SERVICE ORIENTED ARCHITECTURE FOR ONLINE GAMING SERVICE

By

Sasikumar.P
Reg. No. 71204621039

of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE

A PROJECT REPORT
Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree
of

MASTER OF COMPUTER APPLICATIONS

July, 2007

Kumaraguru College of Technology
Coimbatore – 641006.

Department of Computer Applications

Bonafide Certificate

Certified that this project report titled **Service Oriented Architecture for Online Gaming Service** is the bonafide work of **Mr. Sasikumar.P** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide

Head of Department

Submitted for the University Examination held on 03/07/07

Internal Examiner

External Examiner 3/7/07

CARITOR

May 25th, 2007

To whomsoever it May Concern

This is to inform you that **SASIKUMAR.P** has successfully completed his project assignment titled **SERVICE ORIENTED ARCHITECTURE FOR ONLINE GAMING SERVICE** as a part of **MCA** curriculum.

As a Project Trainee, he started this project on **December 18, 2006** and completed it on **May 25, 2007**

Please note, as per the company's policies and practices, the company retains ownership of the intellectual property rights concerning work undertaken during projects and disclosure of the source code and any other relevant information or data out of the organization is strictly prohibited.

SASIKUMAR.P designated, as project trainee will not be delivering the respective source code pertaining to his project.

For Caritor (India) Pvt Ltd,

BHAVANI DEVAIAH
MANAGER – HUMAN RESOURCES

ABSTRACT

Online games refer to the video games that are played over some form of computer networks, most commonly the Internet. Online games started to move from a wide variety of LAN protocols and onto the Internet. The project titled "SERVICE ORIENTED ARCHITECTURE FOR ONLINE GAMING SERVICE "consists of two main divisions they include gaming service and development of the game.

The gaming service consists of providing service to desktop users and web users. The service provider (server) is implemented in such way that it would support both desktop and web users.

This system provide services include new user registration, user authentication, search for games, if it is single-user game user can directly play the game, if it is two-user game user can select the opponent by sending message and then play the game. The service-oriented architecture (SOA) is the state-of-art technology is being implemented. It provides for loose coupling among the communicating software agents.

The gaming service can provide a synchronized way of logging in and playing the games. The score details are stored in the database. It also aims at creating an efficient and user-friendly environment for the users. The project use ASP.NET(C#) as the front end and SQL Server 2005 as the back end, which is highly configured to accommodate the desired function.

ACKNOWLEDGEMENT

I wish to express my sincere thanks to **Dr. JOSEPH V.THANIKAL Ph.D.,** Principal, Kumaraguru College of Technology, Coimbatore, for permitting me to undertake this project.

My deepest acknowledgement to **Dr. M. GURURAJAN Ph.D.,** Head of the Department, Computer Applications, Kumaraguru College of Technology, Coimbatore, for his timely help and guidance throughout this project.

I am greatly indebted to my guide **Mr. A. MUTHUKUMAR M.Sc,MCA,M.Phil.,** Project Coordinator, Assistant professor, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for his valuable guidance and encouragement at every stage of this project work.

I express my sincere thanks to **MR.RAVINDRA MALLIPUDI.,** Project Lead, CARITOR (INDIA) PRIVATE LIMITED, for his support and assistance at various level of my project work.

Finally, I owe a lot to my beloved parents and family members and to my department staffs without their help and co-operation the project would not have taken a final shape.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 ORGANIZATION PROFILE

Caritor was founded as a U.S Corporation in the San Francisco Bay Area, California in 1993. Caritor is a global software services company with operations in the USA, UK, France, Middle East and Asia Pacific with distributed Offshore Development Centers in India at Bangalore and Chennai.

Caritor employs more than 3684 people around the global who are collectively responsible for our successful track record of delivering adaptive software solutions and packaged software products to some of the best-known organizations in the world.

Our software development operations are assessed at SEI-CMM Level 5, PCMM Level 5, CMMI Level 5, ISO 9001 and BS 7799:2002 levels to ensure reliable, high-quality solutions for all our clients.

Industrial Verticals

- Telecommunications
- Healthcare
- Public Sector

Service Offerings

- Architecture & Design
- Systems Integration
- Enterprise Business Solutions

1.2 SYSTEM SPECIFICATION

1.2.1 Software Specification

Front end	:	ASP.NET
Middle Tier	:	Web Services, C#
Back end	:	SQL Server 2005
Operating System	:	Windows XP

1.2.2 Hardware Specification

Processor	:	Pentium IV
RAM	:	512MB
Hard Disk	:	80GB

1.3 PROGRAMMING ENVIRNOMENT

1.3.1 Technologies Used

- **Service Oriented Architecture (SOA)**

SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. SOA is essentially a collection of service. A Service is a unit of work done by a service provider to achieve desired end results for a service consumer. SOA is not tied to a specific technology. SOA supports requirements of business processes and software users.

SOA (service-oriented architecture) has become a buzzword of late. Although the concepts behind SOA have been around for over a decade now, SOA has gained extreme popularity of late due to web services. Before we dive in and talk about what SOA is and what the essentials behind SOA are, it is a useful first step to look back at the evolution of SOA. To do that, we have to simply look at the challenges developers have faced over the past few decades and observe the solutions that have been proposed to solve their problems.

- **The Problem**

Early programmers realized that writing software was becoming more and more complex. They needed a better way to reuse some of the code that they were rewriting. When researchers took notice of this, they introduced the concept of modular design. With modular design principles, programmers could write subroutines and functions and reuse their code. This was great for a while. Later, developers started to see that they were cutting and pasting their modules into other applications and that this started to create a maintenance nightmare;

when a bug was discovered in a function somewhere, they had to track down all of the applications that used the function and modify the code to reflect the fix. After the fix, the deployment nightmare began. Developers didn't like that; they needed a higher level of abstraction.

Researchers proposed classes and object-oriented software to solve this, and many more, problems. Again, as software complexity grew, developers started to see that developing and maintaining software was complex and they wanted a way to reuse and maintain functionality, not just code. Researchers offered yet another abstraction layer to handle this complexity -- component-based software. Component-based software is/was a good solution for reuse and maintenance, but it doesn't address all of the complexities developers are faced with today. Today, we face complex issues like distributed software, application integration, varying platforms, varying protocols, various devices, the Internet, etc. Today's software has to be equipped to answer the call for all of the above. In short, SOA (along with web services) provides a solution to all of the above. By adopting a SOA, you eliminate the headaches of protocol and platforms and your applications integrate seamlessly.

How does SOA achieve loose coupling among interacting software agents? It does so by employing two architectural constraints:

1. A small set of simple and ubiquitous interfaces to all participating software agents. Only generic semantics are encoded at the interfaces. The interfaces should be universally available for all providers and consumers.
2. Descriptive messages constrained by an extensible schema delivered through the interfaces. No, or only minimal, system behavior is prescribed by messages. A schema limits the vocabulary and structure of messages.

- **Key Components of SOA**

The first step in learning something new is to understand its vocabulary. In the context of SOA, we have the terms service, message, and web services. Each of these plays an essential role in SOA.

- **Service**

A service in SOA is an exposed piece of functionality with three properties:

1. The interface contract to the service is platform-independent.
2. The service can be dynamically located and invoked.
3. The service is self-contained. The service maintains its own state.

A platform-independent interface contract implies that a client from anywhere, on any OS, and in any language, can consume the service. Dynamic discovery hints that a discovery service (e.g., a directory service) is available. The directory service enables a look-up mechanism where consumers can go to find a service based on some criteria. For example, if one was looking for a credit-card authorization service, he might query the directory service to find a list of service providers that could authorize a credit card for a fee. Based on the fee, he would select a service . The last property of a service is that the service be self-contained

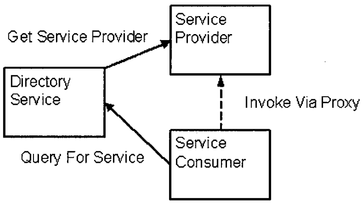
- **Message**

Service providers and consumers communicate via messages. Services expose an interface contract. This contract defines the behavior of the service and the messages they accept and return. Because the interface contract is platform- and language-independent, the technology used to define messages must also be agnostic to any specific platform/language. Therefore, messages are typically constructed using XML documents that conform to XML schema. XML provides all of the functionality, granularity, and scalability

required by messages. That is, for consumers and providers to effectively communicate, they need a non-restrictive type of system to clearly define messages; XML provides this. Because consumers and providers communicate via messages, the structure and design of messages should not be taken lightly. Messages need to be implemented using a technology that supports the scalability requirements of services. Having to redesign messages will break the interface to providers, which can prove to be costly.

- **Web Services**

The World Wide Web is more and more used for application-to-application communication. The programmatic interfaces made available are referred to as Web Services. The goal of Web Services activity is to develop a set of activities in order to lead Web Services to their full potential.



Directory service

Web Services are units of programming application logic that sit on Web Service and can be accessed remotely using standard Internet protocols such as

- XML.
- HTTP.
- SOAP.

Web Services overcomes the problems like getting through firewalls, Communicating across platforms, Keeping system components independent of each other which was faced by middleware platforms such as DCOM and JAVA RMI. Though the CORBA specification allows for the applications to be loosely coupled and be technology-independent, it has other issues due to inconsistent implementation by the vendors.

Web Services are loosely coupled, so that the implementation of either the server or the client could be changed keeping the application unaffected and still functioning. They are supported by all Operating Systems and development platforms.

- **Stateless Service**

Each message that a consumer sends to a provider must contain all necessary information for the provider to process it. This constraint makes a service provider more scalable because the provider does not have to store state information between requests. This is effectively "service in mass production" since each request can be treated as generic. It is also claimed that this constraint improves visibility because any monitoring software can inspect one single request and figure out its intention.

- **Stateful Service**

Stateful service is difficult to avoid in a number of situations. One situation is to establish a session between a consumer and a provider. A session is typically established for efficiency reasons. For example, sending a security certificate with each request is a serious burden for both any consumer and provider. It is much quicker to replace the certificate with a token shared just between the consumer and provider. Another situation is to provide customized service.

Stateful services require both the consumer and the provider to share the same consumer-specific context, which is either included in or referenced by messages exchanged between the provider and the consumer. The drawback of this constraint is that it may reduce the overall scalability of the service provider because it may need to remember the shared context for each consumer. It also increases the coupling between a service provider and a consumer and makes switching service providers more difficult.

- **Asp .Net**

ASP .NET is a part of .NET Framework. It is a programming framework builds on the Common Language Runtime that can be used on a server to build powerful Web applications.

ASP .NET runs alongside ASP. ASP .NET has page file extension as .aspx. It supports multiple languages like VB .NET, C#, jscript, etc. Caching techniques, Session Management, Configuration and Deployment are some of the features. It is a simplified and event based programming model.

- **SQL Server**

Microsoft SQL Server 2005 is a database and data analysis platform for large-scale online transaction processing (OLTP), data warehousing, and e-commerce applications.

Microsoft SQL Server 2005 extends the performance, reliability, availability, programmability, and ease-of-use of SQL Server 2000. SQL Server 2005 includes several new features that make it an excellent database platform. A database system is an overall collection of different database software components and database containing the parts. SQL server provides a mechanism called a trigger for enforcing procedural integrity constraints. A trigger is a mechanism that is invoked when a particular action occurs on a particular table.

- **C#**

The first component-oriented language in the C/C++ family is C#. It is robust and durable software for next generation. Its component concepts include mainly classes and objects. It also includes properties, methods, events, design-time and run-time attributes and integrated documentation using XML.

It can be embedded in web pages. It has no header files. It has improved extensibility and reusability. Pervasive versioning considerations in all aspects of language design. A set of .NET components collectively known as ADO.NET provides efficient access to relational database and variety of data source. Components are also available to allow access to the file system and to directories.

CHAPTER 2

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Service Oriented Architecture for Online Gaming Service is being developed as a new service to provide access to various online games. This uses a Service Oriented Architecture (SOA), which addresses the drawbacks in using the existing technologies mentioned below:

- The technology made use of tightly coupled software in which the processing components were highly interdependent.
- The system would otherwise be vendor specific and platform dependent and need homogenous infrastructure.
- If the implementation of one system changes with time, then all other components should be recompiled.
- There was no facility to search for a particular game as per user's wish and play.
- There was no structure or strategies involved to display the score details.
- Any user can access the system. There was no security facility like authentication.

2.2 PROPOSED SYSTEM

The proposed system aims at fulfilling the needs of the user by browsing through and selecting the game, which the user wishes to play, and thus providing the users flexibility on playing games.

The features of the proposed system provide the consolidated way of synchronizing the users by authentication process for logging in. The users can then visit the site and view the contents. The users can also key in the name of the game if he knows already and then start playing the game he wishes to play. Each user can login in the similar way and play the desired game.

The features also provide a loosely coupled environment. The proposed system is not vendor specific and is platform independent. The infrastructure is heterogeneous. The implementation of one system does not affect the other components and hence they need not be recompiled.

2.3 FEASIBILITY ANALYSIS

Feasibility analysis is the measure of how beneficial or practical the development of Information System will be to the Organization. Once the scope of the problem has been identified and initial analysis of the system has been completed, the feasibility of the project and the software to be developed is put under the test. This is done at this point to basically verify whether the software which is to be developed meets the scope and whether the project is feasible to do. Preliminary investigations would reveal whether the entire project would be possible to carry out in the current environment. The three stages of feasibility analysis are described below.

2.3.1 Technical Feasibility

Technical Feasibility is the measure of practicality of a specific technical solution and the availability of technical resources and expertise. It centers on the existing computer system (hardware, software, etc.) and to what extent it can support the new addition. The technical issues that have to dealt with in this phase are regarding the technology the system uses and its modifiability. It be verified that the system would be within the state of the art and defects, if any can be reduced to a level matching the application needs. Other considerations include:

- ➢ Existing of the necessary technologies.
- ➢ Capacity of the proposed system to hold the required data.
- ➢ Ability of the system to respond accurately to all inquiries.
- ➢ Scope of expandability of the system
- ➢ Technical guarantee of accuracy, reliability, ease of access and data security.

When all of the above mentioned features have been considered and the result is found to be positive, then the system development is said to be technically feasible.

2.3.2 Operational Feasibility

Operational Feasibility asks if the system will work when it is developed and installed. It checks for the support of the management, the current business methods, user's involvement and their attitude towards the proposed system, etc.

The proposed system is said to be operationally feasible only it meets all the requirements of the client. The proposed system has found encouraging from all type users as the system is said to flexible to the users and its operations are transparent as it the makes use of the loosely coupled architecture called as the service oriented architecture (SOA). The operations performed in the system are transparent as it provides for the supports variety of technologies besides the proper handshaking is followed.

2.3.3 Economic Feasibility

Economic feasibility is the measure of the cost-effectiveness of the proposed system. The investment to be made in the proposed system must prove a good investment to the organization by returning benefits equal to or exceeding the costs incurred in developing the system.

The proposed benefits of the system will outweigh the costs to be incurred during system developed since the system does not require procurement of additional hardware facilities it is economically feasible. In addition capability of the system to incorporate future enhancement will improve the performance to suit the future need of the client organization.

2.3.4 Users of the System

The new users are new user, they should be registered first this process involves the details to be furnished to the server. The registered users can directly play the game by giving the login and password details.

CHAPTER 3

3. SYSTEM DESIGN AND DEVELOPMENT

3.1 ELEMENTS OF DESIGN

System Design is the most creative and challenging phase in the development of a software system. Design implies to a description of the final system and the process by which it is developed. The first step is to determine what input data is needed for the system and then to design a database that will meet the requirements of the proposed system. The next step is to determine what outputs are needed from the system and the format of the output to be produced.

During the design of the proposed system some areas where attention is required are:

- What are the inputs required and the outputs produced?
- How should the data be organized?
- What will be the processes involved in the system?
- How should the screen look?

The steps carried out in the design phase are as follows:

- File Design
- Modular Design
- Input Design
- Output Design
- Database Design

3.1.1 Features of a well defined system

In design an efficient and effective system is of great importance to consider the human factor and equipment that these will require to use. System analyst must evaluate the capabilities and limitations of the personal and corresponding factors of the equipment itself.

The characteristics associated with effective system operation are:

- Accessibility
- Decision making ability
- Economy
- Flexibility
- Reliability
- Simplicity

3.1.2 File Design

Software tools are used to describe, manipulate and manage data. Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information. The objectives of the file design are to provide effective auxiliary storage and to contribute to the overall efficiency of the computer program component of the system.

3.1.3 Modular Design

It is always difficult for any System Development team to grasp a system without breaking it into several smaller systems. These smaller systems will be a part of the original system yet they will be independent in the sense that

they will incorporate within them the major functionalities of the proposed system.

A software system is always divided into several subsystems which make it easier to develop and perform tests on the whole system. The subsystems are known as the modules and the process of dividing an entire system into subsystems is known as Decomposition.

The modules identified for the proposed Service Oriented Architecture for Online Gaming Service are as below:

- Authentication
- Gaming Service
- Gaming

MODULE DESCRIPTION

Authentication

Main functionality's are

- User Registration
- User Authentication

User Registration:

Every user should register before using the gaming service. The new user should give their details of user id, first name, last name , password, e-mail id, date of birth, address and gender.

Authentication:

The validation of user id and password of the existing users are authenticated. If the user types in the wrong user id and password means an message would be triggered such that you are an invalid user. These are the process which are being performed in the authentication phase.

Gaming Service

The Gaming Service module main functionality's are

- Search for games
- View game details
- Play Game
- Access through desktop client
- Access through web client
- Scores
- Help

Search for games:

The user can search the games hosted by the service. The user can select the category of games available such as Sports, Puzzles, Action, Others. The result will be the selected game details.

Viewing the game details:

User can select one from the displayed result. User can view the description of the game, which user selected.

Play game:

- **Single user:**

If it is a single user game, user can start playing the game directly.

- **Multi-user:**

If it is a multi-user game the user should select an opponent based on the two statuses 'Available', and 'Playing'. User can send a request to another user to play a multi-user game.

User who receive request message from other user could either accept or reject. If User accept the request both user can start to play the game .If user reject the request user can select another user.

Access through the desktop client:

The desktop clients can access the application directly to the gaming server. The desktop client can access the game through the win forms.

Access through the web client:

The web client can access the application from any system and Play the game. This service is being brought by the web services.

Scores:

When the game gets over, the users can view the score taken by them. The time is also displayed for the users of the game.

Help:

The help pages are available to the users of the game, which will contain details about the game and current page. It also tells the user about how the game can be played.

Game:

In Game module the game named **'Memory Blocks'** is developed. This module has information about how to play the game and the controls used.

Development of the Game:

The features of the game that is included is given as

- Pause
- Stop
- Controls
- Rules
- Scores

Rules of memory block game:

- The tiles are arranged in a matrix format. Flip over any two tiles by clicking on them.
- If the pictures of the tiles are identical then they will disappear. If they are not identical they will be flipped back to the picture-side-down.
- Number of attempts made will be taken into account.
- Continue flipping pairs until all matches have been found and there are more titles.
- Scores are made by two ways. First, by the number of seconds it takes for you to find all of the matches. Second, by the total number of pairs you need to flip in order to find all of the matches.
- The top score are displayed on the high-score list.

3.1.4 Input Design

Input design is the process of converting user-originated inputs to computer-based format. Input data are collected and organized into groups of similar data. Validations are made for each and every data entered in the screens for data accuracy.

The entire project is implemented using ASP.NET. The information such as User Id and Password are given in the input screen, which is verified for correctness and then all the processes are done. All users are registered initially and then logged in. If the input fails, then the user should retry it.

On each click of the controls used the respective screens are opened. Each screen has Textboxes, Label and Buttons. All the screens are interactive with the user in accepting and displaying data requested by the user.

All the data entered are recorded in the respective database table, specified in the SQL statement and stated in the executing system. Each time the data is entered, verifications are done to ensure no errors are committed and to confirm its accuracy.

The first step in design is to design input within predefined guidelines. Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design.

The following forms are used in the project (refer Appendix 3)

- Form 3.1 - Login, is the Login page of the Online Gaming System. This form shows process involved with new user and login existing user.
- Form 3.2 - Registration form, allows the process of registering the details of the new user.
- Form 3.3 - Home page form, allows navigation to various game details and options for searching the new games.
- Form 3.4 - Search page form, allows to search the games and displays the results.
- Form 3.5 – Game page form, allows to play the blocks game and this page also displays the score details, attempts, time made to complete the game.

3.1.5 Output Design

Computer output is the most important and direct source of information to the user. Efficient and intelligent output design should improve the system's relationships with the user and help in decision making. A major form of output is a hard copy from the printer.

Printouts should be designed around the output requirements of the user. The output devices are considered depending on factors such as

compatibility of the device with the system, response time requirements, expected print quality, and number of copies needed.

The types of output used in the system are internal output, interactive output, turn-around output and data item. The internal outputs are those, whose destination are within the organization and are the user's main interface with the Computer.

The standards for printed output suggest the following:

- Give each output a specific name or title.
- Provide a sample of the output layout, including areas where printing may appear and the location of each field.
- State whether each output field is to include significant zeroes, spaces between fields and alphabetic or any other data.
- Specify the procedure for providing the accuracy of output data.

The output design focuses on to serve the intended purposes, to fit the user, deliver the appropriate quality of output. Based on the need and requirements of the various departments, the outputs were designed with much care and consideration.

3.1.6 Database Design

A database is a collection of inter-related data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of database design is to make the data access easy, inexpensive and flexible to the user. An elegantly designed database can play a strong foundation for the whole system. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated as whole. Database Management System allows data to be protected and organized separately from other resources.

The details about the relevant data for the system are first identified. According to their relationship, tables are designed through the following method.

- The data type for each data item in the table is decided.
- The tables are then normalized.

The tables are normalized so that they can provide better response time, have data integrity, avoid redundancy and be secure. Database management system provides a centralized access to the data from the programs. The main objectives of database design are data integration and data independence.

The tables for the Result Analysis system have been Normalized up to the Second Normal Form (2NF). A store of integrated data capable of being directly addressable for multiple users is database. It is organized so that various files can be accessed through a single reference based on the relationship among the records in the file rather than the physical location.

3.1.7 Process Design

In 'Service Oriented Architecture for Online Gaming Service', the user initially logs in to get connected to the Gaming Server. The new user initially registers by giving all the necessary details of the user. The existing user can give the user_id and Password for authentication and then logs in.

In the *Homepage*, the user can choose the game he wishes to play. The name of the game can be entered by the user in the Textbox provided in the form. The user can also select the game by choosing the category. From the displayed game list in the *Data grid* based on a particular category, the user can choose the game.

If he selects single user game he can play game directly. If it is two users he should select their opponents and play the game. The user can select the opponents by sending a 'Request message' to any of the available users.

The user can either reject or accept this Request message received from other users. The user when accepts the message from other user, both the users can start playing the game and they are not available to other users as opponents.

If an opponent rejects the Request message, then the user can send a similar Request message to other users in the same way and choose their opponents.

The description about each and every game will be given. The *Help* option is provided in all pages. The rules of the game are also given. The controls used are specified. The user when completes playing the game, can view the score.

p-1915

CHAPTER 4

4. SYSTEM TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

Testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of any system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancement of the system under consideration. The testing phase conducts test using the Software Requirement Specification as a reference and with the goal to see whether the system satisfies the specified requirements.

Standard procedures have been followed in testing Result Analysis. Test cases are generated for each screen. These test cases will cover every possibility which could result in both positive and negative results. These test plans are maintained for any further testing done on the system. The test plan stores information such as, the test script/input, expected output, actual output, comments and the name of the tester. This plan will be followed for all types of testing done in the system.

The main types of tests carried out are:

- Unit Test
- Integration Test
- System Test

4.1.1 Unit Testing

Module or Unit Testing is the process of testing all the program units that make up a system. Unit testing focuses on an individual module thus allowing one to uncover all the errors made logically and while coding in the module.

Each page is tested separately as a unit. Initially the flow of control and data through that page is checked. When considering a module as a unit, the flow of data and control through the whole module is tested. The result is stored in the test plan. In a page, each control is further tested in unit testing. The process is done in all the pages of the system. Once the errors are rectified, the testing procedure is repeated with same test cases to ensure this hasn't produced new errors. Hence this is a continuous process.

Test cases were generated to test the control flow of each unit or module. Almost all cases needed for testing control flows have been generated.

Test Cases for the Login Screen:-

Sr.No	Test Case	Expected Result	Observed Result	Status
1	User id : G100, Password: XYZ	Invalid Login	Invalid Login	Pass
2	User id : G100, Password: server	Login successful	Login Successful	Pass
3	User id : Xyz, Password: abc	Invalid Login	Invalid Login	Pass

4.1.2 Integration Testing

Integration testing tests the process of integrating the various modules to form the completed system. Integration starts with a set of units each individually tested in isolation and ends when the entire application has been built. Integration testing verifies that the combined units function together correctly. It facilitates in finding problem that occur at interface or communication between the individual parts.

In this project top-down integration testing is followed. Modules were linked to the main menu in a sequence as required in the real time operating mode of the system. Menu items were created as and when required for the integration. The same procedure is followed in other modules in the same level at first. Then the upper level is taken into action. The flow of data through the whole module in the upper level is taken and executed. A change of data made in one screen should have reflected in all other screens.

This process is continued from the page level to module level, finally to the system level. In the final stage, the whole system is taken together and tested for integration. A change in one place should be reflected through out the system. Regression testing is done after each change made into the software. This tests if the change has affected any part negatively after the change was made. The whole set of test cases need to be run again to do the regression testing. Data can be lost across the interfaces, one module may have adverse effect on other. Thus integration testing is a systematic testing for constructing tests to uncover errors within the interface.

In this project each and every module are combined and the program is tested as the whole. Integration testing is for testing the design and construction of the software architecture.

4.1.3 System Testing

System testing is actually a serious of different tests, whose primary purpose is to fully exercise the computer-based system. This helps in verifying that all the system elements have been properly integrated and perform the allocated functions. It verifies the entire product after having integrated all software and hardware components, and validates it according to the original project requirement. The system testing takes into consideration the hardware, and the software. It should be able to be run on the specified hardware for variety of cases. The project is tested against recovery from errors.

4.1.3.1 Security Testing

Security testing is important in system testing. The system in no way shall be accessible to unauthorized users. Testing is done to ensure that a user with respective rights can only view the various forms and reports presented by the system. If users try to perform something beyond his assigned rights corresponding messages should be displayed. In such cases it redirects the user back to the previous page.

Another security issue involves the sensitive data in the system. The system is highly secure with authentication fixed at various levels of the hierarchy. One more level of security is concerned with user rights. Each user is applied rights module wise. The menus can be configured to roles. Users can also be configured to roles. Menu items are assigned to users dynamically based on the roles assigned to menu items as well as users. A match is done before displaying the menu to the user.

4.1.3.2 Stress Testing

Stress Testing executes a system in a manner that demands resources in abnormal quantity, frequency or volume. The project was stress tested in the company by having users simultaneous access to various modules in the system.

4.1.3.3 Regression Testing

The regression testing is next level of testing. This testing involves adding new features to the existing system and then testing them. Addition of new features would bring many new errors. So once a new feature is added, it is checked for regression test. This testing is conducted at the final level after completing all other tests.

4.1.3.4 Validation Testing

Validation testing is where requirements established as part of software requirements analysis is validated against the software that has been constructed. The validation should be identified and tested. This test provides the final assurance that the software meets all the functional, behavioral and performance requirements. The errors which, are uncovered during integration testing, are corrected during this phase.

4.1.3.5 Output Testing

No system could be useful if it does not produce the required output in the specific format. The outputs generated or displayed by the system under consideration are tested along the users about the format required by them. Output testing does not result in any correction in the system.

4.1.3.6 User Acceptance Testing

User acceptance of the system is the key factor of success. The system under consideration is tested for user acceptance by constantly, keeping in touch with the prospective system users at time of developing and making changes whatever required. The input output screen design, online messages to guide the user, menu driven system format of reports are tested.

4.2 SYSTEM IMPLEMENTATION

System Implementation is the part of the software engineering life cycle, where, the design artifacts are converted to a working application. Coding is done in this stage using ASP.NET framework and programming language, which would solve the specific problem the best way. Once the design is coded into a working application, it has to be verified, validated and tested in detail. The tested product if successful is deployed in the user environment.

Apart from the knowledge base maintained in the SQL database, the system also houses SOA knowledge base and that of Web Services. Some of the knowledge and criteria are presented directly in the front-end application, which is ASP.NET in this case.

The classes made use in this application have been included in the Business Logic, which is Web Services in this case. The connection is made to the back-end for data access.

The Desktop client / Web client can access the Web Services directly. The Web Services has functionalities that can access the Gaming Server. The Gaming Server can access the Database.

The concept of this base has been introduced having observed the fact that, it shares, retrieves and manipulate data or knowledge pertaining in form of a relational database.

4.2.1 System Verification

System Verification answers the question "Am I building the product right?" It includes the review of interim work steps and interim deliverables during a project to ensure they are acceptable. Verification also determines if the system is consistent, adheres to standards, uses reliable techniques and prudent practices, and performs the selected functions in the correct manner. In data access, it verifies whether the right data is being accessed, in terms of the right place and in the right way.

For e.g., the drop downs gather data from the database, so each dropdowns should be verified whether they are bound to the correct database field. It is done during development of the key artifacts. Verification is a demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle. In this project, verification is done during the development itself. Each database bindings are verified after binding to test whether the control is bound to the right data field.

4.2.2 System Validation

Validation answers the question "Am I building the right product?" This checks whether the developer is moving towards the right product, whether the development is moving towards the actual intended product that was agreed upon in the beginning. Validation also determines if the system complies with the requirements and performs functions for which it is intended and meets the organization's goals and user needs. It is traditional and is performed at the end of the project. In data access, it checks whether we are accessing the right data, in terms of data required to satisfy the requirement.

Validation is performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment. It determines the correctness of the final software product by a development project with respect to the user needs and requirements.

Functional validation is done in the Online Gaming Service to check whether each of the functions are done correctly as expected in every page. Each control in a Screen is designed to do some function. These functions are checked against the requirements stated for them. For e.g., clicking 'Save' button should take the corresponding action of saving the details into the database. Clicking the Edit icon should allow one to edit the contents that are being currently displayed. This level of validation can continue to all the controls in the system. This checking is usually done after the system is developed so that all activities that are affected can be checked.

Field level validation is done in Online Gaming Service to check whether each of the fields either accepts the data as expected and do the client side validation of data entered. For e.g. a field level validation on a text box would check against the type of data entered and follow rules such as length of entry etc. The data type validation checks are conducted after the form is submitted. It takes place in the Action Form class of the struts framework. If the validation check fails then the processing stops and the control returns back to the original form that was submitted.

The validation is done in a step by step process. First the screen is loaded with the controls. When the user moves between controls on the screen, the validation events for the control that lost the focus are fired and appropriate error messages (if any) are displayed. If the user generates a form save request, the entire form is evaluated for any validation controls that are not valid. If even one control is not valid, the form will not be submitted.

CHAPTER 5

5. CONCLUSION AND FUTURE ENHANCEMENTS

5.1 CONCLUSION

The 'Service Oriented Architecture for Online Gaming Service' is designed with web service, which helps in implement Service Oriented Architecture (SOA) in this project. Using this we can change Backend(C#) without affecting the user interface. The changes are done in Web Method, no other changes in user interface. New type of games can be created and added to the existing game list easily. Many of the current available features are extensible.

The system is tested with sample data and was found to be working efficiently. Sample data are tested for invalid Login and the updating of records. The developed system is flexible and changes can be made easily as and when required. The system is efficient so as to facilitate the interactions of customers.

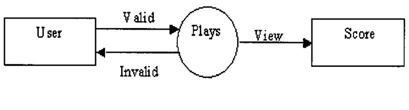
5.2 FUTHER ENCHANCEMENTS

The Gaming Service can be still improved and extended in the future by including chatting between two opponents while playing. In this they can send message between them while playing. It can be developed for both desktop client and web client. AJAX can be included in this project to help chat and for game page. This will help in refreshing the particular component without refreshing the whole page.

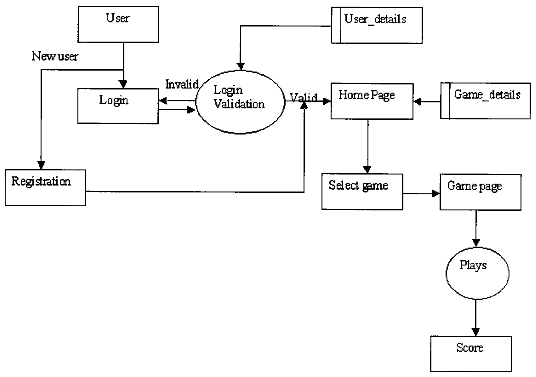
APPENDIX 1

DATAFLOW DIAGRAMS

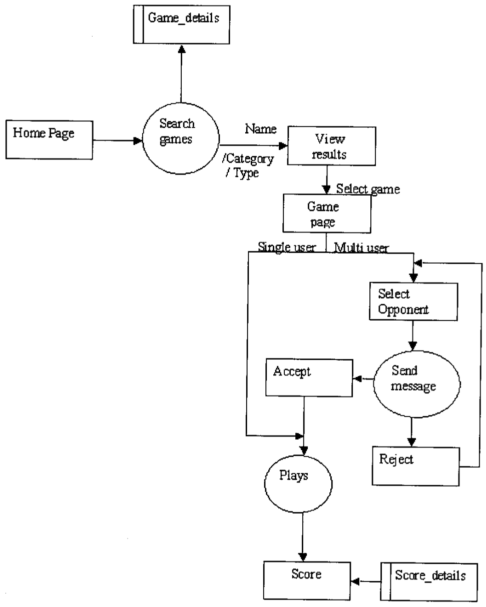
1.1 LEVEL 0



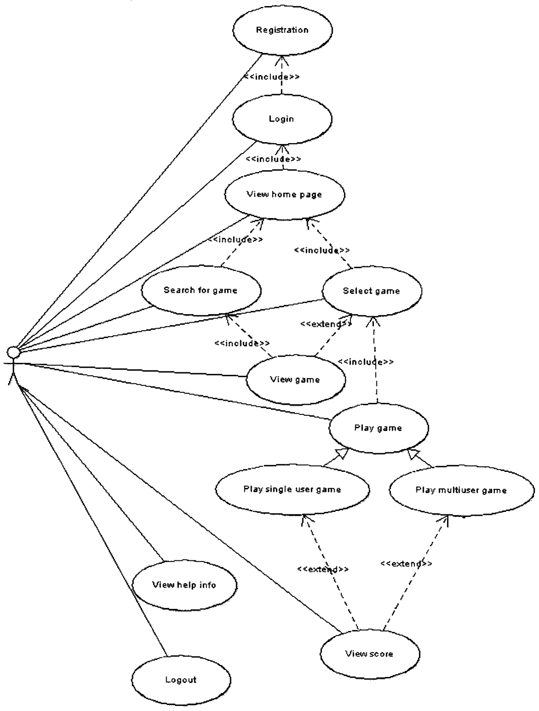
1.2 LEVEL 1

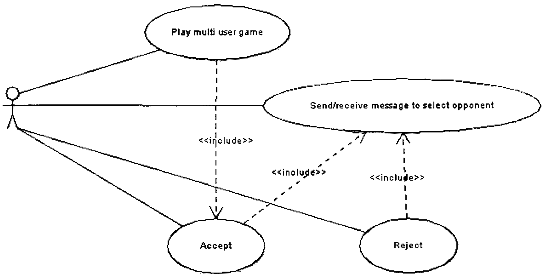


1.3 LEVEL 1.1

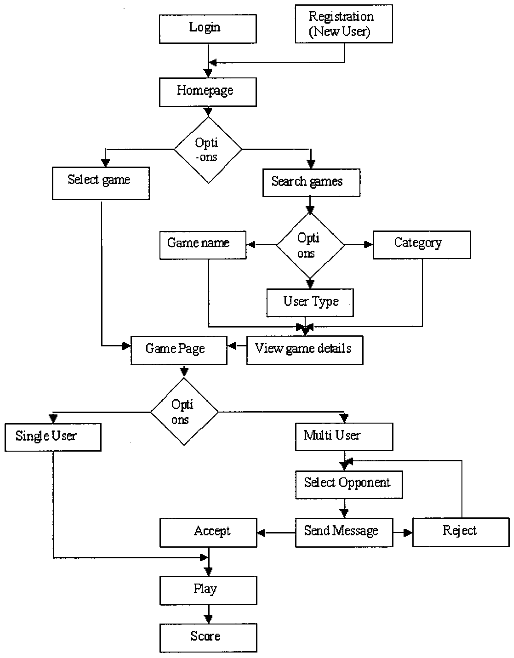


USE-CASE DIAGRAM





SYSTEM FLOW DIAGRAM



APPENDIX 2

TABLES

2.1 Table Name : User Details

Description : This table describes about the details of the new user

Column Name	Data Type	Length	Constraint
User_Id	Varchar	10	Primary key
First_Name	Varchar	10	
Last_Name	Varchar	10	
Password	Varchar	10	
E-mail_Id	Varchar	40	
DOB	Date	8	
Address	Varchar	100	
Gender	Varchar	8	

2.2 Table Name : Score Details

Description : This table describes about the user score

Column Name	Data Type	Length	Constraint
User_Id	Varchar	10	Foreign key
Game_Id	Varchar	10	Foreign key
Score	Numeric	10	

2.3 Table Name : Game Details

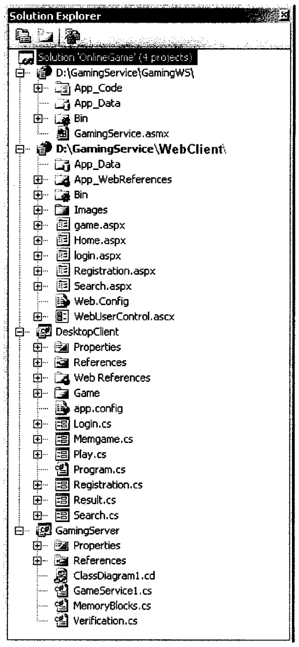
Description : This table describes about the details of the games that are available

Column Name	Data Type	Length	Constraint
Game_Id	Varchar	10	Primary Key
Game_Name	Varchar	20	
Category	Varchar	20	

APPENDIX 3

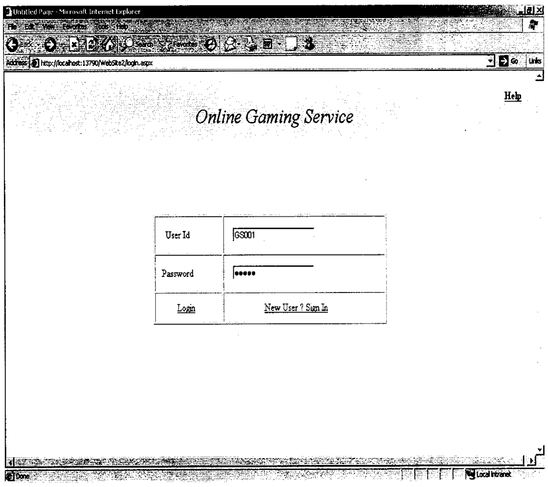
FORMS

Abstract view of the project organization



```
Solution Explorer
Solution 'OnlineGame' (4 projects)
D:\GamingService\GamingWS\
    App_Code
    App_Data
    Bin
    GamingService.asmx
D:\GamingService\WebClient\
    App_Data
    App_WebReferences
    Bin
    Images
    game.aspx
    Home.aspx
    login.aspx
    Registration.aspx
    Search.aspx
    Web.Config
    WebUserControl.ascx
DesktopClient
    Properties
    References
    Web References
    Game
    app.config
    Login.cs
    Memgame.cs
    Play.cs
    Program.cs
    Registration.cs
    Result.cs
    Search.cs
GamingServer
    Properties
    References
    ClassDiagram1.cd
    GameService1.cs
    MemoryBlocks.cs
    Verification.cs
```

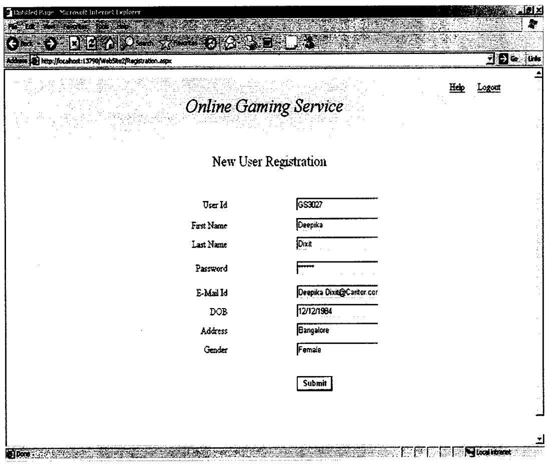
3.1 LOGIN PAGE


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:13790/WebSite2/login.aspx
Go Links
Help
Done
Local intranet

Online Gaming Service

User Id	GS001
Password	●●●●●
Login	New User ? Sign In

3.2 NEW USER REGISTRATION



Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Done
Local intranet

http://localhost:13790/WebSite2/Registration.aspx

Help Logout

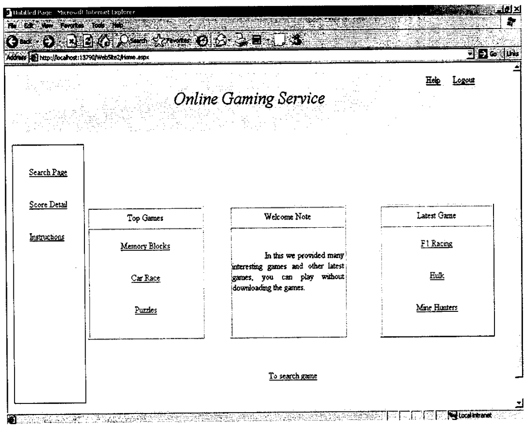
Online Gaming Service

New User Registration

User Id	GS3027
First Name	Deepika
Last Name	Dixit
Password	******
E-Mail Id	Deepika Dixit@Cantor.cor
DOB	12/12/1984
Address	Bangalore
Gender	Female

Submit

3.3 HOME PAGE



Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Local intranet

http://localhost:13790/WebSite2/Home.aspx

Help Logout

Online Gaming Service

Search Page

Score Detail

Instructions

Top Games

Memory Blocks

Car Race

Puzzles

Welcome Note

In this we provided many interesting games and other latest games, you can play without downloading the games.

Latest Game

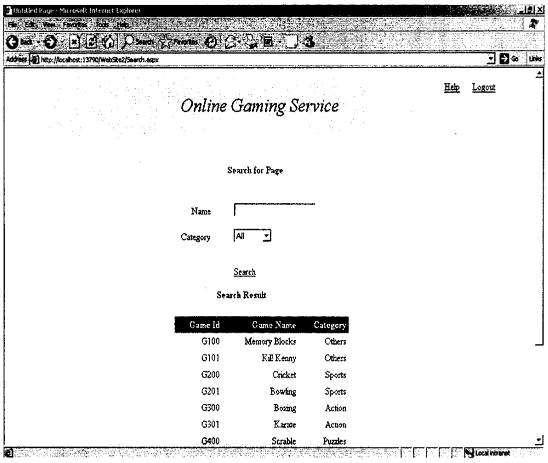
F1 Racing

Hulk

Mine Hunters

To search game

3.4 SEARCH PAGE



Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Local intranet

http://localhost:13790/WebSite2/Search.aspx

Help Logout

Online Gaming Service

Search for Page

Name

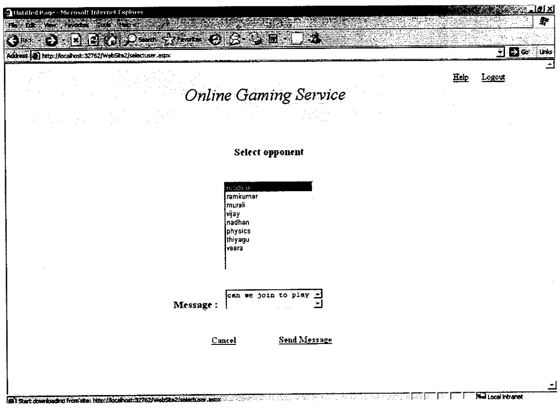
Category All

Search

Search Result

Game Id	Game Name	Category
G100	Memory Blocks	Others
G101	Kill Kenny	Others
G200	Cricket	Sports
G201	Bowling	Sports
G300	Boxing	Action
G301	Karate	Action
G400	Scrable	Puzzles

3.5 SELECT OPPONENT



Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Local intranet

http://localhost:32762/WebSite2/selectuser.aspx

Help Logout

Online Gaming Service

Select opponent

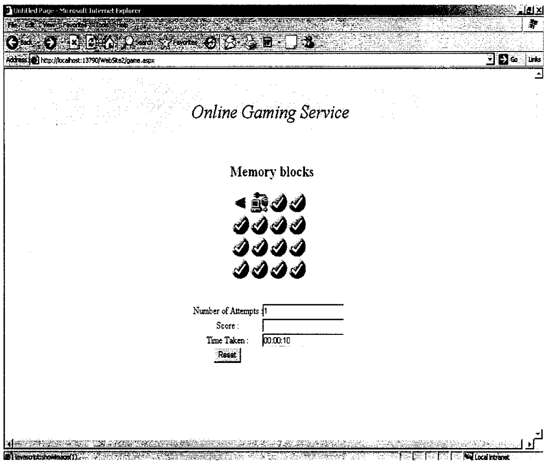
madhan
ramkumar
murali
vijay
nadhan
physics
thiyagu
veera

Message : can we join to play

Cancel Send Message

Start downloading from site: http://localhost:32762/WebSite2/selectuser.aspx

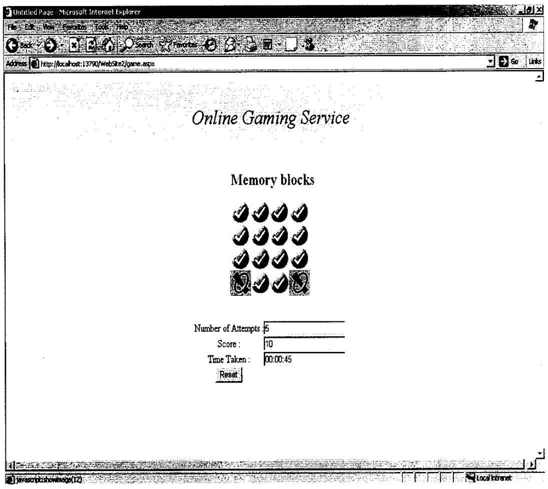
3.6 GAME PAGE


Untitled Page - Microsoft Internet Explorer
http://localhost:13790/WebSite2/game.aspx
Memory blocks
Reset
javascript:showimage(1)
Local intranet

Online Gaming Service

Number of Attempts : 1
Score :
Time Taken : 00:00:10

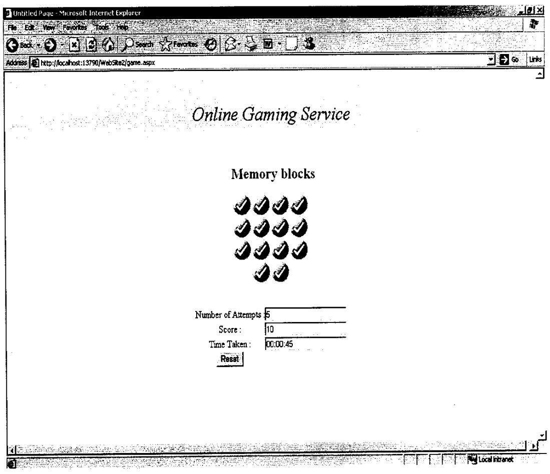
3.7 GAME PAGE


Untitled Page - Microsoft Internet Explorer
http://localhost:13790/WebSite2/game.aspx
Memory blocks
Reset
javascript:showimage(12)
Local intranet

Online Gaming Service

Number of Attempts : 5
Score : 10
Time Taken : 00:00:45

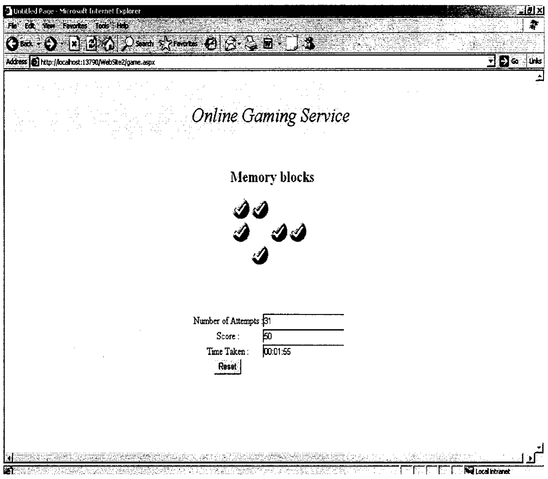
3.8 GAME PAGE


Untitled Page - Microsoft Internet Explorer
http://localhost:13790/WebSite2/game.aspx
Memory blocks
Reset
Local intranet

Online Gaming Service

Number of Attempts : 5
Score : 10
Time Taken : 00:00:45

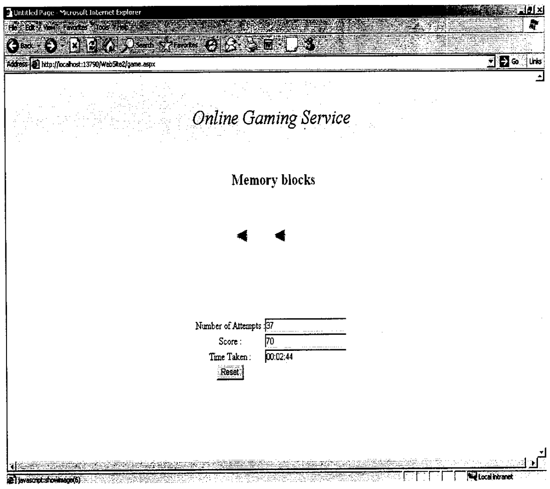
3.9 GAME PAGE


Untitled Page - Microsoft Internet Explorer
http://localhost:13790/WebSite2/game.aspx
Memory blocks
Reset
Local intranet

Online Gaming Service

Number of Attempts : 31
Score : 50
Time Taken : 00:01:55

3.10 GAME PAGE


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:13790/WebSite2/game.aspx
Go Links
javascript:showimage(6)
Local intranet

Online Gaming Service

Memory blocks

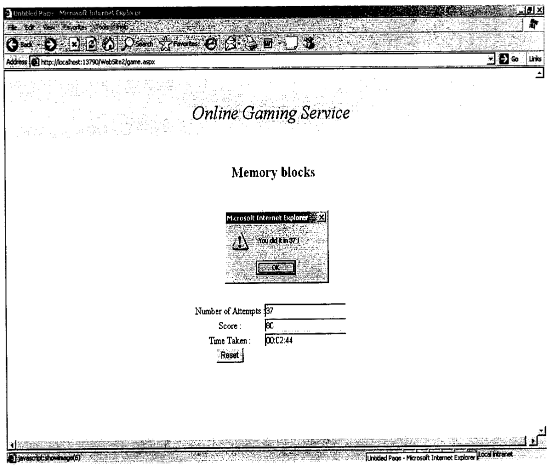
Number of Attempts : 37

Score : 70

Time Taken : 00:02:44

Reset

3.11 GAME PAGE


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:13790/WebSite2/game.aspx
Go Links
javascript:showimage(6)
Untitled Page - Microsoft Internet Explorer
Local intranet

Online Gaming Service

Memory blocks

Microsoft Internet Explorer

You did it in 37 !

OK

Number of Attempts : 37

Score : 80

Time Taken : 00:02:44

Reset

BIBLIOGRAPHY

Reference Books:

1. Mathew Macdonald, Mario Szpuszta (2004), 'ASP.NET 2.0 IN C#2005 APRESS-2005', Tata McGraw Hill Publications.

2. Danny Goodman, Michael Morrison (2003),'JAVA SCRIPT BIBLE ' WILEY PUBLISHING, INC

Reference Sites:

1. www.informit.com
2. http://msdn.microsoft.com
3. www.ondotnet.com
4. http://safari.phptr.com
5. www.w3org/tr/wsdl