Website developer: Web application

Venkata Krishna Reddy Tummeti
WEBSITE DEVELOPER, A WEB APPLICATION

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Venkata Krishna Reddy Tummeti
December 2003
WEBSITE DEVELOPER, A WEB APPLICATION

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Venkata Krishna Reddy Tummeti
December 2003

Approved by:

Dr. Ernesto Gomez
Associate Professor, Computer Science

Dr. Richard Botting
Professor, Computer Science

Dr. David Turner
Assistant Professor, Computer Science

Date: 12/4/03
Unlike content-centric websites, web applications serve the specific tasks of its users. The motivations for using a web application is almost always explicit, precise, and user specific. Whereas content-centric websites provide general services to its users.

A web application behaves like traditional software but the user doesn’t have to install anything. It is a program whose user interface runs entirely from within a browser. That makes it incredibly portable and easily accessible from anywhere one has access to the Internet.

In this project a web application was developed which helps its users create and host their websites. WD (Website Developer) is a user-friendly program that guides its users in a step-by-step process. Its users are not required to have any knowledge of HTML (Hypertext Markup Language) or any other web authoring tools.

WD uses JSP (Java Server Pages) and HTML as its front end, Java for processing, and the data is stored in MySQL database in the back end. Apache Tomcat is used as the web server.
# TABLE OF CONTENTS

ABSTRACT ................................................................. iii

LIST OF TABLES ......................................................... x

LIST OF FIGURES ......................................................... xi

CHAPTER ONE: SOFTWARE REQUIREMENTS SPECIFICATION

1.1 Introduction ......................................................... 1
  1.1.1 Purpose of the Project ................................. 1
  1.1.2 Scope of the Project .................................. 2
  1.1.3 Limitations of the Project ............................ 3
  1.1.4 Definitions, Acronyms, and Abbreviations .......... 5
  1.1.5 Overview .................................................. 8

1.2 Overall Description ............................................. 8
  1.2.1 Product Perspective .................................... 8
  1.2.2 System Interfaces ....................................... 9
  1.2.3 User Interfaces .......................................... 9
    1.2.3.1 WD Home Page .................................... 9
    1.2.3.2 WD Signup Page .................................. 10
    1.2.3.3 WD Login Page ................................... 10
    1.2.3.4 WD Administration Page ......................... 10
    1.2.3.5 WD Template Page ................................ 11
  1.2.4 Hardware Interfaces ...................................... 11
  1.2.5 Software Interfaces ...................................... 11
CHAPTER TWO: SOFTWARE DESIGN

2.1 Overview ........................................ 14
2.2 WD Methodology ................................. 14
   2.2.1 Central Server ............................... 14
   2.2.2 Remote Client ............................... 14
2.3 WD Architecture ................................. 15
   2.3.1 Presentation Layer ......................... 16
   2.3.2 Application Logic Layer .................... 17
   2.3.3 Control Layer ............................... 17
2.4 WD Detailed Design ............................. 19
   2.4.1 Architectural Approach ..................... 19
   2.4.2 WD Presentation Layer ...................... 19
      2.4.2.1 WD Home Page .......................... 20
      2.4.2.2 WD Signup Page ........................ 21
      2.4.2.3 WD Login Page .......................... 23
      2.4.2.4 WD Forgot Password Page .............. 24
      2.4.2.5 WD Password Display Page ............ 26
      2.4.2.6 WD Administration Page .............. 27
2.4.2.7 WD Personal Information Change Page ................................. 30
2.4.2.8 WD Page-Name Change Page ............... 32
2.4.2.9 WD Template Page ......................... 33
2.4.2.10 No-Selection Page ....................... 36
2.4.2.11 Text Display Page ....................... 37
2.4.2.12 Picture Display Page ..................... 39
2.4.2.13 Links Display Page ....................... 41
2.4.2.14 WD Background Select Page ............ 43
2.4.2.15 WD Menu choice Page .................... 44
2.4.2.16 WD Page-Title Change Page ............ 46
2.4.2.17 WD Unit Choice Page .................... 47
2.4.2.18 WD Text Choice Page ..................... 48
2.4.2.19 WD Picture Choice Page ................. 50
2.4.2.20 WD Links Choice Page ................... 51
2.4.2.21 WD Text Preview Page .................... 53
2.4.2.22 WD Picture Preview Page ............... 54
2.4.2.23 WD Picture Upload Page ................. 55
2.4.2.24 WD Links Preview Page ................. 57
2.4.2.25 WD Text Input Page ...................... 58
2.4.2.26 WD Caption Input Page ................. 59
2.4.2.27 WD Links Input Page ..................... 60
2.4.3 WD Control Layer ............................. 62
2.4.4 WD Application Logic Layer .......... 62
  2.4.4.1 UserBean ......................... 63
  2.4.4.2 UserRepository .................... 63
  2.4.4.3 UserRepositoryException .......... 64
  2.4.4.4 PageBean ......................... 65
  2.4.4.5 PageRepository ................... 66
  2.4.4.6 PageRepositoryException .......... 67
  2.4.4.7 TextBean ......................... 67
  2.4.4.8 TextRepository ................... 68
  2.4.4.9 TextRepositoryException .......... 69
  2.4.4.10 PictureBean ...................... 70
  2.4.4.11 PictureRepository ................. 70
  2.4.4.12 PictureRepositoryException ...... 71
  2.4.4.13 LinkBean ......................... 71
  2.4.4.14 LinkRepository ................... 72
  2.4.4.15 LinkRepositoryException .......... 73
  2.4.4.16 PictureUpload .................... 73
  2.4.4.17 PictureUploadException .......... 74
  2.4.5 WD Database Design ................. 74

  2.5 Summary .................................... 79

CHAPTER THREE: TESTING

  3.1 Overview .................................. 78
  3.2 Unit Testing .............................. 78

vii
3.2.1 Test Cases ................................................. 78
3.2.2 Results ...................................................... 79
3.3 Integration Testing ................................. 82
  3.3.1 New User Scenario ......................... 83
    3.3.1.1 Signup Process ......................... 83
    3.3.1.2 Create New Page Process .......... 84
    3.3.1.3 Page-Title Change Process ........ 85
  3.3.2 Registered User Scenario ................. 86
    3.3.2.1 Forgot Password Process .......... 86
    3.3.2.2 Login Process ......................... 88
    3.3.2.3 Update Request Process .......... 89
  3.3.3 Results ................................................. 90
3.4 System Testing ........................................... 90
3.5 Summary ................................................... 91

CHAPTER FOUR: MAINTENANCE MANUAL

  4.1 Overview ............................................... 92
  4.2 WD Server Configuration ....................... 92
    4.2.1 WD Development Environment ........... 92
    4.2.2 WD Application Server .................. 92
    4.2.3 WD Application ............................. 93
    4.2.4 WD Database .................................. 95
  4.2.5 Starting and Stopping
the WD Web Server ................................ 95
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3 WD Client Configuration</td>
<td>96</td>
</tr>
<tr>
<td>4.4 Summary</td>
<td>96</td>
</tr>
</tbody>
</table>

CHAPTER FIVE: CONCLUSIONS AND FUTURE DIRECTIONS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Conclusions</td>
<td>97</td>
</tr>
<tr>
<td>5.2 Future Directions</td>
<td>98</td>
</tr>
<tr>
<td>5.2.1 Application Functionality</td>
<td>98</td>
</tr>
<tr>
<td>5.2.2 HTML Specifications</td>
<td>99</td>
</tr>
<tr>
<td>5.2.2.1 Text Properties</td>
<td>100</td>
</tr>
<tr>
<td>5.2.2.2 Picture Properties</td>
<td>100</td>
</tr>
<tr>
<td>5.2.2.3 Sound</td>
<td>100</td>
</tr>
<tr>
<td>5.2.2.4 Color</td>
<td>100</td>
</tr>
<tr>
<td>5.2.2.5 More Units</td>
<td>100</td>
</tr>
<tr>
<td>5.3 Summary</td>
<td>101</td>
</tr>
</tbody>
</table>

REFERENCES | 102 |
LIST OF TABLES

Table 1. Definitions, Acronyms, and Abbreviations ..... 5
Table 2. WD Database Tables Description .................. 75
Table 3. WD Unit Test Results .................................. 79
Table 4. WD Integration Test Results .......................... 90
Table 5. WD System Test Results ............................... 91
Table 6. WD Editing "server.xml" file ......................... 93
Table 7. "web.xml" file ........................................... 94
LIST OF FIGURES

Figure 1. WD Use Case Diagram ........................................ 3
Figure 2. WD Deployment Diagram ........................................ 9
Figure 3. WD Architecture .................................................. 16
Figure 4. WD Architectural Approach ..................................... 19
Figure 5. WD Home Page .................................................... 21
Figure 6. WD Signup Page ................................................... 23
Figure 7. WD Login Page ..................................................... 24
Figure 8. WD Forgot Password Page ....................................... 25
Figure 9. WD Password Display Page ...................................... 27
Figure 10. WD Administration Page ....................................... 28
Figure 11. WD Personal Information Change page ...................... 31
Figure 12. WD Page-Name Change Page ................................... 32
Figure 13. WD Template Page ............................................... 34
Figure 14. No-selection Page ............................................... 37
Figure 15. Text Display Page ................................................ 38
Figure 16. Picture Display Page ............................................. 40
Figure 17. Links Display Page ............................................... 42
Figure 18. WD Background Select Page ................................... 44
Figure 19. WD Menu Choice Page ......................................... 45
Figure 20. WD Page-Title Change Page ................................... 46
Figure 21. WD Unit Choice Page ............................................ 47
Figure 22. WD Text Choice Page ........................................ 49
Figure 23. WD Picture Choice Page ..................................... 50
Figure 24. WD Links Choice Page ....................................... 52
Figure 25. WD Text Preview Page ....................................... 53
Figure 26. WD Picture Preview Page .................................... 54
Figure 27. WD Picture Upload Page ..................................... 56
Figure 28. WD Links Preview Page ...................................... 57
Figure 29. WD Text Input Page .......................................... 59
Figure 30. WD Caption Input Page ...................................... 60
Figure 31. WD Links Input Page .......................................... 61
Figure 32. UserBean Class ................................................ 63
Figure 33. UserRepository Class ........................................ 64
Figure 34. UserRepositoryException Class .......................... 64
Figure 35. PageBean Class ................................................ 65
Figure 36. PageRepository Class ........................................ 66
Figure 37. PageRepositoryException Class .......................... 67
Figure 38. TextBean Class ................................................. 68
Figure 39. TextRepository Class ......................................... 69
Figure 40. TextRepositoryException Class .......................... 69
Figure 41. PictureBean Class ............................................. 70
Figure 42. PictureRepository Class ...................................... 71
Figure 43. PictureRepositoryException Class ....................... 71
Figure 44. LinkBean Class ................................................. 72
Figure 45. LinkRepository Class ......................... 73
Figure 46. LinkRepositoryException Class ............... 73
Figure 47. PictureUpload Class ............................ 74
Figure 48. PictureUploadException Class ................. 74
Figure 49. Signup Process - Sequence Diagram .......... 84
Figure 50. Create New Page Process - Sequence Diagram ... 85
Figure 51. Page-Title Change Process -
Sequence Diagram ........................................ 86
Figure 52. Forgot Password Process - Sequence Diagram ... 87
Figure 53. Login Process - Sequence Diagram ............. 88
Figure 54. Update Request Process - Sequence Diagram .... 89
CHAPTER ONE

SOFTWARE REQUIREMENTS SPECIFICATION

1.1 Introduction

1.1.1 Purpose of the Project

The purpose of the project is to use JSP technology to create a web application that could be used by any person who has the basic knowledge of browsing through the Web. Many people want to create web pages. There are many programs on the market that aid users in creating web pages. But the process involved in creating web pages using the available software and hosting their website is fairly complex for general people. It involves buying software like Microsoft FrontPage or Macromedia Dreamweaver, installing the software, acquiring the knowledge to use it, setting up a web server, and then hosting the web site.

WD can be accessed on-line, from different operating systems or platforms as long as they have a web browser. The users do not have to worry about any of the above steps. They just follow the directions and select different choices and the application will create a web page. It will even host the web site.
1.1.2 Scope of the Project

WD has the following functionality for its users:

- Create a Web page
- Update an existing Web page
- Delete an existing Web page
- Rename an existing Web page
- View the Web page that’s being created or updated.
- View an Existing Web page.
- Change users personal information

Supporting the above functionality, the Use Case Diagram is shown in Figure 1.
1.1.3 Limitations of the Project

A Web page is a HTML document read and interpreted by a Web browser. HTML specification is big and as a result, the scope of the project was limited to the following HTML functionality:

- Text: The following operations related to a text are supported
  - Fonts (Arial, Arial Black, Comic Sans Ms, Courier, Courier New, Georgia, Helvetica,
Impact, Palatino, Times New Roman, Trebuchet Ms and Verdana)

- Style (Bold, Italic and Underline)
- Alignment (Center, Left, Right and Justify)
- Color (Black, Blue, Green, Red, Yellow and White)
- Size (1 through 7)

- Picture: The following operations related to a Picture are supported
  - Caption (this is the text shown below any Picture)
  - Caption Color (Black, Blue, Green, Red, Yellow and White)

- External Links: The following operations related to a Link operation are supported
  - Fonts (Arial, Arial Black, Comic Sans Ms, Courier, Courier New, Georgia, Helvetica, Impact, Palatino, Times New Roman, Trebuchet Ms and Verdana)
  - Alignment (Center, Left, Right and Justify)
  - Color (Black, Blue, Green, Red, Yellow and White)
• Size (1 through 7)

- Internal Links: Users can choose if they want the links to their other pages to appear or not, on a particular page. If they want the Links to appear, they can choose if it would be buttons or as hyperlinks.

- Background: Users can choose if they want any Background for their Web page. If they want a Background, they can choose a picture or a color.

1.1.4 Definitions, Acronyms, and Abbreviations

The definitions, acronyms, and abbreviations used in the document are described in the Table 1.

Table 1. Definitions, Acronyms, and Abbreviations

<table>
<thead>
<tr>
<th>Browser</th>
<th>A program used to view HTML documents.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client/Server Architecture</td>
<td>A network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.</td>
</tr>
<tr>
<td>Database Server</td>
<td>A database server is a computer system that processes database queries.</td>
</tr>
<tr>
<td><strong>DHTML</strong></td>
<td>Dynamic HTML is an extension of HTML giving greater control over the layout of page elements and the ability to have web pages that change and interact with the user without having to communicate with the server.</td>
</tr>
<tr>
<td><strong>HTML</strong></td>
<td>Hypertext Markup Language is a set of tags and rules for using them in developing hypertext documents.</td>
</tr>
<tr>
<td><strong>HTTP</strong></td>
<td>Hypertext transfer protocol is a protocol (utilizing TCP) to transfer hypertext requests and information between servers and browsers.</td>
</tr>
<tr>
<td><strong>Hypertext</strong></td>
<td>Machine-readable text that is not sequential but is organized so that related items of information are connected.</td>
</tr>
<tr>
<td><strong>Internet</strong></td>
<td>Internet is a computer network consisting of a worldwide network of computers that use the TCP/IP network protocol to facilitate data transition and exchange.</td>
</tr>
<tr>
<td><strong>Java</strong></td>
<td>Java is object-oriented, cross-platform programming language from Sun Microsystems.</td>
</tr>
<tr>
<td><strong>JavaScript</strong></td>
<td>Cross-platform, WWW scripting language, only very vaguely related to Java, JavaScript is intimately tied to the <a href="http://WWW">WWW</a>. It runs in only three environments - as a server-side scripting language, as an embedded language in server-parsed HTML, and as an embedded language run in web browsers where it is the most important part of DHTML.</td>
</tr>
<tr>
<td>Java Servlet</td>
<td>A Java program that runs as part of a network service, typically a web server and responds to requests from clients.</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JSP</td>
<td>JavaServer Pages is a specification for extending the Java Servlet API to generate dynamic web pages on a web server.</td>
</tr>
<tr>
<td>MySQL</td>
<td>A popular database server.</td>
</tr>
<tr>
<td>Operating System</td>
<td>Software that controls the execution of computer programs and may provide various services.</td>
</tr>
<tr>
<td>Platform</td>
<td>A specific combination of hardware and operating system and/or compiler.</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol on top of the Internet Protocol provides a reliable, point-to-point communication channel those client-server applications on the Internet use to communicate with each other. To communicate over TCP, a client program and a server program establish a connection to one another. Each program binds a socket to its end of the connection. To communicate, the client and the server each reads from and writes to the socket bound to the connection.</td>
</tr>
<tr>
<td>UML</td>
<td>The Unified Modeling Language is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It simplifies the complex process of software design, making a &quot;blueprint&quot; for construction.</td>
</tr>
<tr>
<td>Web Application</td>
<td>A program or group of programs designed for end users, which runs on WWW.</td>
</tr>
</tbody>
</table>
A Web page is a HTML document read and interpreted by a Web browser.

A web server is an application, which would serve contents to the WWW.

World Wide Web is a computer network consisting of a collection of Internet sites that offer text and graphics and sounds and animation resources through the HTTP.

World Wide Web Consortium is the main standards body for the WWW.

Extensible Markup Language is an interactive from the W3C defining an "extremely simple" dialect of Standard Generalized Markup Language suitable for use on the WWW.

1.1.5 Overview

The remainder of this chapter defines the functions and specific requirements of WD in a format consistent with the IEEE Std 830-1993 SRS format [1] and IEEE Std 830-1998 SRS format [2].

1.2 Overall Description

1.2.1 Product Perspective

WD is a web application that helps end users create and host their web pages. WD makes use of Client/Server
architecture and serves through the remote server through the WWW.

Figure 2 shows the deployment of WD. WD will contain two central servers – an application server and a database server, both run on server side. On the client side any web browser is used. The server and the clients use TCP/IP protocols to communicate.

![WD Deployment Diagram](image)

Figure 2. WD Deployment Diagram

1.2.2 System Interfaces

Any web browser will provide the interfaces to the system. The recommended browsers will be Internet Explorer version 4.0 or higher and Netscape Navigator version 4.0 or higher.

1.2.3 User Interfaces

1.2.3.1 WD Home Page. The WD Home Page will have a description of the application and it will have a hyperlink
"Signup Now", for the new users to signup. When clicked, it will take the users to WD Signup Page. The WD Home Page will also have a hyperlink "Login". When clicked, it will take the users to the WD Login Page.

1.2.3.2 WD Signup Page. The WD Signup Page will have the following text-fields "First Name", "Last Name", "Username", "Enter Password", "Retype Password", "E-Mail", "Select a Secret Question" (a select list), "Answer to the Secret Question", and a "Signup" button. Once the users click on the "Signup" button, WD will process the registration form and if it is successful, they will go to the File Management Page, where the "Signup Successful" message is displayed.

1.2.3.3 WD Login Page. The WD Login Page will have "Username" and "Password" text-fields and a "Login" button. All registered users can enter their respective username and password and when they click on the "Login" button, WD will process the information and if it is successful, they will go to the WD Administration Page, where the "Login Successful" message is displayed.

1.2.3.4 WD Administration Page. The WD Administration Page will have a list of pages designed by the user. The user can view, update, rename and delete the different
pages. If the user wants to create a new page, he/she can do so by typing a new page name in the text field "Enter Page Name" and clicking on the "Create a New Page" button. Also there will be provision on this page for users to logout and to change their personal information.

1.2.3.5 WD Template Page. The WD Template Page is opened when a user is updating an existing web page or creating a new web page. Here users can manage the web page's background, menu, and content. Users can also view the web page they are working on, at any time. It also has a link for users to logout.

1.2.4 Hardware Interfaces

WD will not directly implement any hardware interfaces. All interfaces to I/O devices will be provided by the operating system.

1.2.5 Software Interfaces

Since WD is a web application, a web server is required to serve the application, JDK 1.2 for software development, and a web browser, which can support Java Foundation Classes.

The following browsers/platforms will be supported at the minimum:
• Operating Systems Supported:
  o Windows 98/NT.
  o LINUX 5.2+.
  o Solaris

• Browsers:
  o Netscape 4.0+
  o Internet Explorer 4.0+

• Servers:
  o Any server that supports JSP.

1.2.6 Communications Interfaces
Since WD is a web application, all updates and connections between the central server and the distributed clients are established by TCP/IP protocols.

1.2.7 User Characteristics
The primary users of WD would be any person who is interested in creating and hosting their own web pages. The users are not expected to be familiar with HTML or any web page development tools. Also, it is not necessary for the users of WD to know Java or any programming language. WD uses the point-and-click technique. Only a basic knowledge of using Internet browser menus according to the system instructions is enough.
1.2.8 Logical Database Requirements

WD is a database driven application. On the server side, there will be a central database server like SQL server to manage the data. On the client side, users just need a web browser to access the WD application. There are no database requirements on the client side.

1.3 Summary

This Chapter describes the software requirements for the WD. It covered the advantages of web applications, scope and limitations of WD, methodology, various technical terms and other basic requirements for the project.

In the next chapter WD architecture is explained in detail and the implementation details are discussed.
CHAPTER TWO
SOFTWARE DESIGN

2.1 Overview

In this chapter WD Methodology, WD Architecture, and WD Detailed Design are explained.

2.2 WD Methodology

WD is developed as a web application to make use of web technology and its advantages. This made WD cross platform application and easily available to its users.

The framework is based on client (remote clients)/server (central server) model. The remote clients would send requests to the central server and the central server would process the requests and publish the response. The central server and the clients use TCP/IP protocol to communicate with each other. The deployment diagram of WD was shown in Figure 2 (CHAPTER ONE).

2.2.1 Central Server

The central server has an application server, a database server and all other required data.

2.2.2 Remote Client

The client could be run from any computer, which has a web browser.
2.3 WD Architecture

The high level architecture of WD has three logical areas, which can be seen as the following three layers:

- The Presentation layer, which controls the look and feel. It also takes the requests and delivers results.
- The Control layer, which controls the application flow.
- The Application Logic layer, which manages application data, performs tasks and communicates with back-end resources.

This three-layer architecture is an implementation of a design pattern in software development - the MVC (Model-View-Controller) pattern [3]. The MVC pattern is concerned with separating the information (the Model) from its presentation (the View).

It is advantageous to treat each layer as an independent portion of the application. Isolating the logical portions of the application helps ensure a robust architecture and lays the groundwork for the implementation. Each layer plays an important role in the
WD's architecture and they are discussed briefly in the following sections. Figure 8 shows the WD Architecture.

![WD Architecture Diagram](image)

Figure 3. WD Architecture

### 2.3.1 Presentation Layer

This layer includes the client side display elements, such as HTML, and JSP. The presentation layer can be thought of as the user interface for the application because it is used to get input from the end user and display the application's results. In the MVC paradigm, the presentation layer fills the role of the View.

The presentation layer is not concerned with how the information is obtained, or from where. Its responsibilities lie only in displaying the information itself while delegating any other activity up the chain to other layers. For example, in an application that involves submitting a search query through a web form, only the form itself and the corresponding results are
the responsibilities of the presentation layer. It is not accountable for what happens in-between, namely the processing of the request and the retrieval of the results.

2.3.2 Application Logic Layer

In the MVC paradigm, the application logic layer fills the role of the Model. The application logic layer is the heart of the application, responsible for actually doing whatever the application is supposed to do. For example, it is responsible for performing queries against a database, calculating sales tax, or processing orders. It is an encapsulation of data and behavior that is independent of its presentation.

Unlike the presentation layer, this layer attends only to storing, manipulating, and generating data, not displaying it. For this reason, components designed to work as application logic can be relocated outside web-based applications, since the behavior they encapsulate isn’t web-centric.

2.3.3 Control Layer

The control layer determines the application’s flow, serving as intermediary between the presentation layer and the application logic layer. This layer serves as the
logical connection between the user’s interaction with the front-end and services on the back end. In the MVC pattern this layer acts as the Controller. It delivers the model to the view and regulates communication between the two layers.

This layer is also responsible for making decisions among multiple presentations, when available. For example, in the process of retrieving forgotten password information, the users enter the data and submit the form. If the information entered is correct, the password information has to be displayed. Otherwise a “no-match found” message has to be displayed.

Each request enters the application through the control layer. The control layer decides how the request should be handled and delivers it to the application logic layer for processing. When the application logic layer has completed its operation, the control layer directs the request back to the user via the presentation layer.
2.4 WD Detailed Design

2.4.1 Architectural Approach

Separating an application into presentation, control and application logic subsystems make it easier to develop, understand and maintain. For that reason, JSP and HTML pages are used to implement the presentation layer, JSP is used to implement the control layer and Java classes are used to implement the application logic layer. MySQL is used as the back-end database server.

![Diagram of Architectural Approach](image)

Figure 4. WD Architectural Approach

2.4.2 WD Presentation Layer

This layer has all the user interfaces. User interfaces are implemented by using JSP, as per the specifications described in section 1.2.3.

JavaScript is used for form validations in user interfaces.
Also HTML pages are used to make interactive help menus.

2.4.2.1 WD Home Page. This page is implemented by the JSP file index.jsp. The WD Home Page (refer to Figure 5) has a description of the application. In order to use this application a user has to register himself and every time he visits the application he has to login. WD Home Page has a hyperlink "Signup Now" which opens WD Signup Page, a hyperlink "Login" which opens WD Login Page, and a hyperlink "Forgot Password?" which opens WD Forgot Password Page. These requests are handled by the WD Home Page itself. It does not forward these requests to the control layer.
Website Developer is a web application, which will aid in design and hosting websites to the end users. Any user who intends to design and host his/her own website without any knowledge of any web tools can do so by logging on to this application. This application will provide a template which gives the users flexibility to choose different combinations of text matter, pictures and links. This template also gives the users flexibility with the layout of the webpage. When the users use this template a website is automatically created. There is also provision for the user to view & edit the page, the moment it is designed. There is also a provision to manage all the pages created.

So just go ahead and Sign up and start creating your web pages.

Figure 5. WD Home Page

2.4.2.2 WD Signup Page. This page is implemented by the JSP file signup.jsp. The WD Signup Page (refer to Figure 6) has the following text-fields “First Name”, “Last Name”, “Username”, “Enter Password”, “Retype Password”, “E-Mail”, “Select a Secret Question” (a select list), “Answer to the Secret Question”, and a “Signup” button.
button. All fields are mandatory. When a user clicks on the “Signup” button, JavaScript function will check if all the fields are filled. If that is the case WD Signup Page sends the request to the signupProcess.jsp in the control layer. Otherwise the request is terminated and the names of the incomplete fields are displayed.

If the new user signup process is successful, signupProcess.jsp will open WD Administration Page. Otherwise a message is displayed on the WD Signup Page.
2.4.2.3 WD Login Page. This page is implemented by the JSP file login.jsp. The WD Login Page (refer to Figure 7) has the text fields “Username”, and “Password” and a “Login” button. All registered users can enter their respective username and password and when they
click on the "Login" button, a request is sent to the loginProcess.jsp in the control layer.

Figure 7. WD Login Page

If the login process is successful, loginProcess.jsp will open WD Administration Page. Otherwise a message is displayed on the WD Login Page.

2.4.2.4 WD Forgot Password Page. This page is implemented by the JSP file forgot.jsp. The WD Forgot
Password Page (refer to Figure 8) has the text fields “Enter your E-Mail Address”, “Select your Secret Question” (a select list), “Answer to Secret Question”, and a “Submit” button. When the user clicks “Submit” button, a request is forwarded to the forgotProcess.jsp in the control layer.

![Forgot Password Page](image-url)

**Figure 8. WD Forgot Password Page**

If the password retrieve process is successful, WD Password Display Page will open and the password
information will be displayed on that page. Otherwise a "Match Not Found" message will be displayed on the WD Forgot Password Page.

2.4.2.5 WD Password Display Page. This page is implemented by the JSP file passwordDisplay.jsp. The WD Password Display Page (refer to Figure 9) displays the username and password of the user. This page also has hyperlinks that open WD Home Page, WD Login Page and WD Signup page.
Here is your Information:

Your Username is: k
Your password is: k

2.4.2.6 WD Administration Page. This page is implemented by the JSP file userAdminPage.jsp. The WD Administration Page (refer to Figure 10) displays the list of web pages designed by the user. The user can view, update, rename and delete these existing web pages. The users can also create a new web page or change their personal information.
Figure 10. WD Administration Page

- **Changing Personal Information:** When the users click on the hyperlink “Change Personal Information”, the WD Personal Information Change Page is opened.

- **View an Existing Web Page:** When the users click on the hyperlink “View”, next to a web page, it will open that web page in a new Window.
• Delete an Existing Web Page: When the users select a web page and Click "Delete" button. The request will be forwarded to the pageDeleteProcess.jsp in the control layer. It will process the request and WD Administration Page is refreshed.

• Change an Existing Web Page's Name: The users can select a web page and click "Change Page Name" button. That opens the WD Page-Name Change Page.

• Update an Existing Web Page: The users can select a web page and click the "Update" button. That will forward the request to userAdminProcess.jsp in the control layer, which in turn opens the WD Template Page.

• Create a New Web Page: The users can type a page name in the text field "Enter Page Name" and click "Create a New Page" button. That will forward the request to userAdminProcess.jsp in the control layer, which in turn opens the WD Template Page.
• Logout: If the users want to logout, they can click on the hyperlink "Logout". That will end the user's session and opens the WD Login Page.

2.4.2.7 WD Personal Information Change Page. This page is implemented by the JSP file personalInfo.jsp. The WD Personal Information Change Page (refer to Figure 11) is similar to the WD Signup Page except for the user's username is displayed instead of a text field "Username". The users can change all other fields and when they click the "Update" button, the request is forwarded to personalInfoProcess.jsp in the control layer.
If the update process is successful, personalInfoProcess.jsp will open WD Administration Page. Otherwise a message is displayed on the WD Personal Information Change Page.
2.4.2.8 WD Page-Name Change Page. This page is implemented by the JSP file linknameChange.jsp. The Page-Name Change Page (refer to Figure 12) displays the page name of the selected web page and it has a text field to fill in the new page name. When the user clicks the “Change” button, the request is forwarded to linknameChangeProcess.jsp in the control layer.

Figure 12. WD Page-Name Change Page

If the update process is successful, linknameChangeProcess.jsp will open WD Administration
Page. Otherwise a message is displayed on the WD Page-Name Change Page.

2.4.2.9 WD Template Page. This page is implemented by the JSP file template.jsp. The WD Template Page (refer to Figure 13) is opened when a user is updating an existing web page or creating a new web page.

The WD Template Page is made up of different pages. It has seven frames. The top frame is implemented by myMenu.jsp. The second frame from top has Page-Title Display Page, which is implemented by pageTitleDisplay.jsp. The left frame has Menu Display Page, which is implemented by menuDisplay.jsp. The main frame has four frames corresponding to four unit locations. Each frame has either No-Selection Page, Text Display Page, Picture Display Page, or Links Display Page, which are implemented by nothing.jsp, textDisplay.jsp, pictureDisplay.jsp, and linksDisplay.jsp respectively.

On the WD Template Page the users can manage the web page’s background, menu, and its content. At any time users can view the web page they are working on, or go to the WD Administration Page, or logout.
Figure 13. WD Template Page

- Change Background: When users click on the hyperlink "Change Background", WD Background Select Page will open in a new window.

- Change Menu: When users click on the "Change Menu" button, WD Menu Choice Page will open in a new window.

- Change Page Title: When users click on the "Change Title" button, WD Page-Title Change Page will open in a new window.
• Manage Content of the Web Page: Based on what users have in the four unit locations, the corresponding display pages (No-Selection Page, Text Display Page, Picture Display Page, or Links Display Page) appear. If the user has just started creating a new web page, No-Selection Page will appear in all the unit locations. The functionality of the above four pages is described in the sections 2.4.2.10 through 2.4.2.13.

• View the Web Page that’s being Created or Updated: At any point users can click on the hyperlink “View”. That will open a new window displaying the web page that users are presently creating or updating.

• Go to Administration Page: When the user click on the hyperlink “My Admin Page”, WD will save the web page they were working-on and open the WD Administration Page.

• Logout: If the user wants to logout, they can click on the hyperlink “Logout”. That will save
the web page they were working-on, end the user’s session, and open the WD Login Page.

2.4.2.10 No-Selection Page. This page is implemented by the JSP file nothing.jsp. The No-Selection Page (refer to Figure 14) has "<< Show Choice" button. If the users want text or a picture or external hyperlinks in that unit location, they can click "<< Show Choice" button and WD Unit Choice Page will open in a new window.
Figure 14. No-Selection Page

2.4.2.11 Text Display Page. This page is implemented by the JSP file textDisplay.jsp. The Text Display Page (refer to Figure 15) has three Buttons followed by the text.
Figure 15. Text Display Page

- The users can change the text by clicking on the “Change >” button. It will open WD Text Input Page in a new window.
- The users can choose a different text by clicking on the “< Text Choice” button. It will open WD Text Choice Page in a new window.
The users can choose a different option (like a picture or links or blank) by clicking on the "<< Other Choice" button. It will open WD Unit Choice Page in a new window.

2.4.2.12 Picture Display Page. This page is implemented by the JSP file pictureDisplay.jsp. The Picture Display Page (refer to Figure 16) has three buttons followed by the caption and the thumbnail picture.
The users can change the caption or caption color by clicking on the "Change >" button. It will open WD Caption Input Page in a new window.

The users can choose a different picture by clicking on the "< Picture Choice" button. It
will open WD Picture Choice Page in a new window.

- The users can choose a different option (like text or links or blank) by clicking on the "<< Other Choice" button. It will open WD Unit Choice Page in a new window.

2.4.2.13 Links Display Page. This page is implemented by the JSP file linksDisplay.jsp. The Links Display Page (refer to Figure 17) has three buttons followed by the external hyperlinks.
The users can change the links by clicking on the "Change >" button. It will open WD Links Input Page in a new window.

The users can choose a different set of links by clicking on the "< Links Choice" button. It will open WD Links Choice Page in a new window.
• The users can choose a different option (like text or a picture or blank) by clicking on the "<< Other Choice" button. It will open WD Unit Choice Page in a new window.

2.4.2.14 WD Background Select Page. This page is implemented by the JSP file background.jsp. The WD Background Select Page (refer to Figure 18) will have option to choose a background color or picture. That will be the new background for the web page.
When users select an option and click "Select" button, the request is forwarded to backgroundProcess.jsp in the control layer. This will update the background, close the window and refresh the WD Template Page.

2.4.2.15 WD Menu Choice Page. This page is implemented by the JSP file menuChoice.jsp. The WD Menu Choice Page (refer to Figure 19) gives the users an
option to choose the style of the internal links that are displayed on the web page.

![WD Menu Choice Page](image)

Figure 19. WD Menu Choice Page

These links connect to the users other web pages. The users select an option - Buttons, Links or Nothing (if they don’t want the links to their other web pages to appear on this web page) and click “Select” Button. Then the request is forwarded to menuProcess.jsp in the control layer. This will update the look of the menu, close the window, and refresh the WD Template Page.
2.4.2.16 WD Page-Title Change Page. This page is implemented by the JSP file pageTitleTemplate.jsp. The WD Page-Title Change Page (refer to Figure 20) has a text field, where the users can enter the new page-title. Then the users can select the font, style, alignment, color and size for this title.

![Figure 20. WD Page-Title Change Page](image)

When the users click "Submit" button, the request is forwarded to pageTitleProcess.jsp in the control layer. This will update the page-title, close the window and refresh the WD Template Page.
2.4.2.17 WD Unit Choice Page. This page is implemented by the JSP file unitChoice.jsp. The WD Unit Choice Page (refer to Figure 21) gives an option for the users to choose if they want a text paragraph, or a picture with caption, or external hyperlinks, or just leave the unit location blank.

Figure 21. WD Unit Choice Page
When users select an option and click "Proceed >" button, the request will be forwarded to unitProcess.jsp in the control layer.

- If the users select "Text", it will open the WD Text Choice Page.
- If the users select "Picture", it will open the WD Picture Choice Page.
- If the users select "Links", it will open the WD Links Choice Page.
- If the users select "Nothing", it will close the window and No-Selection Page will appear in the corresponding unit location, on the WD Template Page.

2.4.2.18 WD Text Choice Page. This page is implemented by the JSP file textChoice.jsp. The WD Text Choice Page (refer to Figure 22) gives the users an option to choose a text paragraph template provided by the WD, or a text paragraph that user has created earlier, or a new text paragraph.
Figure 22. WD Text Choice Page

- If the users want to continue, they select an option and a title, if applicable, and then click "Preview >" button. That will open the WD text Preview Page.
- If the users want to choose a different option (like a picture or links or blank), they can click on the "< Other Choice" button. It will open the WD Unit Choice Page.
2.4.2.19 WD Picture Choice Page. This page is implemented by the JSP file pictureChoice.jsp. The WD Picture Choice Page (refer to Figure 23) gives the users an option to choose a picture provided by the WD, or a picture that user has uploaded earlier.

![Picture Choice Page](image)

Figure 23. WD Picture Choice Page

- If the users want to continue, they select an option and a title (from the list) and then
click "Preview >" button. That will open the WD Picture Preview Page.

- If the users want to upload a new picture, they can click on the "Upload a New Picture from your Computer" button. That will open the WD Picture Upload Page.

- If the users want to choose a different option (like text or links or blank), they can click on the "< Other Choice" button. That will open the WD Unit Choice Page.

**2.4.2.20 WD Links Choice Page.** This page is implemented by the JSP file linksChoice.jsp. The WD Links Choice Page (refer to Figure 24) gives the users an option to choose a set of links provided by the WD, or a set of links that user has created earlier, or a set of new links.
If the users want to continue, they can select an option and a title, if applicable, and then click "Preview >" button. That will open the WD Links Preview Page.

If the users want to choose a different option (like text or a picture or blank), they can click on the "< Other Choice" button. That will open the WD Unit Choice Page.
2.4.2.21 WD Text Preview Page. This page is implemented by the JSP file textPreview.jsp. The WD Text Preview Page (refer to Figure 25) displays the text paragraph that the users choose.

![WD Text Preview Page](image)

Figure 25. WD Text Preview Page

- If they want to continue, they can click the "Customize & Save >" button. That will open the WD Text Input Page.
• If they want to choose a different text paragraph, they can click on the "< Text Choice" button. That will open the WD Text Choice Page.

2.4.2.22 WD Picture Preview Page. This page is implemented by the JSP file picturePreview.jsp. The WD Picture Preview Page (refer to Figure 26) displays the picture that the users choose.

Figure 26. WD Picture Preview Page
• If they want to continue, they can click the "Customize & Save>" button. That will open the WD Caption Input Page.

• If they want to choose a different picture, they can click on the "< Picture Choice" button. It will open the WD Picture Choice Page.

2.4.2.23 WD Picture Upload Page. This page is implemented by the JSP file pictureUpload.jsp. The WD Picture Upload Page (refer to Figure 27) has a text field for title of the picture, a "Browse" button to choose a picture file from the remote user's computer and a text field for caption and a list of colors to choose for the caption.
If the users want to continue, they type and select all the fields and then click the "Upload" button. That will forward the request to pictureUploadProcess.jsp in the control layer.

If the upload process is successful that window will close and Picture Display Page will appear in the corresponding unit location, on the WD Template Page.
If the upload process is not successful, WD Picture Upload Page will open with a message.

- If they want to choose a different picture, they can click on the "< Back" button. That will open the WD Picture Choice Page.

2.4.2.24 WD Links Preview Page. This page is implemented by the JSP file linksPreview.jsp. The WD links preview page (refer to Figure 28) displays the links that the users choose.

Figure 28. WD Links Preview Page
• If they want to continue, they can click the "Customize & Save >" button. That will open the WD Links Input Page.

• If they want to choose different links, they can click on the "< Links Choice" button. That will open the WD Links Choice Page.

2.4.2.25 WD Text Input Page. This page is implemented by the JSP file textTemplate.jsp. The WD Text Input Page (refer to Figure 29) has a text area, where the users can type the text. Then they can select the font, style, alignment, color and size for that text paragraph. When they click "Save >" button, the request is forwarded to textProcess.jsp in the control layer. textProcess.jsp will do the updates, close this window, and refresh the WD Template Page. The updated text is displayed in the corresponding unit location, on the WD Template Page.
Figure 29. WD Text Input Page

2.4.2.26 WD Caption Input Page. This page is implemented by the JSP file pictureTemplate.jsp. The WD Caption Input Page (refer to Figure 30) has a text field for title, a text field for caption and a list of colors for the caption.
The users type in the text fields and select a color. Then when they click "Save >" button, the request is forwarded to pictureProcess.jsp in the control layer. pictureProcess.jsp will do the updates, closes this window, and refresh the WD Template Page. The updated picture caption will appear in the corresponding unit location, on the WD Template Page.

2.4.2.27 WD Links Input Page. This page is implemented by the JSP file linksTemplate.jsp. The WD Links Input Page (refer to Figure 31) has a text field for title, and four pairs of text fields for link name and URL.
The users type in the text fields. Then they can select the font, alignment, color and size for these links. Then when they click “Save >” button, the request is forwarded to linksProcess.jsp in the control layer. linkProcess.jsp will do the updates, closes this window, and refresh the WD Template Page. The updated links will appear in the corresponding unit location, on the WD Template Page.
2.4.3 WD Control Layer


These JSP files make use of different functions of the Java classes in the application logic layer. And they direct the application flow to the appropriate user interface.

2.4.4 WD Application Logic Layer

This layer has various Java class files. All these Java class files are under one package - the wd package.

The wd package is made up of the Java classes - UserBean, UserRepository, UserRepositoryException, PageBean, PageRepository, PageRepositoryException, TextBean, TextRepository, TextRepositoryException, LinkBean, LinkRepository, LinkRepositoryException, PictureBean, PictureRepository,
PictureRepositoryException, PictureUpload, PictureUploadException, Request, and File.

2.4.4.1 UserBean. This class maps with the database table WDUsers. It has getter and setter methods for each parameter. The class diagram is shown in the Figure 32.

Figure 32. UserBean Class

2.4.4.2 UserRepository. This class has the functions that are related to users and they manipulate
the WDUsers table in the database. The class diagram is shown in the Figure 33.

![Class Diagram](image)

**Figure 33. UserRepository Class**

**2.4.4.3 UserRepositoryException.** This class handles the exceptions thrown in the UserRepository class. The class diagram is shown in the Figure 34.

![Class Diagram](image)

**Figure 34. UserRepositoryException Class**
2.4.4.4 PageBean. This class maps with the database table WDPages. It has getter and setter methods for each parameter. The class diagram is shown in the Figure 35.

```
<table>
<thead>
<tr>
<th>Class: PageBean</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_name : String</td>
</tr>
<tr>
<td>page_name : String</td>
</tr>
<tr>
<td>page_title : String</td>
</tr>
<tr>
<td>background : String</td>
</tr>
<tr>
<td>id1 : String</td>
</tr>
<tr>
<td>id2 : String</td>
</tr>
<tr>
<td>id3 : String</td>
</tr>
<tr>
<td>id4 : String</td>
</tr>
<tr>
<td>link_type : String</td>
</tr>
<tr>
<td>last_modified : Date</td>
</tr>
</tbody>
</table>

- getUsername() : String
- getPageName() : String
- getPageTitle() : String
- getBackground() : String
- getId1() : String
- getId2() : String
- getId3() : String
- getId4() : String
- getLinkType() : String
- getLastModified() : Date
- setUsername(String un)
- setPageName(String pn)
- setPageTitle(String pt)
- setBackground(String bg)
- setId1(String id1)
- setId2(String id2)
- setId3(String id3)
- setId4(String id4)
- setLinkType(String lt)
- setLastModified(Date lm)
```

Figure 35. PageBean Class
2.4.4.5 PageRepository. This class has the functions that are related to user web pages and they manipulate the WDPages table in the database. The class diagram is shown in the Figure 36.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getUserPages(String uname)</td>
<td>Retrieve user pages</td>
</tr>
<tr>
<td>deletePage(String username, String selectpage)</td>
<td>Delete page</td>
</tr>
<tr>
<td>createPage(String uname, String page)</td>
<td>Create page</td>
</tr>
<tr>
<td>getBackgrounds()</td>
<td>Get backgrounds</td>
</tr>
<tr>
<td>putPageBean(PageBean pb)</td>
<td>Save page bean to database</td>
</tr>
<tr>
<td>getPageBean(String un, String In)</td>
<td>Get page bean from database</td>
</tr>
<tr>
<td>putPageTitle(String un, String In, String pt)</td>
<td>Save page title to database</td>
</tr>
<tr>
<td>getPageTitle(String un, String In)</td>
<td>Get page title from database</td>
</tr>
<tr>
<td>updatesId(String un, String In, String id, String idval)</td>
<td>Update page id</td>
</tr>
<tr>
<td>checkMatch(String username, String colname, String colvalue)</td>
<td>Check match condition</td>
</tr>
<tr>
<td>getId(String un, String In, String id)</td>
<td>Get page id from database</td>
</tr>
</tbody>
</table>

Figure 36. PageRepository Class
2.4.4.6 PageRepositoryException. This class handles the exceptions thrown in the PageRepository class. The class diagram is shown in the Figure 37.

![PageRepositoryException class diagram](image)

Figure 37. PageRepositoryException Class

2.4.4.7 TextBean. This class maps with the database table WDTExt. It has getter and setter methods for each parameter. The class diagram is shown in the Figure 38.
Figure 38. TextBean Class

2.4.4.8 TextRepository. This class has the functions that are related to the text paragraphs in the user web pages and they manipulate the WDText table in
the database. The class diagram is shown in the Figure 39.

![Figure 39. TextRepository Class](image)

### 2.4.4.9 TextRepositoryException

This class handles the exceptions thrown in the TextRepository class. The class diagram is shown in the Figure 40.

![Figure 40. TextRepositoryException Class](image)
2.4.4.10 PictureBean. This class maps with the database table WDPicture. It has getter and setter methods for each parameter. The class diagram is shown in the Figure 41.

![Class Diagram](image)

Figure 41. PictureBean Class

2.4.4.11 PictureRepository. This class has the functions that are related to the pictures in the user web pages and they manipulate the WDPicture table in the database. The class diagram is shown in the Figure 42.
Figure 42. PictureRepository Class

2.4.4.12 PictureRepositoryException. This class handles the exceptions thrown in the PictureRepository class. The class diagram is shown in the Figure 43.

Figure 43. PictureRepositoryException Class

2.4.4.13 LinkBean. This class maps with the database table WDLinks. It has getter and setter methods for each parameter. The class diagram is shown in the Figure 44.
Figure 44. LinkBean Class

2.4.4.14 LinkRepository. This class has the functions that are related to the external links in the user web pages and they manipulate the WDLinks table in the database. The class diagram is shown in the Figure 45.
Figure 45. LinkRepository Class

2.4.4.15 LinkRepositoryException. This class handles the exceptions thrown in the LinkRepository class. The class diagram is shown in the Figure 46.

Figure 46. LinkRepositoryException Class

2.4.4.16 PictureUpload. This class has the functions that are related to uploading pictures from the remote client computers. The class diagram is shown in the Figure 47.
2.4.4.17 PictureUploadException. This class handles the exceptions thrown in the PictureUpload class. The class diagram is shown in the Figure 48.

Figure 47. PictureUpload Class

Figure 48. PictureUploadException Class

2.4.5 WD Database Design

MySQL database server is used to implement the back-end database. The tables in the WD Database are - WDUsers, WDPages, WDText, WDPicture, WDLinks. These database tables are described in the Tables 2.
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDUsers</td>
<td>user_name</td>
<td>varchar(20)</td>
<td>Primary Key; User’s Login Username</td>
</tr>
<tr>
<td></td>
<td>password</td>
<td>varchar(20)</td>
<td>User’s Login Password</td>
</tr>
<tr>
<td></td>
<td>first_name</td>
<td>varchar(15)</td>
<td>User’s First Name</td>
</tr>
<tr>
<td></td>
<td>last_name</td>
<td>varchar(15)</td>
<td>User’s Last Name</td>
</tr>
<tr>
<td></td>
<td>email</td>
<td>varchar(35)</td>
<td>User’s E-mail</td>
</tr>
<tr>
<td></td>
<td>secret_id</td>
<td>int(11)</td>
<td>Secret Question ID</td>
</tr>
<tr>
<td></td>
<td>secret_ans</td>
<td>varchar(15)</td>
<td>Secret Answer</td>
</tr>
<tr>
<td></td>
<td>Registration _date</td>
<td>timestamp(14)</td>
<td>Registration Time</td>
</tr>
<tr>
<td>WDPages</td>
<td>user_name</td>
<td>varchar(20)</td>
<td>Primary Key; User name</td>
</tr>
<tr>
<td></td>
<td>link_name</td>
<td>varchar(40)</td>
<td>Primary Key; Page Name</td>
</tr>
<tr>
<td></td>
<td>page_title</td>
<td>varchar(44)</td>
<td>Page Title</td>
</tr>
<tr>
<td></td>
<td>background</td>
<td>varchar(64)</td>
<td>Background Path</td>
</tr>
<tr>
<td></td>
<td>id1</td>
<td>varchar(21)</td>
<td>Unit Title</td>
</tr>
<tr>
<td></td>
<td>id2</td>
<td>varchar(21)</td>
<td>Unit Title</td>
</tr>
<tr>
<td></td>
<td>id3</td>
<td>varchar(21)</td>
<td>Unit Title</td>
</tr>
<tr>
<td></td>
<td>id4</td>
<td>varchar(21)</td>
<td>Unit Title</td>
</tr>
<tr>
<td></td>
<td>link_type</td>
<td>varchar(20)</td>
<td>Internal Links Display Type</td>
</tr>
<tr>
<td></td>
<td>last_modified</td>
<td>timestamp (14)</td>
<td>Modification Time</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------</td>
<td>----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>WDText</strong></td>
<td>user_name varchar(20)</td>
<td>Primary Key; User Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>title varchar(20)</td>
<td>Primary Key; Text Title</td>
<td></td>
</tr>
<tr>
<td></td>
<td>text text</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td></td>
<td>font varchar(20)</td>
<td>Text Font</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bold varchar(10)</td>
<td>Text Style</td>
<td></td>
</tr>
<tr>
<td></td>
<td>italic varchar(10)</td>
<td>Text Style</td>
<td></td>
</tr>
<tr>
<td></td>
<td>underline varchar(10)</td>
<td>Text Style</td>
<td></td>
</tr>
<tr>
<td></td>
<td>size varchar(10)</td>
<td>Text Size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>color varchar(10)</td>
<td>Text Color</td>
<td></td>
</tr>
<tr>
<td></td>
<td>align varchar(10)</td>
<td>Text Alignment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>last_modified timestamp (14)</td>
<td>Modification Time</td>
<td></td>
</tr>
<tr>
<td><strong>Picture</strong></td>
<td>user_name varchar(20)</td>
<td>Primary Key; User Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>title varchar(20)</td>
<td>Primary Key; Picture Title</td>
<td></td>
</tr>
<tr>
<td></td>
<td>path varchar(64)</td>
<td>Picture Path</td>
<td></td>
</tr>
<tr>
<td></td>
<td>caption varchar(64)</td>
<td>Picture Caption</td>
<td></td>
</tr>
<tr>
<td></td>
<td>color varchar(10)</td>
<td>Caption Color</td>
<td></td>
</tr>
<tr>
<td></td>
<td>last_modified timestamp (14)</td>
<td>Modification Time</td>
<td></td>
</tr>
<tr>
<td><strong>WDLinks</strong></td>
<td>user_name varchar(20)</td>
<td>Primary Key; User Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>title varchar(20)</td>
<td>Primary Key; Picture Title</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Link_name varchar(60)</td>
<td>Link Name</td>
<td></td>
</tr>
</tbody>
</table>
In this chapter WD methodology, WD architecture, and WD detailed design are explained.

In the next chapter WD testing process is explained with the help of sequence diagrams.
CHAPTER THREE

TESTING

3.1 Overview

Testing is the process of exercising a product to identify differences between expected and actual behavior. It is one of the most important steps in the software life cycle. Testing is necessary to ensure the utility, reliability, robustness, and performance of the software product.

The testing plan for WD was based on typical bottom-up testing plan. It consists of unit testing, integration testing, and system testing. The following sections describe each of them.

3.2 Unit Testing

Unit testing is a process for validating a small part of a complex software system. Unit testing was performed as the different parts of the WD were developed.

3.2.1 Test Cases

Each part (test case) is comprised of one or more program files from different layers (presentation, control and application logic) of the application. The required data was manually inputted in the back-end database.
3.2.2 Results

The different units of WD and the unit test results are shown in Table 3.

Table 3. WD Unit Test Results

<table>
<thead>
<tr>
<th>Unit</th>
<th>Test Performed</th>
<th>Result</th>
</tr>
</thead>
</table>
| Signup Process        | • Check validation of the signup form  
                         • Check signup processing  
                         • Check the application flow | Pass   |
| Login Process         | • Check validation of the login form  
                         • Check login processing  
                         • Check the application flow | Pass   |
| Forgot Password Process | • Check validation of the forgot password form  
                           • Check forgot password processing  
                           • Check the application flow | Pass   |
| Change Personal Information | • Check validation of the personal information change form  
                               • Check forgot personal information change processing  
                               • Check the application flow | Pass   |
| WD Administration Page | • Check if this page displays the user web pages in a table  
                           • Check the validation of this table  
                           • Check the hyperlinks for application flow | Pass   |
<table>
<thead>
<tr>
<th>Unit</th>
<th>Test Performed</th>
<th>Result</th>
</tr>
</thead>
</table>
| View an Existing Web Page    | • Check validation of the hyperlink "View" on the WD Administration Page  
• Check the application flow                                                                                                                                                                                                                                                        | Pass   |
| Delete an Existing Web Page  | • Check page delete processing  
• Check the application flow                                                                                                                                                                                                                                                     | Pass   |
| Change the name of an Existing Web Page | • Check page name change processing  
• Check the application flow                                                                                                                                                                                                                                                      | Pass   |
| Update an Existing Web Page  | • Check the application flow  
• Check if the selected page is displayed on the WD Template Page                                                                                                                                                                                                                     | Pass   |
| Create a New Web Page        | • Check validation of the create new page form  
• Check create new page processing  
• Check the application flow                                                                                                                                                                                                                                                   | Pass   |
| WD Template Page             | • Check the hyperlinks for application flow                                                                                                                                                                                                                                                                                                  | Pass   |
| Page Title Change Process    | • Check the validation of the page title change form  
• Check the page title change processing  
• Check the application flow                                                                                                                                                                                                                                                     | Pass   |
| Background Select Process    | • Check the validation of the background select form  
• Check the background select processing  
• Check the application flow                                                                                                                                                                                                                                                     | Pass   |
| Menu Change Process          | • Check the validation of the menu change form  
• Check the menu change process and application flow                                                                                                                                                                                                                              | Pass   |
<table>
<thead>
<tr>
<th>Unit</th>
<th>Test Performed</th>
<th>Result</th>
</tr>
</thead>
</table>
| Unit Choice Process         | • Check the validation of the unit choice form  
                              • Check the unit choice processing  
                              • Check the application flow     | Pass   |
| Text Manipulation           | • Check the validation of the text choice form  
                              • Check the text choice processing  
                              • Check the text preview process  
                              • Check the validation of the text input form  
                              • Check the text input processing  
                              • Check the text display process  
                              • Check the application flow     | Pass   |
| Picture Manipulation        | • Check the validation of the picture choice form  
                              • Check the picture choice processing  
                              • Check the picture preview process  
                              • Check the validation of the caption input form  
                              • Check the caption input processing  
                              • Check the validation of the picture upload form  
                              • Check the picture upload processing  
                              • Check the picture display process  
                              • Check the application flow     | Pass   |
<table>
<thead>
<tr>
<th>Unit</th>
<th>Test Performed</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Links Manipulation</td>
<td>• Check the validation of the links choice form</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>• Check the links choice processing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Check the links preview process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Check the validation of the links input form</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Check the links input processing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Check the links display process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Check the application flow</td>
<td></td>
</tr>
<tr>
<td>Logout Process</td>
<td>• Check the logout processing</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>• Check the application flow</td>
<td></td>
</tr>
</tbody>
</table>

3.3 Integration Testing

The Integration testing is testing the entire WD application after integrating all the individual units. At this stage of testing, the overall application flow is tested. Also data is transferred between different units.

As a part of integration testing an application flow diagram was developed. This diagram helped in selecting different scenarios to perform integration testing.

Two scenarios were developed, covering the possible paths in the application flow. The different steps involved in each scenario are explained with the help of sequence diagrams.
3.3.1 New User Scenario

3.3.1.1 Signup Process. The steps involved in this process are:

- A new user access WD Home Page.
- The user clicks on the hyperlink “Signup”.
- WD Signup Page opens.
- The user provides all the required information and clicks the “Submit” button.
- WD processes the form and WD Administration Page opens.

This part of the scenario is shown as a sequence diagram in Figure 49.
3.3.1.2 Create New Page Process. The steps involved in this process are:

- On the WD Administration Page, the user enters a page name in the specified text field and clicks "Create a New Page" button.

- WD creates a new page and the WD Template Page opens, displaying the new page.

This part of the scenario is shown as a sequence diagram in Figure 50.
3.3.1.3 Page-Title Change Process. The steps involved in this process are:

- On the WD Template Page, the user clicks "Change Title" button.
- WD Title Change Page opens in a new window.
- The user types a new page title and clicks "Save" button.
- WD processes the request, closes the window and the updated page title is displayed on the WD Template Page.

Figure 50. Create New Page Process - Sequence Diagram
This part of the scenario is shown as a sequence diagram in Figure 51.

Figure 51. Page-Title Change Process - Sequence Diagram

The rest of the processes of this scenario were tested and the results are shown in Table 4 (on page 93).

3.3.2 Registered User Scenario

3.3.2.1 Forgot Password Process. The steps involved in this process are:

- A registered user access WD Home Page.
• The user clicks on the hyperlink "Forgot Password".
• WD Forgot Password Page opens.
• The user provides all the required information and clicks the "Submit" button.
• WD processes the form and the password information is displayed on the WD Password Display page.

This part of the scenario is shown as a sequence diagram in Figure 52.

Figure 52. Forgot Password Process - Sequence Diagram
3.3.2.2 Login Process. The steps involved in this process are:

- On the WD Password Display Page, the user clicks on the hyperlink "Login"
- WD Login Page opens.
- The user provides the username and password and clicks the "Login" button.
- WD processes the information and WD Administration Page opens.

This part of the scenario is shown as a sequence diagram in Figure 53.

![Sequence Diagram](image)

**Figure 53. Login Process - Sequence Diagram**
3.3.2.3 Update Request Process. The steps involved in this process are:

- On the WD Administration Page, the user selects an existing webpage.
- Clicks on the "Update" button.
- WD retrieves the requested page and displays it on the WD Template Page.

This part of the scenario is shown as a sequence diagram in Figure 54.

Figure 54. Update Request Process - Sequence Diagram

The rest of the processes of this scenario were tested and the results are shown in Table 4.
3.3.3 Results

The description of the scenarios and the results of the integration test are shown in the table 4.

Table 4. WD Integration Test Results

<table>
<thead>
<tr>
<th>Scenario Name</th>
<th>Process Names in Order</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>New User</td>
<td>signup, create new web page, change page title, change background, input text, upload a picture, input external links, view the page, go to administration page, create another web page, change background, change page title, select a picture from his previously uploaded picture, view the page, change menu, logout</td>
<td>Pass</td>
</tr>
<tr>
<td>Registered User</td>
<td>retrieves forgot password, login, update an existing page, update page title, update background, update text, update picture caption, update external links, view the page, go to administration page, change page name of an existing page, delete an existing page, change personal information, logout</td>
<td>Pass</td>
</tr>
</tbody>
</table>

Several other scenarios were tested, besides the two described in Table 4, and the results were successful.

3.4 System Testing

System Testing will ensure the WD reliability and integrity. System testing involved testing the database, the server load, the security, and testing the application
on different platforms and operating systems. The System Test results are shown in the Table 5.

Table 5. WD System Test Results

<table>
<thead>
<tr>
<th>Test cases</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing WD application on windows 2000 System. Testing the application with real data to ensure the reliability and integrity of the system.</td>
<td>Pass</td>
</tr>
<tr>
<td>Installing WD application on UNIX Platform. Testing the application with real data to ensure the reliability and integrity of the system.</td>
<td>Pass</td>
</tr>
</tbody>
</table>

3.5 Summary

In this chapter various tests and test cases were explained with the help of sequence diagrams. The test results are presented in various tables.

In the next chapter the development environment and maintenance manual are described.
CHAPTER FOUR

MAINTENANCE MANUAL

4.1 Overview

In this chapter, server and client configuration files are described. The development environment is described. Also configuring the application in a web environment and starting the server is described.

4.2 WD Server Configuration

4.2.1 WD Development Environment

WD uses Java 2 SDK, Standard Edition Version 1.4.2 for the development environment. The Java 2 SDK is a development environment for building applications, applets, and components that can be deployed on the Java platform.

Java 2 SDK can be downloaded from the Sun Java website - http://java.sun.com/downloads/. After download and installing, the Environment Variable "JAVA_HOME" is set to the base directory of the Java 2 SDK.

4.2.2 WD Application Server

WD uses Apache Tomcat 4.1 as its application server. Tomcat is the Servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and
JavaServer Pages specifications are developed by Sun Microsystems under the Java Community Process.

Tomcat can be downloaded from the Jakarta website - http://jakarta.apache.org/site/binindex.cgi. After download and install, the Environment Variable “CATALINA_HOME” is set to the base directory of Tomcat.

To configure the Tomcat web server, edit the file “server.xml” in the Tomcat base directory under “conf” folder. Add the context path in the “server.xml” file.

Table 6. Editing “server.xml” file

```xml
<Context path="/wd" docBase="websitedeveloper" debug="0" reloadable="true" crossContext="true"/>
```

4.2.3 WD Application

Once the Java 2 SDK and Tomcat are setup, create a folder “websitedeveloper” in the Tomcat base directory under “webapps” folder.

- Create a folder “WEB-INF” in the “wd” folder.
  - Create a folder “classes” in the “WEB-INF” folder. Create a folder “wd” in the
“classes” folder. Place all the Java class files in this “wd” folder.

- Place the “web.xml” file in the “WEB-INF” folder (refer to the Table 5).

Table 7. “web.xml” file

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
</web-app>
```

- All the JSP files go under the “websitedeveloper” folder.
- Create a folder “help” in the “websitedeveloper” folder and Place all the HTML files in this “help” folder.
- Create a folder “images” in the “websitedeveloper” folder and Place the WD logo “wd.jpg” in this “images” folder.
• Create a folder "backgrounds" in the "websitedeveloper" folder and Place the WD background images in this "backgrounds" folder.

• Create another folder "users" in the "websitedeveloper" folder. When the users upload their pictures, different username folders are created in this "users" folder. The picture files are saved in those folders.

4.2.4 WD Database

WD uses MySQL 4.0 database server. MySQL is a popular open source database, recognized for its speed and reliability.

MySQL 4.0 can be downloaded from the MySQL website http://www.mysql.com/downloads/mysql-4.0.html. After download and install, start the database server and execute the database.sql file. This will create a new database "WEBSITEDEVELOPER", create tables, and inserts required data in the tables.

4.2.5 Starting and Stopping the WD Web Server

Once all the above steps are followed, open the Command Prompt. Go to the folder "bin" under Tomcat base directory. To start the web server execute "startup.sh" or
“startup.bat” based on what operating system the server has. And to stop the web server execute “shutdown.sh” or “shutdown.bat”.

4.3 WD Client Configuration

There is no configuration required on the client side. All they need is a web browser that supports JDK 1.2. The recommended browsers are Internet Explorer version 4.0 or higher and Netscape Navigator version 4.0 or higher.

4.4 Summary

In this chapter the build environment was described. Client, and server configuration files and details were described. Web server set up and the batch files to start the server, and database set up were described.

In the next chapter conclusions and future directions are discussed.
CHAPTER FIVE

CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Conclusions

The aim of this project is to develop an application that can be easily used by anybody and to make it easily available to its users. WD is developed as a web application, making use of the web technology and its advantages. WD is a tool that helps users create and host web pages.

WD is based on client/server software architecture model. The remote clients communicate with the central server through TCP/IP protocol.

WD application architecture is an implementation of the MVC (Model-View-Controller) design pattern. The Presentation layer (also know as the View) is implemented using JSP and HTML pages. This layer controls the look and feel of the application. It also takes the requests and delivers results. The Control layer (also know as the Controller) is implemented using JSP. This layer controls the application flow. The Application Logic layer (also know as the Model) is implemented using Java class files.
This layer manages application data, performs tasks and communicates with back-end resources.

The three-layered architecture separates the presentation from application logic, thus providing a robust architecture and groundwork for the implementation.

WD is developed to satisfy all its functionality, as specified in the sections 1.1.2 (scope of the project) and 1.1.3 (Limitations of the project). But there is a vast scope to improve the project. Future improvements can be made without redesigning or changing any of the basic architecture. It is possible by just adding more files to the application and/or adding more functionality to the existing files.

5.2 Future Directions

Improvements could be done with respect to two different aspects of the application - the functionality and implementing more HTML specifications.

5.2.1 Application Functionality

Presently WD has the functionality specified in the section 1.1.2 (scope of the project).

Some of the recommended functionality for the users are:
• Upload an HTML file from the user’s computer and host them.

• View and manage the pictures uploaded by the users.

Presently, Administrator can manage the application by having access to the central server. But interfaces could be developed that help the administrator to manage the application by logging on to the application from any remote location.

Some of the recommended functionality for the administrator are:

• View and manage user information.
• View and manage user web pages.
• View and manage pictures uploaded by users.
• Text-search the user web pages.

5.2.2 HTML Specifications

WD incorporates all the HTML specifications discussed in the section 1.1.3 (Limitations of the project). But HTML specification is big and opens a vast scope for improvement. Some of the recommendations are described in the following sections.
5.2.2.1 Text Properties. Presently WD gives the users the option to manipulate the text properties like font, color, style, size and alignment. But all these parameters are the same for an entire text paragraph. Users could be able to select a section of the text from a paragraph and set all the above properties to that particular section.

5.2.2.2 Picture Properties. Users could be able to add a hyperlink for a picture. They could have an option to have a rollover image. They could have an option to manage the alignment of the caption with respect to the picture. Also they could be able to manage the text properties of the caption.

5.2.2.3 Sound. Users could be able to add sounds to their web page.

5.2.2.4 Color. Presently WD gives the users an option to choose a color from a list of about six colors, for any text or background. More colors could be added to this list.

5.2.2.5 More Units. A unit can be a text paragraph or a picture with caption or a set of external links. Presently users can have up to four units in each web page they create. This number could be increased.
5.3 Summary

This chapter has the conclusions and some of the recommended future directions were discussed.
REFERENCES

[1]. "IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1993)"

[2]. "IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)"


