PROJECT REPORT

ON

ONLINE REAL ESTATE WEBSITE

Submitted towords partial fulfillment of the degree of

Bachelor of science

(BSCIT)

by

MR.Omkar Ramchandra Kadam(seat no:1133376)

Mr. Rahul TUKAram Karande

UNDER THE GUIDENCE OF

PROF. AMARESH PATRA

to

university of Mumbai

through

VIKAS college of arts, science &commerce

2014- 2015.

**Acknowledgement**

At every outset I express my gratitude to almighty lord for showering his grace and blessings upon me to complete this project.

Although our name appears on the cover of this book, many people had contributed in some form or the other form to this project Development. We could not done this project without the assistance or support of each of the following we thank you all.

I wish to place on my record my deep sense of gratitude to my project guide, **Prof**.**AMARESH PATRA** for his constant motivation and valuable help through the project work. I would like to thank my entire department of Information Technology for their solitude help to complete my project work. I also extend my thanks to other Faculties for their Cooperation during my Course.

Finally I would like to thank my friends for their cooperation to complete this project.

THANK YOU

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No. | Topic | | Page No. |
| 1 | Introduction  1.1 Introduction of the Project  1.2 Problem Definition | | 1-8  4  6 |
| 2 | Proposed System  2.1 Proposed System  2.2 Scope of the Project  2.3 Advantages of Proposed System  2.4 Feasibility Study  2.5 Technology Used  2.6 Operating Environment | | 9-33  9  12  14  16  20  28 |
| 3 | Data Design  3.1 Database Table Design | | 34-42  39 |
| 4 | 4.1 Gantt Chart | | 43-45 |
| 5 | Analysis and Design  5.2 Activity Diagram  5.1 Use Case Diagram  5.4 Context Diagram  Class Diagram  5.5 Entity Relationship Diagram  5.6 Sequence Diagram | | 46-73  46  53  56  58  63  68 |
| 6 | Testing and Reports  6.1 Testing  6.2 Test Report | | 74-  75 |
| 7 | Hardware Software Requirements  7.1 Hardware & Software Specifications | |  |
| 8 | Screen Layouts  8.1 Screen Shot of Pages | |  |
| 9 | | Conclusion |  |
| 10 | | Bibliography |  |

**INTRODUCTION**

Introduction

The introduction specifies information about the project and also specifies

* What project is this?
* How our project is?
* What are various output of our project?

**What Project is this?**

In this rat race of globalization issue where 3rd world countries like India needs to compete with the first world country to mark its presence all it needs its technology and the closest hit is computers. Hence a computer has invaded all the fields.

Just tried to give it an international touch. It is a Website project which is designed to simplify the process of Real Estate. In Real Estate there are lots of calculations like keeping record of Properties in terms of location, rate, number of registered members, and especially calculations of Loan EMI calculation concept at the time of transaction etc. These calculations are not only complicated but also brain eating. Our project also consists oftables to store**,** individual detail of property, customer’s details, and feedback’s details.

So the owner can have all these details in his personal computers rather than having hundreds of registers.

We are designing our project with the help of VISUAL STUDIO, ASP.NET with C# and Scripting Language and SQL SERVER 2008 as back end. The information about these applications is the following sections.

So the project entitled “Online Real Estate Website” is an end user Website. The Real estate Website we are designing that helps the Developer to handle those task that had been handled manually.

The tasks like:

* + Seeping record of properties available,
  + Stores & gives details of properties,
  + Maintaining record of customers visited his website,
  + Displays information of customers.
  + Displays information properties.
  + Displays information of location.
  + Displays information of location placed by customer.

Introduction Of The Project

This is the form design and display. It is the user interface for the project.

Online Real Estate Website is an academic project developed for the purpose of automating the properties.The organization profile of the system in concern is:

**GUI**

**Connectivity**

**Collation**

**MIS**

**Primary Process**

**User**

Fig. System Flow

Description

* **The User :**

The main input source in the system is the user. The events are created keeping user input in mind.

* **Primary Process:**

A primary process so is the first process or the first event in the system, which come from the external source i.e. input from user, command from user, etc. This is called as initially events, which starts off the system.

* **MIS:**

This is the Management information system. It is the collection of all external, temporal and state events, which can occur in the system and their sources and destinations. It is summarized in the database tables.

* **Collation:**

Collation is the front end validations, which the system is, ask to perform. All external events checks are collated and provided in Data Flow Diagram.

* **Connectivity:**

In the connectivity all the front end events are connected triggers are executed this is the “Back End” part of the project.

* **GUI (Graphical User Interface):**

This is the form design and display. It is the user interface for the project.

**PROBLEM**

**DEFINATION**

PROBLEM DEFINITION

**Working Of The Real Estate Company**

* Customer walks into the developers office asks Developer for required property.
* Developer searches for named properties/location/flats among cluster/group of books.
* After finding out it serves to the customer.
* If customer is interested in buying that flats, Developer sales the property/flat by completing all formalities, and makes entry into the Customer Details Register manually.
* All the permanent data of the order such as the name of the property/flat, Estate\_ID, as well as name of buyer/customer, Member\_ID and so on are maintained in a master file for future reference.
* The customer has the option of paying by cash or check or credit card.
* At the time customer pays the price he is issued a receipt acknowledging the same.

After carefully studying requirements and working of the organization it has been decided to develop an Online Real Estate Website.

An Online Real Estate Website is nothing but a Website that is supported by a (relational or non-relational) database.

It consists of data-entry of forms, in which the customers can fill-up the data.

After validation, the data can be added to the database.

the Owner/Admin can search, view, and records

**The Current/Existing System**

* All the above-mentioned tasks are taken care off manually.
* The details of all the Properties, customers, stocks and feedbacks are maintained in separate registers.

**Limitations ofthe Current/Existing System**

* Maintaining details of the customers in registers is a time-consuming process and error-prone.
* If a particular record is to be searched then one has to search through a lot of physical records.
* No Security has been provided to the data.
* As the work is manual it consumes lot of time and energy.
* There is no facility provided for online booking of property

**Drawbacks of the Current/Existing system To the Admin**

* At present everything is taken care-off manually. The admin does not have access to the information that is generated by the Developer.
* And because the work is done manually there is area possibility of the information being error-prone.
* Thus the admin is not in a position to make quick decisions.

**PROPOSED**

**SYSTEM**

PROPOSED SYSTEM

**Proposed System Feature**

1**.** The proposed system is developed using VISUAL STUDIO, ASP.Net with C#, and Microsoft SQL Server 2008 as a back end database. The proposed system deals with very popular World Wide Web (www) i.e. Internet.

2. The database is used i.e. Microsoft SQL Server 2008 is very faster well designed tool to the user for quick manipulation with it.

3. The proposed system has better both Input/output capabilities of each the user activities while interacting with the system.

4. The search/retrieval of the properties/flats is much faster than the present system. Hence it cause to saving time for the further work.

5. The user can have fast interaction with the system by inserting keywords at the respective places, by clicking on the buttons or links etc. Because not only the front end provides the faster interaction with the records but with back end also provides the proper interaction with the records and gives or prompts the information to the end user if he is making error during work.

6. Searchingfeature is quite faster than current system. Because it searches directly from system that is from the front end.

7. The most important feature of this system is online exhibiting the properties, Thissaves time to visit at the actual place where the property is located.

8. Feedback feature is included which will helps in keeping attention on customers’ needs and requirements. The customer can have user friendliness with the system by allowing him to leave his response, advice or opinion about this system, comments on services provided by the system.

**SCOPE OF THE**

**PROJECT**

SCOPE OF THE PROJECT

The Online Real Estate isa Website design to handle the basic tasks of the Real Estate.

Hence the main function of this system or company is to convert the manual trading service of properties / flats in the better quality service that customers want.

The functionality of this system is as like handling the properties or property management and handles the main function that mentioned above.

The functionality of this system is described as follows:

* Allows the customers to view or get details of various properties/flats available.
* Add the properties/flats to the stock.
* Allow admin to checks the availability of properties/flats.
* Customer can view all properties/flats online.
* Displays the details like customers details, rates of flats, locations, remarks details etc.

This is all the functionality of the Online Real Estate Website.

**ADVANTAGES**

**OF THE**

**PROPOSED SYSTEM**

ADVANTAGES OF THE PROPOSED SYSTEM

* The first advantage of the proposed system is that its front end provide easy and précised information to the user to interact with the system and hence it is faster to complete the work.
* The next advantage of the present system is that its faster capability with the database. Even the database also provides its inbuilt features to maintain the records. It also provides security to the records from the system itself and also from the database itself. Hence it prevents the database conjunction
* The present system provides the better screening of the record, another advantage of the proposed system is its faster capability to search the records from the database.

FEASIBILITY STUDY

**FEASIBILITY**

**STUDY**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operational Feasibility
* Economical Feasibility

**TECHNICAL FEASIBILITY**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

**OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**ECONOMICAL FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

.

**TECHNOLOGY USED**

**Technology Used**

* Server Side

1. ASP.NET
2. C#
3. Ajax

* Data Base

1. Microsoft SQL Server 2008

* Web-Service

1. IIS (Internet Information Services)

**ASP.NET**

ASP.NET is an open source server-side Web application framework designed for Web development to produce dynamic Web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

ASP.NET is in the process of being re-implemented as a modern and modular web framework, together with other frameworks like Entity Framework. The new framework will make use of the new open-source .NET Compiler Platform (code-name "Roslyn") and be cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) will merge into a unified MVC 6.

**C#**

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma and ISO. C# is one of the programming languages designed for the Common Language Infrastructure.

C# is intended to be a simple, modern, general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 5.0, which was released on August 15, 2012.

**AJAX**

**What is AJAX?**

AJAX = Asynchronous JavaScript and XML.

AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.

With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.

AJAX is a web browser technology independent of web server software.

A user can continue to use the application while the client program requests information from the server in the background.

* Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
* Data-driven as opposed to page-driven.

## AJAX is Based on Open Standards

AJAX is based on the following open standards:

* Browser-based presentation using HTML and Cascading Style Sheets (CSS).
* Data is stored in XML format and fetched from the server.
* Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
* JavaScript to make everything happen.

**Microsoft SQL Server 2008**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. Its primary query languages are T-SQL and ANSI SQL.

IIS (Internet Information Service)

Internet Information Services (IIS, formerly Internet Information Server) is an extensible web server created by Microsoft for use with Windows NT family. IIS supports HTTP, HTTPS, FTP, FTPS, SMTP and NNTP. It has been an integral part of the Windows NT family since Windows NT 4.0, though it may be absent from some editions (e.g. Windows XP Home edition). IIS is not turned on by default when Windows is installed. The IIS Manager is accessed through the Microsoft Management Console or Administrative Tools in the Control Panel.

IIS

Microsoft Internet Information S4ervices (IIS; formerly called Server is a set of Internet-based services for servers using Microsoft Windows. It is the world’s second most popular web server in terms of overall websites. As of September 2007 it served 34.94% of all websites and 36.63% of all active websites according to Net craft. The servers currently include FTP, SMTP, NNTOP, and HTTP/HTTPS.

**OPERATING**

**ENVIRoNMENT**

OPERATING ENVIRONMENT

**Microsoft Visual Studio**

Microsoft Visual Studio is Microsoft’s flagship software development product for computer programmers. It centers on an integrated distribution environment which has programmers create stand alone, and web services that run on any platforms supported by Microsoft’s .Net Framework (for all versions after 6). Supported platforms include Microsoft windows, servers and workstations, Pocket PC, Smart Phones and World Wide Web browsers not the Java Virtual Machine that all other java tools target.

**Overview of .Net Framework**

The .NET framework is a new computing platform that simplifies application development in the highly distributed environment of the internet. The .NET framework is designed to fulfill following objectives:

* To provide a consistent object oriented programming environment whether object code is stored and executed locally but internet- distributed or executed remotely.
* To provide a code execution environment that minimizes software deployment and versioning conflicts.
* To provide a code execution environment that guarantees safe execution of code, including code created by an unknown or semi trusted third party.
* To provide a code execution environment that eliminates the performance problem of scripted or interpreted environments.
* To make the developer experience consistent across widely types of application, such as windows based applications and web based applications.
* To build all communication on industry standards to ensure that code based on the .NET framework can integrate with any other code.

The .NET framework has two main components: the common language runtime and the .Net framework class library. The common language runtime is the foundation of the .NET framework. You can think of the runtime as an agent that manages code at execution time, and remoting while also enforcing strict type safely and other forms of code accuracy that ensure security and robustness in fact the concept of code management is a fundamental principle of the runtime.

Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET frameworks is a comprehensive, object-oriented collection reusable types that you can use to develop applications ranging from traditional command line or graphical user interface (FGUI) applications to application base d on the latest innovations provided by ASP.NET, such as web forms and XML web services.

The .NET framework can be hosted by unmanaged component that load the common language runtime into their processes and initiate the execution of managed code. ASP.NET works directly with the runtime to enable ASP.NET application and XML web services, both of which are discussed later in this topic, Internet explorer is an example of unmanaged application that hosts the runtime (in the form of a MIME type extension). Using internet explorer to the host runtime enables you to embed managed components or windows forms controls in HTML documents. Hosting the runtime in this way makes mobile code 9similar to Microsoft Active Xr controls) possible, but with significant improvement that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your application and to the overall system. The illustration also shows how managed code operated with in a larger architecture.

We can use the .NET framework to develop the following types of application and services:

* Console applications.
* Window GUI application (Windows Forms) ASP.NET applications.
* XML Web services.
* Windows services.

**Common Language Runtime (CLR)**

The common language runtime (CLR) is responsible for runt-time services such as language integration; security enforcement; and memory, process and thread management. In addition, it has a roll at development time when features such as life cycle management strong type naming, cross-language exception handling, dynamic binding and so on, reduce the amount of code that a developer must write to turn the business logic the reusable component. The runtime can be hosted by high performance, server-side applications, such as Microsoft Internet Information Services (IIS) for building web applications with ASP.NE and the next release of Microsoft SQL Server. This infrastructure enables you to use code “managed “ by the .NET framework to write your business logic, while still enjoying the superior performance of the industry’s best enterprises servers that support runtime hosting.

**ASP.NET**

ASP.NET is a set of Microsoft.NET framework technologies used for building web applications and XML Web services. ASP.NET page execute on the server and generate markup such as HTML, WML or XML that is sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-driven programming model that improves performance and enables the separation of application logic and user interface. Both ASP.NET pages and ASP.NET web services files contain server-side (as opposed to client side logic) written in ASP.NET, C#.NET or any .NET compatible language, Web applications and XML

Web Services take advantage of the features of the common language runtime, such as type safety, inheritance, language, interoperability, versioning, and integrated security.

**.NET Framework Data Providers**

The .NET Framework Data providers are components that have been explicitly designed for data manipulation and fast, forward-only, read-only access to data. The connection object provides connectivity to a data source. The command object enables access to database commands to return data, modify data, run stored procedures and send or retrieve parameter information.

The Data Adapter provides a high-performance stream of data from the data source. Finally, the Data Adapter provides the bridge between the DataSet object and the data source. The Data Adapter uses command object to execute SQL commands at the data source to both load the DataSet with data and reconcile changes made to the data in the DataSet back to the data source.

**Microsoft Data Access Components(MDAC)**

Microsoft Data Access Components (MDAC) is a collection of core files provided to help applications by providing a means of accessing data. MDAC includes core files for Open Database Connectivity (ODBC), ActiveX Data Objects (ADO), OLEDB, Network libraries and client configuration tool for SQL Server. Depending on your data access strategy, you may need to ensure that MDAC is installed on the client computers, the business servers, the Web servers or the database servers. MDAC 2.6 or later is required by the .NET Framework and at least MDAC 2.7 SP1 is recommended.

**DATA**

**DESIGN**

DATA DESIGN

ADMIN

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Constraints** |
| User name | Varchar(10) | Primary key |
| Password | Varchar(10) | Not null |
| Company | Varchar(30) | Not null |
| Address | Varchar(100) | Not null |
| City | Varchar(20) | Not null |
| Pincode | Int | Not null |
| Phone\_no | Varchar(20) | Not null |
| Contact\_Person | Varchar(20) | Not null |
| Mobile\_no | Varchar(15) | Null |
| Email | Varchar(30) | Null |
| Logo | Varchar(15) | Null |
| Last\_Estate\_id | Int | null |

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Constraints** |
| Estate\_id | Int | Primary Key |
| Member\_id | Int | Foreign key |
| Area | Varchar(30) | Not null |
| Address | Varchar(200) | null |
| Type | Varchar(20) | Not null |
| Construction | Varchar(20) | Not null |
| Size | Real | Not null |
| Aprox\_price | Bigint | Not null |
| Entry\_Date | Datetime | Not null |
| Estate\_view | Varchar(20) | Null |
| Sales\_status | Char(1) | Not null |
| Status\_Date | Datetime | Null |
| Remark | Varchar(200) | Null |

ESTATE MASTER

MEMBER MASTE

|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Constraints** |
| Member\_id | Int | Primary key |
| Member\_type | Varchar(25) | Not null |
| User\_Id | Varchar(15) | Not null |
| Password | Varchar(15) | Not null |
| Name | Varchar(50) | Not null |
| Address | Varchar(200) | Not null |
| City | Varchar(50) | Not null |
| Pincode | Int | Not null |
| Phone\_no | Varchar(15) | null |
| Mobile\_no | Varchar(15) | Null |
| Email | Varchar(50) | null |
| Occupation | Varchar(30) | Not Null |
| Caste | Varchar(20) | Null |

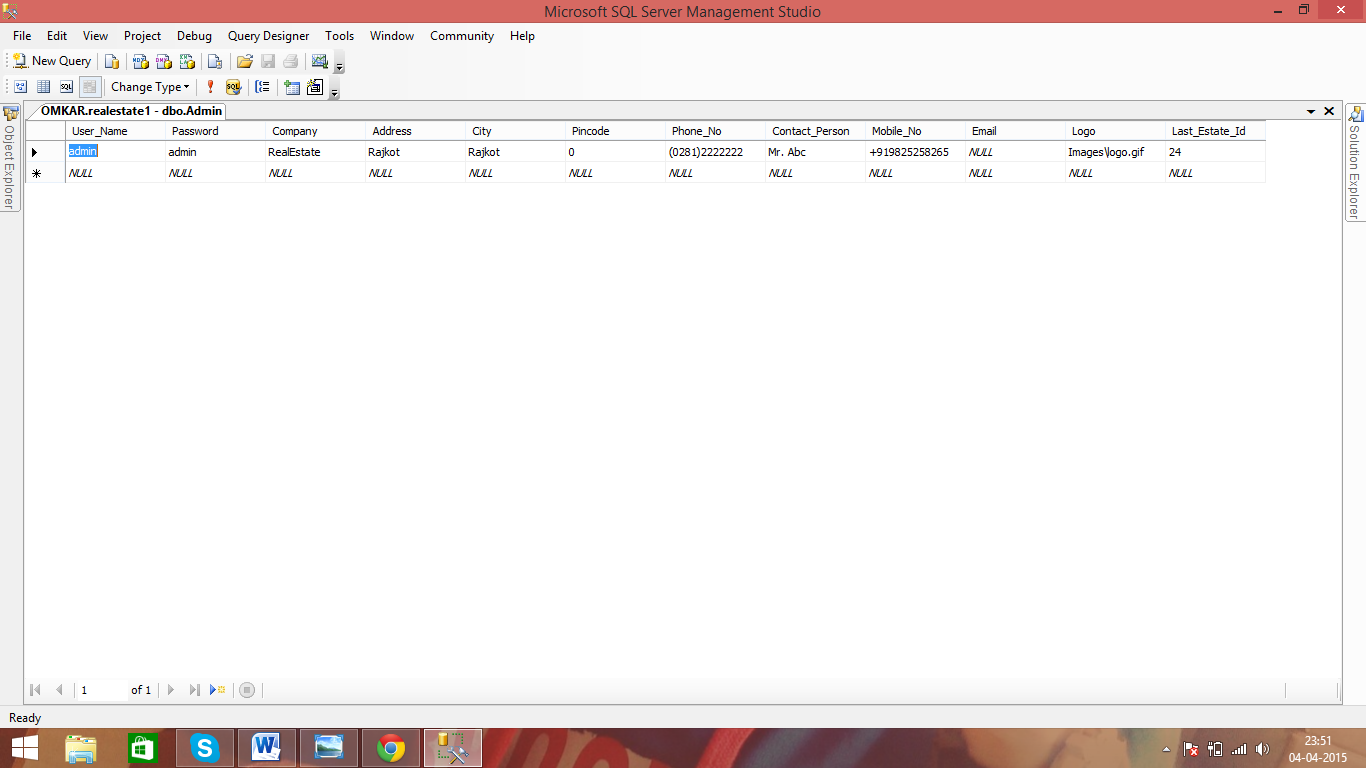
TRANSACTION MASTER

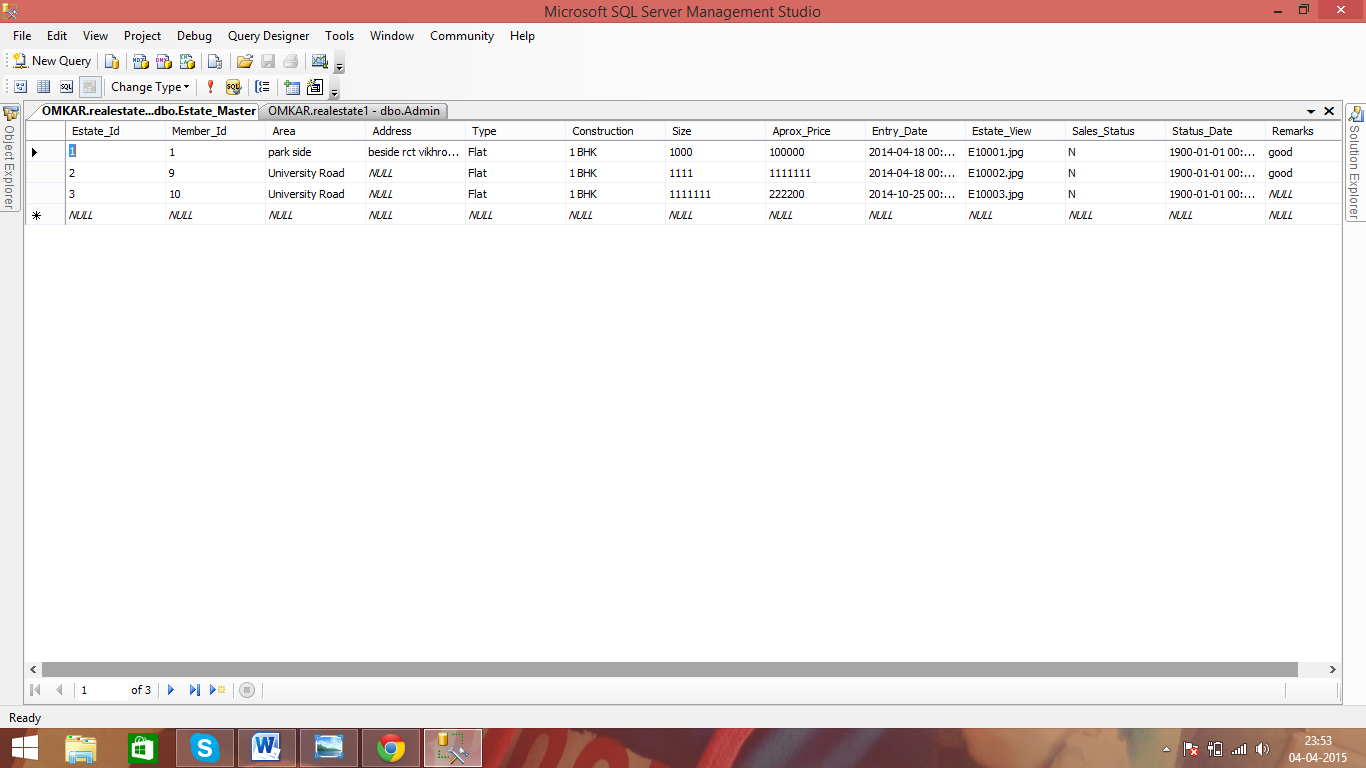
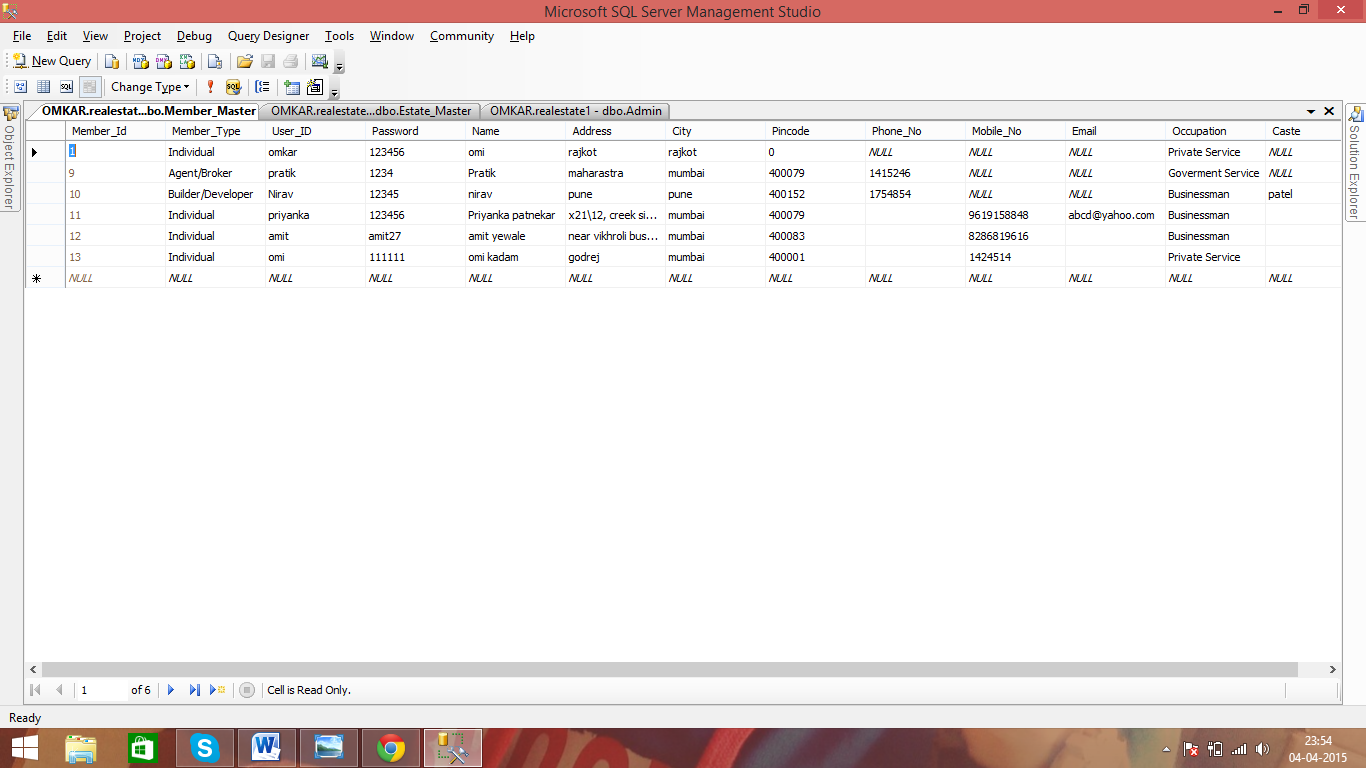
|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Constraints** |
| tran\_id | Int | Primary key |
| Tran\_date | Datetime | Not null |
| member\_Id | Int | Foreign key |
| Tran\_type | Char(1) | Not null |
| Estate\_id | Int | Not null |
| Purchase\_area | Varchar(30) | null |
| Purchase\_type | Varchar(20) | null |
| Construction | Varchar(20) | null |
| Estimate\_budget | Int | null |

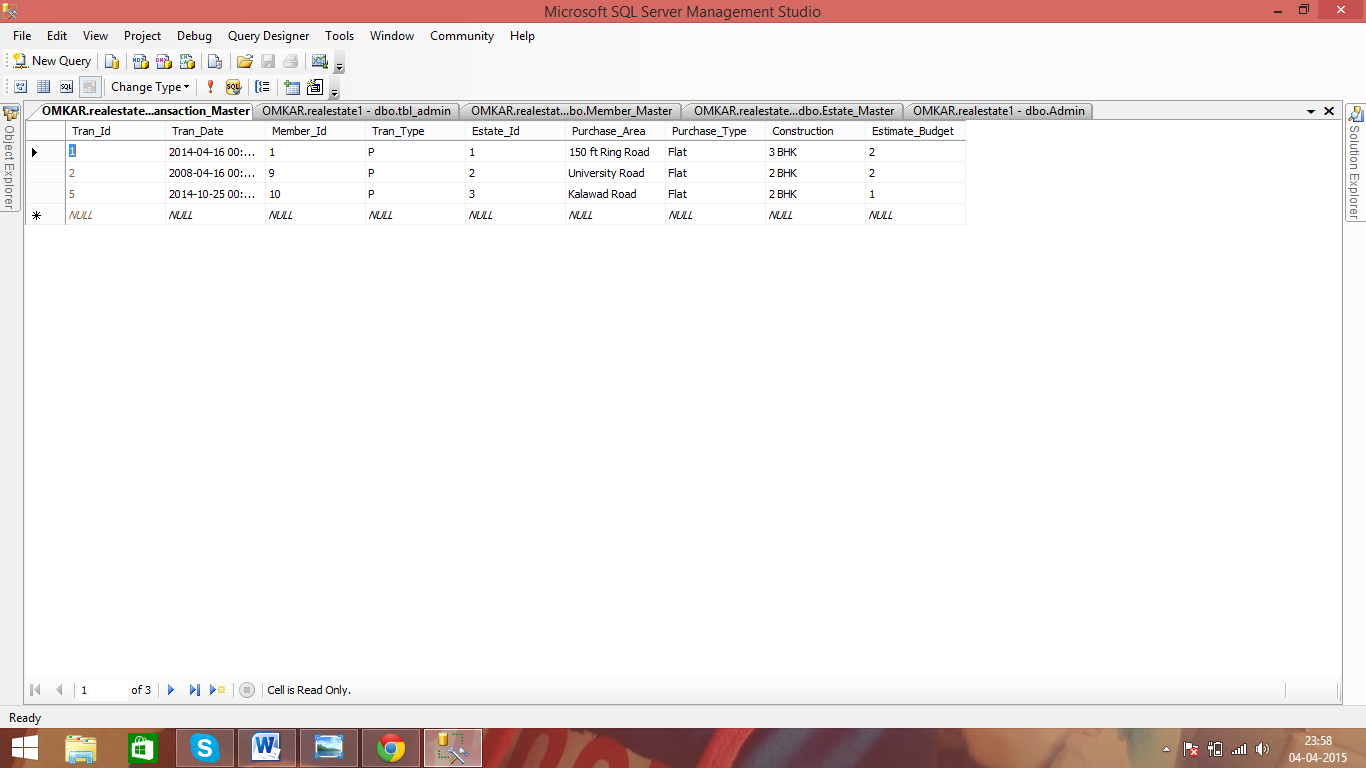
**DATABASE**

**TABLE DESIGN**

DATABASE TABLE DESIGN







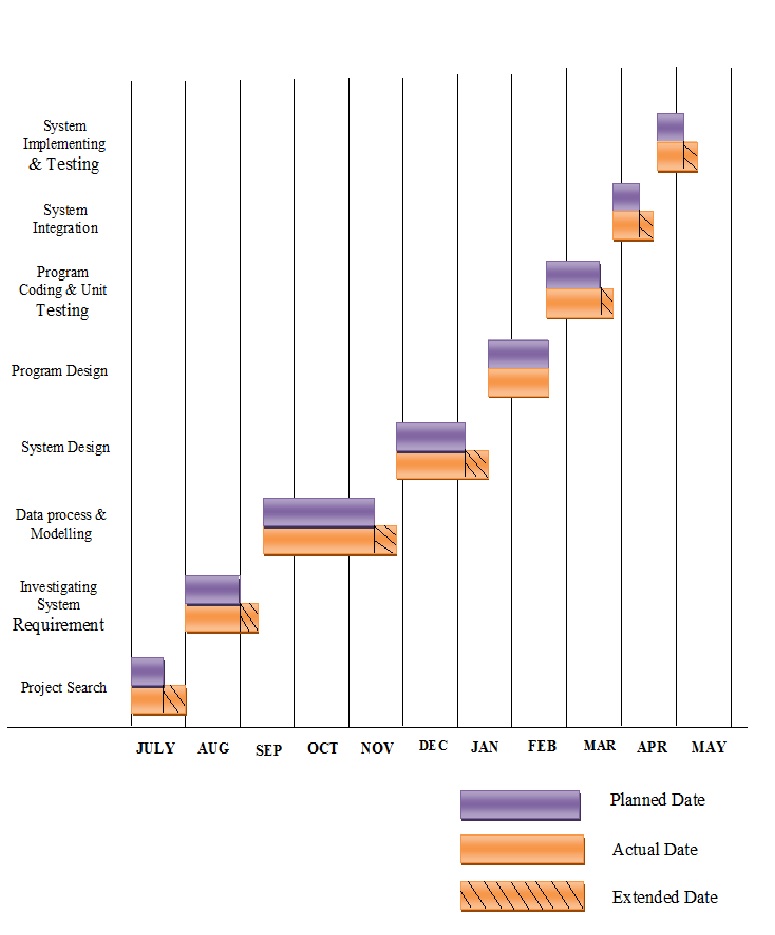
**GANTT**

**CHART**

# What is a Gantt chart?

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows you to see at a glance:

* What the various activities are
* When each activity begins and ends
* How long each activity is scheduled to last
* Where activities overlap with other activities, and by how much
* The start and end date of the whole project

GANTT CHART

**Analysis**

**And**

**Design**

**ACTIVITY dIAGRAM**

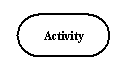
## What is an Activity Diagram?

An activity diagram illustrates the dynamic nature of a system by modelling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation. Because an activity diagram is a special kind of state chart diagram, it uses some of the same modeling conventions.

### Basic Activity Diagram Symbols and Notations

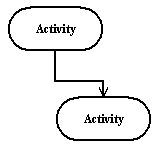
##### **Action states**

Action states represent the no interruptible actions of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.



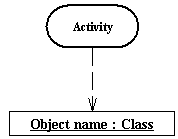
**Action Flow**

Action flow arrows illustrate the relationships among action states.



##### **Object Flow**

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



##### **Initial State**

A filled circle followed by an arrow represents the initial action state.

Initial State

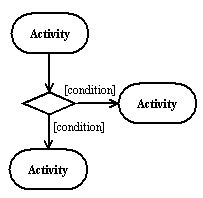
##### **Final State**

An arrow pointing to a filled circle nested inside another circle represents the final action state.

Final State

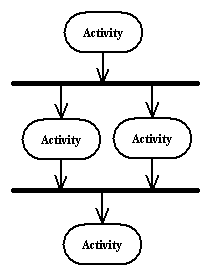
##### **Branching**

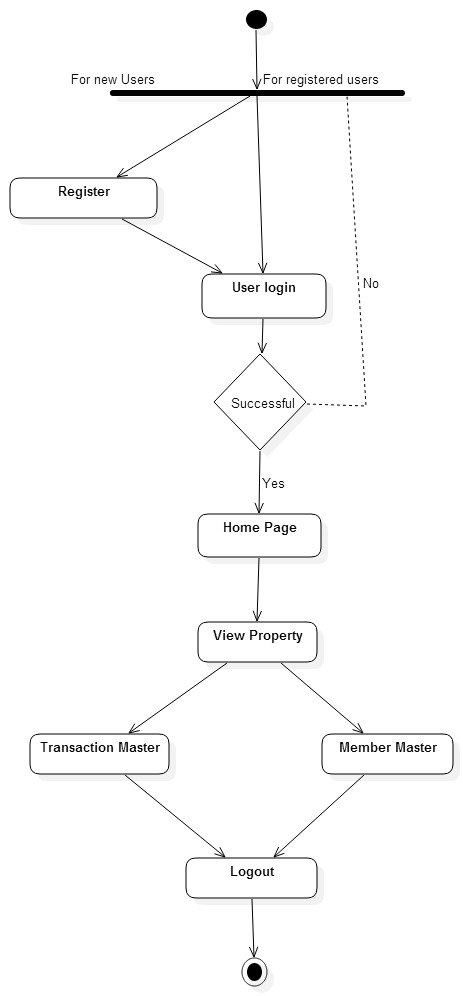
A diamond represents a decision with alternate paths. The outgoing alternates should be labelled with a condition or guard expression. You can also label one of the paths "else."



##### **Synchronization**

A synchronization bar helps illustrate parallel transitions. Synchronization is also called for king and joining.

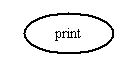


****

**USE CASE**

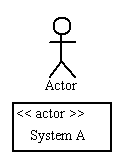
##### **Use Case**

Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.



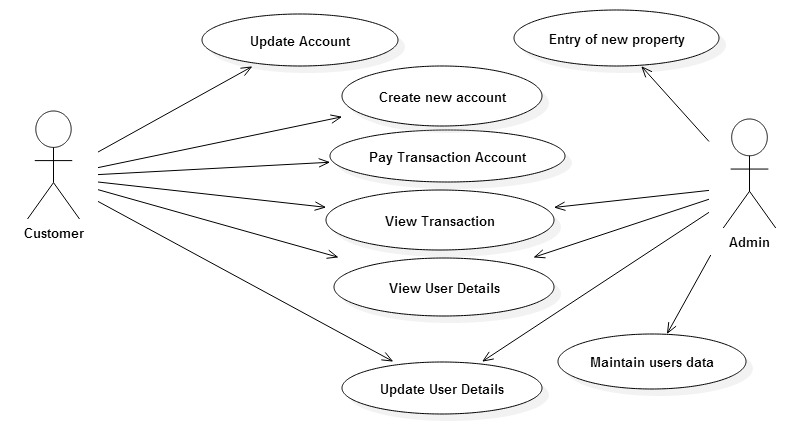
##### **Actors**

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



##### **Relationships**

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labelled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



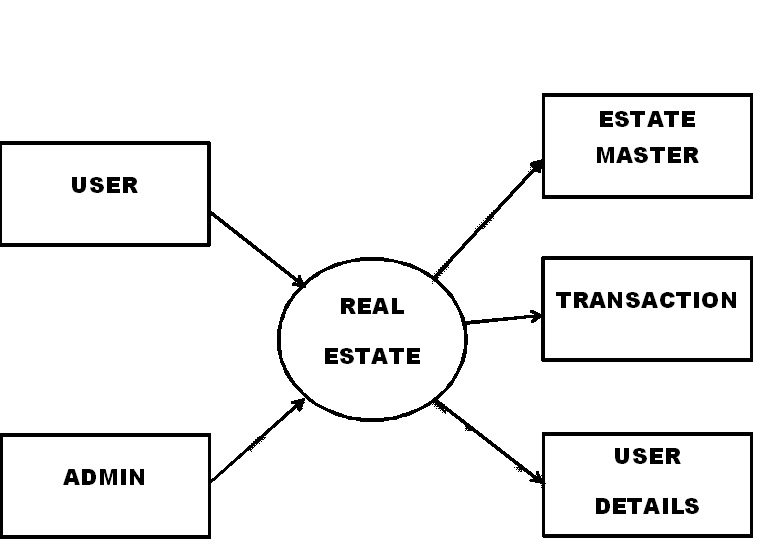
**CONTEXT**

**LEVEL**

**DIAGRAM**

**What is a Context Diagram?**

A Context diagram is a Data Flow Diagram (DFD) at 0 levels that describes the highest-level view of a system. All external agents and all data flows into and out of the system are in one diagram, with the whole system represented as one process. The context diagram is useful for showing system boundaries .The system scope is defined by what is represented within the single process and what is represented as external agent. External agents that supply or receive data from the system are outside of the system scope. Everything else is inside the system scope. The data flows come from the event table; they are triggers and response for all of the events. The context diagram provides a good overview of the scope of the system, showing the system the in “Context”.

****

**Fig.CONTEXT LEVEL DIAGRAM**

**Class Diagram**

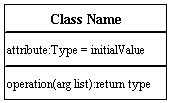
**What is a Class Diagram?**

Class diagrams are the backbone of almost every object-oriented method including UML. They describe the static structure of a system.

**Basic Class Diagram Symbols and Notations**

Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes.

Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition, and write operations into the third.

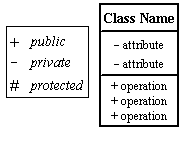


**Active Class**

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.



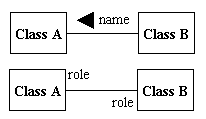
**Visibility**

Use visibility markers to signify who can access the information contained within a class. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class

.

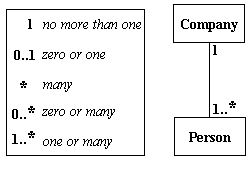
**Associations**

Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each other.



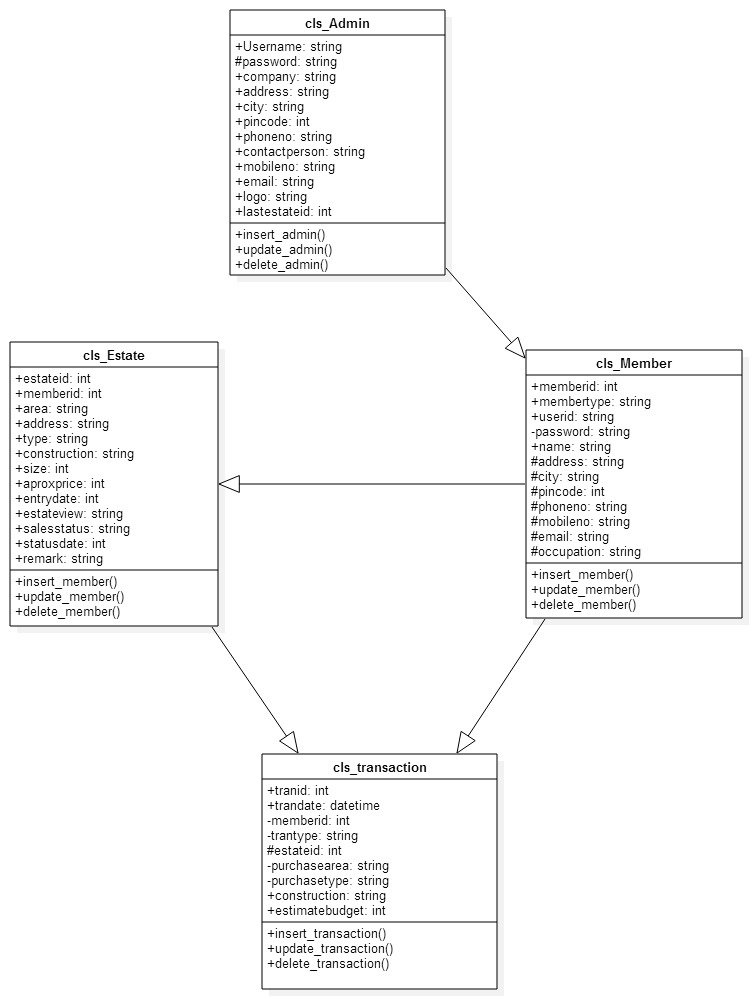
**Multiplicity (Cardinality)**

Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class. For example, one company will have one or more employees, but each employee works for one company only.



**Constraint**

Place constraints inside curly braces {}.



**ENTITY RELATIONSHIP**

**(ER)**

**DIAGRAM**

## What are Entity Relationship Diagrams?

Entity Relationship Diagrams (ERDs) illustrate the logical structure of databases.

### Entity Relationship Diagram Notations

**Entity**

An entity is an object or concept about which you want to store information.

Entity

##### **Weak Entity**

A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

Weak Entity

##### **Key attribute**

A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

Key attribute

##### **Multivalued attribute**

A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

Multivalued attribute

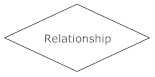
##### **Derived attribute**

A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

Derived attribute

##### **Relationships**

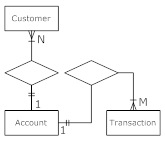
Relationships illustrate how two entities share information in the database structure.



##### **Cardinality**

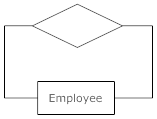
Cardinality specifies how many instances of an entity relate to one instance of another entity.

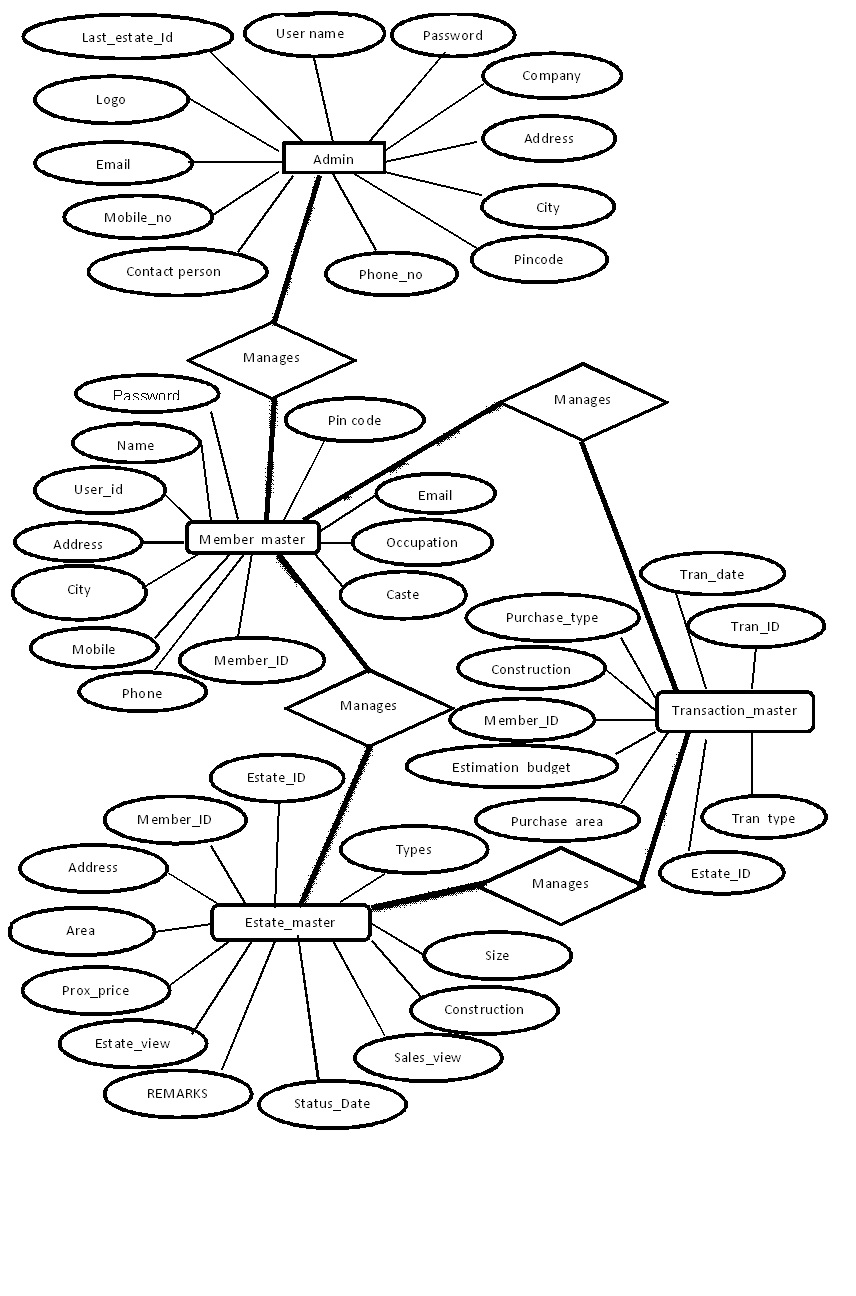
Cordiality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, cordiality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and cordiality specifies the absolute minimum number of relationships.



##### **Recursive relationship**

In some cases, entities can be self-linked. For example, employees can supervise other employees.





**SEQUENCE DIAGRAM**

## What is a Sequence Diagram?

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

### Basic Sequence Diagram Symbols and Notations

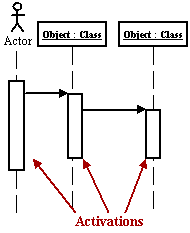
##### **Class roles**

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

Class roles

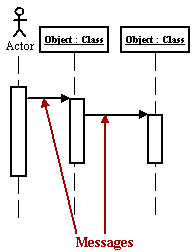
##### **Activation**

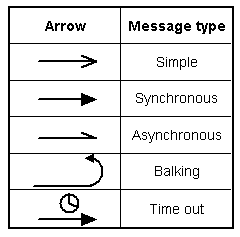
Activation boxes represent the time an object needs to complete a task.



**MESSAGES**

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent

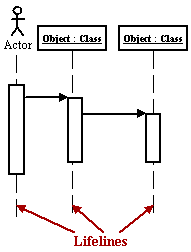
Asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.



Various message types for Sequence and Collaboration diagrams

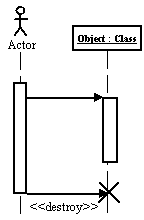
##### **Lifelines**

Lifelines are vertical dashed lines that indicate the object's presence over time.



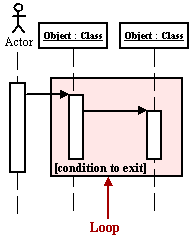
##### **Destroying Objects**

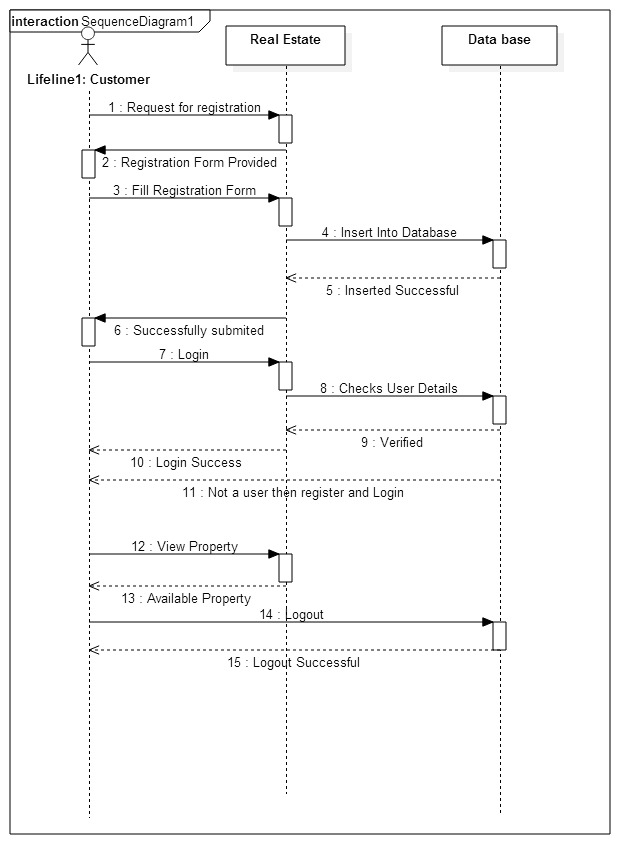
Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X.



##### **Loops**

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [ ].





**SYSTEM TESTING**

System Testing

**Testing:**

After completing any project it is necessary to test that particular project. The need of testing the projects comes because of the Varity, that the System is working properly or not whether it can give the output as per our requirement or not.

Testing of project is classified in two types:

* Logical Test
* Error Test

**Logical Test:**

As per the logical test concern project is tested fully with the help of logical flow of data. In this required output is getting or not. If the required output is matched with the expected output then only we can declare that the System is working properly.

After completing these types of testing the project is supposed to be logically fit for the further process.

**Error Test:**

All the errors in our project (if any) should be tested with the help of this error test. In this test we have to check, if some data is missing, decreasing, deleting and if not linked properly .So all such types of error must be removed in the developed system.

System Development Model

WaterfallModel:

The waterfall model also called the linear sequential model or

Classic life cycle model, is the traditional life cycle model, where each phase has a defined a start point and an end point, and clear deliverables from one phase to the next.

The sequential move from one phase to the next gave rise to name waterfall model. This model is useful because of the specific goal at the end of each phase, rather than only at the end of the cycle.

**Main basis reason behind the choice of using the waterfall model is of followingfacts:**

1. Testing is inherent to every phase of the waterfall model.
2. It is an enforced disciplined approach.
3. It is documentation driven that is documentation produced at every stage.

## The stages of “The Waterfall Model” are:

Feasibility Study &

Problem Definition

Requirement Analysis &

Specification

Design &

Specification

Coding &

Module Testing

Integration &

System Testing

Delivery

(Implementation)

Maintenance

Fig. Phases of Waterfall Model

* **Requirements analysis:**

This first step is also the most important, because it involves gathering information about what the customer needs and defining, in the clearest possible terms, the problem that the product is expected to solve. Analysis includes understanding the customer's business context and constraints, the functions the product must perform, the performance levels it must adhere to, and the external systems it must be compatible with. Techniques used to obtain this understanding include customer interviews, use cases, and “selecting properties” of website features. The results of the analysis are typically captured in a formal requirements specification, which serves as input to the next step.

* **Design:**

This step consists of "defining the hardware and software architecture, components, modules, interfaces, and data. To satisfy specified requirements. It involves defining the hardware and software architecture, specifying performance and security parameters, designing data storage containers and constraints, choosing the IDE and programming language, and indicating strategies to deal with issues such as exception handling, resource management and interface connectivity. This is also the stage at which user interface design is addressed, including issues relating to navigation and accessibility. The output of this stage is one or more design specifications, which are used in the next stage of implementation.

* **Implementation:**

This step consists of actually constructing the product as per the design specification(s) developed in the previous step. Typically, this step is performed by a development team consisting of programmers, interface designers and other specialists, using tools such as compilers, debuggers, interpreters and media editors. The output of this step is one or more product components, built according to a pre-defined coding standard and debugged, tested and integrated to satisfy the system architecture requirements. For projects involving a large team, version control is recommended to track changes to the code tree and revert to previous snapshots in case of problems.

* **Testing:**

In this stage, both individual components and the integrated whole are methodically verified to ensure that they are error-free and fully meet the requirements outlined in the first step. An independent quality assurance team defines "test cases" to evaluate whether the product fully or partially satisfies the requirements outlined in the first step. Three types of testing typically take place: unit testing of individual code modules; system testing of the integrated product; and acceptance testing, formally conducted by or on behalf of the customer. Defects, if found, are logged and feedback provided to the implementation team to enable correction. This is also the stage at which product documentation, such as a user manual, is prepared, reviewed and published.

* **Installation:**

This step occurs once the product has been tested and certified as fit for use, and involves preparing the system or product for installation and use at the customer site. Delivery may take place via the Internet or physical media, and the deliverable is typically tagged with a formal revision number to facilitate updates at a later date.

* **Maintenance:**

This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. Typically, every change made to the product during the maintenance cycle is recorded and a new product release (called a "maintenance release" and exhibiting an updated revision number) is performed to enable the customer to gain the benefit of the updates.

**REPORTS**

REPORTS

Testing report are created at the end of the test cycle. Sometimes test interim test report prepared at define frequency when testing is an on going activity. Its purpose is to communicate about the progress achieve by test term and any impediments by the stakeholder of the project.

Few of the test report and description below:

**1.Unit Test Report:**

Unit test report are generated at the end of the unit testing activity. As the unit testing may happen as when the unit are completed it may not be formlized many places unit tests log it may itself act as unit test.

**2.Integration Test Report:**

Integration test report are generated after the integration testing activities are completed and one can say something about the software tested when the units are bundled together test logs at act as a test report.

**3.System Test Report:**

There may be many cycles of system testing are defined by project.Plan and Activity the test plan. It includes all the special and planned iteration of testing.

**4.Acceptance Test Report:**

Acceptance test report are generally in two phases. Alpha maintenance phase report when testing is done at development and beta acceptance test report when business pilot is conducted.

**5.Various Interim Report:**

There are many interim test reports as individual steps in testing are complete. This includes unit test report and design review report.

**SYSTEM**

**REQUIReMENTS**

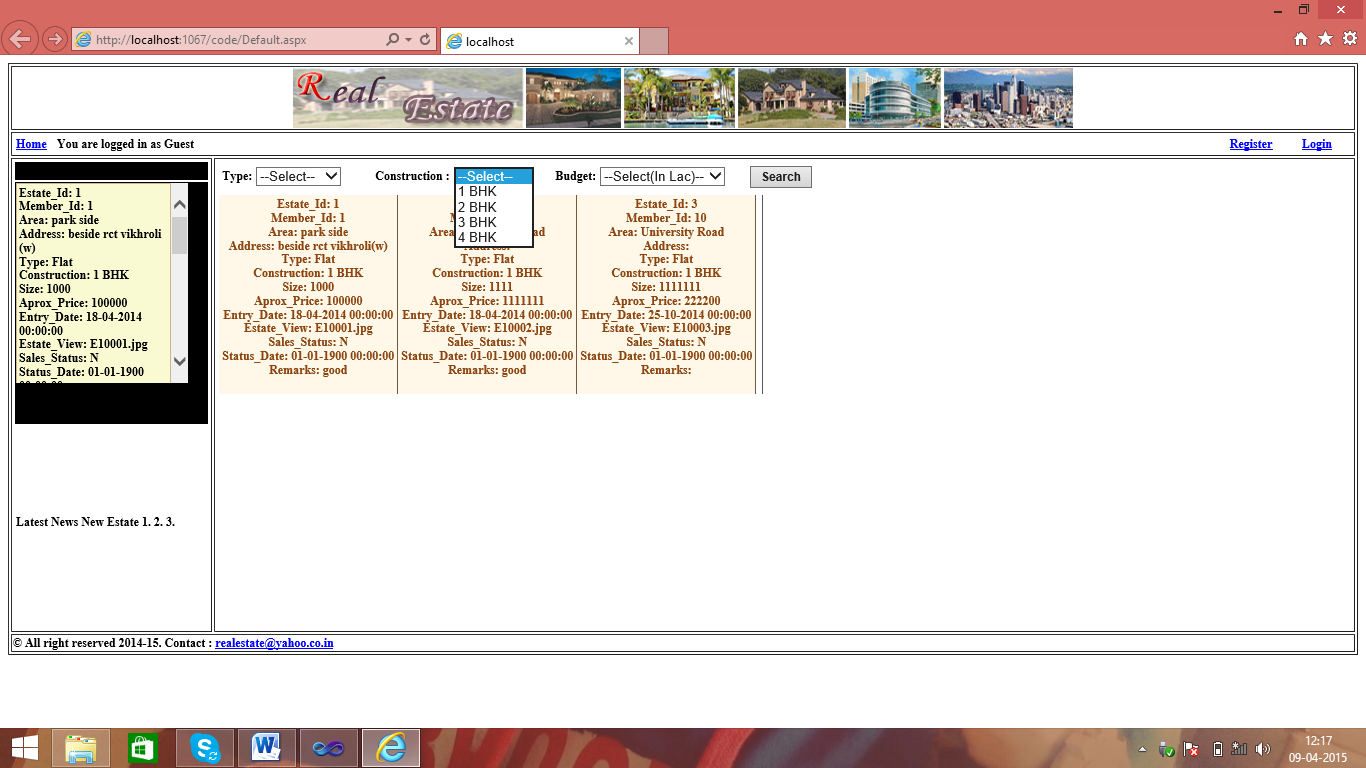
SYSTEM REQUIREMENTS

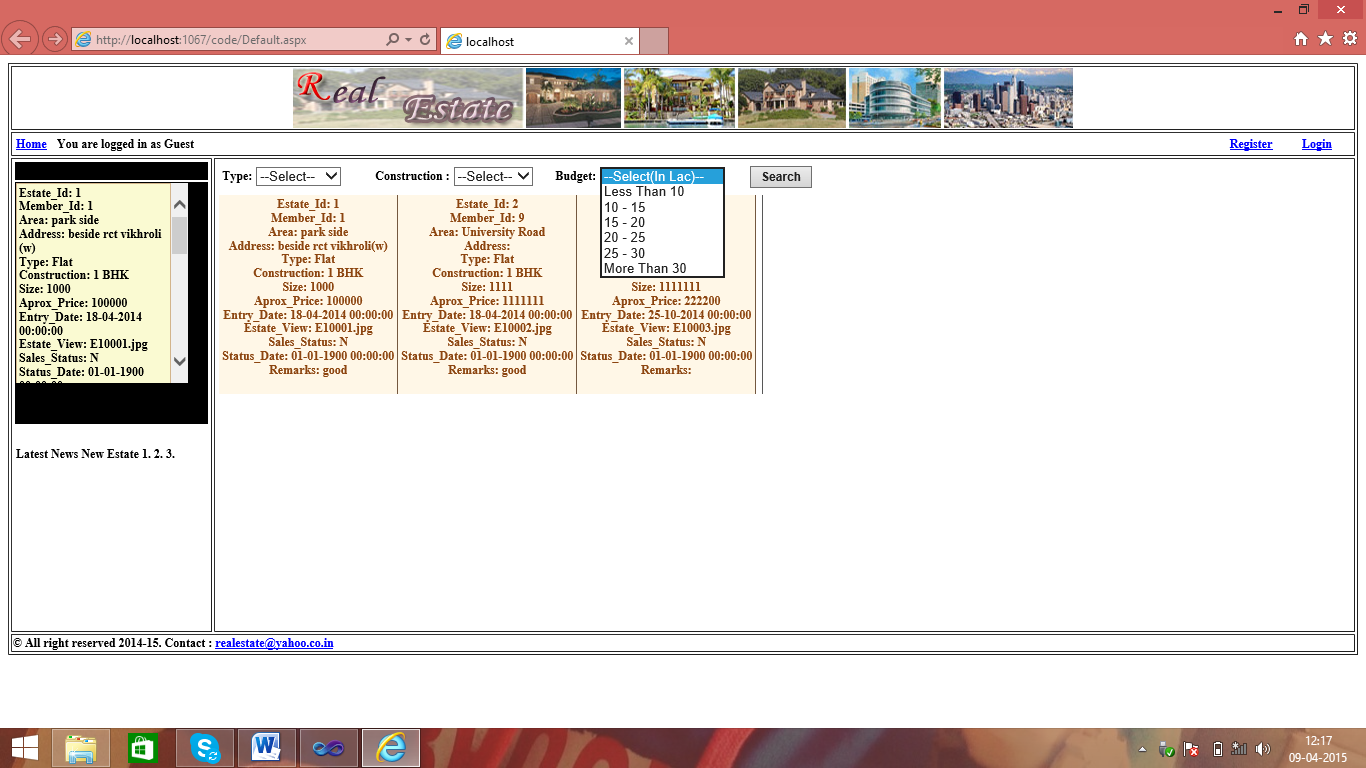
|  |  |
| --- | --- |
| **Title of the Project** | **REAL ESTATE** |
| **Software Requirements** | **Operating System Server: Windows 7**  **Database Server: Microsoft SQL Server-2008**  **Client: Microsoft Internet Explorer**  **Tools: Microsoft Visual Studio .Net-2010**  **User Interface: Asp. Net with Ajax**  **Code Behind: C#.Net** |
| **Hardware Requirements** | **Processor: Pentium IV 2GHz and Above**  **Ram: 1GB Ram**  **Hard Disk: PC with 20GB** |

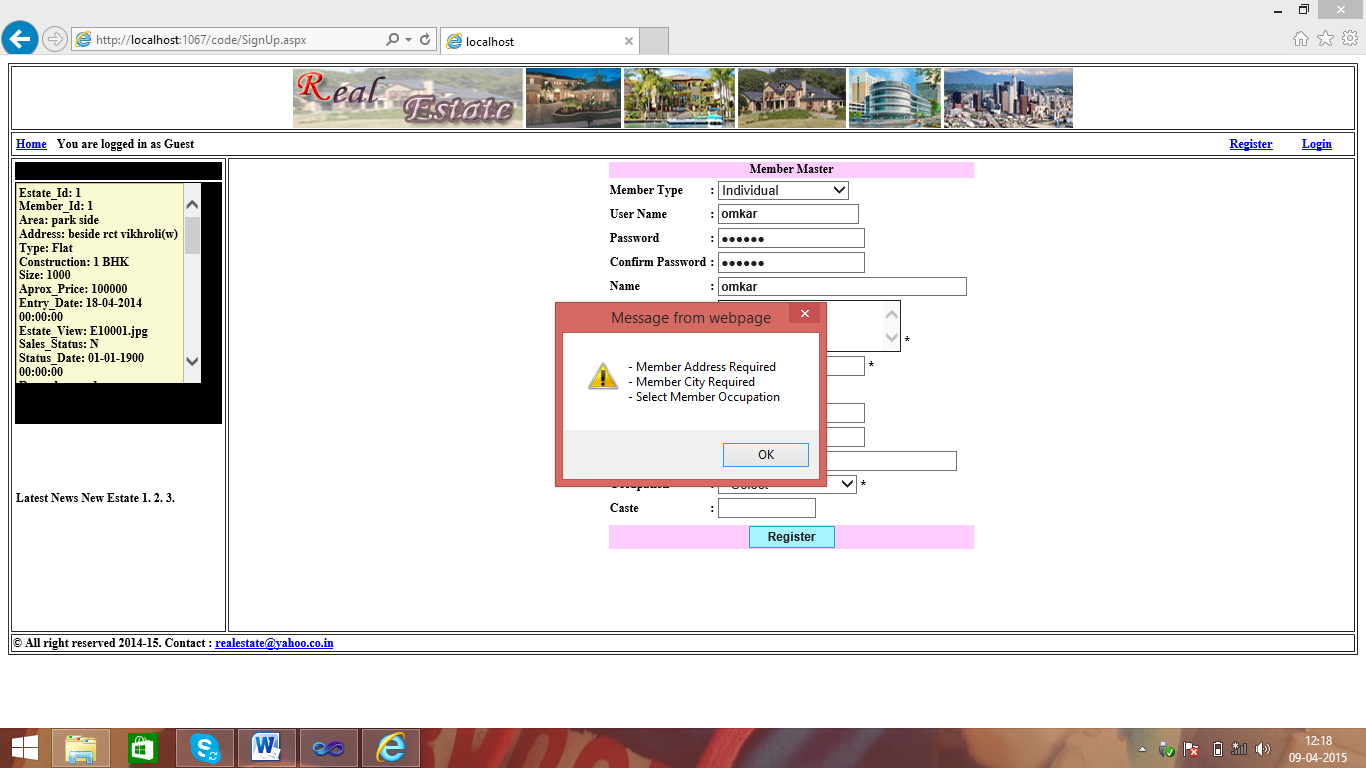
**SCREEN SHOT**

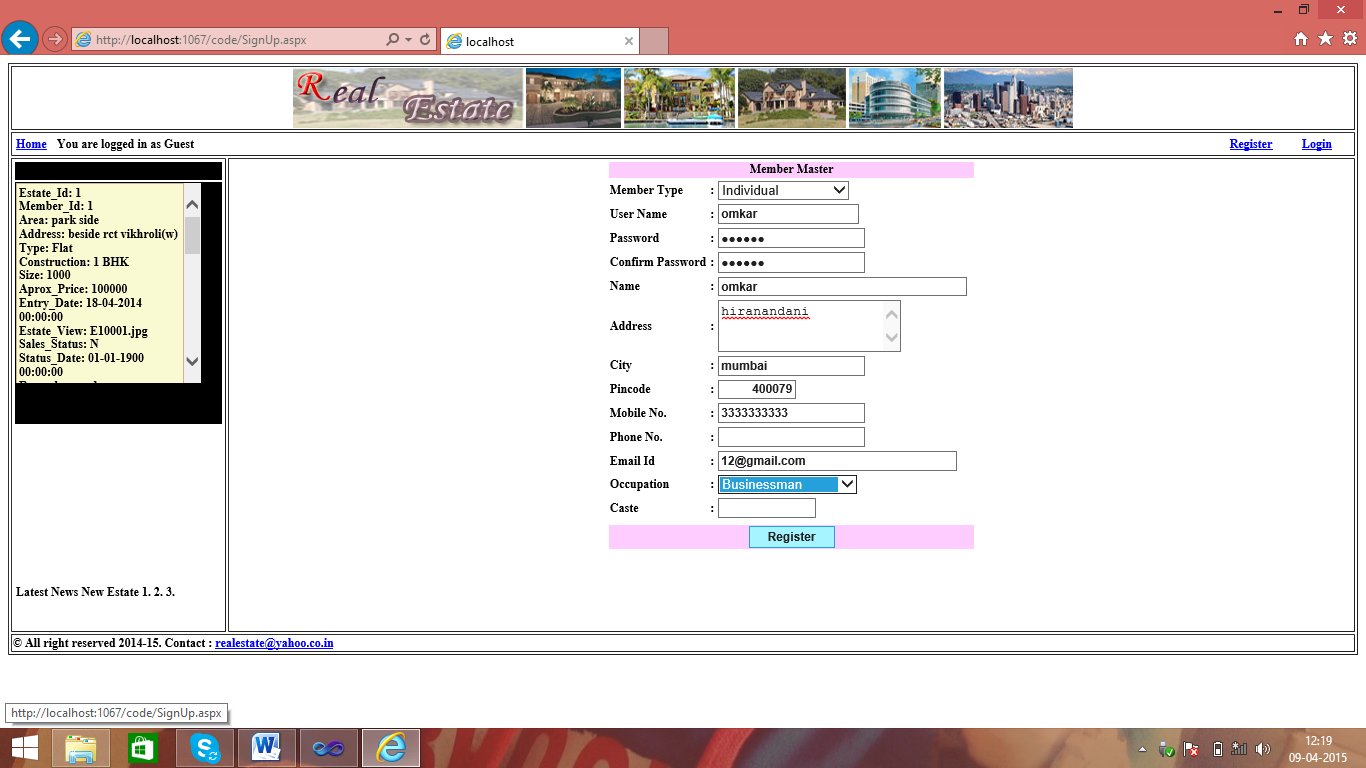
SCREEN SHOT OF THE USER

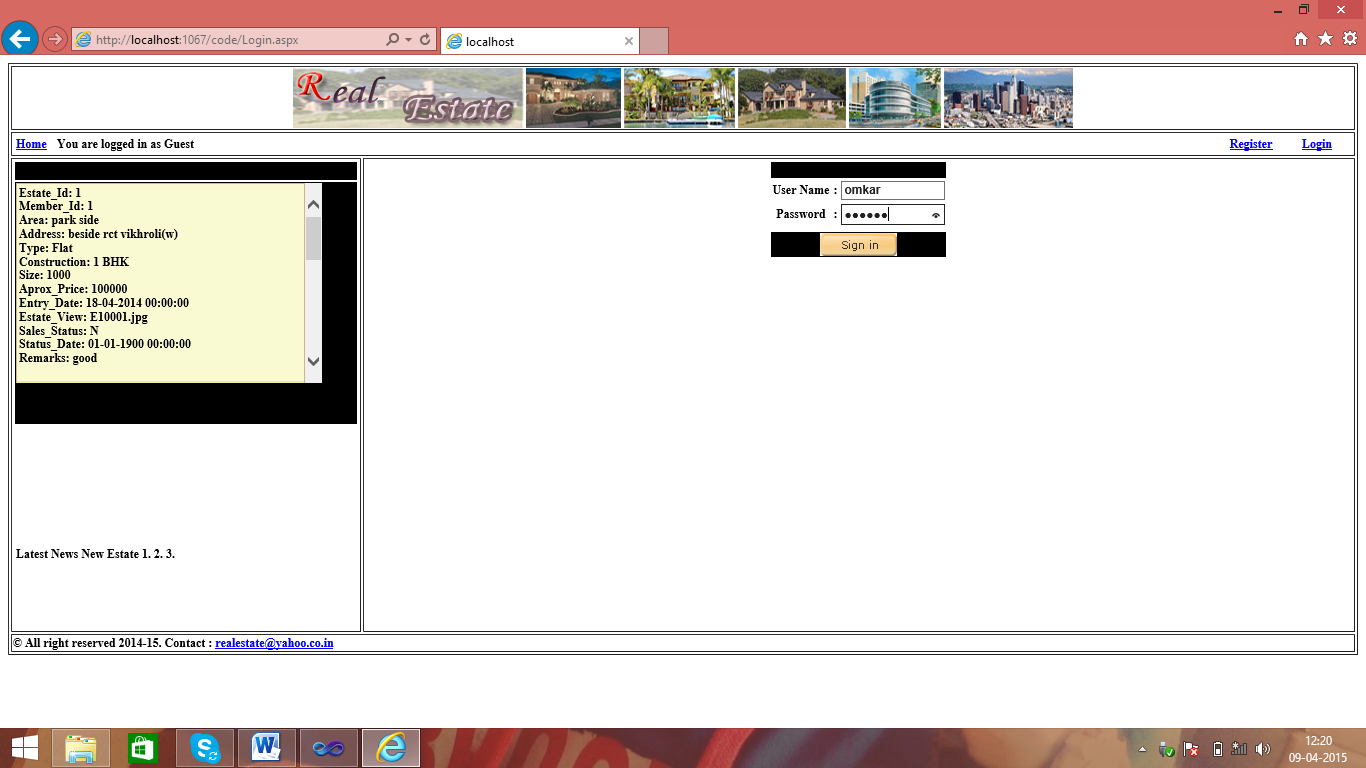


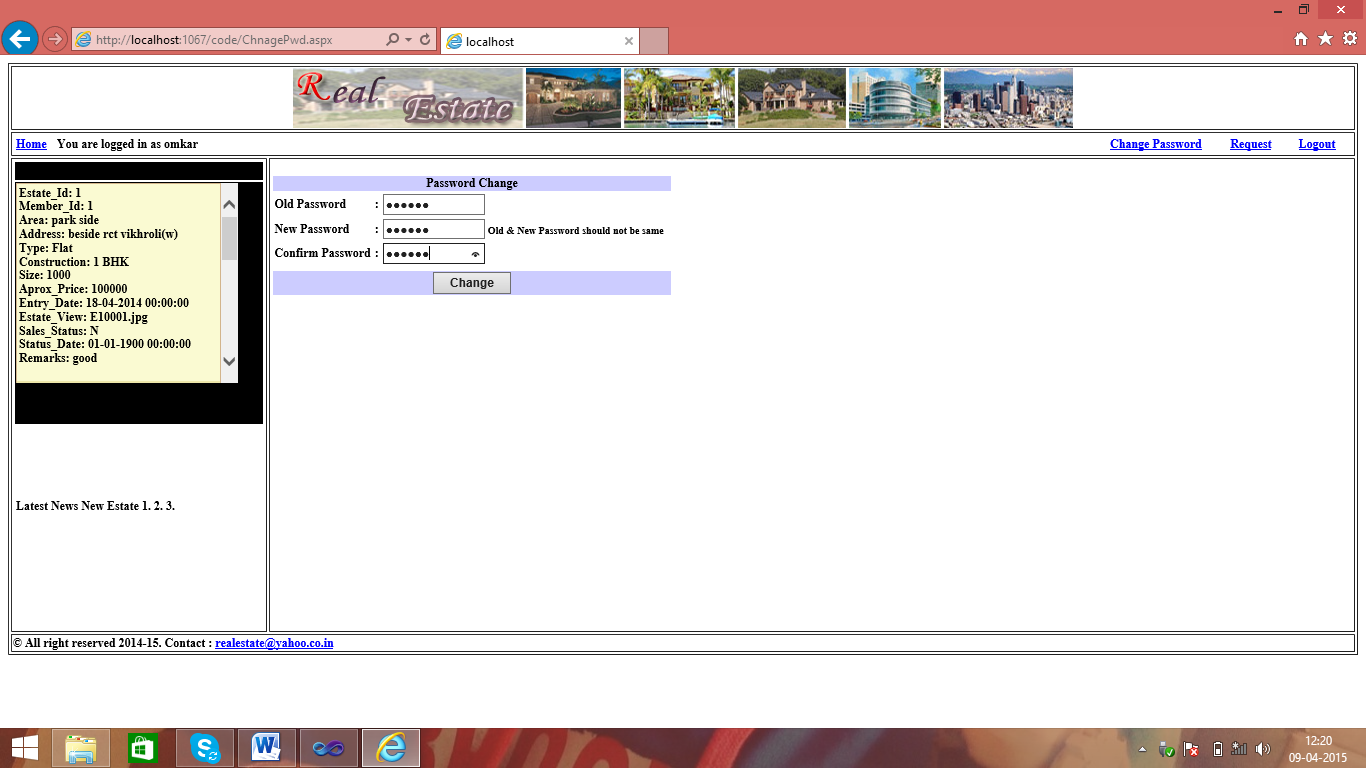


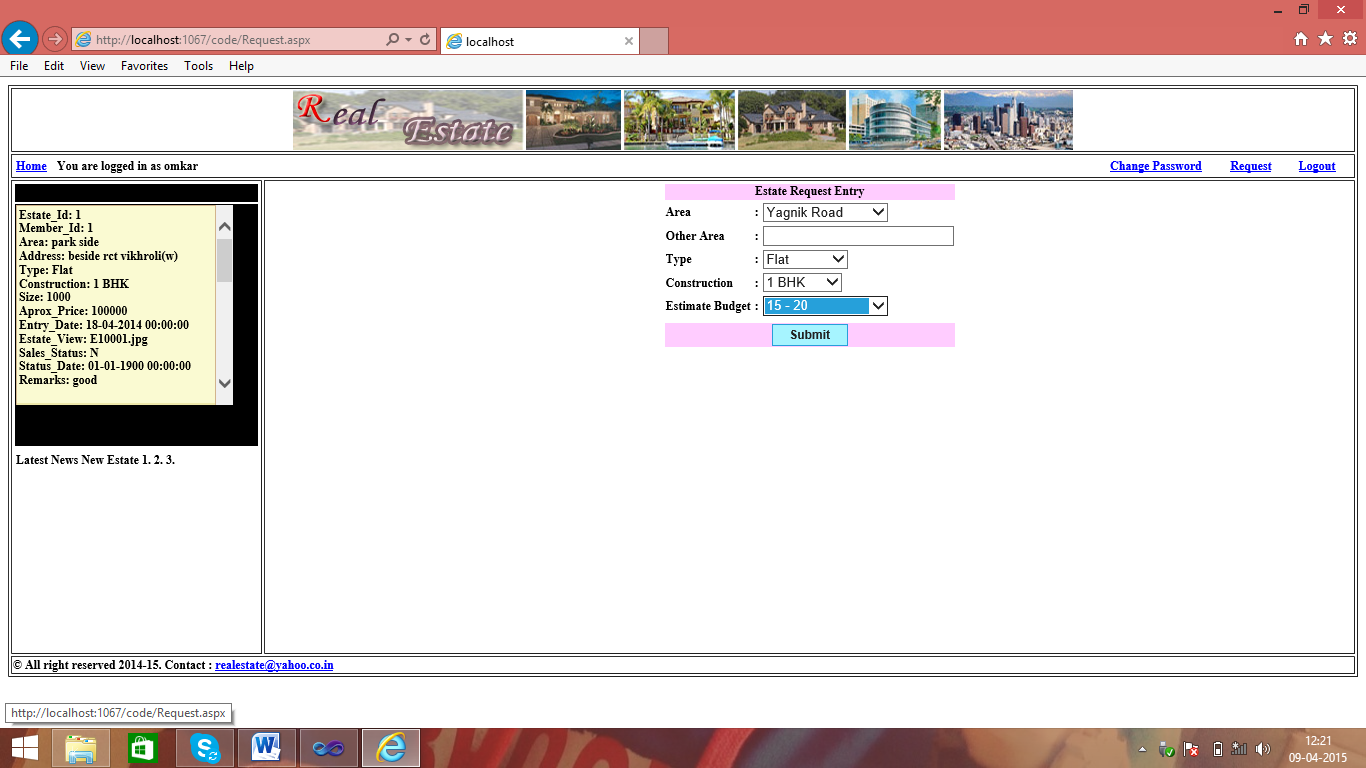


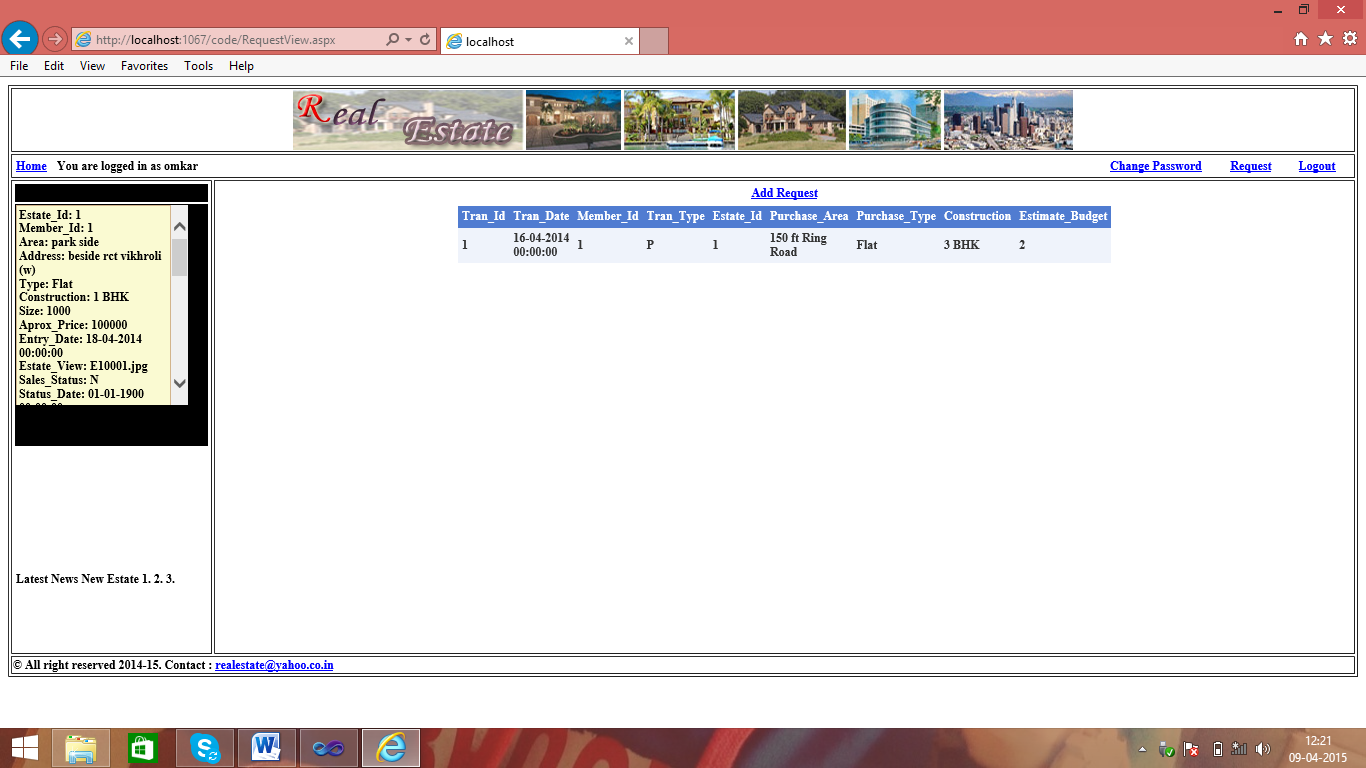












**CONCLUSION**

CONCLUSION

The project has given lot of practice in a very wide subject like Web Design. This project has given us a great opportunity to explore our skills, to think towards problem,To accept them and solved them efficiently. It made us implement everything in practice that we had learned and also other more useful parts.

The project has given a professional touch towards our learning. It has made us implements not only WEB DESIGNING but also our knowledge on different subject like ERP and CRM and others, It also increases our communication skill, made us handle different types of client. It has played a major role in our academic career. In technical term to say it has acted as an “interface” between our academic and industry. It has made us more productive, respecting a need of user, understanding the problems and making them understand the project, and it has acted as “gateway” towards the corporate world.

It made us understand different factors which are required to build fully fledged project.

On personal level it has flourished our skills made us more courage to tackle the problems, made us given the best in us. After completing this project we felt, we could be called as “True I.T professionals”.

**BIBLIOGRAPHY**

Bibliography

**Books**

* Windows programming : Charles Petzoid
* ASP. NET Unleashed : Stephen Walther
* Visual Basic .NET Black Book : Steven Holzer
* Professional ASP. NET Web services : Alex Homer, Dave Sussman
* SQL Server Essential Reference : Sharon Dooley

**Websites**

* [www.google.com](http://www.google.com)
* [www.microsoft.com](http://www.microsoft.com)
* [www.msdn.microsoft.com](http://www.msdn.microsoft.com)
* [www.asp.net](http://www.asp.net)
* [www.asptoday.com](http://www.asptoday.com)