



APRIL-MAY 2023

PROPERTY DEALING WEBSITE

A PROJECT REPORT

SUBMITTED BY-

ABHINAV SINGH (201500012)

AYUSH RAJPUT (201500182)

NAGENDRA DUBEY (201500421)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

Computer Engineering and Application

GLA University, Mathura

CERTIFICATE

This is to certify that the project entitled “**PROPERTY DEALING WEBSITE**” has been developed by BTECH students of GLA University, Mathura towards partial fulfilments of the requirements for the award of the degree of Bachelor of technology is a genius record of the work carried by Abhinav Singh, Ayush Rajput, Nagendra Dubey, under my supervision and guidance and the project report is the original work of the students. They worked hard on the project Property Dealing System. They have used CSS, HTML, JAVASCRIPT AND REACT as Database for the project. Their work is satisfactory. I wish them all the best for their bright future.

Date:

Signature of Guide:

ACKNOWLEDGEMENT

The beatitude, bliss & euphoria that accompany the successful completion of any task would be incomplete without the expression of the appreciation of simple virtues to the people who made it possible. We acknowledge all those whose guidance and encouragement has made successful in winding up this.

We owe a huge debt of thanks too many people without whom none of this would have been possible. We are thankful to **Mr. Ankit Arora (Project Guide)** for valuable suggestions and enthusiastic interest during the entire session.

Finally, we are very grateful to the Institute and all the Faculty members, without their personal attention and time to time help and care, it would not have been possible for us to complete this report.

We have this opportunity as a big milestone in our career development. We will strive to use gained skills and knowledge in the best possible way, and we will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

DECLARATION

We hereby declare that the work in the project report entitled “**Property Dealing Website**” **GLA University, Mathura** for the award of degree of “BTECH” is an authentic record of my work carried out during the sixth semester Third year, 2023 under the supervision of **Mr. Ankit Arora**. The matter embodied in this project report has not been submitted elsewhere by anybody for the award of any degree/diploma.

NAME	UNIVERSITY ROLL NO	CLASS
Abhinav Singh	201500012	BTECH 6 th SEM, 3 rd year
Ayush Rajput	201500182	BTECH 6 th SEM, 3 rd year
Nagendra Dubey	201500421	BTECH 6 th SEM, 3 rd year

PREFACE

With the fast development of computer technology, the software projects are growing and complexity. Software experts have already have recently sought to develop a more systematic and formal approach in the design, development and implementation of their software. This new approach has become necessary because the traditional methods of system development often yielded software characterized by late diversity, costliness, unreliability and non-maintainability and ability.

In this new age of computing everything has been computerized, so how can we become isolate and untouched from this environment. That's why keeping this thing in mind and as opportunity or probably a creativity to do such a task different and unique from others, we thought a way to develop this software.

This project has been developed in aim to aid and computerize a flight booking. While keeping in mind the user will find an easy and friendlier user interface to perform their task. The software has been made so user friendly that any person can use ti easily without having any computer experience.

Table of Contents

List of Figures	i
------------------------	----------

List of Tables	ii
-----------------------	-----------

Particulars	Page No.
--------------------	-----------------

1. Introduction.....	11
1.1 Motivation.....	11
1.2 Project Overview.....	11-12
1.2.1 Project Description.....	11-12
1.2.2 Problems with existing Systems.....	12-13
1.2.3 Objective	13-14
1.2.4 Architecture.....	14
1.3 Requirement Specifications.....	14
1.3.1 Scope.....	14
1.3.2 Goal.....	15
1.3.3 Assumptions.....	15
1.3.4 Environment.....	15
2. Technologies.....	16
2.1 Tools and technologies.....	16-17
2.2 Microsoft Visual Studio 2005.....	17
3. System Architecture.....	18
3.1 System Architecture.....	18-19
3.2 Architecture of Property Dealing Website.....	20

3.2.1 Presentation Tier.....	21
3.2.2 Middle Tier.....	22
3.2.2.1 Use Case Diagram.....	22
3.2.2.2 Class Diagram.....	23-26
3.2.3 Database Tier.....	27
3.2.3.1 ER Diagram.....	27
4. Testing.....	29
4.1 Unit Testing.....	29-31
4.2 Performance Testing.....	31
4.3 Testing Samples.....	32
4.3.1 Search Web pages.....	33
4.3.1.1 Test case 1.....	34
4.3.1.2 Test case 2.....	35
4.3.1.3 Test case 3.....	35
4.3.2 Details web pages.....	36
4.3.2.1 Test case 1.....	36
4.3.2.2 Test case 2.....	37-38
4.3.2.3 Test case 3.....	38-39
4.4 Test Analysis.....	39-42
4.5 Screen Shots of Tested Web Pages.....	42
4.5.1 About Us Webpage.....	43
4.5.2 Search Webpage.....	43-44
4.5.3 Details Webpage.....	45

5. Project Metrics and Experience.....	46
5.1 Project Metrics.....	46
5.2 Overall Experience.....	47-48
6. Conclusion and Future Works.....	49
6.1 Conclusions.....	49
6.2 Future Extensions to the Project.....	50
6.3 Using Object-Oriented Domain Model.....	50-51
6.4 Providing Web Services.....	51
6.5 Functionality Extensions.....	52
References.....	53-55

List of Figures

Particulars of figures	Page no.
Figure 3.1 3-Tier Architecture.....	
Figure 3.2.1 Page Flow Diagram.....	
Figure 3.2.2 Use Case Diagram.....	
Figure 3.2.3 Class Diagram.....	
Figure 3.2.3.1 Property Listing Class.....	
Figure 3.2.3.2 Listing Photo Class.....	
Figure 3.2.3.3 Listing Feature Class.....	
Figure 3.2.3.4 Listing Type Class.....	
Figure 3.2.3.5 Listing Zip code Class.....	
Figure 3.2.3.6 ER Diagram.....	
Screenshots.....	

List of Tables

Particulars of tables	Page no.
Table 4.1 List of Pages being planned for the Page Flow Diagram.....	
Table 4.2 System Configuration.....	
Table 4.3 Test Case Summary.....	
Table 4.4 Search Page Test Case 1.....	
Table 4.5 Search Page Test Case 2.....	
Table 4.6 Search Page Test Case 3.....	
Table 4.7 Details Page Test Case 1.....	
Table 4.8 Details Page Test Case 2.....	
Table 4.9 Details Test Case 3.....	
Table 4.10 Average Response Time Summary.....	
Table 4.11 System Configuration Testing for Performance Upgrade.....	
Table 5.1 Project Lines of Code.....	
Table 5.2 Project Planning Phase.....	

CHAPTER-1

INTRODUCTION

1.1 Motivation

The motivation to develop Property Dealing Website comes from my urge to learn Visual Studio .NET 2005 for building the business logic of the application, SQL server 2005 for database designing and using new web technologies like AJAX, Java Script for website designing. The most influential factor for selecting this application is to add some innovative features to the search engine of a Property Dealing Website which can make the task of a property buyer easy to search for property listings. This application provides a wide scope to include number of web controls to develop the search engine and allow them to implement using AJAX functionality. Performance of the search engine is the main factor which allows the buyers to search for listings with different combination features. Growing buyer needs should be taken in to concern with the new features to be included. Another challenge in this application is the effective storage / retrieval of property images from the SQL database. Thus, the development of powerful search engine, displaying detailed summary of listings on the customized grid and drag - drop tool to select the properties are the main motivations for the project.

1.2 Project Overview

1.2.1 Project Introduction

There are large numbers of commercial real estate online information service providers offering a suite of commercial properties and services tailored to the national and local needs of thousands of commercial the commercial investments

industry. These online marketplace have real estate properties for sale and lease under various categories including commercial office space, industrial, single - family, multi - family, land, etc both for sale and lease as well. Their purpose is to attract community of industry professionals including investors, property managers, landlords, appraisers, local and national buyers to select the properties with desired features. In this increasingly demanding scenario for a platform which could help buyers to have a look at the available property listings with all its photographs, necessary specification and description, this application is an effort to provide such a web interface with some attractive and innovative features using the latest technologies. Property Dealing Website is an AJAX enabled website with the latest AJAX controls giving attractive and interactive look to the web pages. The most significant feature in these websites is the interactive search criterion which lets the buyer specify their requirements to get the correct set of records from the database. The search tool should be strong enough to include all the required features which a buyer may desire. Also, each search requires a post back call to the database to retrieve some set of records but making the website AJAX enabled prevents the annoying post backs. Thus, each call refreshes just the data grid to display the listings rather the whole web page. Some other new features like drag and drop tool to save the listings, retrieving related results for each search, etc make this application more feature rich.

1.2.2 Problems with existing systems

The purpose of creating this Property Dealing Website is to outcast the discrepancies in hundreds of such existing systems on the World Wide Web. One of the basic problems with the existing systems is the non-interactive environment they provide to the users. Most of the applications involved in Real Estate business use some web template to put the content specific to their company and make it

communicate with the database to search the listings. These templates simply use basic web controls to do this task making the web page non-interactive. On the other hand, the motive of this Property Dealing Website is to allow the user to play with the search tool and create different combinatorial search criterion to perform exhaustive search. Another problem in such applications designed so far is the use of traditional user interfaces which make continuous post backs to the server; each post back makes a call to the server, gets the response and then refreshes the entire web form to display the result. This scenario adds an extra trade off causing a delay in displaying the results. Making such applications AJAX enabled gets rid of these unnecessary delays letting the user to perform exhaustive search. The users of this application can easily feel the difference between the Ajax-empowered user interfaces vs. traditional user interfaces.

1.2.3 Objectives

The foremost objective of this project was to give a different visualization styles to the Property Dealing Website which has more features, attractive animations and all together a new look in contrast to the already existing websites. Usually in a real estate website, the property search page consist of traditional search style i.e. a set of textboxes / drop-downs to select a particular county or MLS# for the property to search along with other web controls to specify the number of beds / baths or any additional features they are looking in the property. In contrast, this website provides altogether a new visualization of the search page i.e. AJAX enabled controls with no post backs and additional functionalities like sliding bars to select the number of beds, baths, footage and price, drag and drop functionalities, animation enabled collapsible styles, etc. The next objective of importance was to build an AJAX enabled real estate website which not only

reduces the annoying post backs and loss of control focus, but also gives a faster and more interactive user interface. Moreover to make the website more features rich, features like customized grid, drag and drop tool, accordion panels and sliding bars were added to the website. The third objective was to include a new feature of saving the selected property listing during a session in a drag – drop tool which similar websites don't have. Usually a real estate website forces a user to login as a buyer to save the selected listings so that when the user logs in the next time, those saved listings are accessible. But this site simply provides a toolbox to drag a selected number of listings and save them during the entire session.

1.2.4 Architecture

The Property Dealing Website is built using a layered architecture where the total functionality can be divided into layers having different functionalities. The main layers include the database access layer, business logic layer and the presentation layer. Thus, this application follows the 3-Tier Architecture. The System Architecture is explained in detail in the further sections of this document.

1.3 Software Requirement

1.3.1 Scope

The scope of MS project “Property Dealing Website” is to enable the buyers to search for property listings online. The motive of developing this application is to design a feature rich search engine which can make the search of commercial land/properties an easy task.

1.3.2 Goal

Goal is to gain a good knowledge of the complete life cycle of the project development starting from the Requirement Gathering Phase to the Testing Phase. The implementation of this application also gives a hands-on experience in ASP.NET, C# .NET programming and also design efficient databases in SQL Server.

1.3.3 Assumptions

- User will have Internet Connection while using Property Dealing Website.
- User will have Internet Explorer 5.0 or high IE version while using Property Dealing Website.
- User will upload valid images for the Real Estate Web Application.

1.3.4 Environment

- The Real Estate Web Application will be written in C# .NET language.
- The development environment will be Microsoft Visual Studio .NET.
- The Real Estate Web Application will be tested on Windows XP platform.

CHAPTER-2

TECHNOLOGIES

This chapter includes the details of the latest technologies and tools used to build this application, their benefits and implementation details. The chapter also describes the interaction of these tools with the .NET Framework.

2.1 Tools and Technologies

The latest tools and technologies involved in building this website are: ASP.NET 2.0, Microsoft Visual Studio 2005, and Java Script.

2.1.1. ASP.NET 2.0

ASP.NET 2.0 is the next generation of Microsoft's Active Server Page [ASP] technology which provides a web application framework that allows programmers to dynamically build web pages, web applications and create web services. ASP.NET supports code written in compiled languages such as Visual Basic, C++, C# and Perl, and it features server from the content. The basis for ASP.NET 2.0 is built on the Common Language Runtime [CLR] described above. The dynamic content supported by ASP.NET is contained in the ASPX web forms which contain the static XHTML markup, as well as server-side Web Controls and User Controls where the developers place all the required static and dynamic content for the web page. The major aspects of this technology include ASP.NET Compiler which includes strong typing, performance optimization and early binding. The task of

this compiler is to compile all application components including pages and controls into an assembly which can then be used to service user requests. The Page and Controls Framework runs on the web server which then dynamically produces and render ASP.NET web pages which are completely object-oriented. Other services of ASP.NET include Security infrastructure, State-management facilities, Application configuration, Debugging support and XML Web Services framework. Some of the new features of ASP.NET 2.0 are the introduction of Master Pages and Themes which allow the developer to define a common structure and interface elements, to be shared by many pages in the website. Standard controls for navigation, security, support for XML standards and interoperability with AJAX make web development easier and quicker.

2.2 Microsoft Visual Studio 2005

Microsoft Visual Studio provides an Integrated Development Environment (IDE) for programming that enables developers to quickly create data-driven and distributed applications using C# techniques and reusable controls. It can be used to develop both console and Graphical User Interface applications. The GUI applications include Windows Form applications, web applications and web services. Visual Studio comprises of a Code Editor, Debugger, Designer and other tools making the environment very interactive and easy to use. VS 2005 was upgraded to support all new features introduced in .NET Framework 2.0 as described above. The new features like intelligence and code refactoring supported by code editor, support for SQL Server 2005 databases, an improved environment for web publishing and load testing to see application performance under various sorts of user loads [2].

CHAPTER - 3

SYSTEM ARCHITECTURE

This chapter provides an Architectural Design for the Property Dealing Website which represents a three tier architecture comprising of the Presentation Tier: list of web pages planned to be implemented in the application, the Middle Tier: class diagrams and description of each class and the Data Tier: comprising of the ER diagram and database schema.

3.1 System Architecture

The architecture of the system is based on the three-tier architecture. There are three logical layers: the Presentation Tier, the Business Tier, and the Data Tier. Each layer in the system is a reusable portion of code that performs a specific function. These layers are not only responsible for performing these functions but also interact with other layers to perform specific goals [4]. Like, in this application when the presentation layer needs to extract information from the backend database, it would utilize a series of layers to retrieve the data, rather than having the database calls embedded directly within it. The ASP.NET web site/web form is called the presentation layer because the content within it is viewable to the users of the site. The middle tier called the Business Tier communicates between the presentation tier and the data tier. As mentioned above, the Presentation Layer could communicate with the data access layer directly, it usually goes through the business layer. This layer then validates the input conditions before calling a

method from the data layer. This ensures the data input is correct before proceeding, and can often ensure that the outputs are correct as well. Most of the business logic of the application lies within the business layer, which makes this logic reusable across applications. This layer helps move logic to a central layer for “maximum reusability.” The Data Tier is the key component for this application as it is responsible for storing the data corresponding to each property listing.. Through this approach, data can be logically reused, meaning that a portion of an application reusing the same query can make a call to one data layer method, instead of embedding the query multiple times.. The database retrieves the data corresponding to the query and submits the results back to the presentation tier for display. The Figure 3.1 shows the three-tier architecture of the Property Dealing Website.

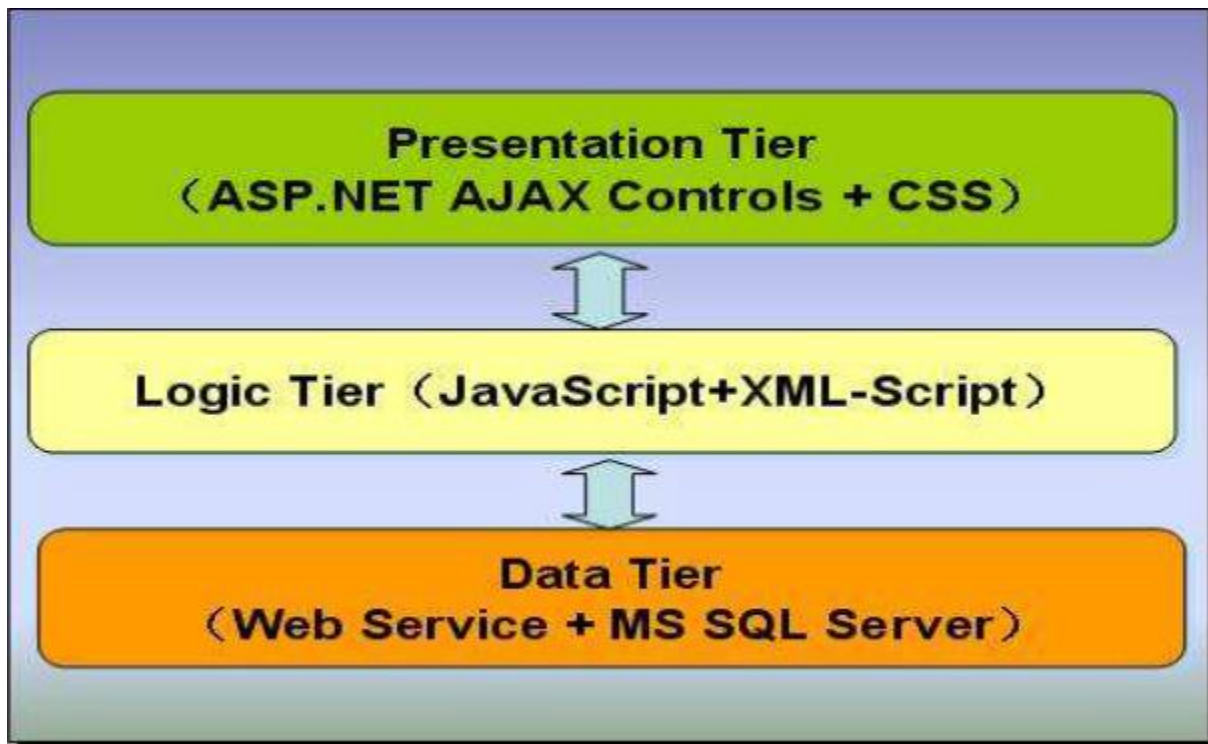


Figure 3.1 3-Tier Architecture

3.2 Architecture of Property Dealing Website

The Property Dealing Website is based on the three tier architecture namely the presentation tier, the middle tier, and the data tier.

3.2.1 Presentation Tier

The Presentation Tier of this application consists of ASP.NET web forms which contain ASP web controls, user controls and AJAX controls. The web pages are designed to make the website look attractive and interactive. The objective of this layer was to take the full advantage of the latest AJAX animated controls like Collapsible panels, sliding bars, watermark textboxes, etc. Visual Studio .NET 2005 version is being used to design and code the ASP.NET pages. ASP.NET 2.0 has an extra support of master pages and themes making the site pages look uniform and have common structure throughout. The banners and footers are included in the master page; the Tab Container loads the rest of the web pages. The content is added by the user in the search engine using the AJAX controls which causes the data grid present in the Update Panel being refreshed. The application constitutes some of the basic pages needed by a Property Dealing Website. This layer is responsible for the results/output from the business layer and transforms the results into a readable format for the end user. The following Page Flow Diagram presents the navigation flow in the application:

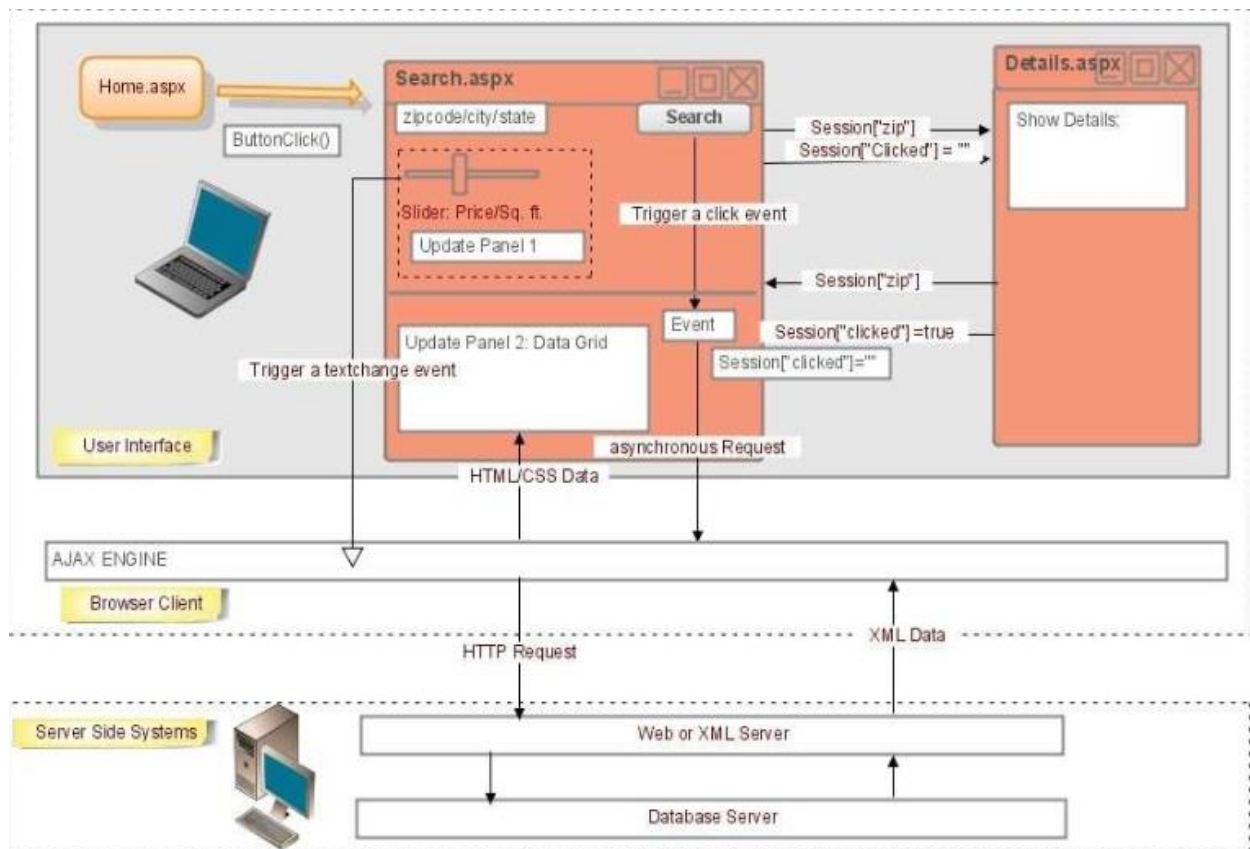


Figure 3.2.1 Page Flow Diagram

Table 4.1 represents the list of ASP.NET Pages implemented in the Property Dealing Website and the purpose of each page.

ASP.NET Pages	Purpose
HomePage.aspx	Information about the Real Estate Company.
Search.aspx	Provides a search engine to the buyer, results are displayed on the data grid.
UploadListing.aspx	Uploads the photographs of the property.
Details.aspx	Displays the property details.

Table 4.1 List of pages being planned for the Page Flow Diagram

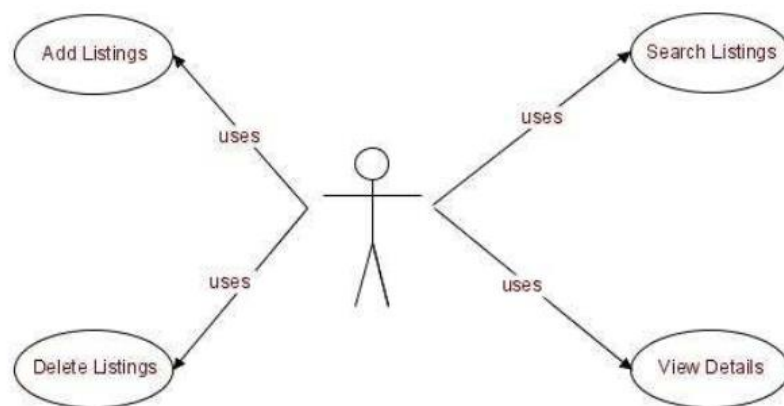
3.2.2 Middle Tier

The Middle Tier called as the Business Logic Layer allows the users to share and control the business logic by isolating it from the other layers of the application. This tier increases the code transparency and supports changes in the data layer and is also responsible for altering the database. This tier contains classes for Property Listings, Property Photo, Property Features, Property Type and Zip codes and accesses the databases. A better understanding of the middle tier can be obtained with the help of the Use Case Diagram and Class Diagram.

3.2.2.1 Use Case Diagram

The buyer can search the property listings, view property details, enter the selected listings in the buyer cart and can delete the listings from the buyer cart. **Figure**

3.2.2.1 Use Case Diagram



3.2.2.2 Class Diagram

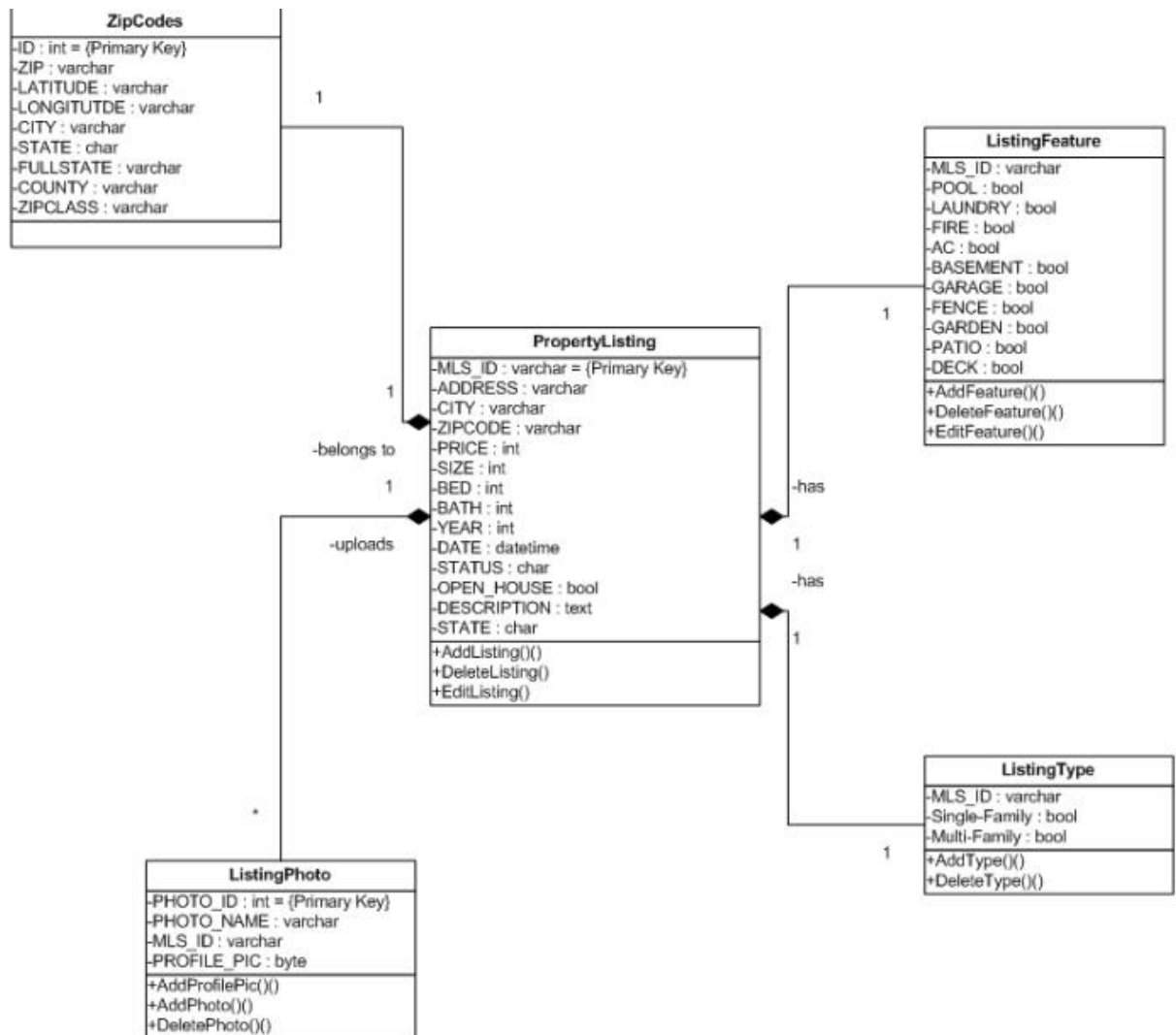


Figure 3.2.2.2 Class Diagram

The description for each of the classes in the Class Diagram is given below:

Property Listing Class: The Property Listing class is the basic class for this application having attributes and description for each property listing which help the buyer to select the property desired for his need. This class includes the basic

attributes to describe the listing like MLS#, listing_name, listing_address, listing_description, etc. The methods included in this class are: Adding a new property listing to the database, Editing the existing property listing. Uploading photographs for each property listing ,Deleting the photographs for each property listing.

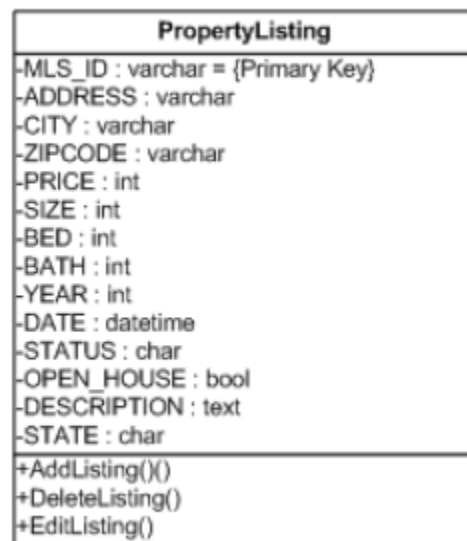


Figure 3.2.3.1 Property Listing Class

ListingPhoto Class:

The ListingPhoto class associates itself with the PropertyListing. It is used to define the attributes of the photo that belongs to each Property Listing. The Real Estate Company can upload any number of photos but can have only one picture that acts as its profile picture. The rest of the pictures corresponding to the listing can be viewed in the photo album. The methods corresponding to this class are AddPhoto and DeletePhoto. The Class Diagram for ListingPhoto is shown below:

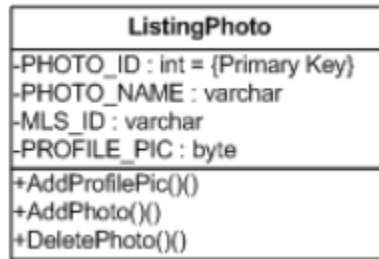


Figure 3.2.3.2 Listing Photo Class

ListingFeatureClass:

The ListingFeature Class corresponds to the super class PropertyListing. This class contains the attributes which describe the extra features of the properties present in the PropertyListing Class. The methods AddFeatures and DeleteFeatures add information corresponding to the property present in the PropertyListing Class. Thus, all the attributes combined from the PropertyListing and PropertyPhoto Classes completely describe the property which a buyer is looking for. The Class Diagram for ListingFeature is shown below:

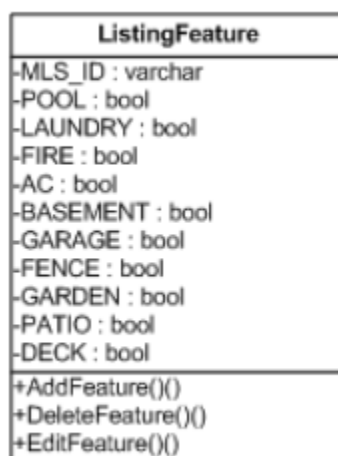


Figure 3.2.3.3 Listing Feature Class

PropertyTypeClass:

PropertyType also corresponds to the super class PropertyListing. This class contains the attributes which describe the type of the property the buyer is looking for. The type of the property can be single-family or multi-family. The operations on the class named AddType and DeleteType determine the type of the property listing. Class Diagram for PropertyType is shown below:

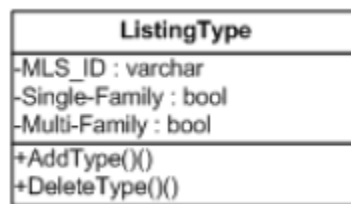


Figure 3.2.3.4 Listing Type Class

ZipCodesClass:

The ZipCode class supports the PropertyListing class. One of the attribute of the PropertyListing class is zip code which identifies the place where the property is located. ZipCode class contains all the US zip codes and supports the PropertyListing class by locating all the zip codes in the vicinity of a particular zip code. Class Diagram for ZipCode is shown below:

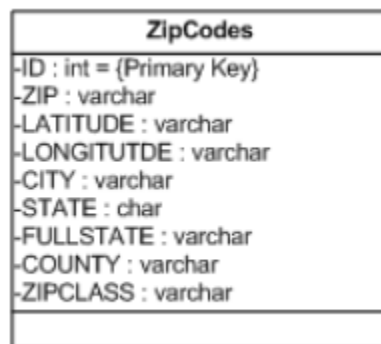


Figure 3.2.3.5 Zip Code Class

4.2.3 Database Tier

The Property Dealing Website is supported by the SQL Server 2005 and its database. SQL Server provides a good response time of the data being stored making the search effective, convenient way for storing the photographs of the properties and storing the entire description and features of the Property Listings. The database schema for this application consists of five tables out of which the Listing is the main table to store the primary details of the property. Listing_type and Listing_feature reference the MLS# in the Listing table to describe more features of the listings. Listing_photo stores the photographs of the properties in the form of binary data. The MLS_ID column name is the attribute in the Listing table which is referenced by all the other tables.

4.2.3.1 ER Diagram

The Entity-Relationship model of the database being displayed is shown below:

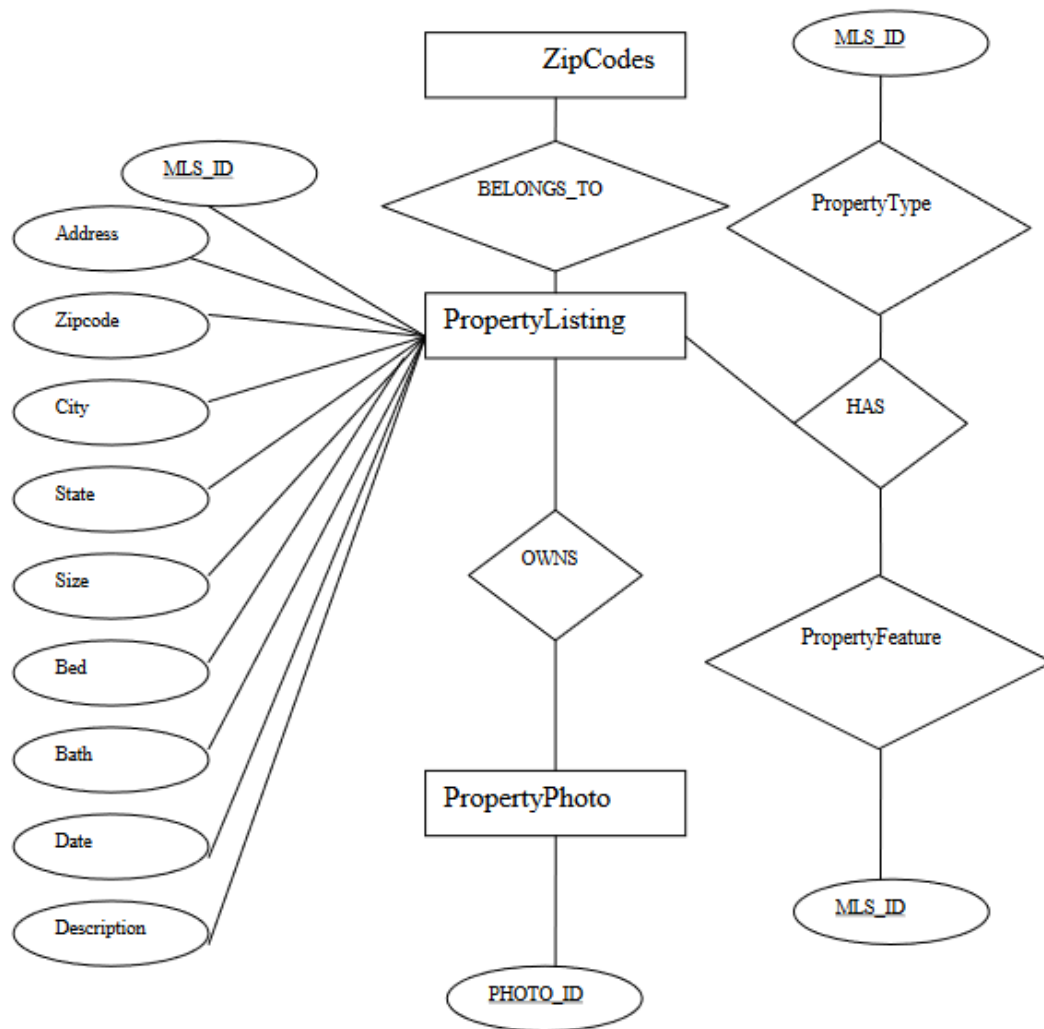


Figure 3.2.3.6 ER Diagram

CHAPTER 4

TESTING

This chapter discusses the various tests performed on the web-application with a set of test scenarios. Both unit and load testing is performed on this application. Testing is done on a local machine with the following system configuration.

Processor	Intel Pentium®
Processor Speed	2.20GHz
Physical Memory	3.00 GB of RAM
Operating System	Windows XP Professional

Table 4.1: System Configuration

4.1 Unit Testing

Unit test is performed to test and validate the individual units of source code. It is the code wrapper around the application code that permits test tools to execute them for fail-pass conditions. Unit testing is important as it gives the code authors and reviewers' confidence that patches produce the correct results. The test cases are a good impetus for developers to discover edge cases. The Property Dealing Website uses NUnit Framework to perform Unit Testing. NUnit Framework is port of JUnit framework from java and Extreme Programming (XP) and is an open source product. This framework is developed to make use of .NET framework functionalities and uses an Attribute based programming model. The nunit.exe program is a graphical runner that shows the test cases in an explorer-like browser

window and provides visual indication of success or failure of the tests. Following are the test cases executed against the web-application using NUnit:

Bed_BathTest: This test case validates the functional unit in the application that is responsible for querying the database to retrieve the number of beds and baths from the database corresponding to the number entered by the user in the search criterion. This functional test checks for the valid beds and baths combination that can let the user retrieve the results from the database.

MLS_IDTest: This test case tests the functional unit that is responsible for fetching the details of the property listing such as city, state, zipcode, beds, baths, etc and displays it to the user in the data grid. This function ensures whether the search entered in the search criterion.

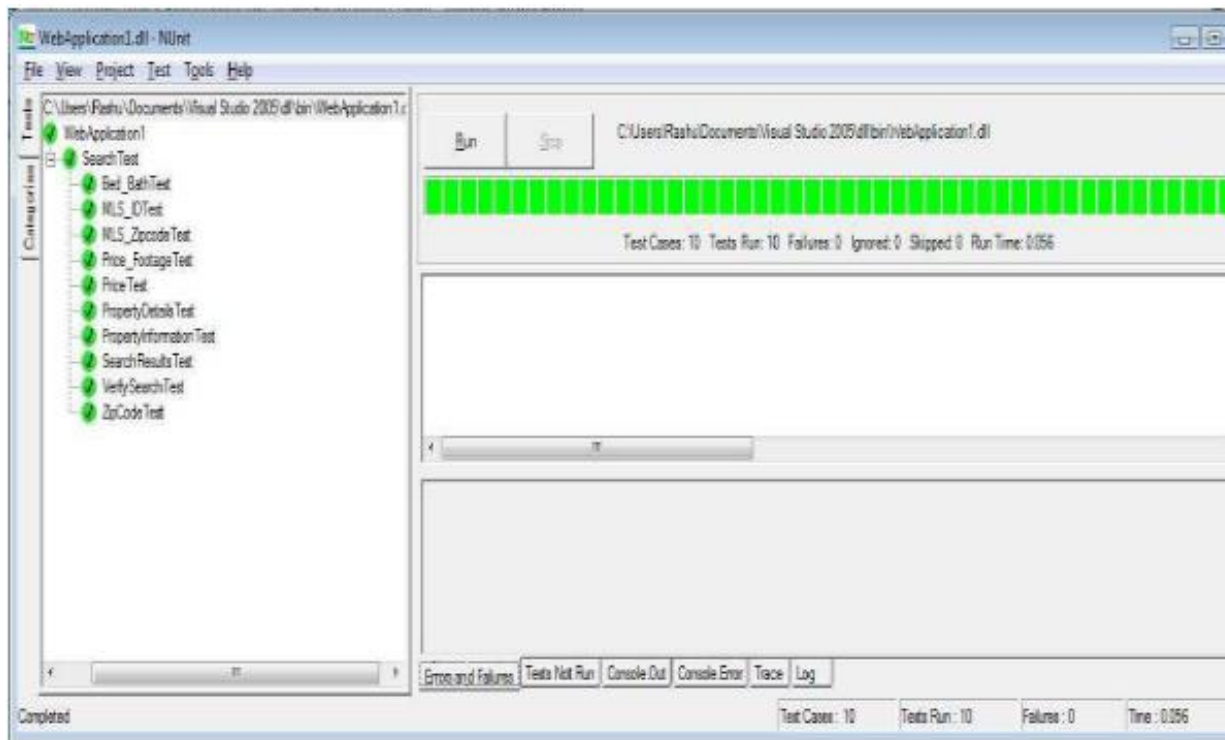
MLS_ZipCodeTest: This test case validates the functional unit in the application that is responsible for querying the database to retrieve the property listing details from the database based on the combinatorial search for the MLS_ID and ZipCode. This functional test checks for the valid MLS_ID and ZipCode combination that can let the user retrieve the results from the database. The result of this combinatorial search results in one row corresponding to the unique MLS_ID entry in the database.

PriceFootageTest: This test case validates the functional unit in the application that is responsible for querying the database to retrieve the property listing details where the price is greater than the price entered by the user and the footage is greater than the square feet miles entered by the user. This query may result in many rows satisfying the search criterion.

PriceTest: This test case tests the functional unit that is responsible for fetching the details of the property listing such as city, state, zipcode, beds, baths, etc and displays it to the user in the data grid where price is greater than the price entered by the user. This function ensures whether the search entered in the search criterion is correct.

PropertyDetailsTest, PropertyInformationTest, SearchResultsTest, VerifySearchTest and ZipCodeTest are other functional unit tests that validate the correctness of the results

corresponding to different combinatorial search criteria. The screenshot of NUnit test cases execution is shown in the Figure 4.1.



4.2 Performance Testing

Performance of a web-application can be tested done by applying the Load and Stress Testing on it. Load testing is subjecting the application to a statistically representative load. This testing is done in support of the application reliability testing and in performance testing. In performance testing, the load is varied from minimum (zero) to the maximum level the application can handle without application-specific excessive delay. In this testing the realistic workloads are characterized, simulated and submitted to the system under test. Load Testing verifies the acceptability of the target-of-test's performance behavior under varying operational conditions (such as number of users/threads, number of loop counts,

etc.) while the configuration remains constant. While the Stress Testing verifies the acceptability of the target-of-test's performance behavior when abnormal or extreme conditions are encountered, such as diminished resources or extremely high number of users. Extreme loads are used in load stress testing – to find the breaking point and bottlenecks of tested system. In load stress testing we have tried breaking the application by an extreme load and expose bugs that are likely to appear under stress, such as data corruption, buffer overflows, large number of users, etc. Load analysis needs to proceed by having a special purpose browser act like a human user. This assures that the performance checking experiment indicates true performance – not performance on simulated but unrealistic conditions. The testing tools try to perform Load Simulation to re-create a realistic scenario and so the results obtained assures that they can be used to analyze the performance characteristics of the web site under high load conditions. The Property Dealing Website uses the Jakarta JMeter as the Website load testing tool as it can simulate thousands of users, came from the same quantity of IPs and analyze, how many concurrent users can handle this web site. Apache JMeter is a 100% pure Java Desktop application designed to load test functional behavior and measure performance of a web application. This tool is strong enough to test perform both on static and dynamic resources such as files, Servlets, Java Objects, Data Bases and Queries, FTP Servers and more. JMeter is used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. It can be used to obtain a graphical analysis of performance behavior under heavy concurrent load [6]. Using the JMeter, I have performed several tests taking different readings and found out the peak number of users that can run effectively with this tool. The hardware capabilities of my machine limit the number of users to 1000. The testing results are shown in a tabular form containing the testing figures for various web pages.

The testing is done keeping the loop count constant i.e. 16000 and Ramp-up period as 5 seconds and varying the number of users.

Users	Ramp-Up Period (sec)	Loop Count
100	5	16000
500	5	16000
1000	5	16000

Figure : Test Case Diagram

Testing is performed on the following pages:

- HTML Page (Home Page)
- Database Intensive Page (Search Page)
- Business Logic Page (Details Page)

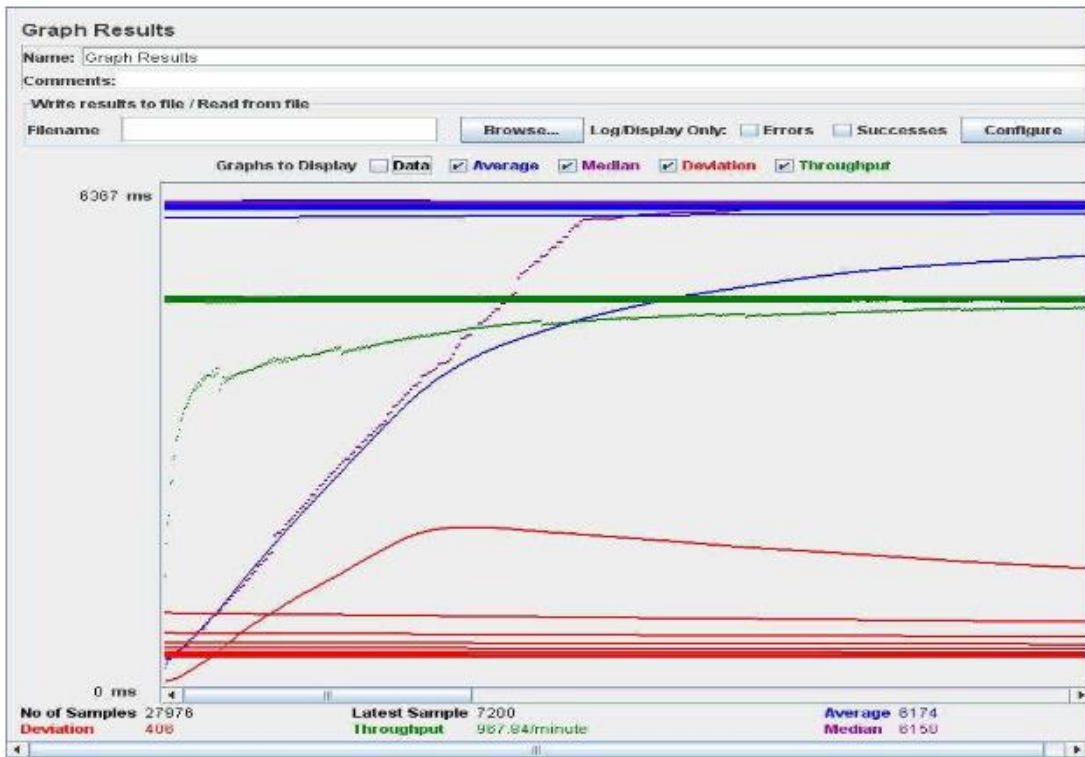
4.3 Performance testing

4.3.1 Search web page

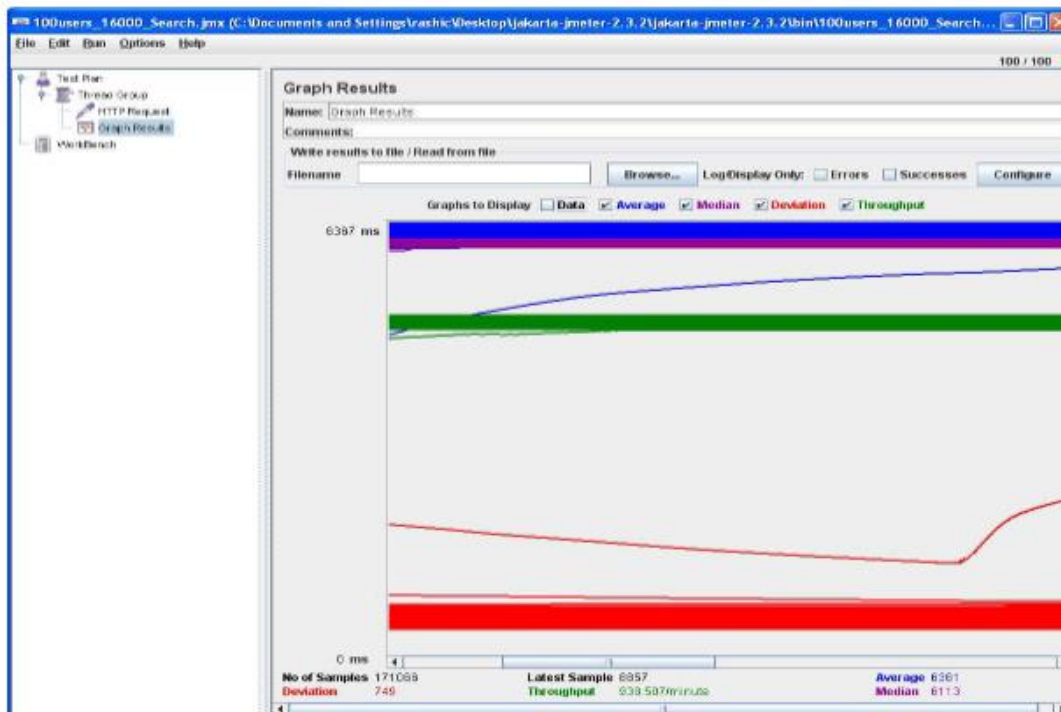
4.3.1.1 Test Case 1

Users	Loop Count	Ramp-up period (sec)	Average Response Time (ms)	Throughput
100	16000	5	6174	967.84/min

Table 4.4 Test Case 1



Search Web Page Test Case 1.1

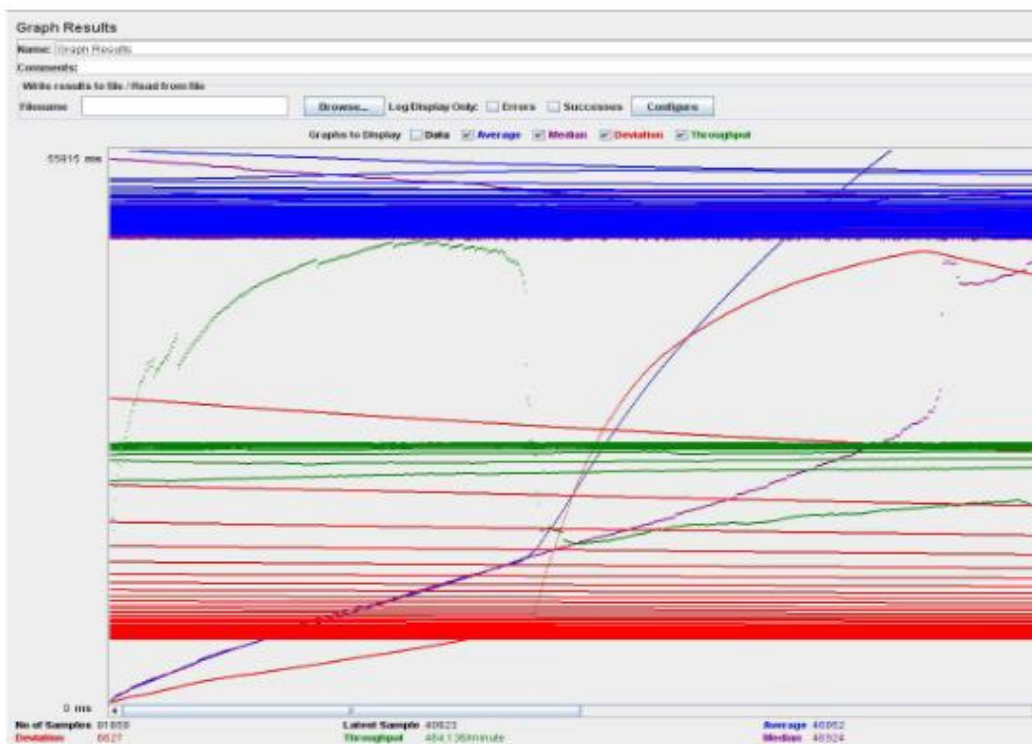


Search Web Page Test Case 1.2

4.2.1.2 Test Case 2

Users	Loop Count	Ramp-up period (sec)	Average Response Time (ms)	Throughput
500	16000	5	55915	967.84/min

Table 4.5 Test Case 2



4.2.1.3 Test Case 3

Users	Loop Count	Ramp-up period (sec)	Average Response Time (ms)	Throughput
1000	16000	5	33243	364.157/min

Table 4.6 Test Case 3

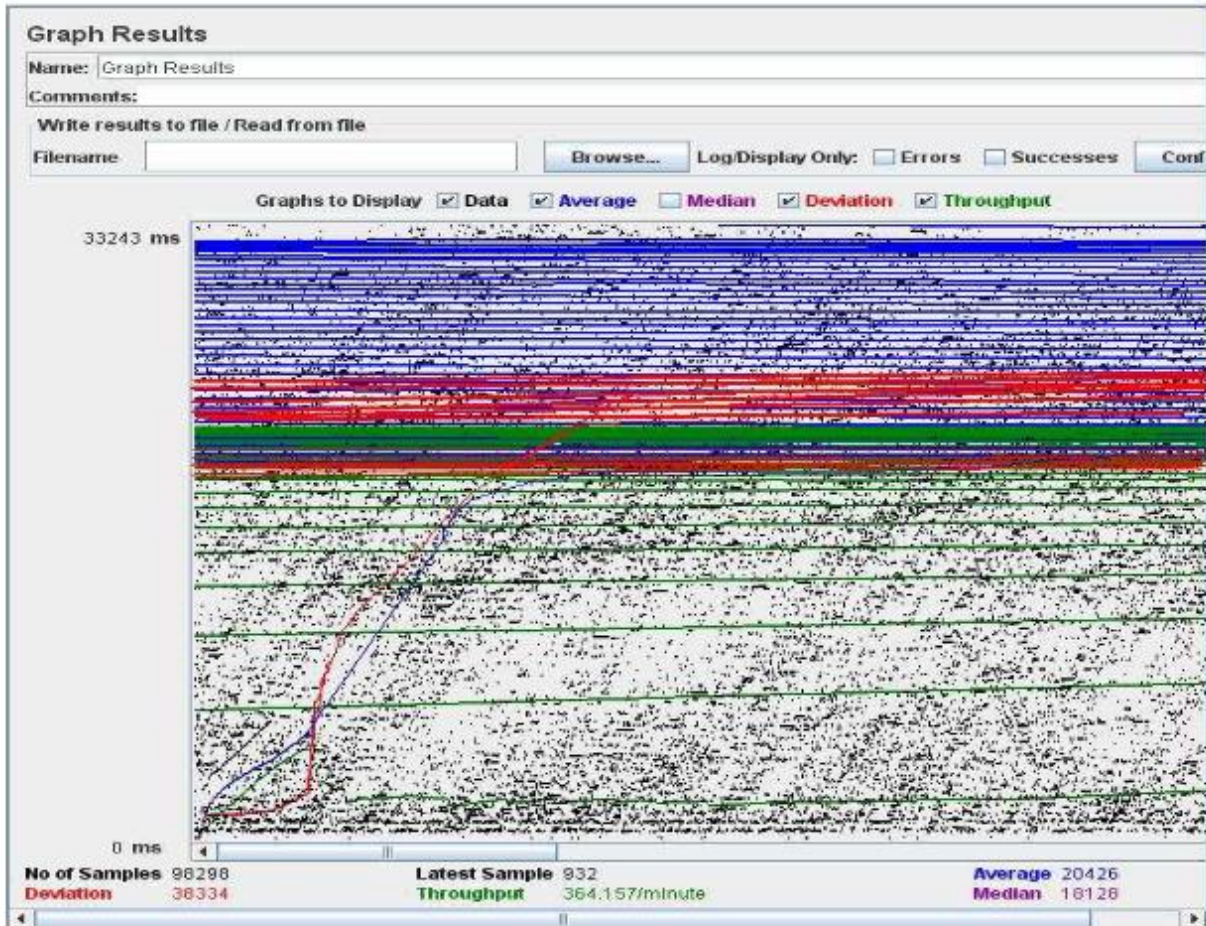


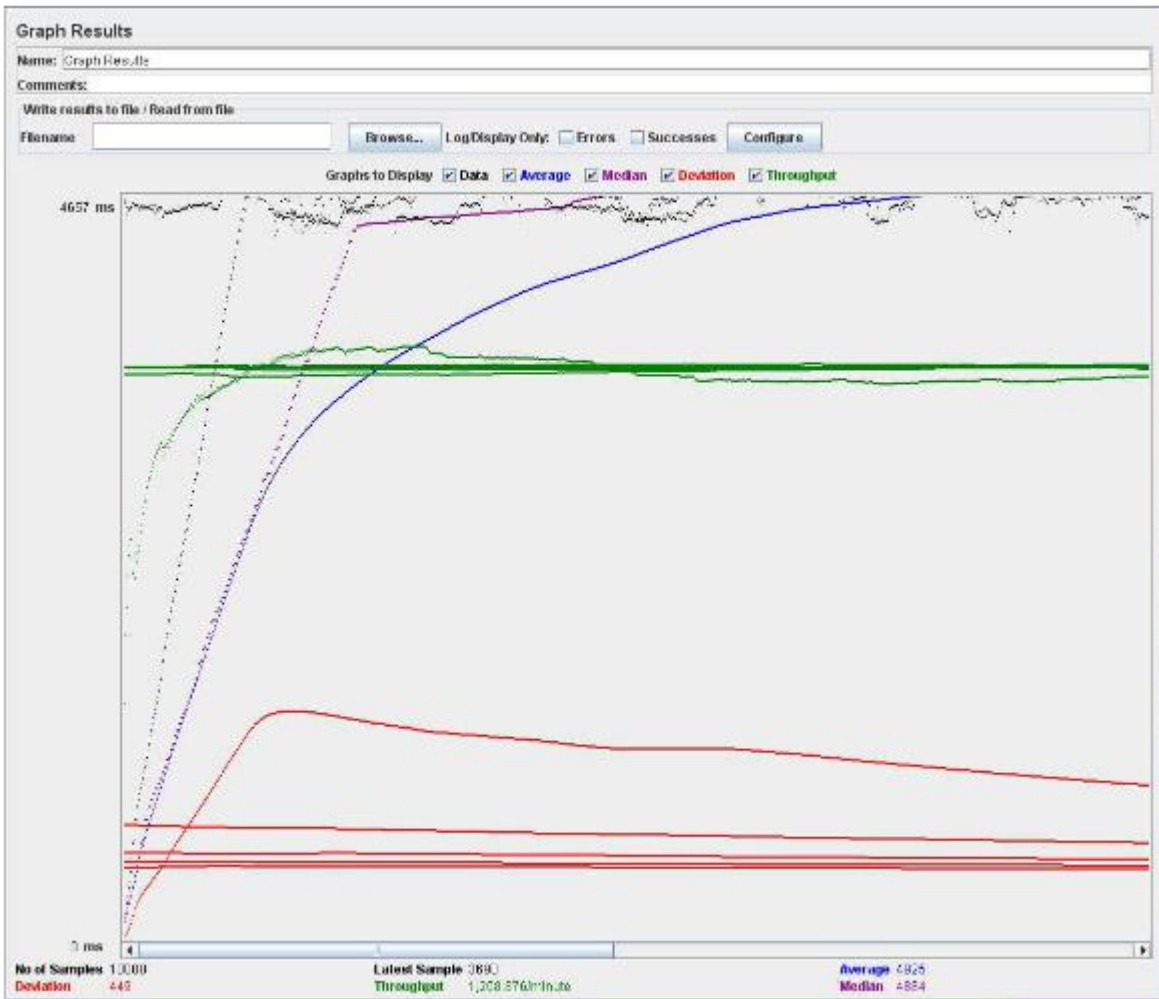
Figure 5.5 Search Page Test Case 3.1 - Graph Results

4.2.2 Details Web page

4.2.2.1 Test Case 1

Users	LoopCount	Ramp-up period (sec)	Average Response Time (ms)
100	16000	5	4657

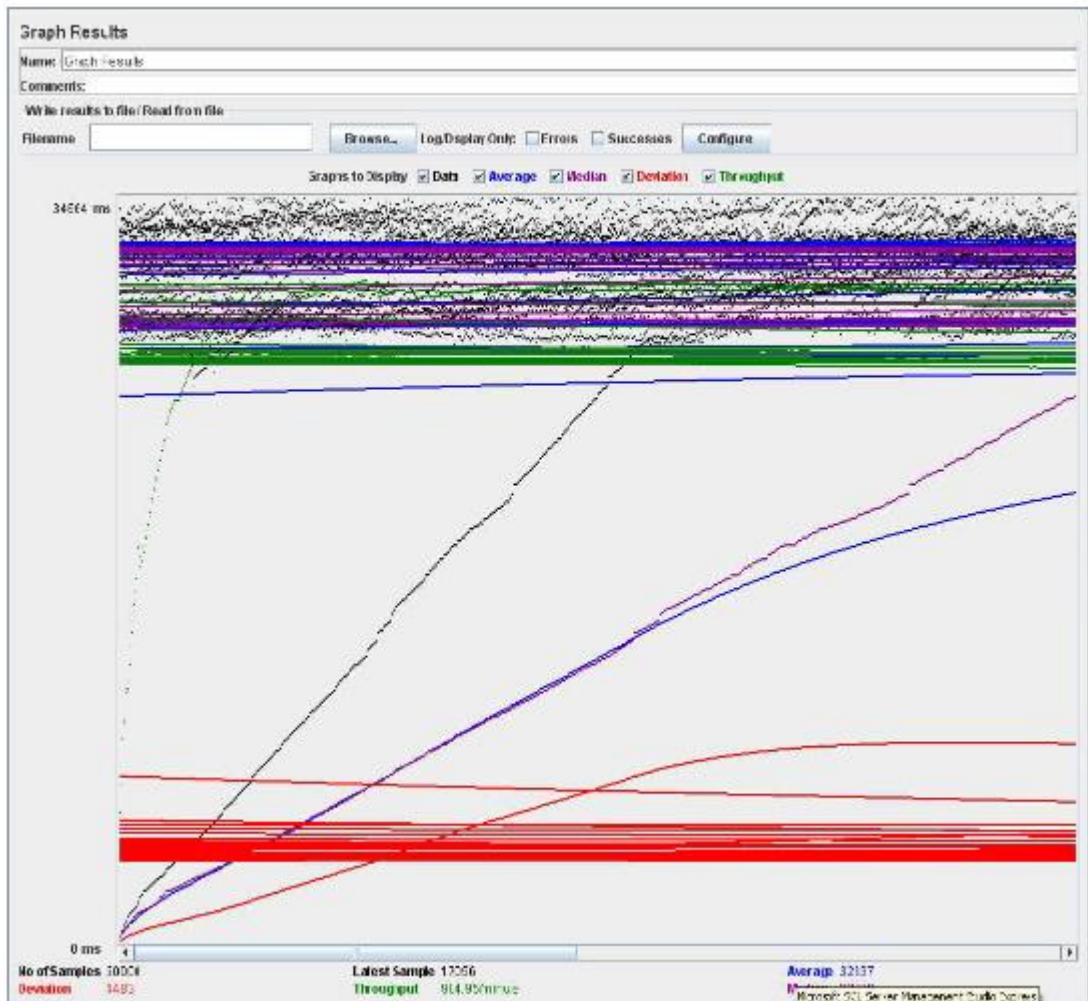
Table 4.7 Test case 1



4.2.2.2 Test Case 2

Users	LoopCount	Ramp-up period (sec)	Average Response Time (ms)
500	16000	5	34664

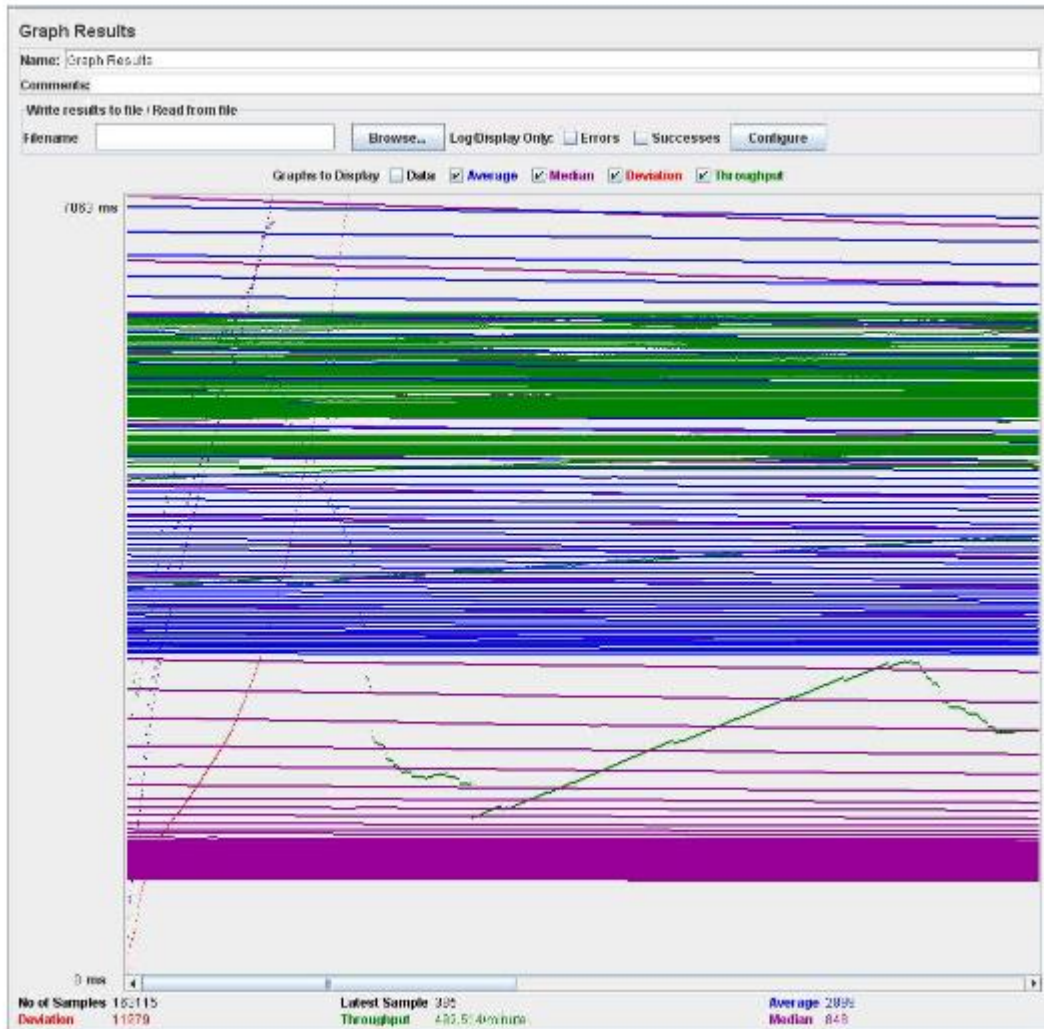
Table 4.8 Test case 2



4.2.2.3 Test Case 3

Users	LoopCount	Ramp-up period (sec)	Average Response Time (ms)
1000	16000	5	70063

Table 4.8 Test case 2



4.4 Test Analysis

The main purpose of performing JMeter Testing is analyzing the response time requirements of the web applications. In context of web applications, response time refers to the time elapsed between the submission of a request and the receipt of the resulting HTML. While calculating the response time from JMeter, it does not take into account the rendering time required by the browser to load the HTML. Thus, Response Time is just the exact time between sending the request and

receiving it. The Red Line represents the Deviation which is the deviation from the average response times which can take place if some percent of the samples have less response times while others have high. The main statistics for the analysis of the load on the web application comes from the Average Response Time and the Deviation readings. Analyzing the search page results we find that, when the number of users are too less i.e. 10 the Response time comes up to approx .5 seconds which is reasonable enough. When the number of users are increased to 100 the Response Time increases to approx 6 seconds indicating more load on the application. Initially when the application starts, the number of users are less for the first few seconds so the Average and Deviation increase parallel but as the number of users are increased the deviation starts decreasing to a constant line and at the same time the Average increases to a constant line. Thus, the Average Time comes up to 6 seconds and Deviation to .4 seconds. Gradually, when the load on the application is too high the Deviation shoots up and the readings come close to Average line. Increasing more number of users i.e. 500 the Average Time further increases to 5 seconds over 80,000 samples. The same scenario can be scene with this case with Average and Deviation moving parallel and then deviation reduces when number of samples increase and finally the deviation shoots upcoming parallel to Average. Finally, incrementing the number of users to 1000 takes the maximum Average Response Time and puts more load on the application. Analyzing the Throughput i.e. number of requests per minute we figure out that it is increases and then comes to a constant and a reasonable value. The results can be explained for the Details page. The figures and Tables above indicate the Response Time and Throughputs for the same number of users i.e. 100, 500, 1000. Looking at the results and the analysis it is evident that for this application and the above mentioned system configuration the application works fine with 100 users but as we increase the number of users

the application starts hanging up indicating the increase of load at the server. The main reason for these performance results is the fault of the system level and the probable reason for it is its configuration. If we have to increase the performance of the application we need to work on the system configuration and upgrade it but before that it is important to figure out which part of the system would shoot up the application's performance. The main factors which might be cause the application to have low performance are: CPU, RAM and Hard Disk. Out of the three if the performance degradation takes place because of the low CPU processing then it is because of Thrashing and so the better CPU configuration is required to handle more users. In order to confirm this assumption we need to look for the evidence and figure out which one needs to be improved. Since the application was not able to handle the load properly for 100 users, I performed the tests for 10, 40 and 50 users. To confirm if the RAM was degrading the performance, I performed the same tests simultaneously at two machines with the same configuration but different RAM's. Both these machines have the application deployed in it and could make out that the performance was still the same. I tested it for CPU by running this application on my laptop and then taking the results in JMeter, when the number of users were increased from 40 to 50 and so on the CPU started blinking indicating the increase of load put on it and also the Task Manager gave a good indication about the amount of time the CPU is taking for testing the application. Table 4.9 show the increase in Response Times with the increase in load i.e. number of users.

Users	Average Response Times (ms)
10	532
40	2543
50	3120

Table 4.9 Average Response Time Summary

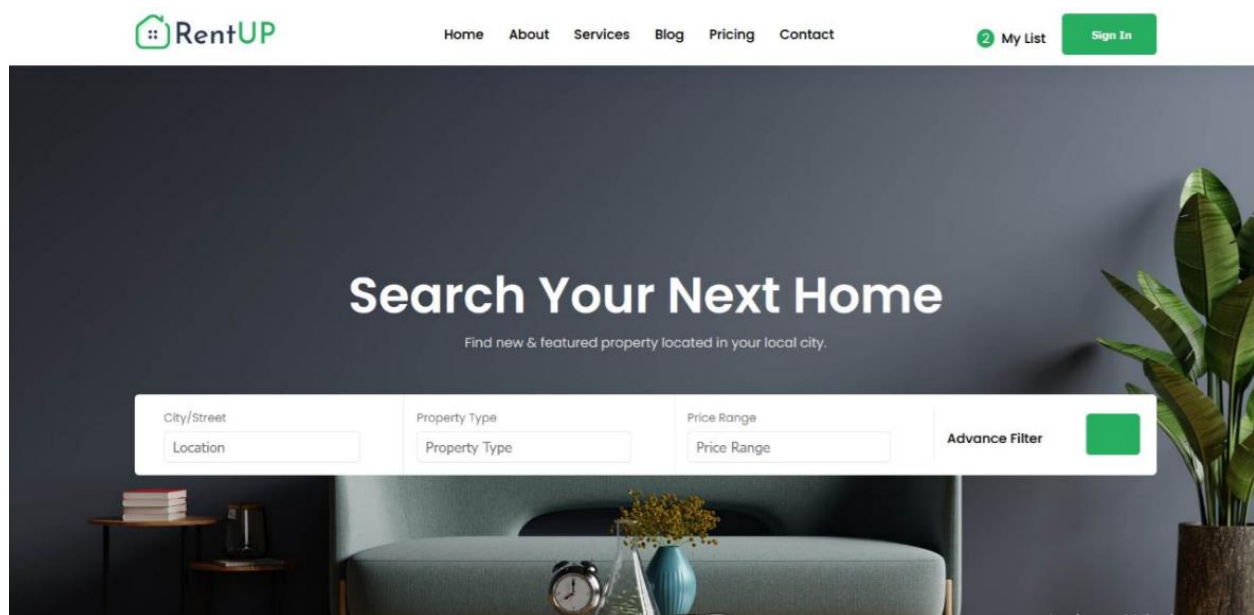
This confirms that the CPU needs to get upgraded with high configuration in order to get good performance results for more users. The CPU can be upgraded with a better processor speed i.e. more than 2 GHz in order to increase the performance load. Table 5.10 shows the results of the system configuration testing.

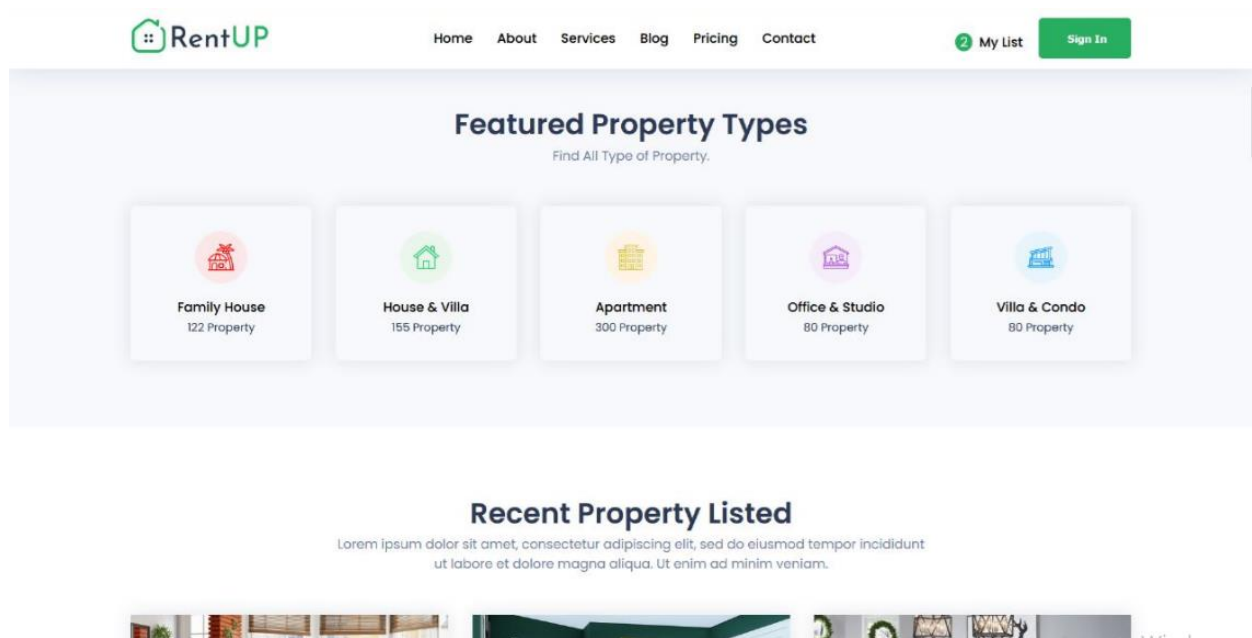
System Part	CPU	Hard Disk	RAM	Reason
Performance	Low	High	High	Low Processor Speed

Table 4.10 System Configuration Testing for Performance Upgrade

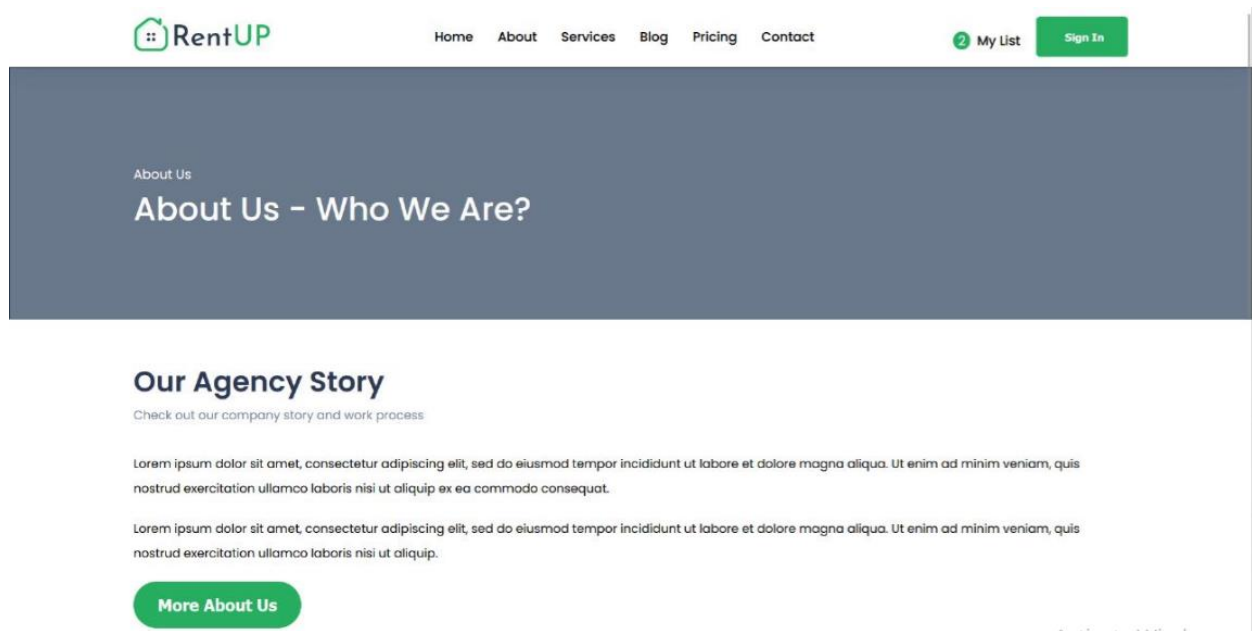
4.5 Screen Shot of System Web Pages

4.5.1 Sign In Webpage

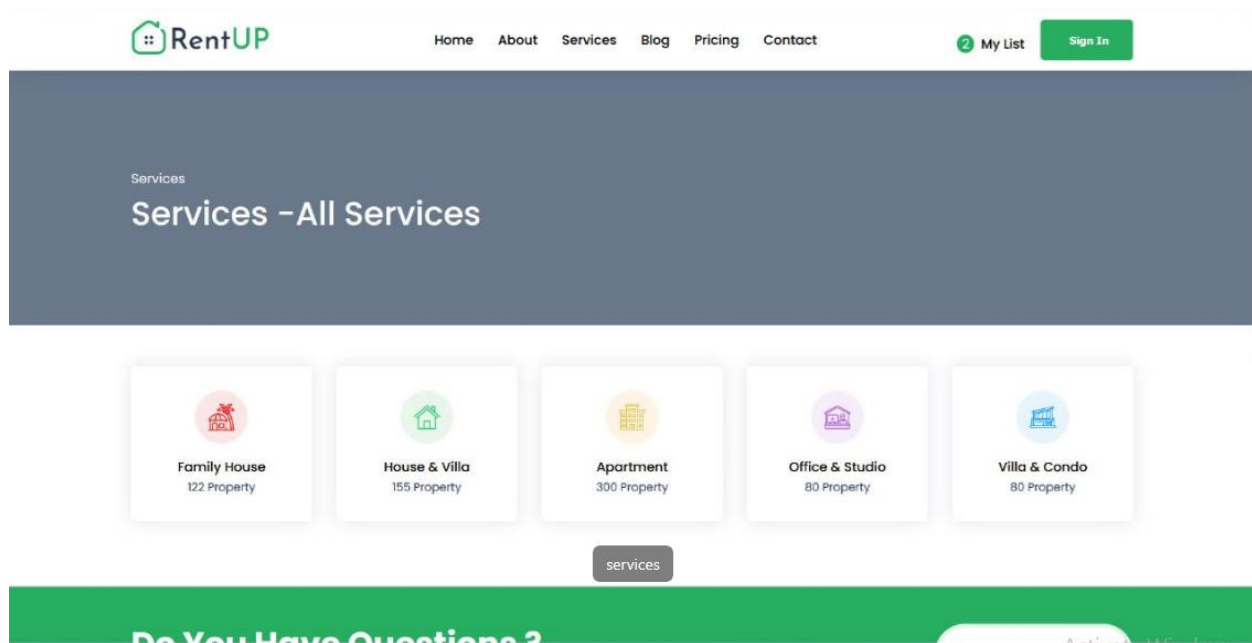




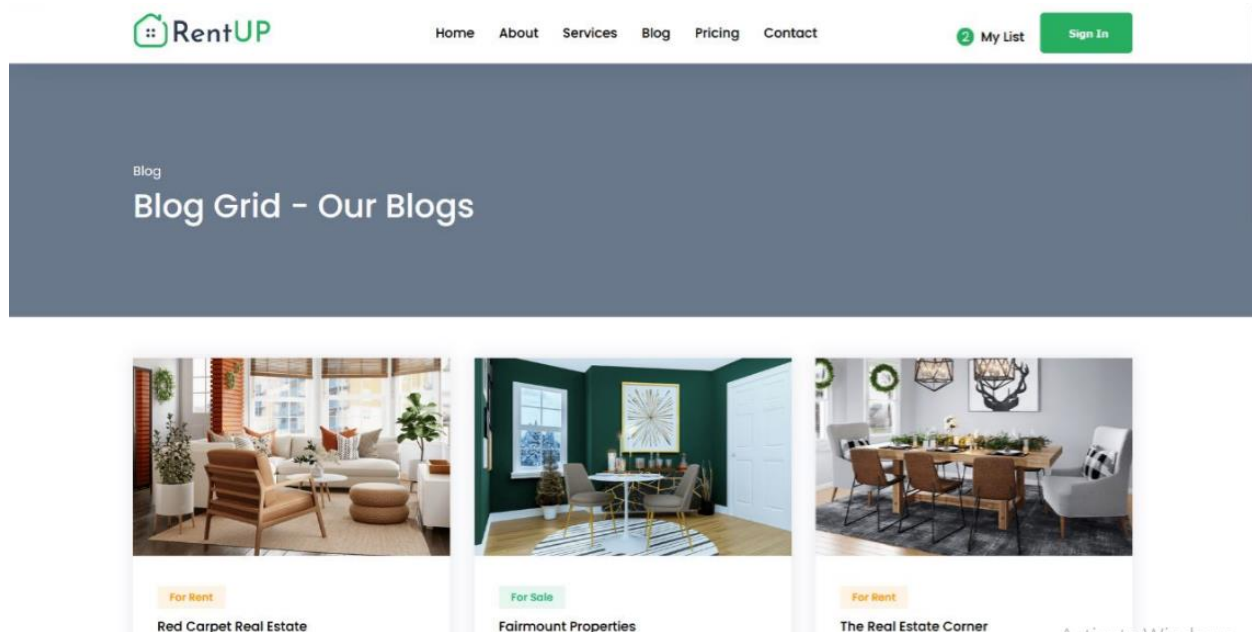
4.5.2 About Webpage




4.5.3 Services Webpage



4.5.4 Blog Webpage



4.5.5 Pricing Webpage

[Home](#) [About](#) [Services](#) [Blog](#) [Pricing](#) [Contact](#) 2 My List [Sign In](#)

30 days money back guarantee

No Extra Fees. Friendly Support

Basic

Rs 2900

per user, per month

- 99.5% Uptime Guarantee
- 120GB CDN Bandwidth

Best Value

Standard

Rs 4900

per user, per month

- 99.5% Uptime Guarantee


Platinum

Rs 7900

2 user, per month

- 100% Uptime Guarantee
- 200GB CDN Bandwidth

4.5.6 Contact Webpage

[Home](#) [About](#) [Services](#) [Blog](#) [Pricing](#) [Contact](#) 2 My List [Sign In](#)

Contact Us

Get Helps & Friendly Support

Fillup The Form

[Submit Request](#)

CHAPTER 5

PROJECT METRICS AND EXPERIENCE

This chapter presents the project metrics showing the number of hours spent completing each phase of the project. It also summarizes the experiences gained during the entire life-cycle of the project.

5.1 Project Metrics

Project Metrics discussed in this document are source lines of code and the amount of time spent during the entire project span. Table 1 and Table 2 represent the project metrics: source lines of code and project phases and their duration respectively.

ASP.NET & C# Server Side Code	Handwritten C# Code – 1180 lines approx. Auto-generated C# Code – 750 lines approx.
SQL Code	300 lines
CSS Code	500 lines

Table 5.1: Project Lines of Code

Learning Project Technologies	1 Week
Requirement Gathering and Design	2 Week
Implementation	4 Weeks
Testing	1 Week
Documentation	2 Weeks

Table 5.2: Project Planning Phase

5.2 Overall Experience

The idea of developing Property Dealing Website originated when an immediate requirement aroused from local Real Estate Company to develop a new website for them. This company had its own website in production but wanted to re-design it with some attractive features which could help them fetch more business. The project started with an intention to develop an application specific to the needs of this company but later due to the lack of funds the project could not be continued. But with the continuous support of Dr. Andresen this application came into development. There are hundreds of real estate websites on the World Wide Web with same features but the intention of building this application was to design something new and innovative and include some cool features which have not been incorporated in these websites so far. The biggest challenge involved in this project was to gather the requirements and design its structure so that it can altogether have a new look. It also involved thinking about new features which can be incorporated in this application and could make the search of listings much easier for the real estate buyers. Understanding the structure of the application was the biggest problem on the start of the project. Finally, the scope of this application was defined which greatly helped in understanding what all features have to be included in the project. The whole emphasis in this application is given on the search criteria to help buyers to search for new property listings. After the Requirement Specification, learning .NET 2005 and new AJAX controls was the second phase which gave me a great learning experience. From my past knowledge of .NET, working with the new version of Microsoft Visual Studio was not a big problem but incorporating AJAX features to it was a deal. I spent around two weeks of time learning these technologies with the help of online tutorials and sample applications. This learning brought me in a very comfortable position to

think about what new AJAX features I can put up in the application. Another problem I faced in my project was the implementation of the drag and drop tool which was required to store the saved listings by the buyer. Traditionally, to save the listings the buyer has to register to the website so that the selected listings can be saved corresponding to the account of the user but this project implements this using the session. As soon as the user starts a search a new session is created and during this session the buyer can store the listings in the tool by dragging and dropping it on the listing cart. Once the user resets the search or starts a new search, a new session is created and the saved listings are lost. This session based application does not force the buyer to log in or register to just select the property listings. I have learnt also of new things while developing dynamic web-applications using Microsoft ASP.NET and AJAX. This project gave me an opportunity to learn a lot of new concepts in programming and web development such as Java Script, AJAX controls and ASP.NET 2.0 which would help me throughout my career.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter describes the future scope and extensions for the project. There is still a huge scope of implementing something new and more to the project which can make it to the level of a commercial product. This section also concludes stating the advantages and applications of this Real Estate Web Application.

6.1 Conclusion

This Property Dealing Website is a typical .NET web application using ASP.NET 2.0 and SQL 20005 in the C# programming language. It uses a client/server architecture based on the HTTP protocol. It is developed in Microsoft's Visual Studio .NET programming environment. The buyer performs a search for the property listings by putting either Zip code/City/State or MLS# in the search textbox. The business logic tier communicates with the database tier requesting the results of the query sent by it. The results obtained by the database are displayed on the data grid, by refreshing the grid rather than refreshing the entire web page. Efficiency of the application is improved by the use of web methods that help in separating Application Tier from the Presentation Tier. The performance of this application is evaluated by rigorously testing it against various test scenarios. Efficiency and correctness of the application is evaluated with the help of various test cases. Some ways in which this system could be enhanced with additional functionalities are discussed in the section.

6.2 Future Extensions to the Project

This project is developed as a master's project and still gives lot of scope for its extension which could be made to the project if it is going to be developed as commercial product. We can use pure object-oriented domain model to deal the database access tier using Visual Studio 2005 and SQL Server 2005. In doing this, we can get a better architecture design which will improve code efficiency and running performance. Besides, we can build XML web service programming model that enables other applications to consume real estate web services built by us using standard protocol such as HTTP, XML, XSD, SOAP and web services description language (WSDL). This project just deals with the Home page and Search page to search for property listings, more functionality can be added for searching the agents and offices making it a complete application. With the advancement of technology, dynamic maps can be generated using AJAX which can help the buyer locate a particular area where the property is located in the Google Map. Inclusion of all these features would make the application feature rich. The advantages of putting these functionalities in the project are described in detail in the following sections.

6.2.1. Using Object-Oriented Domain Model

Visual Studio .NET 2005 add a significant feature of object-oriented domain model. Instead of using traditional Relational Database Model, the database access layer treat each table as class model and each row as instance object. It gives a

friendly development thought to the developer and makes the business logic more convenient when interacting with the back end. Currently, there is a third-party tool called NEO which could generate the domain model for database automatically. In using object-oriented domain model, the database is transported to the developer and the application using these domain classes to handle data processing instead of using disconnected Dataset or typed Dataset.

6.2.1.1 Providing Web Services

XML and web services are used to build highly scalable, loosely coupled, distributed application using standard web protocols such as HTTP, XML, and SOA. An XML web service is a component that implements the program logic and provides functionality to other applications. There are some examples such as Amazon.com, Google.com. WSDL is used to describe XML web service and XML-based messaging is used to send and receive data. Visual Studio .NET IDE has a powerful support to develop XML web services. We can create a real estate web application which provides application logic to a client application. For example, I can create a search function to search the houses based on the feature. The client can call my web service and provide parameters to the search function and the web service will return dataset containing the result house records list back to the client. We can create other convenient functions and the client can easily build their own custom web application while still using the programming logic from the server web services.

6.2.1.2 Functionality Extensions

If we are going to develop commercial real estate web application based on this project, we can add some more feature as the following:

1. We can provide a login page for buyers and store their username and password in the database and can save the search criteria for the buyers. This will help the buyers to use the same search criteria rather than creating a new criterion every time the buyer performs the search.
2. By using the Ad Rotator web control we could add more advertisement on the web site or links to other web site.
3. Add agent and company search functionality.
4. Displaying the search results on Google Maps locating the area where the property listing is located.
5. Refine this web site and make it friendly and pretty.

References

- [1] ASP.NET and Web Development Overview,
<http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx>

- [2] Microsoft Visual Studio Overview,
http://en.wikipedia.org/wiki/Microsoft_Visual_Studio#Visual_Studio_2005

- [3] Microsoft SQL Server 2005,
http://en.wikipedia.org/wiki/Microsoft_SQL_Server

- [4] 3-Tier System Architecture,
http://en.wikipedia.org/wiki/Multitier_architecture

- [5] Introduction to ADO.NET,
<http://en.wikipedia.org/wiki/ADO.NET>

- [6] Load Testing using Apache JMeter Testing Tool,
<http://jakarta.apache.org/jmeter/>

- [7] Introduction to .NET Architecture,
<http://www.devtopics.com/what-is-net/>

[8] Common Language Infrastructure Definition,

http://en.wikipedia.org/wiki/Common_Language_Infrastructure

[9] AJAX – Bridging the Thin-Client Performance Gap,

<http://www.ironspeed.com/articles/ajax-bridging%20the%20thin-client%20performance%20gap/article.aspx>

[10] 3-Tier Architecture

<http://www.c-sharpcorner.com/>

