving of room** | **Staff** | **Generation of bill** | **Bill payment received from customer** | **Customer** |

**Use Case Diagram**

A **use case diagram** in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by

a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram

displays the relationship among actors and use cases. The two main components of a use case diagram are use

cases and actors.



An actor is represents a user or another system that will interact with the system you are modeling. A use case

is an external view of the system that represents some action the user might perform in order to complete a task.

**When to Use: Use Cases Diagrams**

Use cases are used in almost every project. They are helpful in exposing requirements and planning the project.

During the initial stage of a project most use cases should be defined, but as the project continues more might

become visible.

For Reservation/ Booking Room

Customer Staff

Ordering Item\Menu

Customer User

For Payment /Bill

Customer User

**Entity-Relationship Diagram**

Data models are tools used in diagrams to describe the data requirements and assumptions in the system from a top down perspective.

An entity relationship diagram is a graphical\cal representation of an organization’s data storage requirements.

Entity relationship diagrams are abstractions of the real world which simplify the problem to be solved while retaining its essential features. Entity relationship diagrams are used to identify the data that must be captured, stored and retrieved in order to support the business activities performed by the organization; and identify the data required to derive and report on the performance measures that an organization should be monitoring.

There are three basic elements in ER models:

**Entities:** Entities are the “things” about which we seek information.

**Attributes**: Attributes are the data we collect about the entities.

**Relationships**: Relationships provide the structure needed to draw

information from multiple entities.

Developing an ERD requires an understanding of the system and its components.

ERD brings out issues:

i) Ambiguities

ii) Entities and their relationships

iii) What data needs to be stored?

iv) The degree of a relationship

|  |  |  |
| --- | --- | --- |
| Symbols | Symbol Name | Meaning |
|  | Entity | An object that exist and is  distinguishable from all  other objects. |
|  | Weak Entity | An entity set with no  sufficient attributes to  form a primary key. |
|  | Relationship | An association among  Several entities. |
|  | Identifying Relationship | A relationship associating  weak entity set with  identifying entity set |
|  | Attribute | A specification that  defines the property of an  Object, element of file. |
|  | Primary Key | A candidate key that has  been selected as the  identifier for an entity  Type. Primary key values  May be null. It is also  Called as identifier. |
|  | Multi-valued Attribute | An attribute that can have  more than one value for  each entity instance. |
|  | Discriminating Attribute | A set of attributes that  allows distinguishing  among the entities |

**Activity Diagram**

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something.

The diagrams describe the state of activities by showing the

sequence of activities performed. Activity diagrams can show activities that are conditional or parallel.

**When to Use: Activity Diagrams**

Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams.

The main reason to use activity diagrams is to model the workflow behind the system being designed.

Activity Diagrams are also useful for: analyzing a use case by describing what actions need to take place and when they should occur; describing a complicated sequential algorithm; and modeling applications with parallel processes.

However, activity diagrams should not take the place of interaction diagrams and state diagrams.

Activity diagrams do not give detail about how objects behave or how objects collaborate

**Below are figures:**

The given below is figure for the activity diagram of the system

Customer Staff Computer System

Start Process

Contact to Hotel System

yes No For each order

Verify Pay

Enter Pay

Pay For Bill

Make Bill

End Order

End order

Add Order

Enter Order

Request order item

Create a new order list

Take Order

Enter Customer Info InfoCustomerInformation

Verify Customer

Create new Customer

Display Customer Info

**Class diagram**

The class diagram is the main building block in object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are

Represented with boxes which contain three parts:

• A class with three sections.

• The upper part holds the name of the class

• The middle part contains the attributes of the class.

•The bottom part gives the methods or operations the class can take or undertake.

In the system design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the statically relations between those objects. With detailed modeling, the classes of the conceptual design are often split in a number of subclasses.

In order to further describe the behavior of systems, these class diagrams can be complemented by state diagram or UML state machine. Also instead of class diagrams Object role modeling can be used if you just want to model the classes and their relationships.

1 m M 1 m m 1 m m 1 m m 1 1 M 1 m

**Maintenance**

**Admin&Account**

**Banquet**

**Restau-rent**

**Kitchen**

**House**

**Keeping**

**Customer details**

**1. custID 2. Cname**

**3. Contact no. 4 Room no.**

**5. Check IN 6.Room Type**

**7. No.of Room 8. Rent**

**9. Service**

**Employee**

**1. EmpID 2.Name**

**3. Address 4. DOB**

**5. Gender**

**Restaurant Details**

**1. CustID 2.EmpID**

**3.Code 4.Pay type**

**Clean Status**

**1.Room no.**

**2. Floor no.**

**3. Status**

**4.Assigned to**

**Room Reservation**

**1. RID 2. CustID**

**3. DOR 4. RMtype**

**5. No. of person**

**Menu Item**

**1. Code**

**2. Item name**

**3. Rate**

**Dept**

**1 .Dept ID**

**2. Dept name**

**3. Dept head**

**4. No. of employee**

**Payment**

**1. Pay type**

**2. Cash**

**3. Card no.**

**4. Cust\_ID**

**5. Total Amt**

**sequence diagram**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.

It is a construct of a Message Sequence Chart.

Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams.

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.

Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.

However, an organization's business staff can find sequence diagrams useful to communicate how the business currently works by showing how various business objects interact.

Besides documenting an organization's current affairs, a business-level sequence diagram can be used as a requirements document to communicate requirements for a future system implementation.

During the requirements phase of a project, analysts can take use cases to the next level by providing a more formal level of refinement.

When that occurs, use cases are often refined into one or more sequence diagrams.

An organization's technical staff can find sequence diagrams useful in documenting how a future system should

behave.

During the design phase, architects and developers can use the diagram to force out the system's object interactions, thus fleshing out overall system design.

One of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement.

Use cases are often refined into one or more sequence diagrams.

In addition to their use in designing new systems, sequence diagrams can be used to document how objects in an existing (call it "legacy") system currently interact.

This documentation is very useful when transitioning a

system to another person or organization.

**Sequence Diagram**



**HOTEL MANAEMENT SYSTEM**

CUSTOMER

**Request for new room**

**Reservation window**

**Customer information**

**Update customer information**

**And update room availability**

**Book room (cust\_no)**

**Type of room (single / double etc)**

**Confirm booking**

**Check IN**

**Look up for all rooms**

**Facilities/service give to customer**

**Accept the service**

**Check OUT details**

**Payment summary**

**Payment type**

**Accept payment**

**Check OUT**

**State diagram**

State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented in series of events that could occur in one or more possible states. Hereby "each diagram usually represents objects of a single class and tracks the different states of its objects through the system".

State diagrams can be used to graphically represent finite state machines. This was introduced by Taylor Booth in his 1967 book "Sequential Machines and Automata Theory". Another possible representation is the State transition table.

Start Process Service Started Cancelled End Process

Ask For Booking

Full

Check For Availability of room

Bill Payment

Order Menu

**Booking Reservation**

Reservation Confirm

Types Of Room

**State Diagram**

**SYSTEM DESIGN**

**Converting ERD to Tables**

**Customer Master:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustId | CustName | Gender | Address | City | State |
|  |  |  |  |  |  |
| Country | Nationality | Passport | Contact |
|  |  |  |  |

**Employee Master:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eid | EName | Gender | DOB | Address | City |
|  |  |  |  |  |  |
| State | Country | Nationality | Contact | Department | Salary |
|  |  |  |  |  |  |
| Joining Date |
|  |

**Item Master:-**

|  |  |  |  |
| --- | --- | --- | --- |
| Mid | Type | Name | Price |
|  |  |  |  |

**Room Master:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rid | RoomNo | Floor | CustId | RoomType | Price |
|  |  |  |  |  |  |
| Status |
|  |

**User Master:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UId | FName | LName | UserName | Password | User Type |
|  |  |  |  |  |  |

**CheckiIn:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustId | CustName | CGender | Address | PassNo | ContactNo |
|  |  |  |  |  |  |
| Checkin | Staying | Checkout | RoomType | RomNo | Floor |
|  |  |  |  |  |  |
| Amount | Paid | Remain | Total | Status |
|  |  |  |  |  |

**BillMenu:-**

|  |  |  |
| --- | --- | --- |
| CustId | Mid | TPrice |
|  |  |  |

**CheckOut:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustId | CustName | CustGender | Address | PassNo | ContactNo |
|  |  |  |  |  |  |
| Checkin | Staying | Checkout | RoomType | RomNo | Floor |
|  |  |  |  |  |  |
| Amount | Status | ManuBill | Paid | Remain | TotalBill |
|  |  |  |  |  |  |

**Package Diagram:-**

In addition to the standard UML Dependency relationship, there are two special types of dependencies defined between packages:

• Package import

• Package merge

A package import is "a relationship between an importing namespace and a package, indicating that the importing namespace adds the names of the members of the package to its own namespace."

By default, an unlabeled dependency between two packages is interpreted as a package import relationship.

A package merge is "a directed relationship between two packages that indicates that the contents of the two packages are to be combined. It is very similar to Generalization in the sense that the source element conceptually adds the characteristics of the target element to its own characteristics resulting in an element that combines the characteristics of both".

View Layer Domain Layer Domain Layer Data Access Layer

RoomMstr

DB

ItemMstr

DB

EmpMstr

DB

CustMaster

DB

Enter

Customer

Enter Menu

Enter Employee

Room Details

Menu

Details

Employee

Details

Customer

Details

Enter

Room

Data access layer **Program Flow Chart:-**

**START**

READ HOTEL, ROOM & MENU INFORMATION

Customer Enters in Hotel

Is Customer Enter in Hotel

Provide Lodging to Customer

Request for Item

Request For Room

Provide Item

No

Customer waiting for Room

Is Room Available

Is leaving Restaurant

Provide Details

Provide Rooms

Is CheckOut

Bill Generation

**System Flow Chart:-**

Transaction Details

Search Details

Employee/Menu/ Customer/Room Details

RESTAURENT SYSTEM

SEARCH

BILLING REPORT

**ROOM**

**MENU**

**EMPLOYEE**

**CUSTOMER**

Check IN/ Check OUT Details

Menu Details

MASTER

LODGING SYSTEM

**TRANSACTION**

**Structure Chart:-**

**SYSTEM CODING**

**Menu Tree:-**

**List Of Tables with Attributes and constraint**

**Customer Details Table:**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| CustId | Numeric | Primary key |
| CustName | Char |  |
| Gender | Char |  |
| Address | Char |  |
| City | Char |  |
| State | Char |  |
| Country | Char |  |
| Nationality | Char |  |
| Passport | Numeric |  |
| Contact No. | Numeric |  |

**Employee Master:-**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| EId | Numeric | Primary key |
| EName | Char |  |
| Gender | Char |  |
| DOB | DateTime |  |
| Address | Char |  |
| City | Char |  |
| Country | Char |  |
| Nationality | Char |  |
| Contact No. | Char |  |
| Department | Char |  |
| Salary | Numeric |  |
| Joining Date | DateTime |  |

**Menu Master**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| CustId | Numeric | Primary key |
| Mid | Numeric |  |
| Type | Char |  |
| Name | Char |  |
| Price | Numeric |  |

**RoomMaster:-**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| Rid | Numeric | Primary key |
| RoomNo | Numeric |  |
| Floor | Char |  |
| RoomType | Char |  |
| RoomPrice | Numeric |  |
| Status | Char |  |

**User Master:-**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| Id | Numeric | Primary key |
| FName | Char |  |
| LNAme | Char |  |
| UsetName | Char |  |
| UserPassword | Char |  |
| UserType | Char |  |

**MenuBill:-**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| CustId | Numeric | Primary key |
| Mid | Numeric |  |
| TPrice | Numeric |  |

**CheckIn:-**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| CustId | Numeric | Primary key |
| CustName | Char |  |
| Gender | Char |  |
| Address | Char |  |
| PassNo | Numeric |  |
| contactno | Numeric |  |
| Checkindate | DateTime |  |
| Staying | Numeric |  |
| checkoutDate | DateTime |  |
| Roomtype | Numeric |  |
| RoomNo | Numeric |  |
| Floor | Char |  |
| Amount | Numeric |  |
| Paid | Numeric |  |
| Remain | Numeric |  |
| Total | Numeric |  |
| status | Char |  |

**CheckOut:-**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Description |
| CustId | Numeric | Primary key |
| CustName | Char |  |
| Gender | Char |  |
| Address | Char |  |
| PassNo | Numeric |  |
| contactno | Numeric |  |
| Checkindate | DateTime |  |
| Staying | Numeric |  |
| checkoutDate | DateTime |  |
| Roomtype | Numeric |  |
| RoomNo | Numeric |  |
| Floor | Char |  |
| Amount | Numeric |  |
| status | Char |  |
| MenuBill | Numeric |  |
| Paid | Numeric |  |
| Remain | Numeric |  |
| Total | Numeric |  |

**Program Description**

**Splash Screen:-**

It provides a splash screen showing the system and its name.

**LoginForm:-**

This form shows username and password to enables MDI\_form.

**MDI\_Form:-**

It is the main form having manus.

**CustMaster:-**

This form gives us information about customer.

**EmployeeMaster**:-

This form gives us information about employee.

**ItemMstr:-**

This form gives us information about Menus.

**Room Master:-**

This form gives us information about rooms

**CheckIn:-**

This form gives us information about customer checkin.

**CheckOut:-**

This form gives us information about customers checkout.

**View All Rooms:-**

This form gives us information about all the rooms and also shows that which are available and which are allocated.

**Naming Conventions**

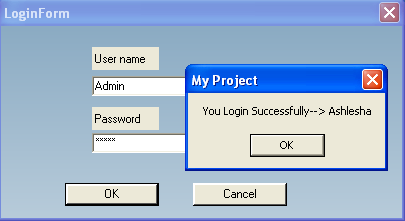
|  |  |
| --- | --- |
| **Name** | **Conventions** |
| Form | frm |
| Text Box | txt |
| Button | Btn |
| ComboBox | Cbo |
| DataGrid | DG |
| Label | lbl |
| DateTimePicker | Dtp |
| ImageList | img |

**Screen Layout**

**Splash Screen:-**

****

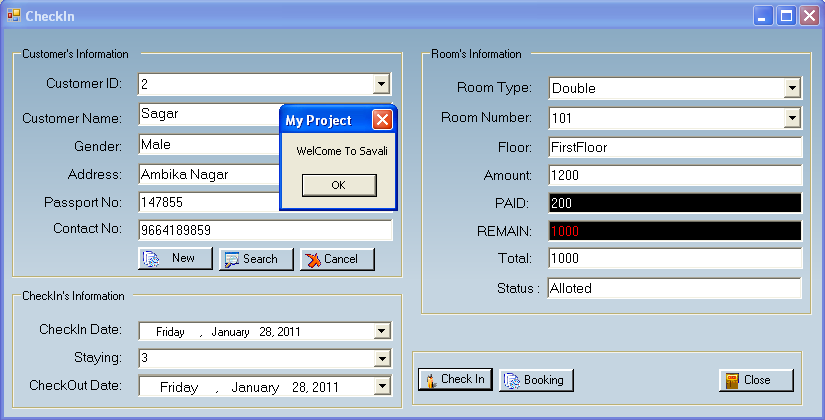
**Login Form:-**

****

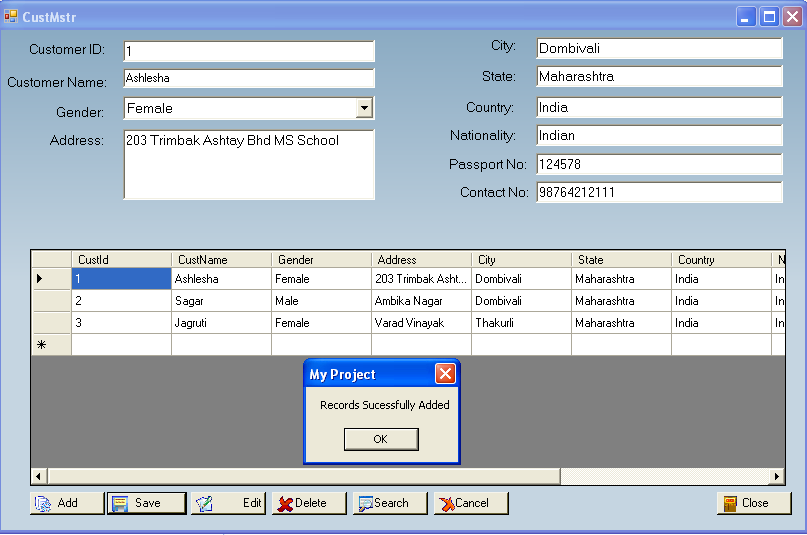
**MDIForm:-**

****

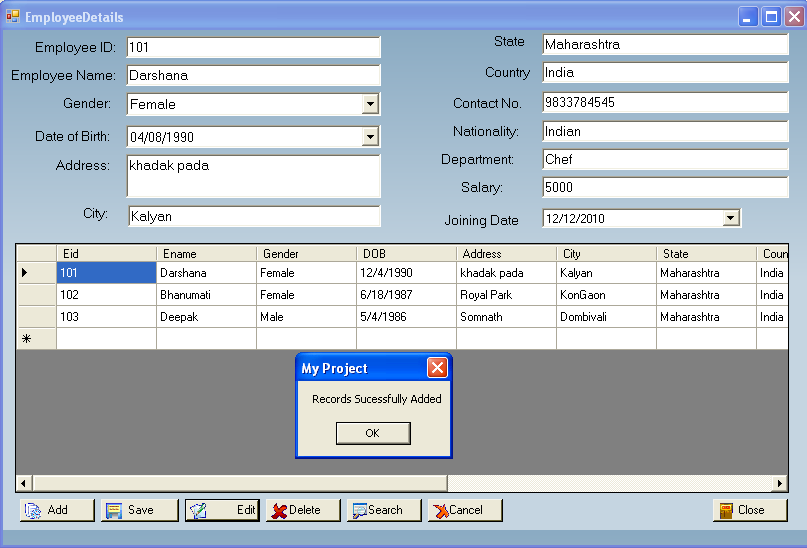
**CheckIn:-**

****

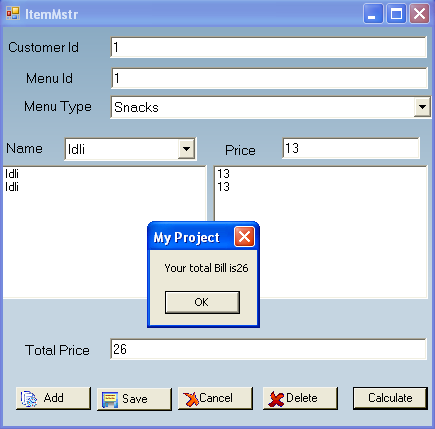
**CustomerMaster:-**

****

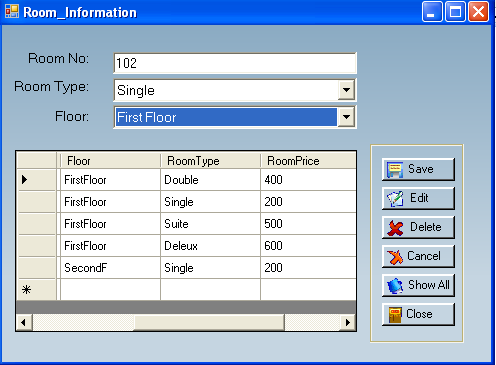
**EmpliyeeDetails:-**

****

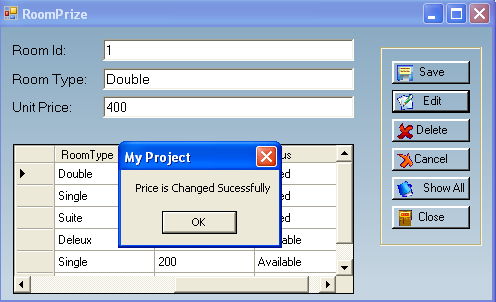
**ItemMaster:-**

****

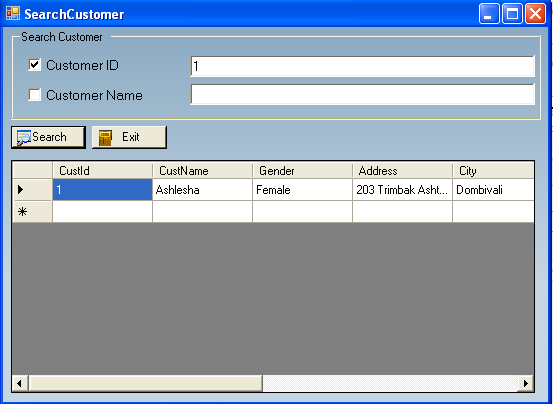
**RoomInformation:-**

****

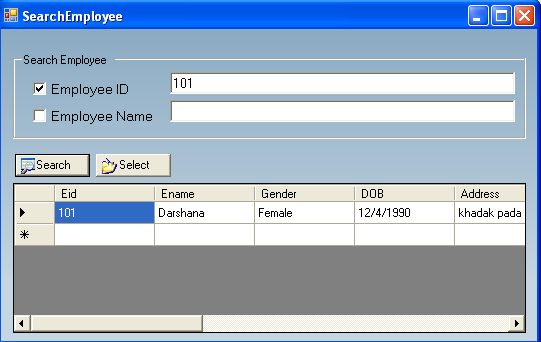
**Set Room Price:-**

****

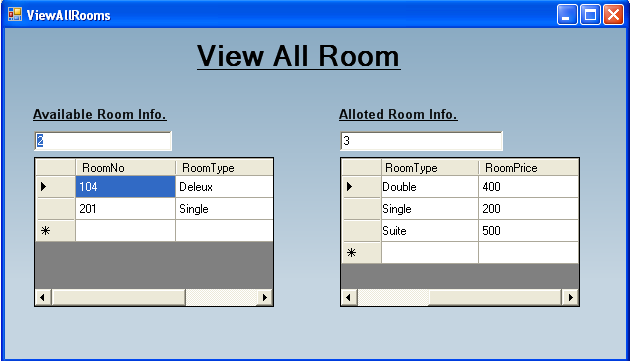
**SearchCustomer:-**

****

**Search Employee:-**

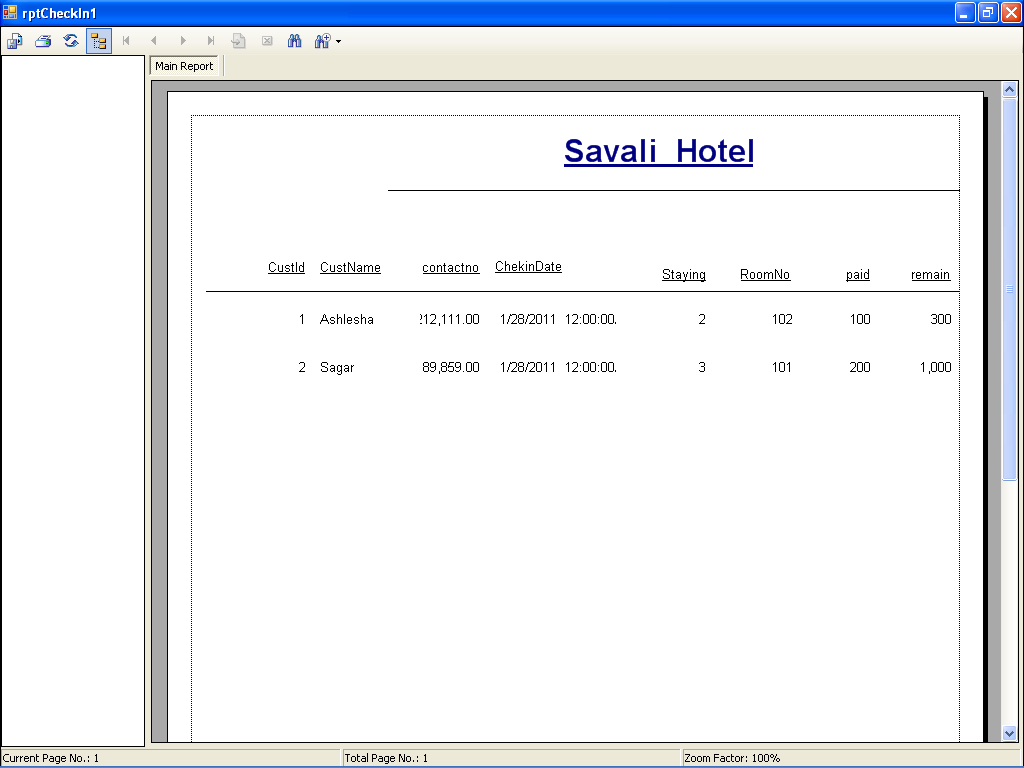
****

**View All Rooms:-**

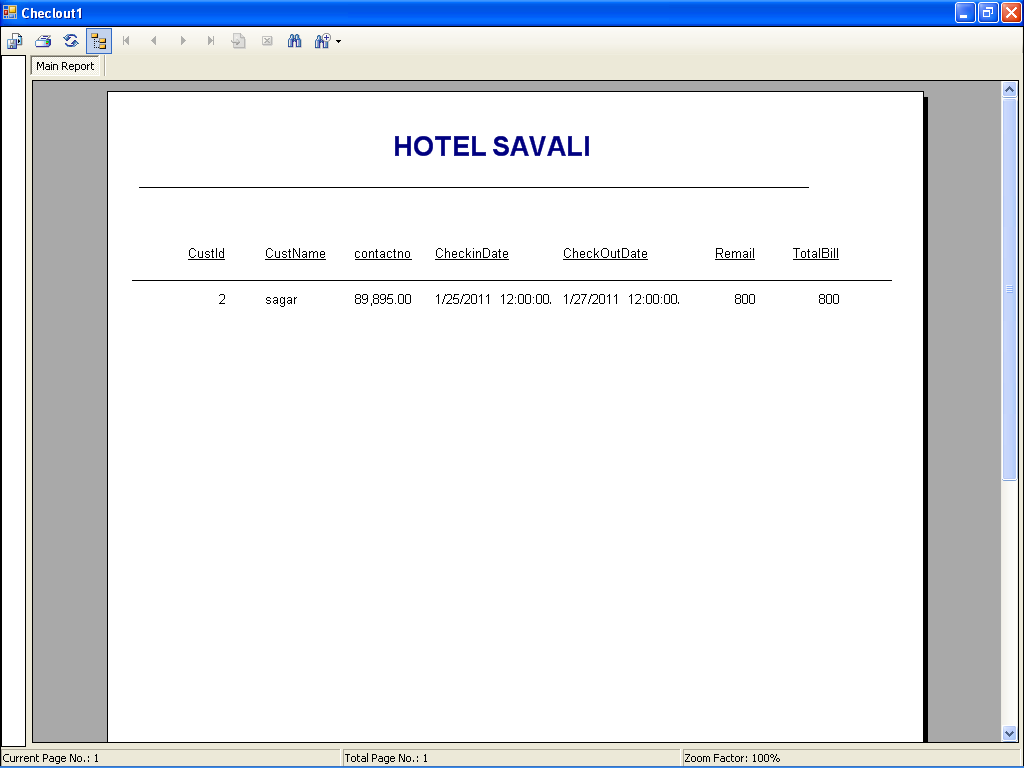
****

**Report Layout**

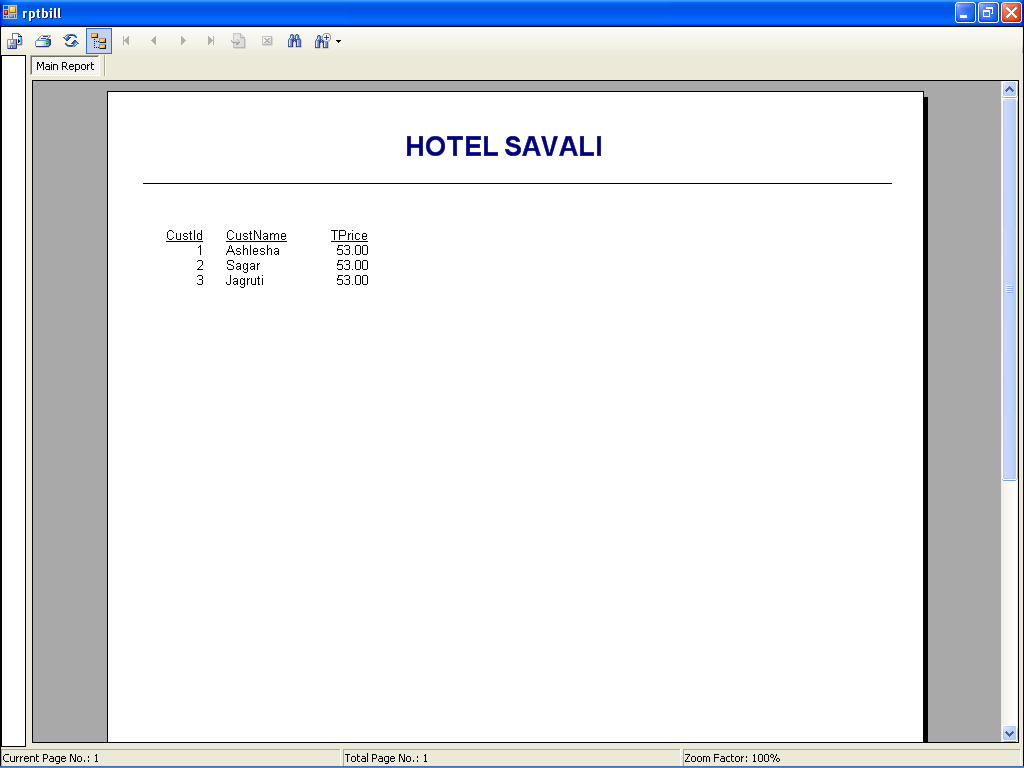
**CheckIn Report:-**

****

**CheckOutReport:-**

****

**BillReport:-**

****

**Validations:-**

**Testing:-**

Software testing is the process used to help, identify the correctness, completeness, security, quality of developed computer software. Testing is a process of technical investigation performed on behalf of stake holders that is intended to reveal quality of related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to the process of executing a program or application with the intend of finding errors.

**Test Levels:**

**Unit Testing**: Testing of the minimal software components and sub-components or modules by the programmer.

**Integration Testing**: Exposes defects in the interface and interaction between integrated components (modules).

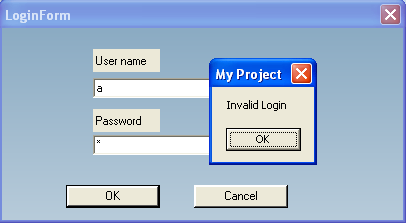
**Functional Testing**: Tests the product according to programmable work.

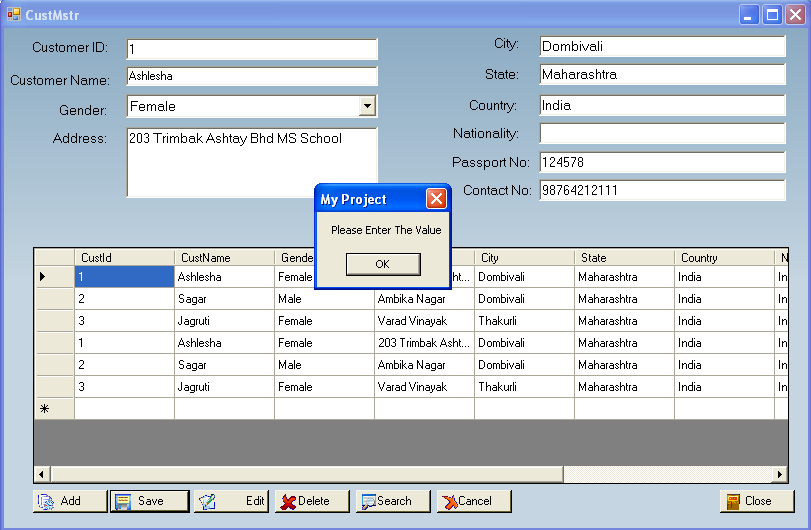
**System Testing**: Test an integrated system to verify /validate that it makes it requirements.

**Acceptance Testing**: Can be conducted by the client. It allows the end user or customer or client to decide whether or not to accept the product.

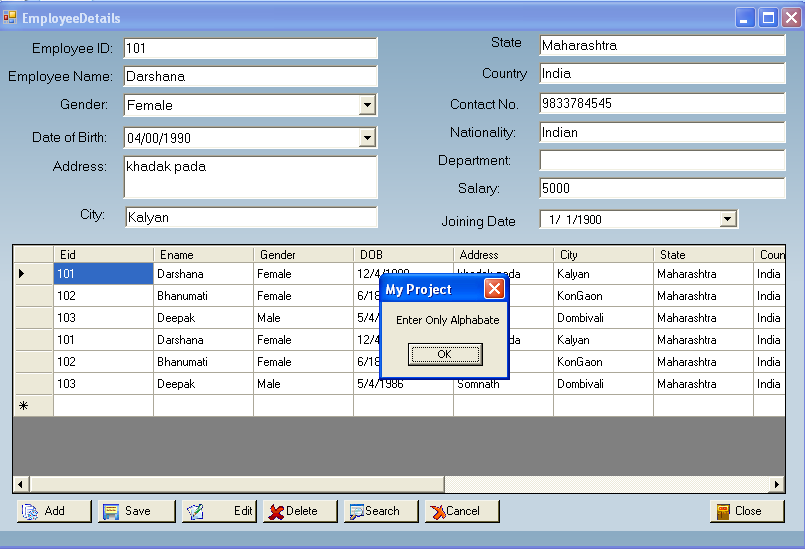
Acceptance testing may be performed after the testing and before the implementation phase.

**Login:**

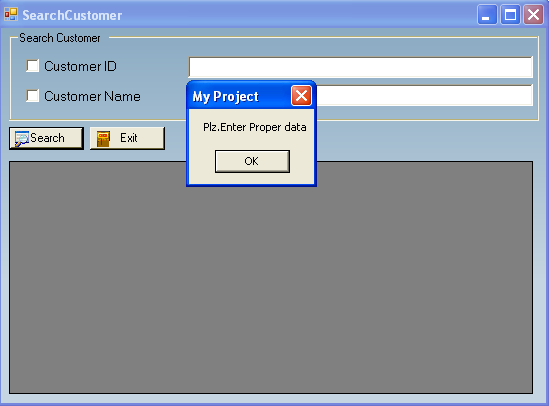
** CustomerMstr:**

****

**EmployeeMaster:**

****

**SearchCustomer:**

****

**TestData:-**

**Customer Master:-**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Validation** | CustId | CustName | Gender | Address | City | State |
| **Valid** | Numeric | String | String | String | String | String |
| **Invalid** | String | Numeric | Numeric | Numeric | Numeric | Numeric |
| **Validation** | Country | Nationality | Passport | Contact | -\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_ |
| **Valid** | String | String | Numeric | Numeric |  |  |
| **Invalid** | Numeric | Numeric | String | String |  |  |

**Employee Master:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Validation** | **Eid** | **EName** | **Gender** | **DOB** | **Address** |
| **Valid** | Numeric | Numeric | Numeric | Numeric | Numeric |
| **Invalid** | String | String | String | String | String |
| **Validation** | **City** | **State** | **Country** | **Nationality** | **Contact** |
| **Valid** | Numeric | Numeric | Numeric | Numeric | Numeric |
| **Invalid** | String | String | String | String | String |
| **Validation** | **Department** | **Salary** | **Joining Date** |  |  |
| **Valid** | Numeric | Numeric | Numeric |  |  |
| **Invalid** | String | String | String |  |  |

**Item Master:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Validation** | Mid | Type | Name | Price |
| **Valid** | Numeric | String | String | Numeric |
| **Invalid** | String | Numeric | Numeric | String |

**Room Master:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Validation** | Rid | RoomNo | Floor | CustId | RoomType |
| **Valid** | Numeric | Numeric | String | Numeric | String |
| **InValid** | String | String | Numeric | String | Numeric |
| **Validation** | Price | Status |  |  |  |
| **Valid** | Numeric | String |  |  |  |
| **InValid** | String | Numeric |  |  |  |

**User Master:-**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Validation** | UId | FName | LName | UserName | Password | User Type |
| **Valid** | Numeric | String | String | String | String | String |
| **Invalid** | String | Numeric | Numeric | Numeric | Numeric | Numeric |

**Checkin:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Validation** | **CustId** | **CustName** | **CustGender** | **Address** | **PassNo** |
| **Valid** | Numeric | String | String | String | Numeric |
| **Invalid** | String | Numeric | Numeric | Numeric | String |
| **Validation** | ContactNo | Checkin | Staying | Checkout | RoomType |
| **Valid** | Numeric | Numeric | Numeric | Numeric | String |
| **Invalid** | String | String | String | String | Numeric |
| **Validation** | **RomNo** | **Floor** | **Amount** | **Status** | **Paid** |
| **Valid** | Numeric | String | Numeric | String | Numeric |
| **Invalid** | String | Numeric | String | Numeric | String |
| **Validation** | **Remain** | **TotalBill** |  |  |  |
| **Valid** | Numeric | Numeric |  |  |  |
| **Invalid** | String | String |  |  |  |

**BillMenu:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **Validation** | **CustId** | **Mid** | **TotalBill** |
| **Valid** | Numeric | Numeric | Numeric |
| **Invalid** | String | String | String |

**CheckOut:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Validation** | **CustId** | **CustName** | **CustGender** | **Address** | **PassNo** |
| **Valid** | Numeric | String | String | String | Numeric |
| **Invalid** | String | Numeric | Numeric | Numeric | String |
| **Validation** | **ContactNo** | **Checkin** | **Staying** | **Checkout** | **RoomType** |
| **Valid** | Numeric | Numeric | Numeric | Numeric | String |
| **Invalid** | String | String | String | String | Numeric |
| **Validation** | **RomNo** | **Floor** | **Amount** | **Status** | **ManuBill** |
| **Valid** | Numeric | String | Numeric | String | Numeric |
| **Invalid** | String | Numeric | String | Numeric | String |
| **Validation** | **Paid** | **Remain** | **TotalBill** |  |  |
| **Valid** | Numeric | Numeric | Numeric |  |  |
| **Invalid** | String | String | String |  |  |

**Source Code:-**

**LoginForm:**

Imports System.Data

Imports System.Data.SqlClient

Public Class LoginForm

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

Dim da As SqlDataAdapter

Dim ds As New DataSet

aa()

Dim sqlq As String

sqlq = "select \* from UserMaster where UserName='" & UsernameTextBox.Text & "' and UserPassword='" & UsernameTextBox.Text & "'"

' constr = "Server=ABHI;Database=HTDB;Trusted\_Connection=True"

'con = New SqlConnection(constr)

'con.Open()

da = New SqlDataAdapter(sqlq, con)

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

MsgBox("You Login Successfully--> " & ds.Tables(0).Rows(0).Item(1))

MsgBox("Welcome" & ds.Tables(0).Rows(0).Item(1))

usertype = ds.Tables(0).Rows(0).Item(1)

'SplashScreen1.Show()

MDIParent1.Show()

Else

MsgBox("Invalid Login")

End If

End Sub

End Class

**MDI Form:**

Imports System.Windows.Forms

Public Class MDIParent1

Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs)

' Create a new instance of the child form.

Dim ChildForm As New System.Windows.Forms.Form

' Make it a child of this MDI form before showing it.

ChildForm.MdiParent = Me

m\_ChildFormNumber += 1

ChildForm.Text = "Window " & m\_ChildFormNumber

ChildForm.Show()

End Sub

Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs)

Dim OpenFileDialog As New OpenFileDialog

OpenFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

OpenFileDialog.Filter = "Text Files (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*"

If (OpenFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

Dim FileName As String = OpenFileDialog.FileName

' TODO: Add code here to open the file.

End If

End Sub

Private Sub SaveAsToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Dim SaveFileDialog As New SaveFileDialog

SaveFileDialog.InitialDirectory = My.Computer.FileSystem.SpecialDirectories.MyDocuments

SaveFileDialog.Filter = "Text Files (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*"

If (SaveFileDialog.ShowDialog(Me) = System.Windows.Forms.DialogResult.OK) Then

Dim FileName As String = SaveFileDialog.FileName

' TODO: Add code here to save the current contents of the form to a file.

End If

End Sub

Private Sub ExitToolsStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Me.Close()

End Sub

Private Sub CutToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

' Use My.Computer.Clipboard to insert the selected text or images into the clipboard

End Sub

Private Sub CopyToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

' Use My.Computer.Clipboard to insert the selected text or images into the clipboard

End Sub

Private Sub PasteToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

'Use My.Computer.Clipboard.GetText() or My.Computer.Clipboard.GetData to retrieve information from the clipboard.

End Sub

Private Sub ToolBarToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

End Sub

Private Sub StatusBarToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

End Sub

Private Sub CascadeToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Me.LayoutMdi(MdiLayout.Cascade)

End Sub

Private Sub TileVerticalToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Me.LayoutMdi(MdiLayout.TileVertical)

End Sub

Private Sub TileHorizontalToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Me.LayoutMdi(MdiLayout.TileHorizontal)

End Sub

Private Sub ArrangeIconsToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

Me.LayoutMdi(MdiLayout.ArrangeIcons)

End Sub

Private Sub CloseAllToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs)

' Close all child forms of the parent.

For Each ChildForm As Form In Me.MdiChildren

ChildForm.Close()

Next

End Sub

Private m\_ChildFormNumber As Integer

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

CustMstr.Show()

End Sub

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

EmployeeDetails.Show()

End Sub

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

Room\_Information.Show()

End Sub

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

ItemMstr.Show()

End Sub

End Class

**CustomerMAster:**

Imports System.Data

Imports System.Data.SqlClient

Public Class CustMstr

Dim da As SqlDataAdapter

Dim ds As New DataSet

Dim sqlq As String

' Dim dt As DataTable

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

Call aa()

ds = New DataSet

Dim s As String

s = "select \* from CustMaster"

da = New SqlDataAdapter(s, con)

ds = New DataSet

da.Fill(ds)

txtCustID.Text = ds.Tables(0).Rows(0)(0).ToString

TextBox1.Text = ds.Tables(0).Rows(0)(1).ToString

cboGender.Text = ds.Tables(0).Rows(0)(2).ToString

txtAddress.Text = ds.Tables(0).Rows(0)(3).ToString

txtCity.Text = ds.Tables(0).Rows(0)(4).ToString

txtState.Text = ds.Tables(0).Rows(0)(5).ToString

txtCountry.Text = ds.Tables(0).Rows(0)(6).ToString

txtNationality.Text = ds.Tables(0).Rows(0)(7).ToString

txtPassNo.Text = ds.Tables(0).Rows(0)(8).ToString

txtContactNo.Text = ds.Tables(0).Rows(0)(9).ToString

da.Fill(ds)

DG2.DataSource = ds.Tables(0)

End Sub

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

If CheckBlank(txtCustID.Text) = False Then

If CheckBlank(TextBox1.Text) = True Or CheckBlank(cboGender.Text) = True Or CheckBlank(txtAddress.Text) = True Or CheckBlank(txtCity.Text) = True Or CheckBlank(txtState.Text) = True Or CheckBlank(txtCountry.Text) = True Or CheckBlank(txtNationality.Text) = True Or CheckBlank(txtPassNo.Text) = True Or CheckBlank(txtContactNo.Text) = True Then

MsgBox("Please Enter The Value")

Else

sqlq = "Insert into CustMaster values(" & txtCustID.Text & ",'" & TextBox1.Text & "','" & cboGender.Text & "','" & txtAddress.Text & "','" & txtCity.Text & "','" & txtState.Text & "','" & txtCountry.Text & "','" & txtNationality.Text & "'," & txtPassNo.Text & "," & txtContactNo.Text & " )"

da = New SqlDataAdapter(sqlq, con)

ds = New DataSet

da.Fill(ds)

Call gridfill()

MsgBox("Records Sucessfully Added")

End If

End If

End Sub

Public Sub gridfill()

Dim s As String

s = "select \* from CustMaster"

da = New SqlDataAdapter(s, con)

ds = New DataSet

da.Fill(ds)

DG2.DataSource = ds.Tables(0)

End Sub

Private Sub DG2\_CellClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG2.CellClick

txtCustID.Text = DG2.CurrentRow.Cells(0).Value

TextBox1.Text = DG2.CurrentRow.Cells(1).Value

cboGender.Text = DG2.CurrentRow.Cells(2).Value

txtAddress.Text = DG2.CurrentRow.Cells(3).Value

txtCity.Text = DG2.CurrentRow.Cells(4).Value

txtState.Text = DG2.CurrentRow.Cells(5).Value

txtCountry.Text = DG2.CurrentRow.Cells(6).Value

txtNationality.Text = DG2.CurrentRow.Cells(7).Value

txtPassNo.Text = DG2.CurrentRow.Cells(8).Value

txtContactNo.Text = DG2.CurrentRow.Cells(9).Value

End Sub

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

If CheckBlank(txtCustID.Text) = False Then

If CheckBlank(TextBox1.Text) = True Or CheckBlank(cboGender.Text) = True Or CheckBlank(txtAddress.Text) = True Or CheckBlank(txtCity.Text) = True Or CheckBlank(txtState.Text) = True Or CheckBlank(txtCountry.Text) = True Or CheckBlank(txtNationality.Text) = True Or CheckBlank(txtPassNo.Text) = True Or CheckBlank(txtContactNo.Text) = True Then

MsgBox("Please Enter The Value")

Else

sqlq = "update CustMaster set CustName= '" & TextBox1.Text & "',Gender= '" & cboGender.Text & "',Address= '" & txtAddress.Text & "',City= '" & txtCity.Text & "',State= '" & txtState.Text & "',Country='" & txtCountry.Text & "',Nationality= '" & txtNationality.Text & "',Passport= " & txtPassNo.Text & ",Contact= " & txtContactNo.Text & " where CustId=" & txtCustID.Text & ""

da = New SqlDataAdapter(sqlq, con)

ds = New DataSet

da.Fill(ds)

Call gridfill()

MsgBox("Records Sucessfully Added")

End If

End If

End Sub

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

Me.Hide()

Search\_Customer.Show()

Call gridfill()

End Sub

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

Call gridfill()

Call blank()

End Sub

Public Sub blank()

txtCustID.Text = ""

TextBox1.Text = ""

cboGender.Text = ""

txtAddress.Text = ""

txtCity.Text = ""

txtState.Text = ""

txtCountry.Text = ""

txtNationality.Text = ""

txtPassNo.Text = ""

txtContactNo.Text = ""

End Sub

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

Me.Hide()

End Sub

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

sqlq = "delete from CustMaster where CustId= " & txtCustID.Text & ""

da = New SqlDataAdapter(sqlq, con)

ds = New DataSet

da.Fill(ds)

Call gridfill()

End Sub

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

Call blank()

Dim s As String

Dim da As SqlDataAdapter

Dim ds As DataSet

s = "select case when count(CustId)=0 then 1 else (max(CustId)+1) end from CustMaster"

da = New SqlDataAdapter(s, con)

ds = New DataSet

da.Fill(ds)

txtCustID.Text = ds.Tables(0).Rows(0)(0).ToString

End Sub

Private Sub txtPassNo\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtPassNo.KeyPress

onlynumbers(e)

End Sub

Private Sub TextBox1\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress

onlyAlphabate(e)

End Sub

Private Sub txtCity\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtCity.KeyPress

onlyAlphabate(e)

End Sub

Private Sub txtState\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtState.KeyPress

onlyAlphabate(e)

End Sub

Private Sub txtCountry\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtCountry.KeyPress

onlyAlphabate(e)

End Sub

Private Sub txtContactNo\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtContactNo.TextChanged

'onlynumbers(e)

End Sub

Private Sub txtCustID\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtCustID.KeyPress

onlynumbers(e)

End Sub

End Class

**Module1:**

Imports System.Data

Imports System.Data.SqlClient

Module Module1

Dim da As SqlDataAdapter

Dim dt As DataTable

Public con As SqlConnection

Public usertype As String

Public Sub aa()

con = New SqlConnection("Server=ABHI;Database=HTDB;Trusted\_Connection=True")

con.Open()

End Sub

End Module

**Validations:-**

Imports System.Text.RegularExpressions

Module validations

Function CheckBlank(ByVal blank As String) As Boolean

If blank = "" Then

CheckBlank = True

Else

CheckBlank = False

End If

End Function

Public Function CheckNumeric(ByVal strVal As String) As Boolean

If IsNumeric(strVal) = True Then

CheckNumeric = True

Else

CheckNumeric = False

End If

End Function

Public Sub onlynumbers(ByVal e As System.Windows.Forms.KeyPressEventArgs)

If Not Asc(e.KeyChar) = 8 Then 'For Backspace Work.

If InStr("0123456789", e.KeyChar) = 0 Then

e.Handled = True

'Same Like Keyascii=0 in vb6.0

MsgBox("Enter Only Numbers")

End If

End If

End Sub

Public Sub onlyAlphabate(ByVal e As System.Windows.Forms.KeyPressEventArgs)

If Not Asc(e.KeyChar) = 8 Then 'For Backspace Work.

If InStr("abcdefghijklmnopqrstuvwxyz .", e.KeyChar, CompareMethod.Text) = 0 Then

e.Handled = True

MsgBox("Enter Only Alphabate") 'Same Like Keyascii=0 in vb6.0

End If

End If

End Sub

End Module

**ViewAllRooms:**

Imports System.Data

Imports System.Data.SqlClient

Public Class ViewAllRooms

Dim sqlq As String

Dim da As SqlDataAdapter

Dim ds As DataSet

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

Call aa()

Dim sqlq As String

sqlq = "select count(\*) from RoomMstr where status='Available'"

Dim da As SqlDataAdapter

Dim ds As DataSet

da = New SqlDataAdapter(sqlq, con)

ds = New DataSet()

da.Fill(ds)

TextBox1.Text = ds.Tables(0).Rows(0)(0)

Dim s1 As String

s1 = "select RoomNo,RoomType,RoomPrice from RoomMstr where Status='Available '"

Dim da1 As SqlDataAdapter

Dim ds1 As DataSet

da1 = New SqlDataAdapter(s1, con)

ds1 = New DataSet()

da1.Fill(ds1)

DataGridView1.DataSource = ds1.Tables(0)

Dim s4 As String

s4 = "select RoomNo,RoomType,RoomPrice from RoomMstr where Status='Alloted '"

Dim da4 As SqlDataAdapter

Dim ds4 As DataSet

da4 = New SqlDataAdapter(s4, con)

ds4 = New DataSet()

da4.Fill(ds4)

DataGridView2.DataSource = ds4.Tables(0)

Dim s As String

s = "select count(\*) from RoomMstr where status='Alloted'"

Dim da2 As SqlDataAdapter

Dim ds2 As DataSet

da2 = New SqlDataAdapter(s, con)

ds2 = New DataSet()

da2.Fill(ds2)

TextBox2.Text = ds2.Tables(0).Rows(0)(0)

End Sub

Public Sub gridfill()

Dim s As String

s = "select \* from RoomMstr"

da = New SqlDataAdapter(s, con)

ds = New DataSet

da.Fill(ds)

DataGridView1.DataSource = ds.Tables(0)

End Sub

Private Sub DataGridView1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

aa()

Dim sqlq As String

sqlq = "select \* from RoomMstr where Status='Available '"

'sqlq = "select RoomNo,RoomType,RoomPrice from RoomMstr where Status='Available '"

Dim da As SqlDataAdapter

Dim ds As DataSet

da = New SqlDataAdapter(sqlq, con)

ds = New DataSet()

da.Fill(ds)

Call gridfill()

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

End Sub

End Class

**System Implementation:-**

**Implementation**

The following steps were carried out in implementation phase.

* Conduct Training: The training was conducted for the employees of the company to make them familiar with the system.
* Bug fixing and documentation: Any errors that occurred were solved and documented.
* Install the system: The system was then installed.

**Maintenance**

**SYSTEM MAINTENANCE:**

The maintenance of software is the time period in which the software is software product performs useful work. Maintenance activities involve making enhancement activities to the, adapting product to new environment and correcting problems. Software enhancement may involve providing new functional capabilities, improving user displays and modes of interaction.

Adaptation of software to a new environment may involve moving the software to a different machine. Problem correction involves modification and revalidation of software to correct errors. The four types of maintenance activities are described below:

🡪 Corrective Maintenance

🡪 Preventive Maintenance

🡪 Perfective Maintenance

🡪 Adaptive Maintenance

Corrective Maintenance:

Corrective maintenance can be defined as the [maintenance](http://en.wikipedia.org/wiki/Maintenance) which is required when an item has failed or worn out, to bring it back to working order.

Corrective maintenance is the most commonly used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometime it causes spreading of damage to other parts. In most cases this is costly business. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial.

Adaptive Maintenance:

Adaptive maintenance is an activity that modifies software to properly interface with the changing environment.

Perfective Maintenance:

Perfective maintenance is performed to satisfy user requests such as new Capabilities, modifications to existing functions and general enhancements.

Preventive Maintenance:

Preventive maintenance occurs when software is changed to improve future maintainability or to provide a better basic for future enhancements.

**Conclusion**

An attempt is made in all its earnest towards the successful completion of the project. This system was verified with valid as well as with invalid data.

This system is user friendly since it has been developed in Visual Studio 2008, a successful GUI environment. Since the connection can be extended to any database. The control will be more powerful.

Connecting it to any type of database extends the development control. Any suggestions for future development of the system are welcome.

Upgrading the system can be done without affecting the proper functioning of system.

**BIBLIOGRAPHY:**

**For Database:**

1. System Analysis and Design –By Satzinger
2. Software Engineering—By McGrill Hill 7th Edition

**For VB.NET:**

1. Munarch VisualBasic 2008,Anne Boehm, Munarch

[www.google.com](http://www.google.com)

[www.w3schools.com](http://www.w3schools.com)