CHAPTER - 1
INTRODUCTION

1.1 OVERVIEW:

The purpose of this document is to define scope and requirements of an Online Collection Combinations – OCC for a leading consultancy who wanted to empower its consultants in structured decision-making. Currently the employees waste significant time to getting the different combinations of the collections and making as meaningful decisions. As a first step, the proposed system – OCC will provide an online structured assistance to quickly prioritize items such as actionable items, initiatives, or to-dos. This document is the primary input to the development team to architect a solution for this project.

1.2 ABOUT THE PROJECT:

OCC will provide a web-based formal tool for consultants to quickly prioritize items such as actionable items, initiatives, or to-dos using a structured pair-wise comparison technique. It is a technique used in decision-making process to rank or prioritize items on a relative scale of importance against a criterion. Let us assume that there are “n” items to be ranked. In this technique, all unique pairs of “n” items are formed. Note there will be nC2 pairs for “n” items. The user is presented each pair sequentially and asked to select (against the criterion) his/her “preferred item” from the pair. Once all pairs have been evaluated, it is very easy to score how many times an item was selected as “preferred”. At this stage, each item presented in the descending order of its score – the ranked list! The technique works well for a small list of items typically less than 10.For example if we give five tasks for a day then the comparisons are done in between the five tasks and produced ten types of combinations in those combinations we have to give the priority among one of the comparison like the way we have chosen all the combinations.

1.3 PURPOSE:

The purpose of this document is to define scope and requirements of an Online Collection Combinations – OCC for a leading consultancy who wanted to empower its consultants in structured decision-making.

1.4 FUNCTIONAL COMPONENTS OF THE PROJECT:

Following are the functional needs of the software:-

1. Designer and consultant must have a valid user ID and password to login to the system.

2. After a valid log in, the system shows the upload design page to the designer and view designs and rate page to the consultant.

3. Designer can add design and the consultant can rate the design.

4. The addition of designers and consultants is managed by the admin.

CHAPTER - 2
SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

On the basis of mathematical formula of combinations, we have to get the pair of combinations among the given number of items, here item in the sense the number of tasks. Early days, we have done the comparisons on the basis of the following formula nCr = n!/(r! * (n-r)!). It's very typical tasks for the users to get the pair of combinations manually. For the sake of simplification we have to use technology like java and j2ee. By using these technologies we have done the combinations programmatically for the user convenience.

2.2 PROPOSED SYSTEM:

It is a technique used in decision-making process to rank or prioritize items on a relative scale of importance against a criterion. Let us assume that there are "n" items to be ranked. In this technique, all unique pairs of "n" items are formed. Note there will be n C2 pairs for "n" items. The user is presented with each pair sequentially and asked to select (against the criterion) his/her "preferred item" from the pair. The system operation is outlined below:

1. OCC accepts the list of items from the user; editing is allowed at this stage. The list can be saved as a "draft" for later use.

2. User can click on "start comparison" button on the "item list" page to start the pair-wise comparison process. OCC automatically forms the pairs at the backend, and presents each pair for user's preference selection.

3. Once the all the pairs are exhausted, OCC automatically computes the score and presents the ranked list. Such a list is automatically saved for later reference. OCC will maintain a view of all such lists for later reference for each user. Upon login, the user is directed to his/her lists view, where she/he either works on an existing list or create a new list from scratch.

2.3 BENEFITS OF THE SYSTEM:

- Quick and authenticated access to designs/combinations via Desktop.
- Easy ranking method.

- Suitable for quick decision making.
- Minimize Storage Space.

CHAPTER - 3
FEASIBILITY REPORT

3.1 UNDERSTANDING FEASIBILITY:

Feasibility study means the analysis of problem to determine if it can be solved effectively. In other words it is the study of the possibilities of the proposed system it studies the work ability, impact on the organization ability to meet user's need and efficient use of resources.
Three aspects in which the system has to be feasible are:-

3.1.1 ECONOMICAL FEASIBILITY:

The economic analysis checks for the high investment incurred on the system. It evaluates development &Implementing charges for the proposed "Online Collection Combinations". The Software used for the development is easily available at minimal cost & the database applied is freely available hence it results in low cost implementation.

3.1.2 TECHNICAL FEASIBILITY:

This aspect concentrates on the concept of using Computer Meaning, "Mechanization" of human works. Thus the automated solution leads to the need for a technical feasibility study.
The focus on the platform used database management and users for that Software.
The proposed system doesn't require an in depth technical knowledge as the system development is simple and easy to understand. The Software used makes the system user friendly (GUI).

3.1.3 BEHAVIOURAL FEASIBILITY:

Behavioral feasibility deals with the runtime performance of the Software, the proposed system must score higher than the present in the behavioral study. The Software should have end user in mind when the system is designed while designing Software the programmer should be aware of the conditions user's Knowledge input, output, calculations etc.

CHAPTER - 4
SOFTWARE REQUIREMENT SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS:

Processor	:	Intel Core i3 and above.
RAM	:	2GB and above.
Hard disk	:	250 GB and above.

4.2 SOFTWARE REQUIREMENTS:

Operating System	:	Any Graphical OS.
Language	:	JAVA & HTML.
Scripting	:	JavaScript.
Web Server	:	Apache Tomcat.
RDBMS/Back End	:	SQL-Server/Oracle.
Back-End Software	:	JSP.

4.3 JAVA TECHNOLOGY:

Java technology is both a programming language and a platform.

4.3.1 THE JAVA PROGRAMMING LANGUAGE:

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

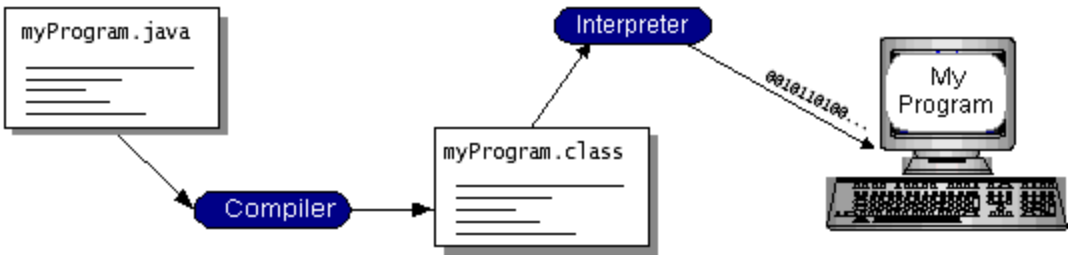


Figure 4.3.1.1- Working of Java

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (JVM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the JVM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. That means that as long as a computer has a JVM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

4.3.2 THE JAVA PLATFORM:

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

The Java Virtual Machine (JVM)

The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

The next section, **What Can Java Technology Do?**, highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

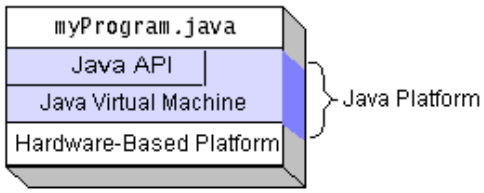


Figure 4.3.2.1- The Java Platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

4.4 INTRODUCTION TO JSP:

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP, but it uses the Java programming language.

To deploy and run Java Server Pages, a compatible web server with a Servlet container, such as Apache Tomcat or Jetty, is required.

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP servlet is cached and re-used until the original JSP is modified.

JSP can be used independently or as the view component of a server-side Model–View–Controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture (Figure 4.4.1).

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java byte code rather than machine code. Like any other Java program, they must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well.

The Web container creates JSP implicit objects like pageContext, servletContext, session, request & response.

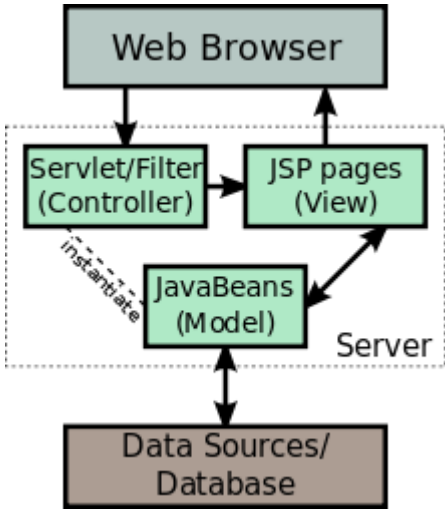


Figure 4.4.1- JSP Model 2-Architecture

4.5 HTML:

Web pages are written in HTML - a simple scripting language.

HTML is short for Hyper Text Markup Language.

- Hypertext is simply a piece of text that works as a link.
- Markup Language is a way of writing layout information within documents.

Basically an HTML document is a plain text file that contains text and nothing else. When a browser opens an HTML file, the browser will look for HTML codes in the text and use them to change the layout, insert images, or create links to other pages. Since HTML documents are just text files they can be written in even the simplest text editor. A more popular choice is to use a special HTML editor - maybe even one that puts focus on the visual result rather than the codes - a so-called WYSIWYG editor ("What You See Is What You Get").

Some of the most popular HTML editors, such as FrontPage or Dreamweaver will let you create pages more or less as you write documents in Word or whatever text editor you're using. However, there are some very good reasons to create your own pages - or parts of them - by hand. It is possible to create Web pages without knowing anything about the HTML source behind the page. There are excellent editors on the market that will take care of the HTML parts. All you need to do is layout the page.

4.5.1 HTML TAGS:

- HTML tags are used to mark-up HTML elements.
- HTML tags are surrounded by the two characters < and > are called angle brackets, the first tag in a pair is the start tag and the second tag is the end tag.
- The text between the start and end tags is the element content.
- HTML tags are not case sensitive; <b> means the same as <B>.

4.6 JAVA SCRIPT:

JavaScript is a scripting language that will allow you to add real programming to your Web pages. You can create small application type processes with JavaScript, like a calculator or a primitive game of some sort.

However, there are more serious uses for JavaScript:

- **Browser Detection:**
  Detecting the browser used by a visitor at your page. Depending on the browser, another page specifically designed for that browser can then be loaded.
- **Cookies:**
  Storing information on the visitor's computer, then retrieving this information automatically next time the user visits your page. This technique is called "cookies".
- **Control Browsers**:
  Opening pages in customized windows, where you specify if the browser's buttons, menu line, status line or whatever should be present.
- **Validating Forms:**
  Validating inputs to fields before submitting a form.
  An example would be validating the entered email address to see if it has a @ in it, since if not, it's not a valid address.

4.6 JAVA SERVER PAGES

Introduction:

Jsp technology enables you to mix regular static html with dynamically generated content from servlets. Separating the static html from the dynamic content provides a number of benefits over servlets alone.

Why use JSP:

Jsp is easy to learn and allows developers to quickly produce wed sites and application in an open and standard way. Jsp is based on java, an object-oriented language. Jsp offers a robust platform for web development.

Main reasons to Jsp:

a. Multi-platform
b. Component reuse by using java beans and Ejb
c. Advantages if java

We can take one Jsp file and move it to another platform, web server or Jsp servlet engine.

JSP compared to ASP:

Jsp and Asp are fairly similar in the functionality that they provide. Jsp may have slightly higher learning curve. Both allow embedded code in an html page, session variables Platform i.e., NT, JSP can operate on any platform that conforms to the J2EE specification. Jsp allow component reuse by using JavaBeans and Ejbs. Asp provides the use of Com/activeX controls.

JSP compared to servlets:

A servlet is java class that provides special server side service. It is hard to write HTML code in servlets. In servlets you need to have lots of println statement to generate HTML.

Description:

JSP looks like html, but they get compiled into java servlets the first time they are invoked. The resulting servlet is a combination of the html from the jsp file and embedded dynamic content specified by the new tags. That is not to say that jsp must contain html. Some of them contain only java code; this is particularly useful when the jsp is responsible for a particular task like maintaining application flow.

Everything in Jsp can be broken into 2 categories:

a. Elements that are processed on the server.
b. Template data or everything other than elements that the engine processing the Jsp ignores.

A Jsp page is executed by a Jsp engine or container, which is installed on a web server, or on an application server. When the client asks for a jsp resource the engine wraps up that request and response object to incorporate the communication with the client. The container then wraps up the underlying layer for a jsp that of a servlet implementation. The abstractions of the request

and response are the same as javax.servlet.ServletRequest and javax.servlet.ServletReesponse respectively.

JSP Architecture:

Jsp are built on top of sun's servlet technology. Jsp is essentially an html page with special jsp tags embedded. These jsp tags can contain java code. The jsp file extension is .jsp rather than .htm or .html. The jsp engine parses the .jsp and creates a java servlet source file. It then compiles the source file into a class file; this is done the first time and this why the jsp is probably slower the first time it is accessed. Any time after this, the special compiled servlet is executed and is therefore returns faster.

Steps required to JSP request:

a. The user goes to a web site made using jsp. The user goes to a jsp page. The web browser makes the request via the internet.
b. The jsp request gets sent to the web server
c. The web server recognizes that the file required is special (.jsp), therefore passes the jsp file to the jsp servlet engine.
d. If the jsp file has been called the first time, the jsp file is parsed, otherwise go to step 7
e. The next step is to generate a special servlet form the jsp file. The entire html required is converted to println statements.
f. The servlet source code is compiled into a class
g. The servlet is instantiated, calling the init and service methods
h. Html from the servlet output is sent via the internet
i. Html results are displayed on the user's web browser.

Servlets are server-side java programs that can be deployed on a web server.

The servlet interface provides the basic frame work for coding servlets.

Java Server Pages, SQL, HTML Forms and Databases

This section examines how to communicate with a database from Java. We have already seen how to interface an HTML Form and a JSP. Now e has to see how that JSP can talk to a database. The objectives of this section are to understand how to:

- Administratively register databases.
- Connect a JSP to an Access database.
- Insert records in a database using JSP.
- Inserting data from HTML Form in a database using JSP.
- Delete Records from Database based on Criteria from HTML Form.
- Retrieve data from a database using JSP – result sets.
- Apply SQL operations like sort, create table, remove table, delete, and Access-based arithmetic functions.

JSP Declarations:

Used to define page level variables and methods are placed with in the <%!and %> tags and always end with a semicolon.

Example:

```
<%!
    Int I=0;
    Int j=0;
    Int z=0 ;%>
```

JSP Scriptlets:

Consists of valid code snippets enclosed with in the <% and %> JSP tags.

Example:

To accept the user name and display the name 10 times:

```
<%@ page import="java.util.*" %>
<%@ page import="java" %>
```

```
<HTML><BODY>
  <%out.println (“<HTML>”);
      out.println (“<BODY>”);
      out.println (“<BODY>”);
      out.println (“<HTML>”);%>
</BODY></HTML>
```

CHAPTER - 5
SYSTEM DESIGN

5.1 INTRODUCTION:

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

5.1.1 3-TIER ARCHITECTURE WEB APPLICATION:

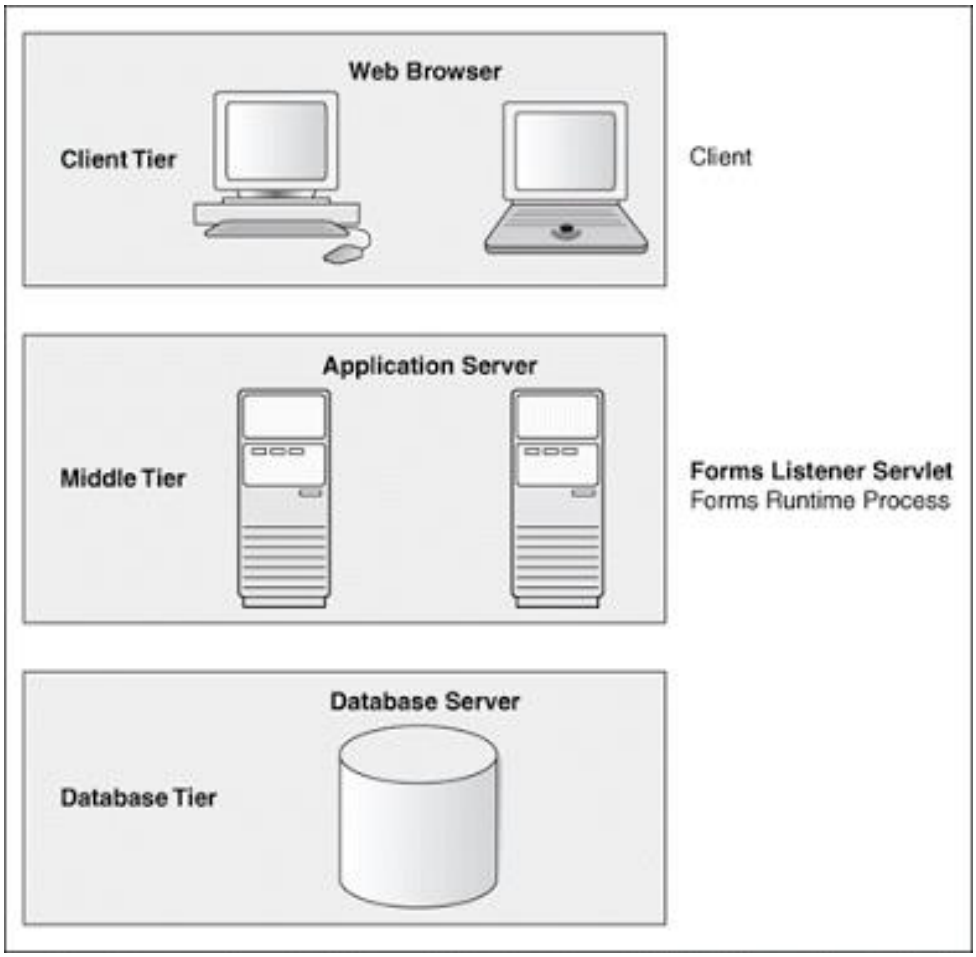


Figure 5.1.1.1 - 3-Tier Architecture

5.1.2 PHYSICAL CONTROL FLOW ARCHITECTURE:

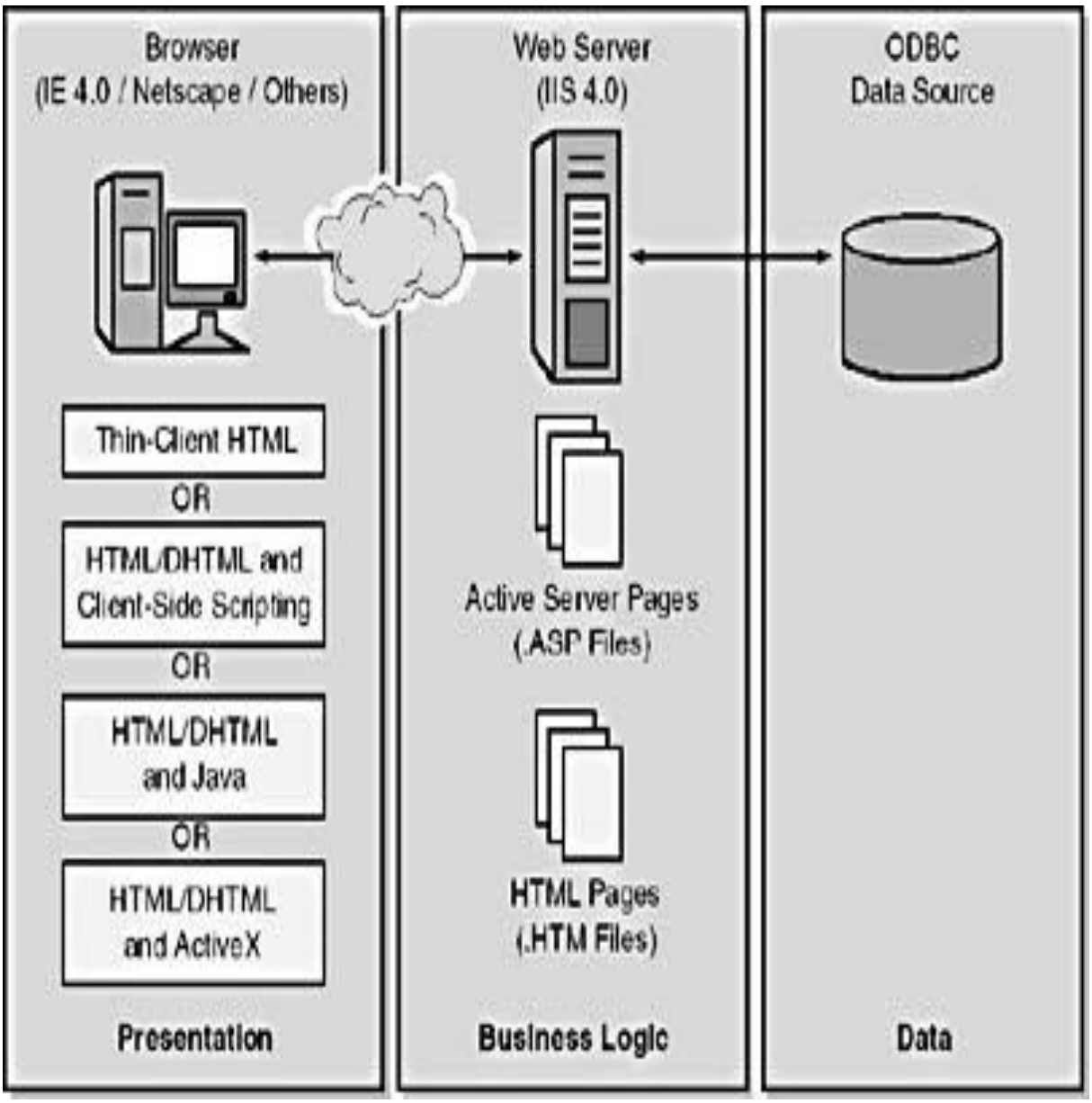


Figure 5.1.2.1 – Control Flow Architecture

5.2 MODULES:

The System consists of 3 Modules:

1 .Admin Module

2. Designer Login Module

3. Consultancy Login Module

5.2.1 ADMIN MODULE:

1. Register New Designer
2. Register New Consultant
3. View Ratings

5.2.2 DESIGNER LOGIN MODULE:

4. Add New Design

5.2.3 DESIGNER LOGIN MODULE:

5. View All Designs
6. View By Designer
7. Rate Items

CHAPTER - 6
UML DIAGRAMS

6.1 UNIFIED MODELING LANGUAGE:

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules.
A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

- **User Model View:**
  i. This view represents the system from the user's perspective.
  ii. The analysis representation describes a usage scenario from the end-users perspective.
- **Structural model view:**
  i. In this model the data and functionality are arrived from inside the system.
  ii. This model view models the static structures.

- **Behavioral Model View:**
  i. It represents the dynamic behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- **Implementation Model View:**
  i. In this the structural and behavioral as parts of the system are represented as they are to be built.

- **Environmental Model View:**
  i. In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

6.2 UML DAIGRAMS:

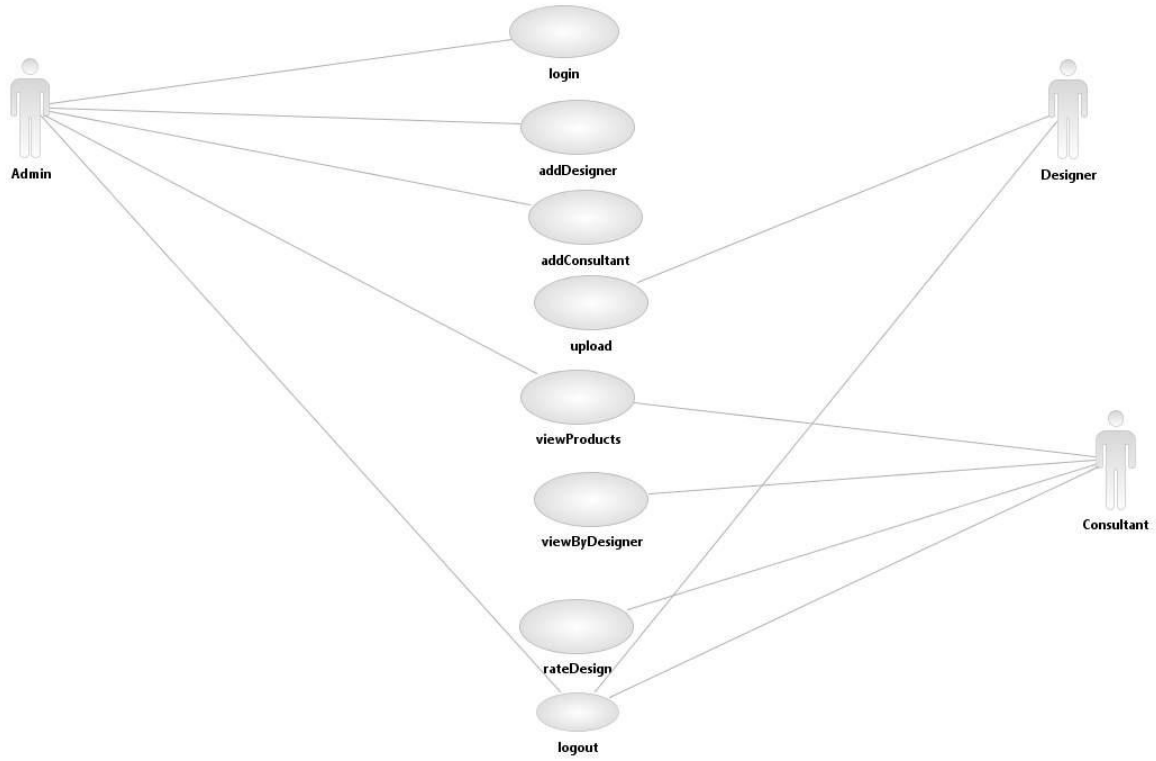


Figure 6.2.1 - Over All Use Case Diagram

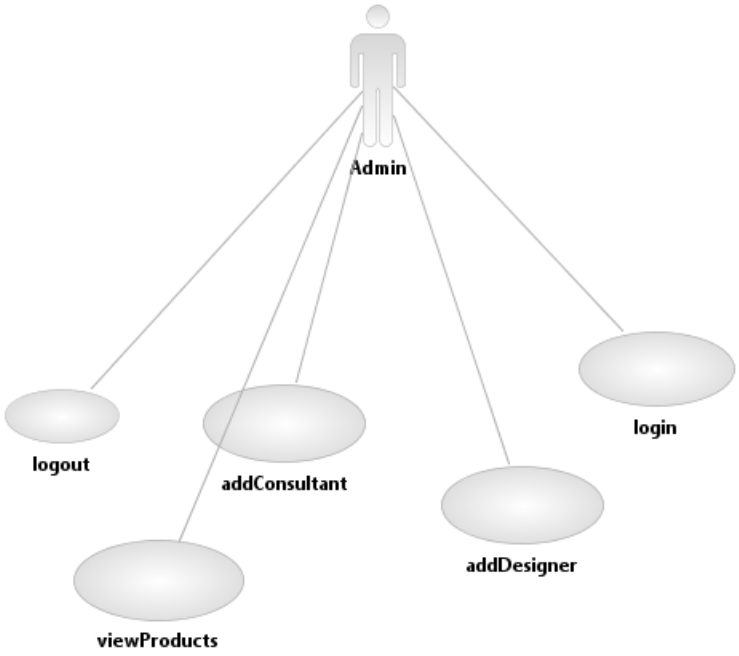


Figure 6.2.2 – Administrator Use Case Diagram

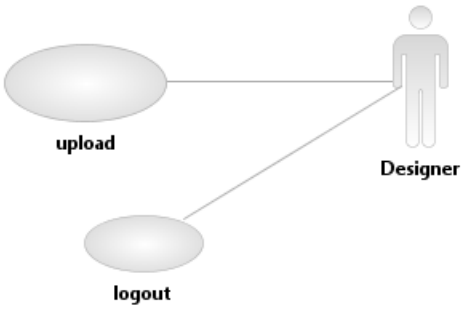


Figure 6.2.3 – Designer Use Case Diagram

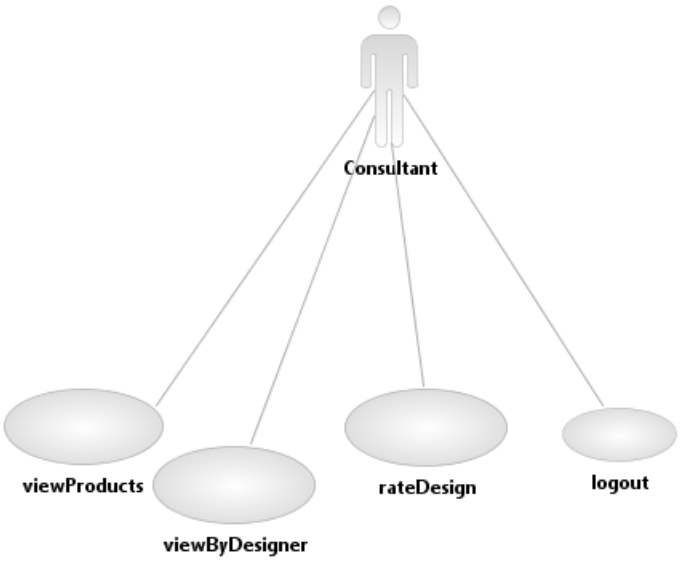


Figure 6.2.3 – Consultant Use Case Diagram

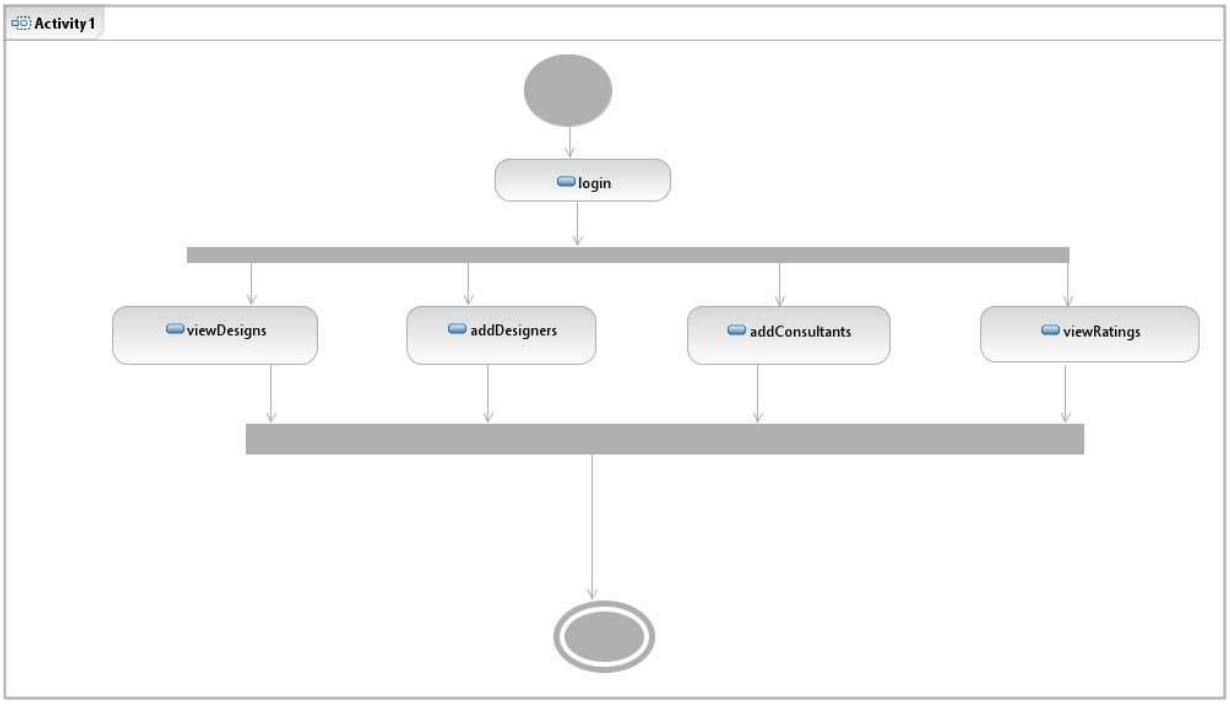


Figure 6.2.4 – Activity Diagram for Admin

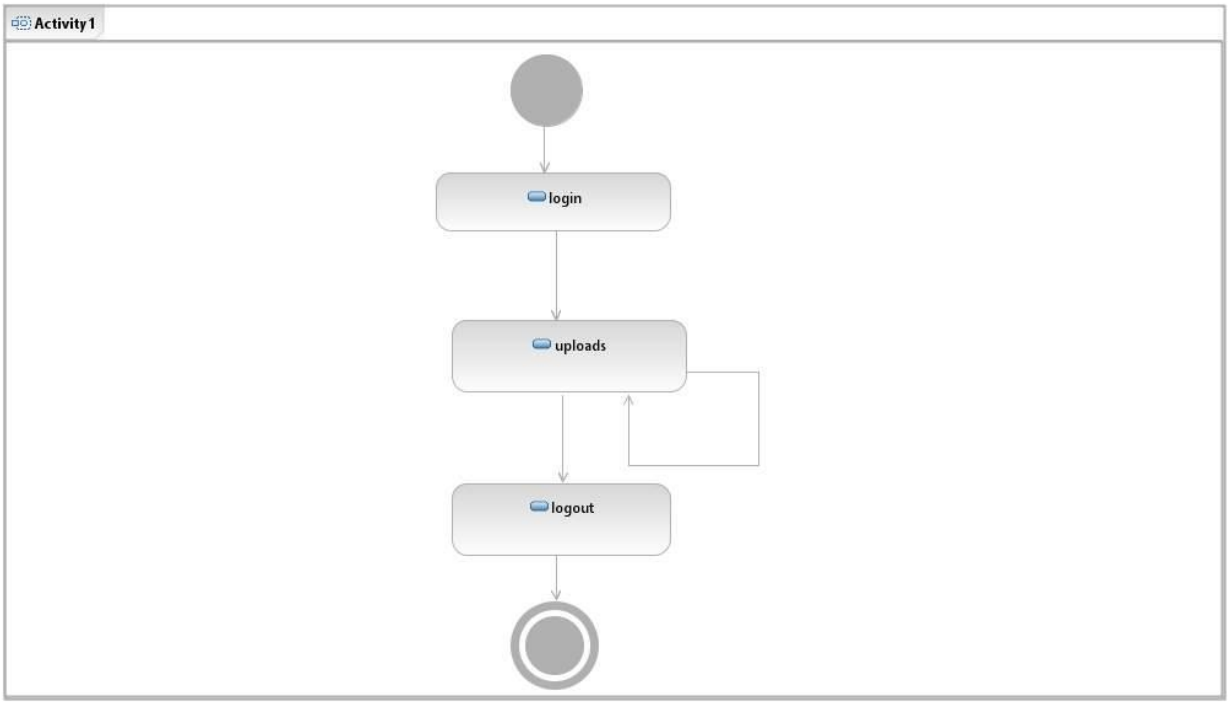


Figure 6.2.4 – Activity Diagram for Designer

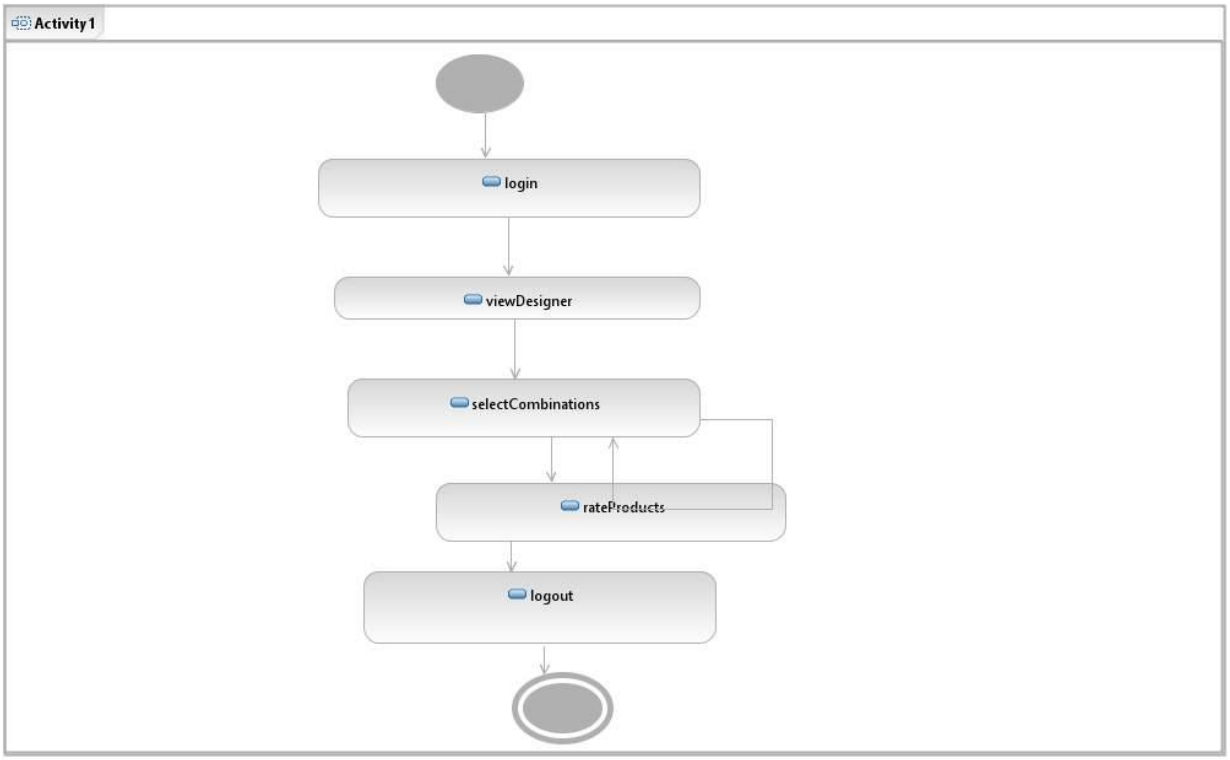


Figure 6.2.4 – Activity Diagram for Consultant

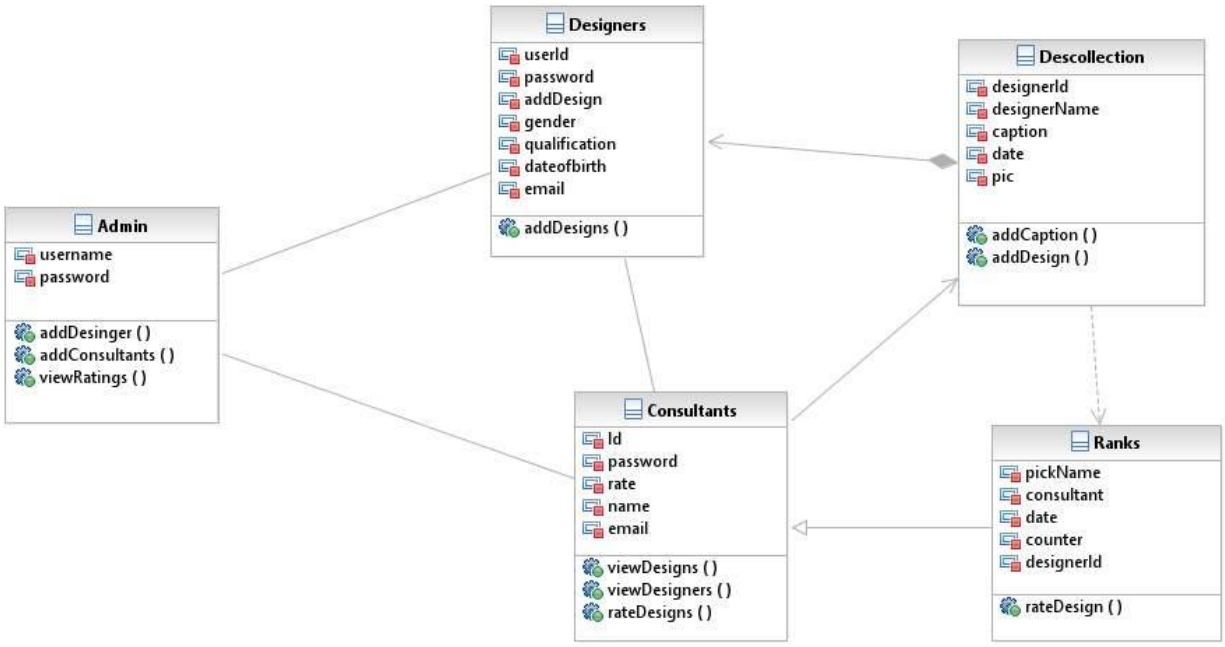


Figure 6.2.5 – Class Diagram

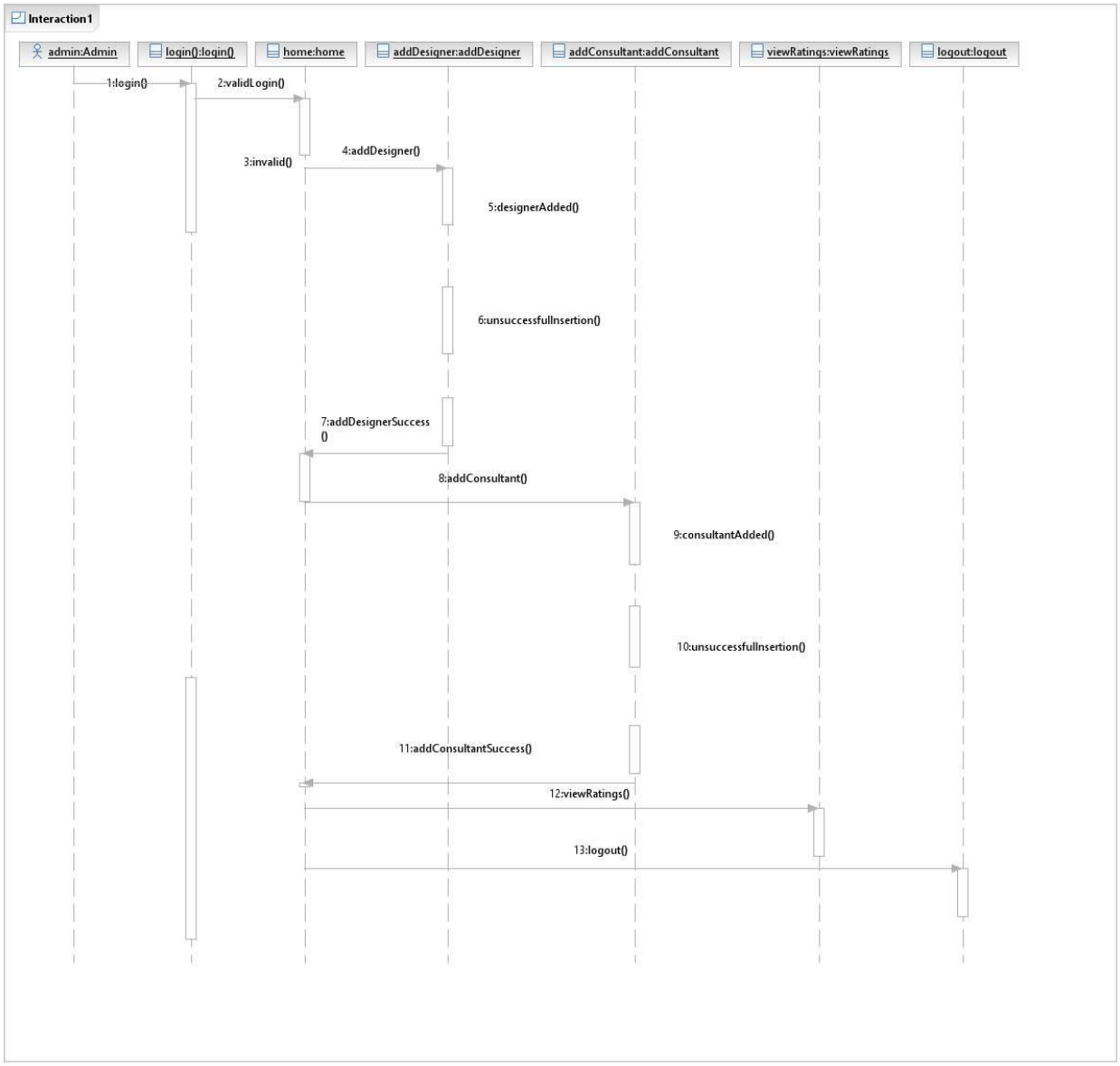


Figure 6.2.6 - Sequence Diagram for Admin

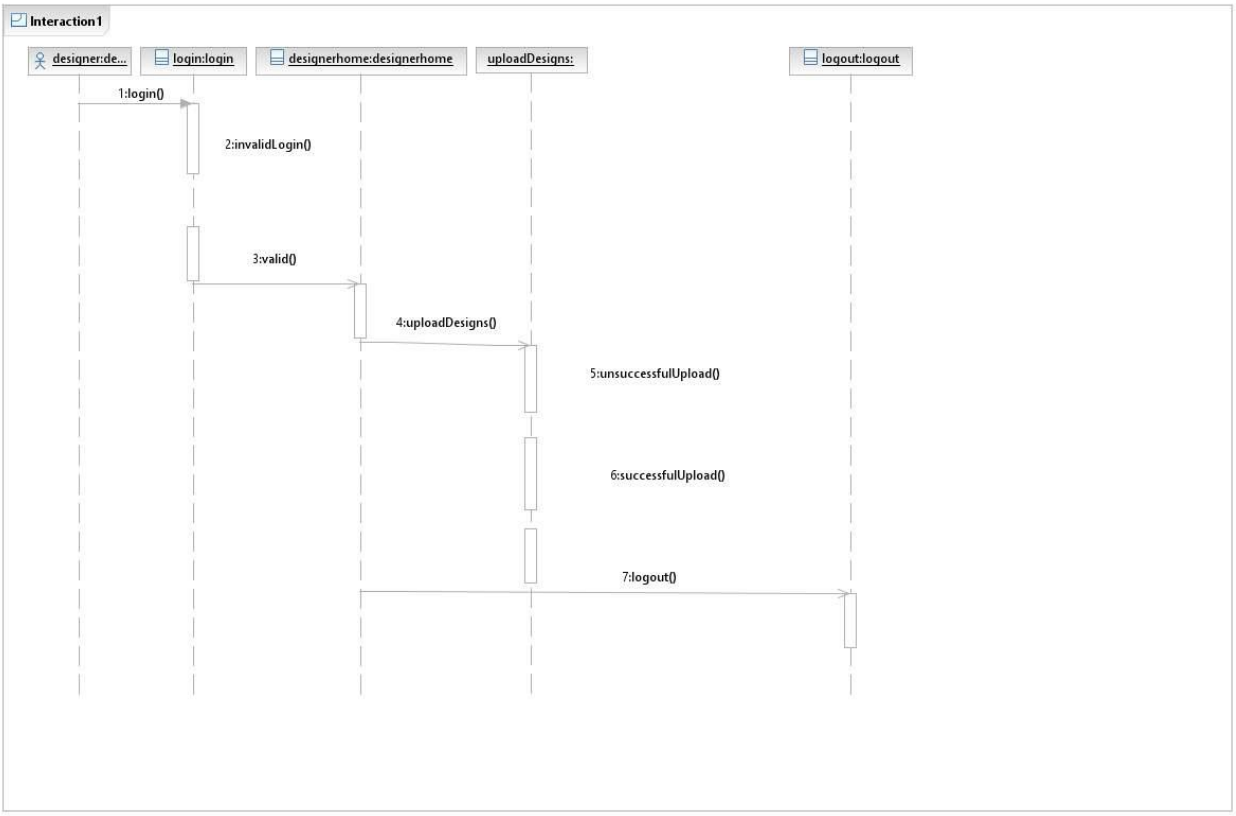


Figure 6.2.7 - Sequence Diagram for Designer

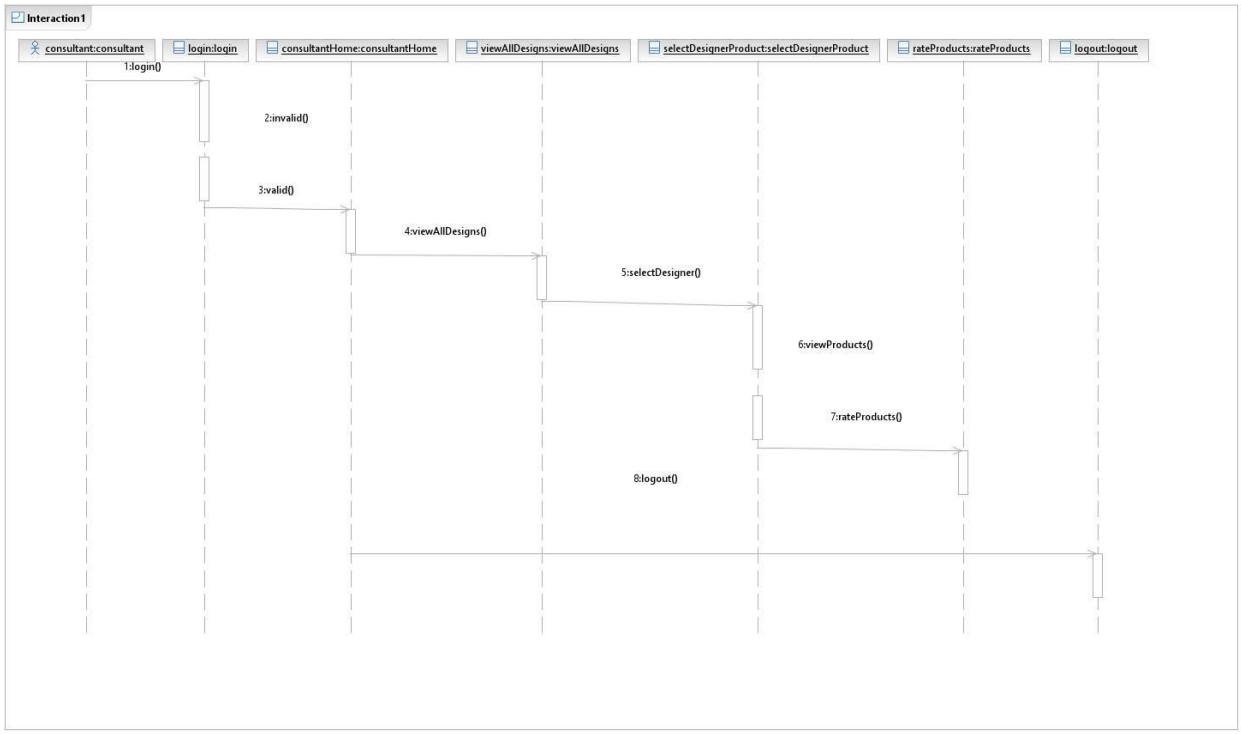


Figure 6.2.7 - Sequence Diagram for Consultant

CHAPTER-7
CODING AND IMPLEMENTATION

7.1 INTRODUCTION:

Coding and Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

7.2 DATABASE CONNECTION:

In computer science, a database connection is the means by which a database server and its client software communicate with each other. The term is used whether or not the client and the server are on different machines.

The client uses a database connection to send commands to and receive replies from the server. A database is stored as a file or a set of files on magnetic disk or tape, optical disk, or some other secondary storage device. The information in these files may be broken down into records, each of which consists of one or more fields.

In our project we used MYSQL Database, is a relational database management system (RDBMS). In July 2013 it was the world's second most widely used RDBMS, and the most widely used open-source RDBMS.

The following Program is used to connect with MYSQL Server:

```
package databaseconnection;

import java.sql.*;


public class databasecon

{

        static Connection con;
```

```
public static Connection getconnection()
{

        try
        {
                Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/onlinecollection","root","root");
        }
        catch(Exception e)
        {
                System.out.println("class error"+e);
        }
        return con;
}

}
```

7.3 ADMIN LOGIN PAGE :(adminlogin.jsp)

```
<%@ include file="aheader.jsp"%>
<%
```

```
String msg = request.getParameter("msg");

if(msg != null){

out.println("<script>alert('Once again verify  Admin and Password....!!!!')</script>");

}

%>

        <div id="body">

                <div id="featured">

                        <div class="first">

                                <div>

                                        <h2>World&#8217;s Finest and Elegant Tuxedo</h2>

                                        <p>

                                                You Could Say I Have A Passion For Handmade Fashion!  You Could Say I Have A Passion For Handmade Fashion!

                                        </p>

                                        <a href="product.jsp" id="shopnow">View More</a>

                                </div>

                                <span><img src="images/tuxedo1.jpg" alt="Image"></span>

                        </div>

                        <div class="last">

                                <h3>Admin Login:</h3>

                                <form action="adminvalidation.jsp" method="post">
```

```
<table>
<tr>
<td><h4>Admin Name:</h4></td><td><input type="text" id="newsletter"  name="aname" required/></td>
</tr>
<tr>
<td><h4>PassWord:</h4></td><td><input type="password" id="newsletter"  name="apass" required/></td>
</tr>
<tr>
<td><input type="submit" class="myButton" value="Login" /></td><td><input type="reset"  class="myButton" value="Reset"/></td>
</tr>
</table>
</form>
</div>
</div>
<div id="content">
<div id="home">
<marquee behavior="scroll" direction="left" scrolldelay="20"><font color="yellow"><img src="images.png" width="50" height="50" alt="Natural" />You Could Say I Have A Passion For Handmade Fashion!<img src="images.png" width="50" height="50" alt="Natural" /> I Enjoy Testing Out, Trying And
```

Tweaking Designer Patterns On The Market As Well As Creating Designs Of My Own! Check Out My Online Pattern Store To Learn How To Create Simple And Flattering Designs Perfect For Beginning Your Own Handmade Wardrobe!</marquee>

<%@ include file="footer.jsp"%>

</body>

</html>

7.4 DESIGNER LOGIN :(designerlogin.jsp)

<%@ include file="aheader.jsp"%>

<%

String msg = request.getParameter("msg");

if(msg != null){

out.println("<script>alert('Once again verify Admin and Password....!!!!')</script>");

}

%>

```
<div id="featured">
    <div class="first">
        <div>
            <h2>World&#8217;s Finest and Elegant Tuxedo</h2>
            <p>
                You Could Say I Have A Passion For Handmade Fashion!  You Could Say I Have A Passion For Handmade Fashion!
            </p>
            <a href="product.jsp" id="shopnow">View More</a>
        </div>
        <span><img src="images/tuxedo1.jpg" alt="Image"></span>
    </div>
    <div class="last">
        <h3>Designer Login:</h3>
        <form action="desvalidation.jsp" method="post">
        <table>
        <tr>
        <td><h4>Designer Id:</h4></td><td><input type="text" id="newsletter"  name="desid" required/></td>
        </tr>
        <tr>
```

```
<td><h4>PassWord:</h4></td><td><input type="password" id="newsletter" name="despass" required/></td>

</tr>

<tr>

<td><input type="submit" class="myButton" value="Login" /></td><td><input type="reset" class="myButton" value="Reset"/></td>

</tr>

</table>

</form>

</div>

</div>

<div id="content">

<div id="home">

<marquee behavior="scroll" direction="left" scrolldelay="20"><font color="yellow"><img src="images.png" width="50" height="50" alt="Natural" />You Could Say I Have A Passion For Handmade Fashion!<img src="images.png" width="50" height="50" alt="Natural" /> I Enjoy Testing Out, Trying And Tweaking Designer Patterns On The Market As Well As Creating Designs Of My Own!<img src="images.png" width="50" height="50" alt="Natural" /> Check Out My Online Pattern Store To Learn How To Create Simple And Flattering Designs Perfect For Beginning Your Own Handmade Wardrobe!<img src="images.png" width="50" height="50" alt="Natural" /></font></marquee>

</div>

</div>
```

```
        </div>


<%@ include file="footer.jsp"%>


</body>

</html>
```

7.5 CONSULTANT LOGIN(conslogin.jsp):

```
<%@ include file="aheader.jsp"%>

<%

String msg = request.getParameter("msg");

if(msg != null){

out.println("<script>alert('Once again verify  Admin and Password....!!!!')</script>");

}

%>

        <div id="body">

                <div id="featured">

                        <div class="first">

                                <div>

                                        <h2>World&#8217;s Finest and Elegant Tuxedo</h2>

                                        <p>
```

```
                                    You Could Say I Have A Passion For Handmade Fashion!  You Could Say I Have A Passion For Handmade Fashion!
                                    </p>
                                    <a href="product.jsp" id="shopnow">View More</a>
                                </div>
                                <span><img src="images/tuxedo1.jpg" alt="Image"></span>
                            </div>
                            <div class="last">
                                                <h3>Consultant Login:</h3>
                                <form action="consvalidation.jsp" method="post">
                                <table>
                                <tr>
                                <td><h4>Consultancy Mail Id:</h4></td><td><input type="email" id="newsletter"name="cemail" required/></td>
                                </tr>
                                <tr>
                                <td><h4>Mobile Number:</h4></td><td><input type="text" id="newsletter" name="mno" maxlength="10"required/></td>
                                </tr>
                                <tr>
                                <td><input type="submit" class="myButton" value="Login" /></td><td><input type="reset" class="myButton" value="Reset"/></td>
```

```
                    </tr>

                    </table>

                    </form>

                </div>

            </div>

            <div id="content">

                <div id="home">

                    <marquee behavior="scroll" direction="left" scrolldelay="20"><font color="yellow"><img src="images.png" width="50" height="50" alt="Natural" />You Could Say I Have A Passion For Handmade Fashion!<img src="images.png" width="50" height="50" alt="Natural" /> I Enjoy Testing Out, Trying And Tweaking Designer Patterns On The Market As Well As Creating Designs Of My Own!<img src="images.png" width="50" height="50" alt="Natural" /> Check Out My Online Pattern Store To Learn How To Create Simple And Flattering Designs Perfect For Beginning Your Own Handmade Wardrobe!<img src="images.png" width="50" height="50" alt="Natural" /></font></marquee>

                </div>

            </div>

        </div>


<%@ include file="footer.jsp"%>


</body>

</html>
```

7.6 HOME PAGE(index.jsp):

```
<%@ include file="header.jsp"%>


<%
String desmsg = request.getParameter("desmsg");
if(desmsg != null){
out.println("<script>alert('Once again verify  Designerid and Password....!!!!')</script>");
}
%>
        <div id="body">
                <div id="featured">
                        <div class="first">
                                <div>
                                        <h2>World&#8217;s Finest and Elegant Tuxedo</h2>
                                        <p>
                                                You Could Say I Have A Passion For Handmade Fashion!  You Could Say I Have A Passion For Handmade Fashion!
                                        </p>
                                        <a href="product.jsp" id="shopnow">View More</a>

                        </div>
```

```
                <span><img src="images/tuxedo1.jpg" alt="Image"></span>
            </div>
            <div class="last">
                <h3>Handcrafted and made from 100% Wool</h3>
                <ul>
                    <li>
                        <a href="product.jsp"><img src="images/tuxedo2.jpg" alt="Image"></a>
                        <p>
                            This is just a place holder
                        </p>
                    </li>
                    <li>
                        <a href="product.jsp"><img src="images/tuxedo3.jpg" alt="Image"></a>
                        <p>
                            This is just a place holder
                        </p>
                    </li>
                </ul>
            </div>
        </div>
```

```
<div id="content">

<div id="home">

<marquee behavior="scroll" direction="left" scrolldelay="20"><font color="yellow"><img src="images.png" width="50" height="50" alt="Natural" />You Could Say I Have A Passion For Handmade Fashion!<img src="images.png" width="50" height="50" alt="Natural" /> I Enjoy Testing Out, Trying And Tweaking Designer Patterns On The Market As Well As Creating Designs Of My Own!<img src="images.png" width="50" height="50" alt="Natural" /> Check Out My Online Pattern Store To Learn How To Create Simple And Flattering Designs Perfect For Beginning Your Own Handmade Wardrobe!<img src="images.png" width="50" height="50" alt="Natural" /></font></marquee>

</div>

</div>

</div>


<%@ include file="footer.jsp"%>


</body>

</html>
```

7.7 Data Base Tables:

Consults table

```
Field         Type
------------  -------------
consult_name  varchar(100)
email         varchar(100)
mno           varchar(10)
addr          varchar(1000)
```

designers table

Field	Type
designerid	varchar(10)
name	varchar(30)
l_name	varchar(30)
password	varchar(20)
emailid	varchar(100)
dateofbirth	varchar(30)
gender	varchar(10)
edu_qualifi	varchar(20)
mno	varchar(10)
addr	varchar(1000)
pics	varchar(200)

descollection Table

Field	Type
designerid	varchar(10)
name	varchar(30)
l_name	varchar(30)
password	varchar(20)
emailid	varchar(100)
dateofbirth	varchar(30)
gender	varchar(10)
edu_qualifi	varchar(20)
mno	varchar(10)
addr	varchar(1000)
pics	varchar(200)

ranks Table

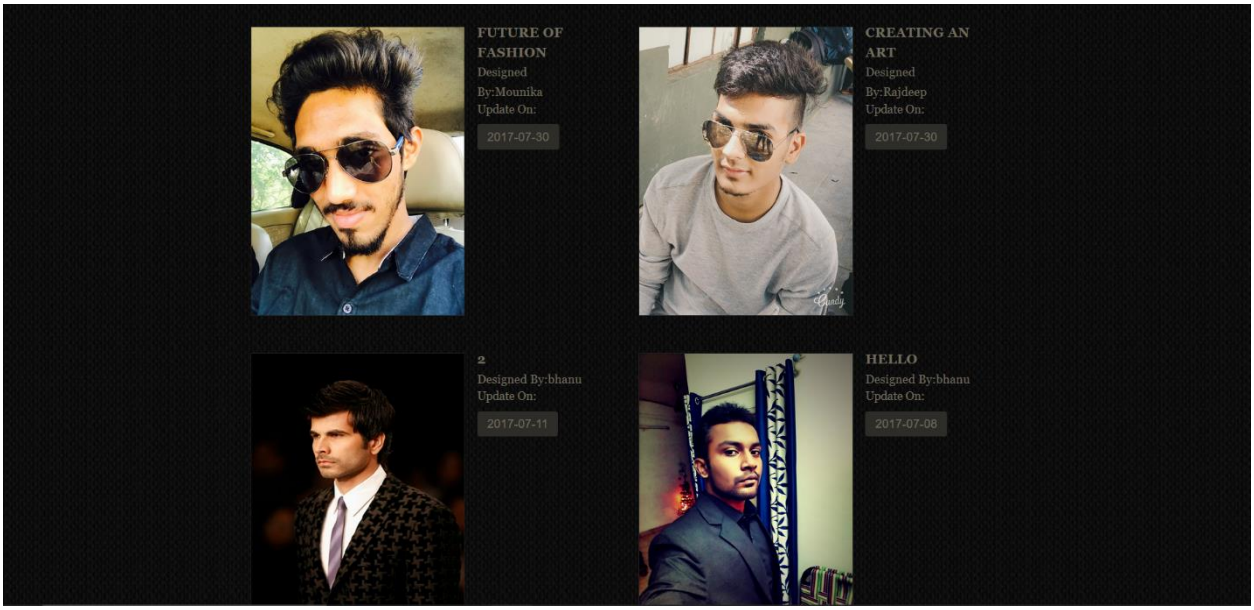
Field	Type
picname	varchar(500)
consults	varchar(100)
date	varchar(20)
counter	int(10)
designerid	varchar(100)

Table 7.1 Database Tables Specifications

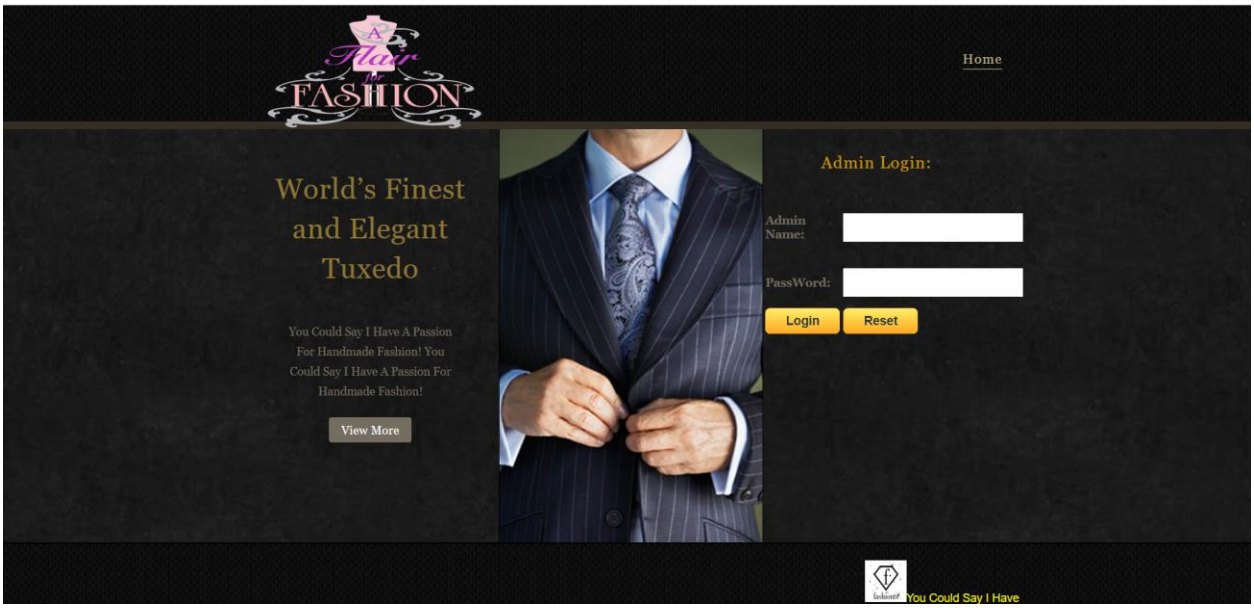
CHAPTER-8

SCREENSHOTS

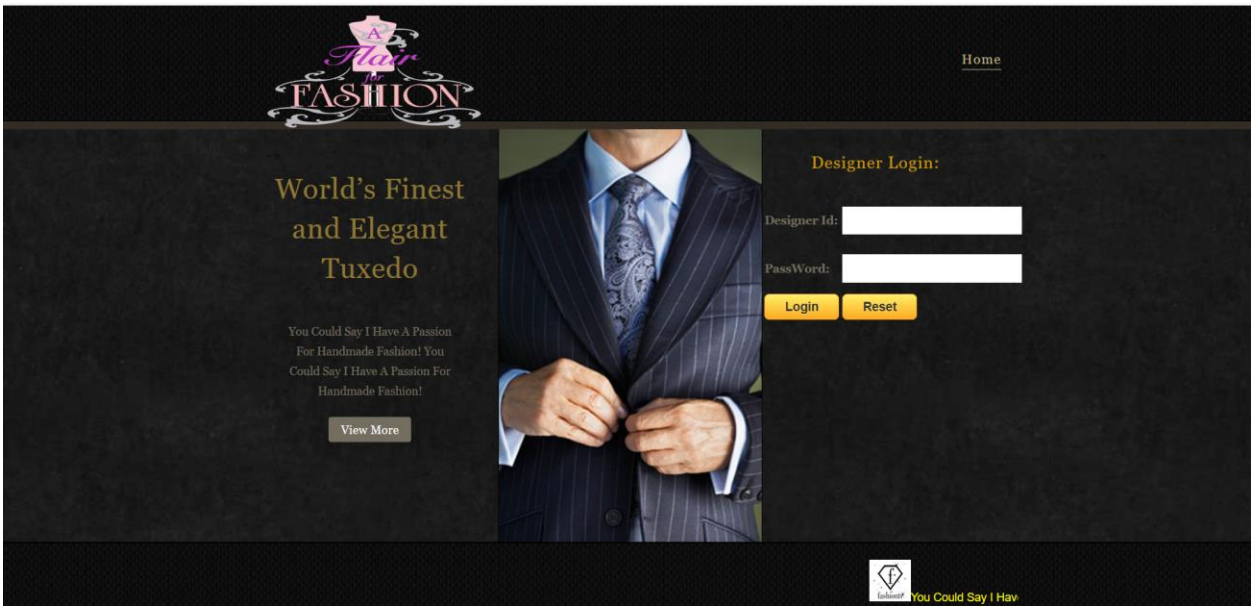
Screenshot 8.1 - Home page



Screenshot 8.2 – View Products page



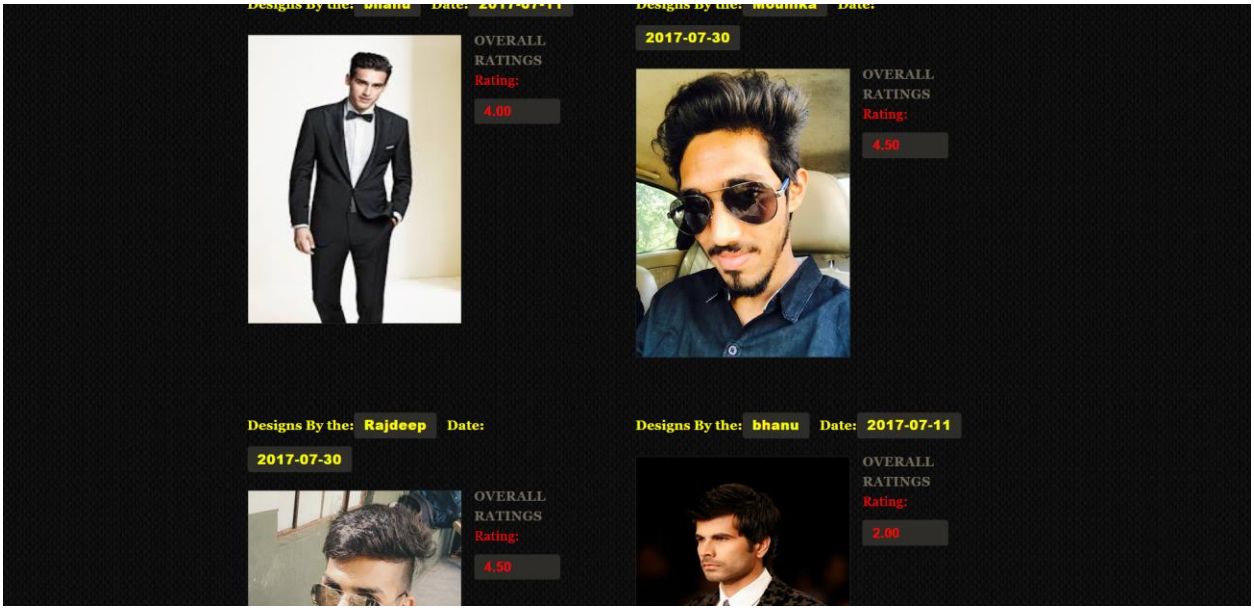
Screenshot 8.3 – Admin Login page



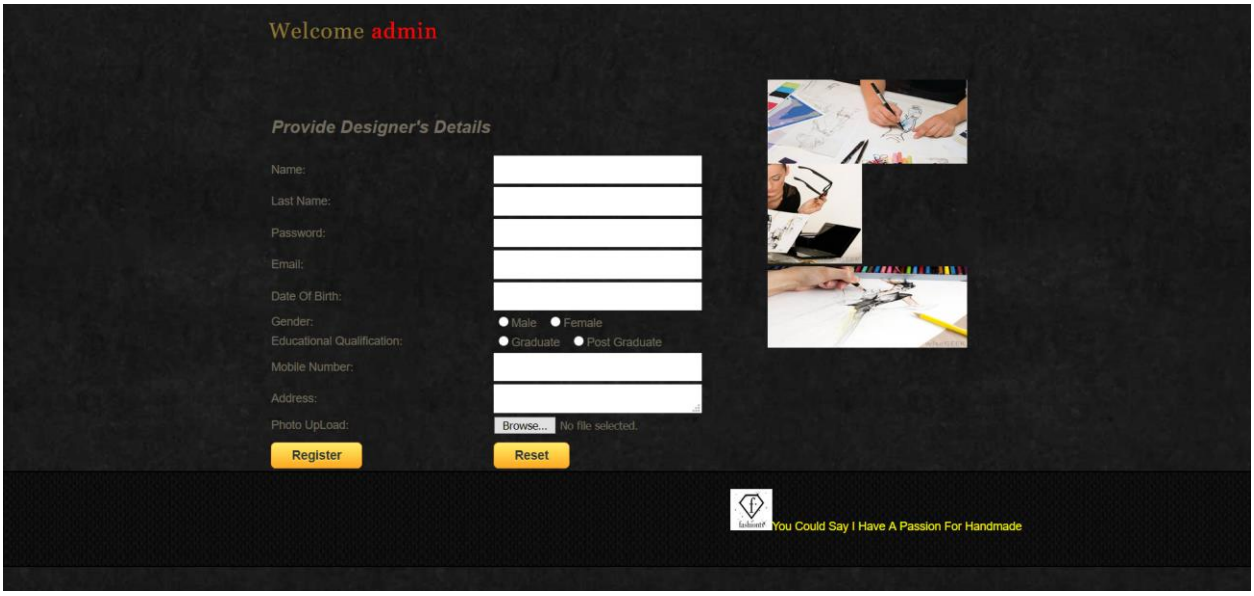
Screenshot 8.4 – Designer Login page

Screenshot 8.5 – Consultant Login page

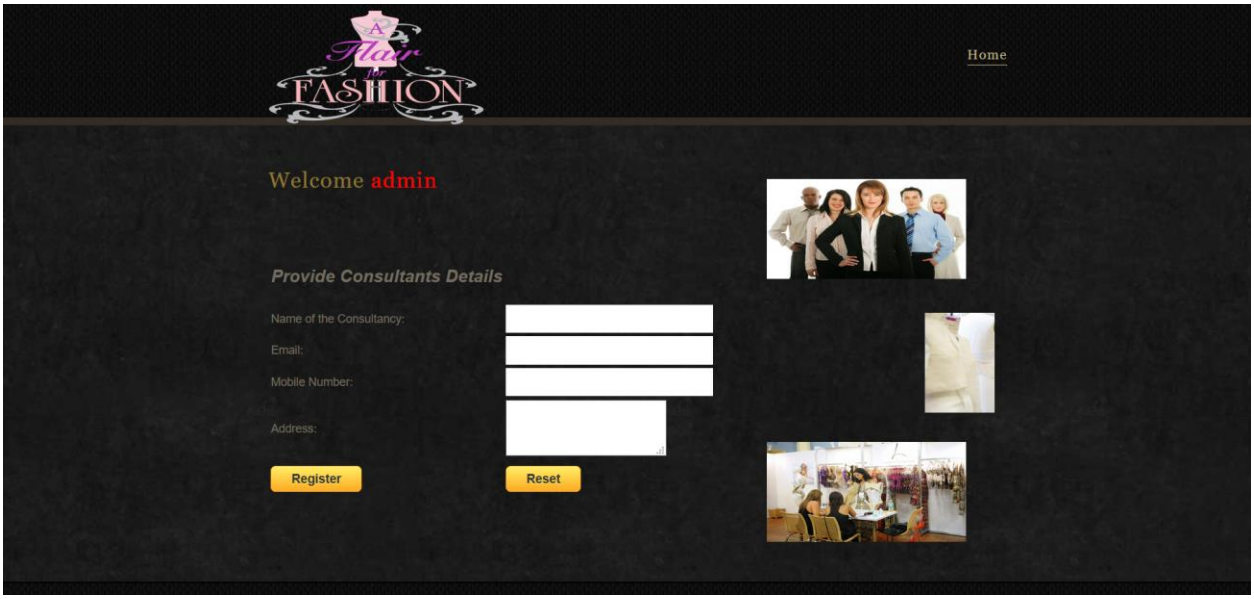
Screenshot 8.6 – Admin Home page



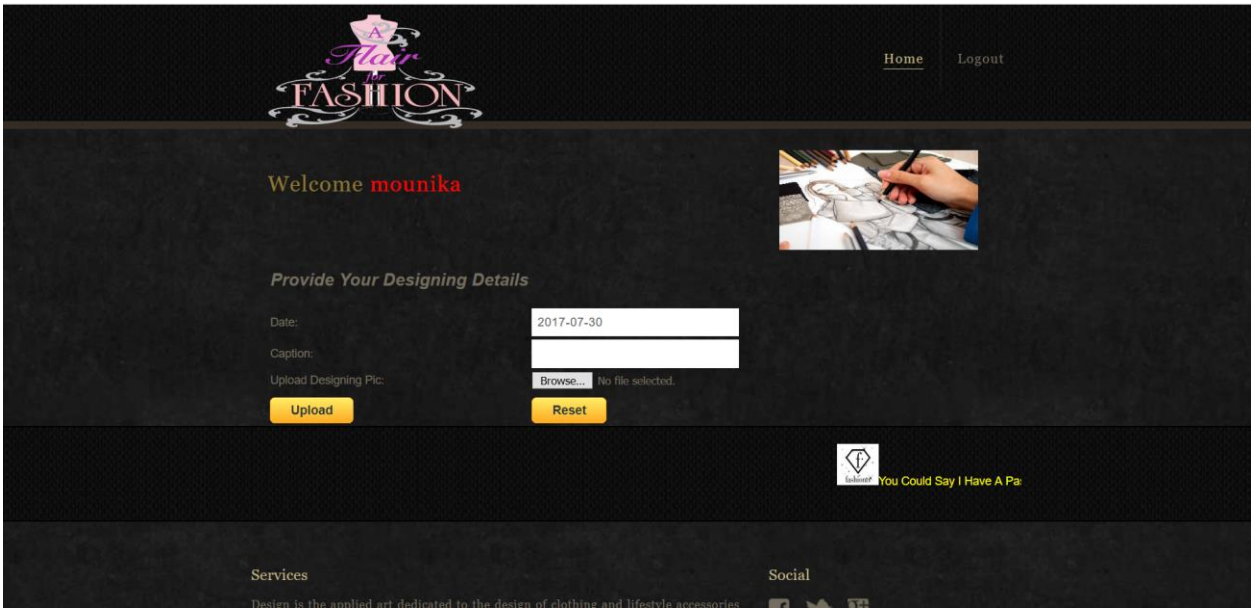
Screenshot 8.7 – View Rating page



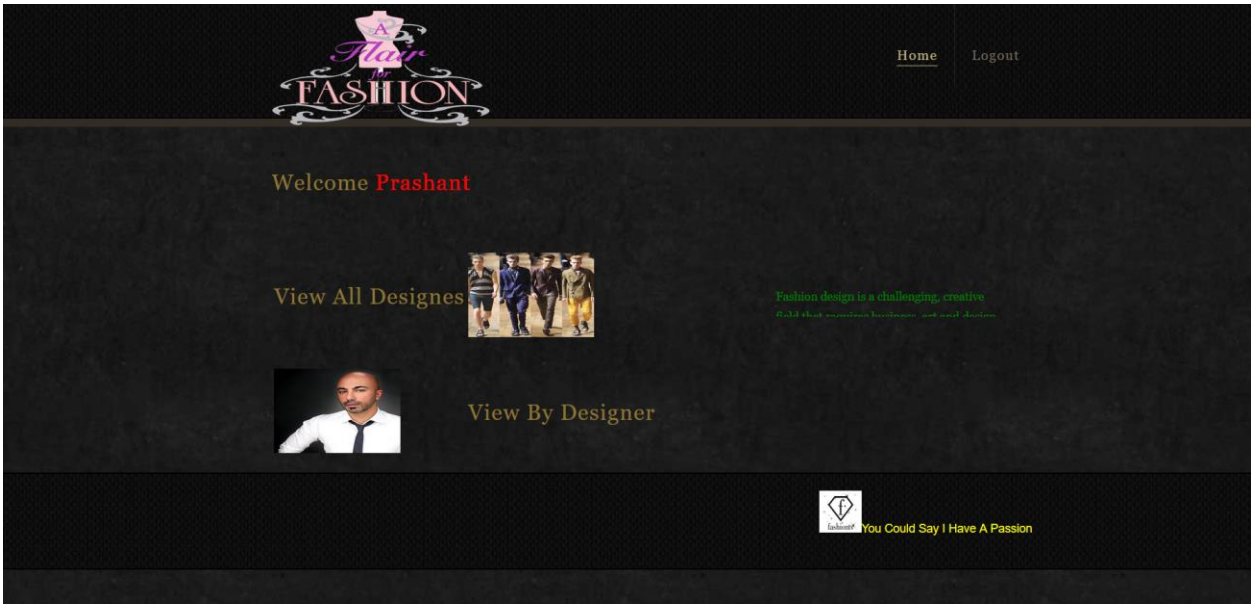
Screenshot 8.8 – Add Designer page



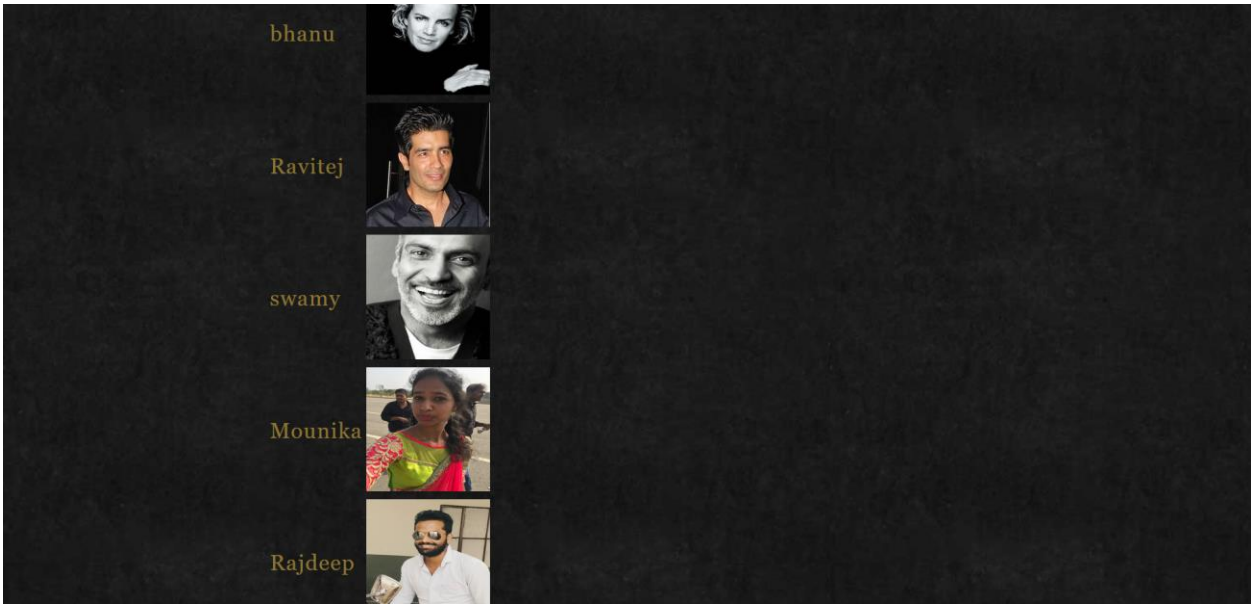
Screenshot 8.9 – Add Consultant page



Screenshot 8.10 – Designs Upload/Designer Home page



Screenshot 8.11 – Consultant Home page



Screenshot 8.12 – View Designers page

CHAPTER-9

SYSTEM TESTING

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

Testing objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

A successful test is one that uncovers an as yet undiscovered error.

A good test case is one that has probability of finding an error, if it exists.

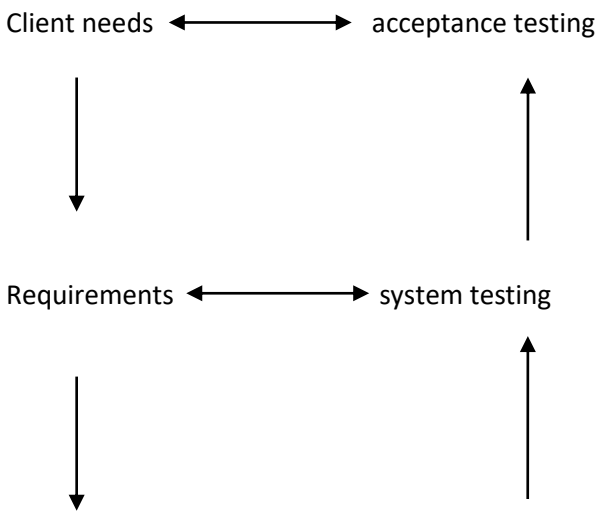
The test is inadequate to detect possibly present errors.

The software more or less confirms to the quality and reliable standards.

Levels of Testing:

In order to uncover present in different phases we have the concept of levels of testing.

The basic levels of Testing:



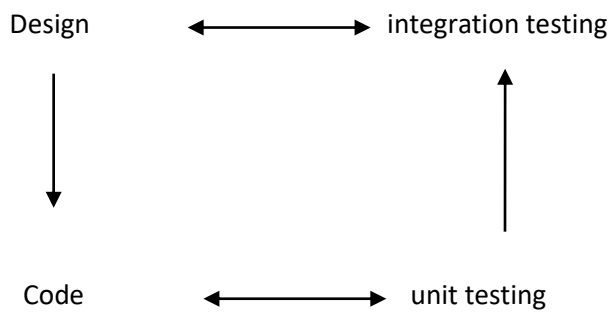


Figure 7.1: Levels of Testing

Code testing:

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

Specification Testing:

Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

Unit testing:

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

9.1 BLACK BOX TESTING

What is Black Box Testing?

Black box testing is a software testing techniques in which **functionality of the software under test (SUT) is tested without looking at the internal code structure**, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

The above Black Box can be any software system you want to test. For example : an operating system like Windows, a website like Google ,a database like Oracle or even your own custom application. Under Black Box Testing , you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Black box testing - Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

Types of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

- **Functional testing** – This black box testing type is related to functional requirements of a system; it is done by software testers.
- **Non-functional testing** – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- **Regression testing** – Regression testing is done after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

9.2 WHITE BOX TESTING

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as **clear, open, structural, and glass box testing**.

It is one of two parts of the **"box testing" approach** of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "whitebox" was used because of the see-through box concept. The clear box or whitebox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

What do you verify in White Box Testing ?

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes

- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

How do you perform White Box Testing?

To give you a simplified explanation of white box testing, we have divided it into **two basic steps**. This is what testers do when testing an application using the white box testing technique:

STEP 1) UNDERSTAND THE SOURCE CODE

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

Step 2) CREATE TEST CASES AND EXECUTE

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

System testing:

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top down testing, which began from upper level to lower level module, was carried out to check whether the entire system is performing satisfactorily.

There are three main kinds of System testing:

i. Alpha Testing
ii. Beta Testing
iii. Acceptance Testing

Alpha Testing:

This refers to the system testing that is carried out by the test team with the Organization.

Beta Testing:

This refers to the system testing that is performed by a selected group of friendly customers

Acceptance Testing:

This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.

Table.9.1: Acceptance Testing:

Test Condition ID	Description of coverage	Expected results	Covered by script
1.	Verification of a particular record	If a particular record already exists it displays a message	This type of test in {verify} procedure in every Jsp file where a record is inserted via an interface
2.	Updating of a particular record	All the details should not be updated.	This type of test is covered in all the Asp files where updations are made.
3.	Validity of login	Only the authorized persons must access system.	This is covered in the login procedure for the validity of a user

Integration Testing:

Data can be lost across an interface, one module can have an adverse effort on the other sub functions, when combined, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. The testing was done with sample data. The developed system has run successfully for this sample data. The need for integrated test is to find the overall system performance.

Output testing:

After performance of the validation testing, the next step is output testing. The output displayed or generated by the system under consideration is tested by asking the user about the format required by system.

Test plan:

The test-plan is basically a list of testcases that need to be run on the system. Some of the testcases can be run independently for some components (report generation from the database, for example, can be tested independently) and some of the testcases require the whole system to be ready for their execution. It is better to test each component as and when it is ready before integrating the components. It is important to note that the testcases cover all the aspects of the system (ie, all the requirements stated in the RS document).

Figure.9.2: Test plan

Sno	**Test case Title**	**Description**	**Expected Outcome**	**The requirement in RS that is being tested**	**Result**
1	Successful User Verification	The login to the system should be tried with the login assigned by the admin and the correct password	Login should be successful and the user should enter in to the system	RS1	Passed
2	Unsuccessful User Verification due to wrong password	Login to the system with a wrong password	Login should fail with an error 'Invalid Password'	RS1	Passed
3	Unsuccessful User Verification due to invalid login id	Login to the system with a invalid login id	Login should fail with an error 'Invalid user id'	RS1	Passed

CONCLUSION

10.1 CONCLUSION:

It is a technique used in decision-making process to rank or prioritize items on a relative scale of importance against a criterion. Let us assume that there are “n” items to be ranked. In this technique, all unique pairs of “n” items are formed. Note there will be n C2 pairs for “n” items. The user is presented each pair sequentially and asked to select (against the criterion) his/her “preferred item” from the pair.

BIBLIOGRAPHY

- **FOR JAVA INSTALLATION:**
    - https://www.java.com/en/download/
- **REFERENCE BOOKS:**

- Raghu Ramakrishnan Gehrke, 2000 - Database Management System, Johannes Mc Graw-Hil International Edition.
- Roger S.Pressman, 1997 - Software Engineering, Fourth Edition, Mc Graw-Hill International Edition.
- Herbert Schildt, 2011 - The Complete Reference Java, Mc Graw-Hill International Edition.
- A. A. Puntambekar, 2015 - Web Technologies, Technical Publications.