

**A PROJECT REPORT
ON
“Web Development On Web”**

Version: 1.0

**FOR
“ Persistent Systems Limited ”**

**BY
MILIND H GOKHALE
AMIT S GOSAVI
AARTI S MAVLINGKAR
MANASI A SAHASRABUDHE**

**DEPARTMENT OF COMPUTER ENGINEERING
P.E.S MODERN COLLEGE OF ENGINEERING
PUNE- 411005
* [2008 – 2009] ***



Guide:

- Internal Guide Name: Ms Deipali Gore
- External Guide Name: Mr. Pravin Kalel

Presented By:

Date	Version	Title	Authors
10-04-2009	1.0	Web Development On Web	Milind Gokhale Amit Gosavi Aarti Mavlingkar Manasi Sahasrabudhe



Progressive Education Society's
Modern College of Engineering
Shivajinagar, Pune – 411005.

CERTIFICATE

This is to certify that the following students of Final Year Computer Engineering have successfully completed the project entitled "**WEB DEVELOPMENT ON WEB**" for the organization "**Persistent Systems Limited**".

The Group Members names are:

Milind Hemant Gokhale
Amit Shrinivas Gosavi
Aarti Satish Mavlingkar
Manasi Aniruddha Sahasrabudhe

This is in partial fulfillment of Bachelor of Computer Engineering under University of Pune.

Date: 10-04-2009

Internal Guide
(Name of Guide)

Head of Dept
(Computer Engineering)
(Prof. Mrs. S. A. Itkar)

External Examiner



April 28, 2009

To Whomsoever it May Concern

This is to certify that following students have completed their final year B.E. project at Persistent Systems Ltd. for academic year 2008-09.

The project details are as below:

Project Title : *Web Development On Web*
Project Guide : *Pravin Kalel*
College Name : *P.E.S Modern College of Engineering, Pune*
Branch : *B.E. (Computer Science)*

Name of Students:

- i. *Milind H Gokhale*
- ii. *Amit S Gosavi*
- iii. *Aarti S Mavlingkar*
- iv. *Manasi A Sahasrabudhe*

For Persistent Systems Ltd.

Baliram KM.
Baliram Mutagekar
Assistant Manager - Human Resource

Acknowledgements

We would like to express our heartfelt gratitude to Persistent Systems Ltd and Mr. Pravin Kalel (industrial mentor) for giving us the opportunity to work under their guidance and help us gain immensely enriching professional experience. Our sincere thanks to Project In charge and our internal project guide, Ms. D. V. Gore for giving us valuable inputs and ideas right from the selection of topic for project till its successful completion.

We thank Prof. Mrs. S. A. Itkar (HOD computer department) for her ongoing support and encouragement in every aspect. Last but not the least, entire staff of Department of Computer Engineering for guiding our thoughts and vision.

The successful completion of our project would not have been possible without the dedicated support from all our mentors, family and friends.

Milind Gokhale

Amit Gosavi

Aarti Mavlingkar

Manasi Sahasrabudhe

ABSTRACT

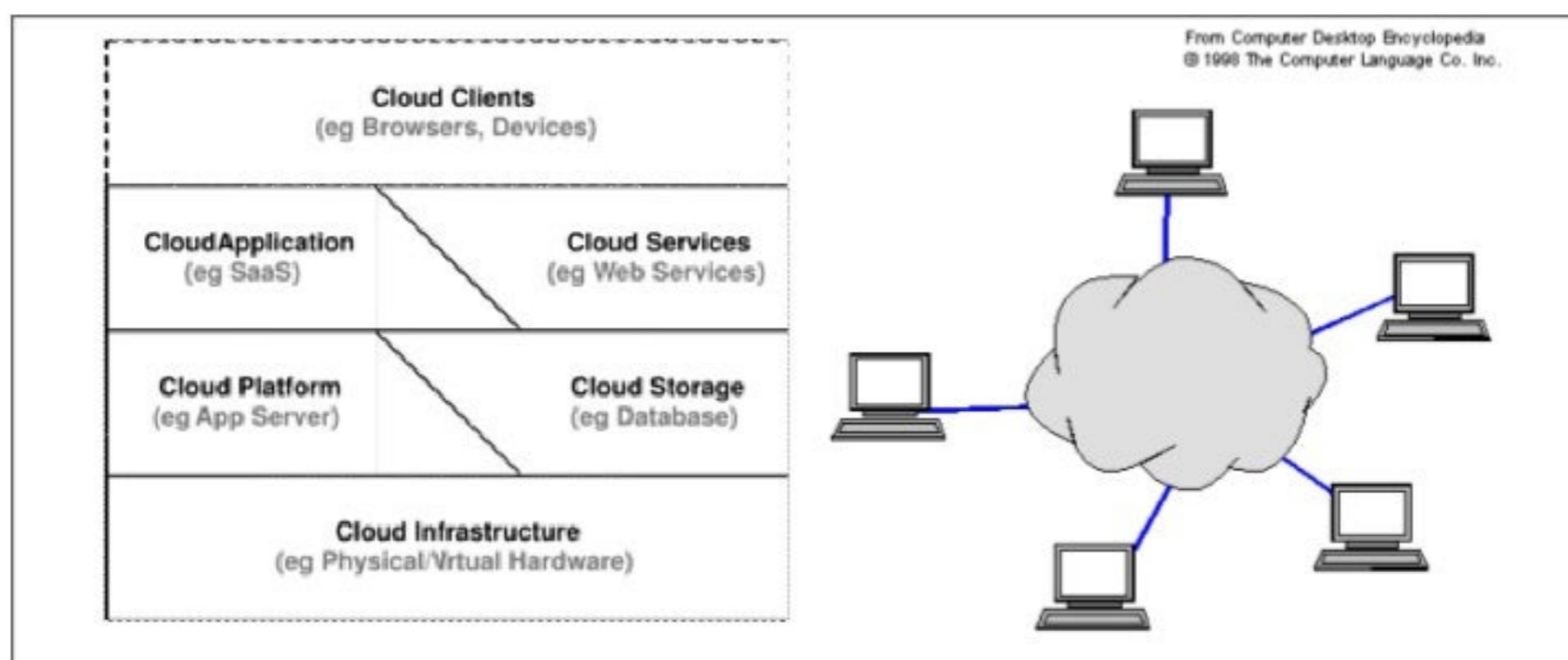
Problem statement -

Develop an application which facilitates creation of web pages without having a need to install any html editor based software and also which can be used by any novice user. That is developing web pages on the fly (online).

Problem Solution -

Software as a service is a model of software deployment where an application is hosted as a service provided to customers across the Internet. By eliminating the need to install and run the application on the customer's own computer, **SaaS** alleviates the customer's burden of software maintenance, ongoing operation, and support.

Using WebyWeb application, user can develop their own web pages very easily and with minimum technical knowledge. It eliminates the need of writing lengthy lines of codes, getting entangled in error fixing process.



Cloud computing

Functional Description -

1. Registration is offered. Then a registered user logs into our site.
2. The WYSIWYG editor dashboard is displayed to user on which pages are developed.
3. Automatic code generation of designed page and HTML page is saved on the server.
The user may preview the page on a range of browsers.

Work Area –

1. **Microsoft Silverlight** - Microsoft Silverlight is a programmable web browser plug-in that enables features such as animation, vector graphics and audio-video playback that characterizes rich Internet applications.

Aim of the project -

- Create web pages online.
- Aesthetic user interface, making it an enriching experience for creation of web pages.

Key Features and scope of the project

1. WYSIWYG Presto-based editor.
2. Preview websites in multiple browsers.
3. Implementation of basic HTML tags and –
 - a. Tables
 - b. Style formatting

Images, buttons etc

Table of Contents

1	Introduction	1
1.1	Problem Statement.....	1
1.2	Project Objectives.....	1
1.3	Project Scope.....	2
1.4	Goals	2
1.5	Assumptions	3
1.6	Constraints.....	4
2	Project Plan.....	5
2.1	Project Plan	5
2.2	Task Sheet	6
2.3	Project Management Approach.....	7
2.3.1	Project Roles and responsibilities.....	7
3	Requirement Analysis	8
3.1	Introduction	8
3.1.1	Purpose.....	8
3.1.2	Intended Audience and reading suggestions	8
3.1.3	Project Scope.....	9
3.2	Overall Description.....	9
3.2.1	Product Perspective.....	9
3.2.2	Product Features	9
3.2.3	User Classes and Characteristics	10
3.2.4	Operating Environment.....	10
3.2.5	Design and Implementation Constraints	10
3.2.6	User Documentation	11
3.2.7	Assumptions and Dependencies	11
3.3	System Features.....	11
3.3.1	WYSIWYG Presto-based editor (High Priority).....	11
3.3.2	The Design View for webpage creation.....	11

3.4 External Interface Requirements	11
3.4.1 User Interfaces	11
3.4.2 Hardware Interfaces	12
3.4.3 Communications Interfaces	12
3.5 Other Nonfunctional Requirements	13
3.5.1 Performance Requirements	13
3.5.2 Safety Requirements	13
3.5.3 Software Quality Attributes	14
3.6 Project Architecture	15
3.7 Technology Used	16
3.7.1 Microsoft Silverlight	16
3.8 Libraries Used	17

4 Design Document.....18

4.1 Introduction	18
4.1.1 Background	18
4.2 Architecture	19
4.2.1 Introduction	19
4.2.2 Data	20
4.2.2.1 File and Data Formats	20
4.3 Code	20
4.3.1 Introduction	20
4.3.2 Modules	20
4.3.2.1 Module A: The Database	20
4.3.2.1.1 Internal Functions	20
4.3.2.2 Module B: The Main Editor	21
4.3.2.2.1 Internal Functions	21
4.3.3 Interfaces	22
4.4 Operation	23
4.4.1 User Types	23
4.4.2 Scenarios	23
4.4.3 Installation	23

4.4.4 Licensing	23
4.4.5 Upgrades	24
4.4.6 Uninstall	24
4.5 Development	24
4.6 Miscellaneous / Appendices	26
4.6.1 Conformance with standards	26
4.6.2 Interoperability with other systems.....	26
4.6.3 Security	27
4.6.4 Open Issues.....	27
4.7 Analysis Models	28

5 Testing34

5.1 Test Plan.....	34
5.1.1 Test Plan Identifier.....	34
5.1.2 Introduction	34
5.1.3 Test Items	34
5.1.4 Features to be tested.....	35
5.1.5 Features not to be tested.....	35
5.1.6 Approach/Strategy	35
5.1.7 Item Pass/Fail Criteria.....	36
5.1.8 Suspension Criteria and Resumption Requirements.....	36
5.1.9 Test Deliverables	36
5.1.10 Test Environments	37
5.1.11 Staffing and Training Needs.....	37
5.1.12 Schedule	37
5.1.13 Risks and Contingencies	37
5.1.14 Approvals	38
5.1.15 Test Plan Identifier.....	38
5.1.16 Introduction	38
5.1.17 Features and Functions to Test	38
5.1.18 Features and Functions not to Test	41
5.2 Test Procedure.....	42
5.2.1 Black Box Testing	42

5.2.2 Regression Testing.....	42
5.2.3 Integration Testing	42
5.3 Test Cases.....	43
5.3.1 “Sign In Module” Test Cases	43
5.3.2 Test case for Sign Up module with specific test data	45
5.3.3 Black Box Testing	49
5.3.4 GUI Testing.....	50
5.4 Test Oracle	51
5.4.1 Test Design.....	51
• Future Enhancement.....	55
• Conclusion.....	57
• Appendix A: Glossary	58
• Bibliography.....	59

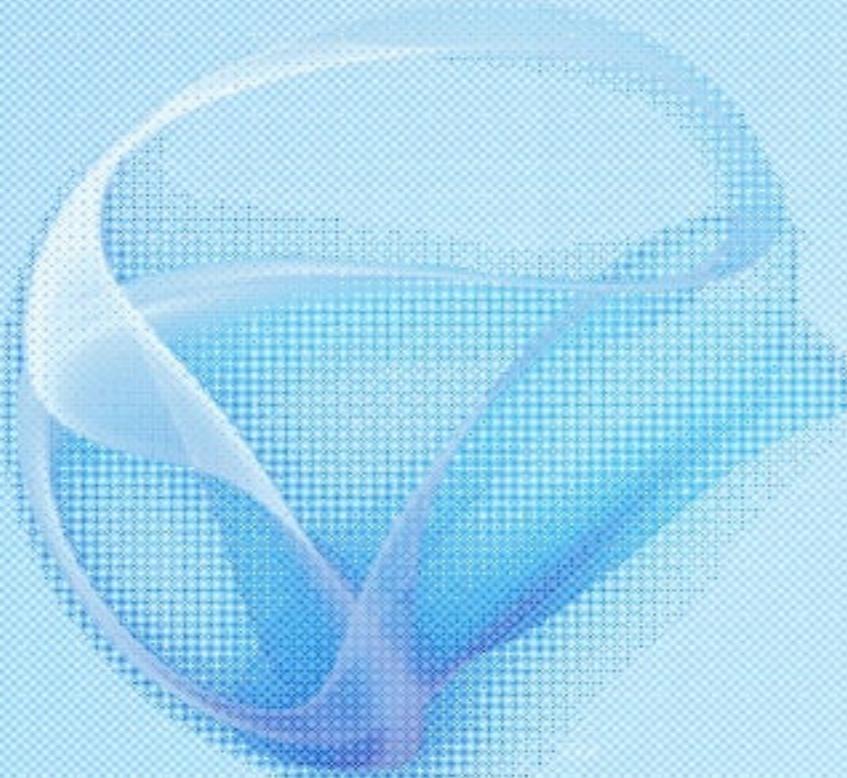
Table of Contents (Tables and figures)

List of Figures

Sr. No.	Name of Figure	Page Number
3.1	WebyWeb UI	12
3.2	WebyWeb Architecture	15
4.1	Silverlight Architecture	17
4.2	Interpretability with other systems	26
4.3	Use Case Diagram	28
4.4	Class Diagram	29
4.5	Class diagram for interfaces and classes	30
4.6	Sequence diagram	31
4.7	State Chart diagram	32
4.8	Component diagram	33
5.1	Dreamweaver installation	52
5.2	WebyWeb Homepage	52
5.3	Insert in Dreamweaver	53
5.4	Insert in WebyWeb	53
5.5	Insert in Google	54
5.6	Insert in WebyWeb	54
6.1	Syntax highlighting	55
7.1	Sample Webpage development in WebyWeb	57

List of Tables

Sr. No.	Name of Table	Page Number
2.1	Project Plan	5
2.2	Task Sheet	6
2.3	Project Roles and Responsibilities	7
3.1	Libraries used	17
4.1	Silverlight across various browsers	27
5.1	Test case for sign-in module	44
5.2	Test case for sign-up module	46
5.3	Black box testing	49
5.4	GUI testing	50



1.

Introduction

1. Introduction

1.1 Problem Statement

Develop an application which facilitates creation of web pages without having a need to install any HTML (Hypertext Markup Language) [1].editor based software and also which can be used by any novice user (No HTML knowledge needed). That is developing web pages on the fly (online).

1.2 Project Objectives

The purpose of WebyWeb is to allow the user to design web pages on the fly.

Existing guidelines for Web usability hinder web-based application usability since they are primarily based on interactions within a browsing metaphor. The main difference is that with traditional in-house software the user only focuses on the application itself. The application is stand-alone, installed on the users system and the IT department or network administrator takes care of the management, monitoring, updates and all other tasks related to using software in a business environment.

A SaaS (Software as a Service) [2] application also focuses on this primary application part, but incorporates tasks that all together make up the SaaS application and business model. These tasks include registering, paying for the software, using support and customizing the application. A SaaS application is much more specifically directed to the customer instead of a traditional desktop application directed at the mass. This results in a higher value of design and interaction.

This achieves following -

- Centralized Updation
- No installations
- Portable
- Cross Browser
- No piracy

1.3 Project Scope

The software WebyWeb will revolutionize the way web pages are designed. With this software, websites can be designed in cloud without the requirement of costly website designing softwares and installations. It will provide a user friendly dashboard which will serve for all user needs. All the HTML tags may be implemented. WebyWeb is a What You See Is What You Get (WYSIWYG) editor, and will also give a code view of the page being designed. WYSIWYG means that the finished page will be display exactly the way it was designed. WebyWeb generates HTML tags while user points and clicks on the desired features. Thus, user can create web page without learning HTML.

Scope-

- WYSIWYG editor
- Dashboard interface
- Style formatting
- Hyperlinks
- Inserting images
- Design and code views
- User Sign In and Sign Up
- Inserting HTML form elements (radio button, checkbox)
- Inserting tables and editing it
- Zoom in and Zoom out

1.4 Goals

SaaS is defined by not only delivery via the Internet, but by subscription and periodic payment. Especially these properties differ SaaS from standard web-based software. SaaS is a model of software deployment where an application is hosted as a service provided to customers across the Internet. By eliminating the need to install and run the application on the customer's own computer, SaaS alleviates the customer's burden of software maintenance, ongoing operation, and support. Using SaaS also can reduce the up-front expense of software purchases, through less costly, on-demand pricing [3].

By implementation of the WebyWeb project, we aim to take the above concept further and provide the user with the ability to develop web pages on internet without having the need of installations of the softwares that facilitate this process. It is a self contained product which provides improvement in the current web development scenarios.

- **User control and freedom**

When a user makes a mistake, he will be able to go back and forth within an application to correct. Therefore, WebyWeb will support undo and redo.

- **Aesthetic design**

WebyWeb aims the user work environment and experience to be aesthetically rich. Hence WebyWeb is designed as RIA (Rich Internet Application) [4].

- **Help and documentation**

WebyWeb will provide simple and concise instructions, prompts, embedded in the application itself. User manual will be provided for easy work environment.

1.5 Assumptions

1.5.1 About SaaS-

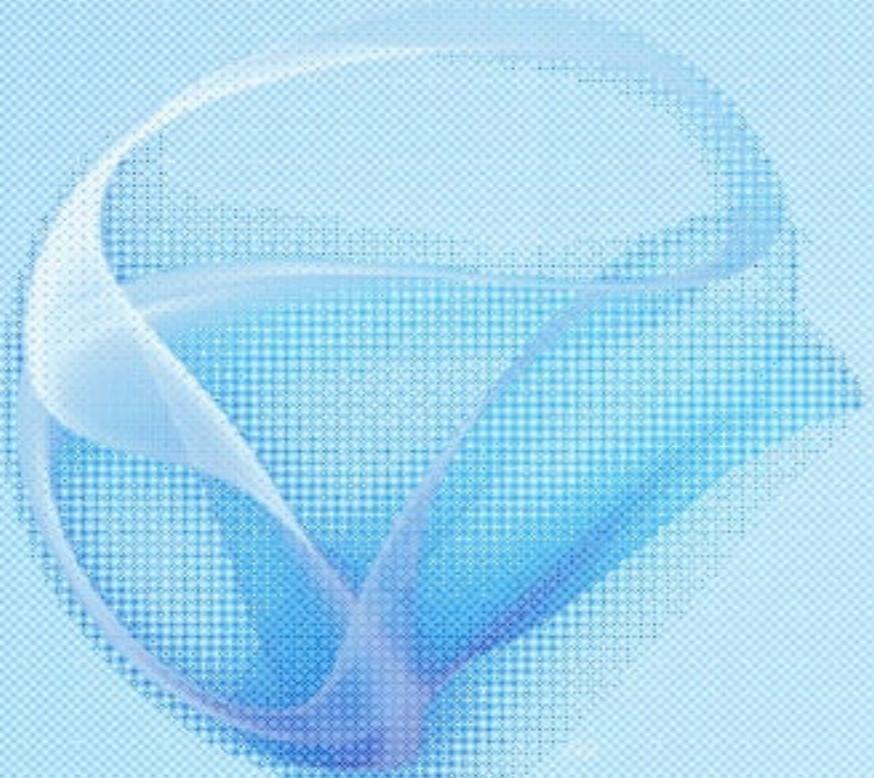
SaaS is based upon the assumption that the services provided are commonplace and well defined, hence economies of scale and balancing of supply and demand becomes possible. This assumption holds true for those areas of IT that are ubiquitous, a cost of doing business and commodity-like. SaaS is therefore not suitable for innovative or highly specialized niche systems, though SaaS may be used to provide one or more components in such systems.

1.5.2 About WebyWeb -

1. Silverlight 2 plug-in installed.
2. The PC is connected to internet/ connected in intranet.
3. 512 GB RAM.
4. 1024 X 768 monitors resolution with 24 bit color.

1.6 Constraints

Minimum number of computers is at least 2 which will limit the development process if they are not available. Since Silverlight provides rich web based interactions, resolution of screen and graphics aspects of hardware are extremely important. Installations of Microsoft expression blend, visual studio 2008, Silverlight are essentials which enable the developer to develop their application.



2.

Project Plan

2.1 Project Plan

		Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Review of different domains	2 days	Wed 23-07-08	Thu 24-07-08		Seminar at company
2		Select project for pursuit	2 days	Thu 14-08-08	Fri 15-08-08		Internet
3		- Present project for discussion	3 days	Mon 18-08-08	Wed 20-08-08		
4		Scope	1 day	Mon 18-08-08	Mon 18-08-08		
5		Resources	1 day	Tue 19-08-08	Tue 19-08-08		
6		Benefits	1 day	Wed 20-08-08	Wed 20-08-08		
7		Strategy	1 day	Wed 20-08-08	Wed 20-08-08		
8		External Assistance	1 day	Thu 21-08-08	Thu 21-08-08		Mentor
9		Initial Meeting	1 day	Thu 21-08-08	Thu 21-08-08		
10		Finalise scope	1 day	Thu 21-08-08	Thu 21-08-08		
11		Define desired input output and Co	1 day	Fri 22-08-08	Fri 22-08-08		
12		- Feasibility	1 day	Fri 22-08-08	Fri 22-08-08		
13		Risk Assessment	1 day	Fri 22-08-08	Fri 22-08-08		
14		Critical success factor	1 day	Fri 22-08-08	Fri 22-08-08		
15		project requirement	1 day	Fri 22-08-08	Fri 22-08-08		
16		- Kick off project	1 day	Sat 23-08-08	Sat 23-08-08	10,11,12,13,14,15	
17		Roles	1 day	Sat 23-08-08	Sat 23-08-08		
18		Schedules	1 day	Sat 23-08-08	Sat 23-08-08		MS project planner
19		Work plan	1 day	Sat 23-08-08	Sat 23-08-08		
20		Finalise contract	1 day	Mon 25-08-08	Mon 25-08-08	16	
21		Abstract and final project	1 day	Tue 26-08-08	Tue 26-08-08		
22		Appointment of internal guide	1 day	Tue 26-08-08	Tue 26-08-08	3	
23		- Acquisition of software	23 days	Wed 27-08-08	Fri 26-09-08		Internet
24		Review of software versions	4 days	Wed 27-08-08	Mon 01-09-08		
25		Downloads from internet	15 days	Tue 02-09-08	Mon 22-09-08		
26		Distribution of software among	4 days	Tue 23-09-08	Fri 26-09-08	25	
27		- Detail design of project	10 days	Wed 01-10-08	Tue 14-10-08		ebooks,white pap
28		Object oriented analysis	5 days	Wed 01-10-08	Tue 07-10-08		
29		Study of modules	5 days	Wed 08-10-08	Tue 14-10-08		
30		- Design of interface	14 days	Fri 16-01-09	Wed 04-02-09		
31		Dashboard	10 days	Fri 16-01-09	Thu 29-01-09		
32		WYSIWYG editor	2 days	Fri 30-01-09	Mon 02-02-09		
33		Menus	2 days	Tue 03-02-09	Wed 04-02-09		
34		Coding	30 days	Wed 04-02-09	Tue 17-03-09		
35		Development of test cases	2 days	Wed 18-03-09	Thu 19-03-09		
36		validation and verification	3 days	Fri 20-03-09	Tue 24-03-09		
37		Acquisition of web space	2 days	Wed 25-03-09	Thu 26-03-09		
38		Development of user manuals	2 days	Wed 25-03-09	Thu 26-03-09		
39		Deployment	1 day	Fri 27-03-09	Fri 27-03-09		
40		Beta version testing	3 days	Mon 30-03-09	Wed 01-04-09	39	
41		Maintenance	4 days	Thu 02-04-09	Tue 07-04-09	39	
42		Final report preparation	1 day	Wed 08-04-09	Wed 08-04-09		

Table 2.1 Project Plan

2.2 Task Sheet

Sr. No.	Task	Sub – Task	Date		Person(s) Allotted
			Start	End	
001	Research on Project related technologies	Microsoft Silverlight, Visual Studio 08, Expression Blend	25-8-2008	27-8-2009	Milind Amit Aarti Manasi
002	Acquisition of software	Review of software versions, Downloads from internet, Distribution of software among project members.	27-8-2008	26-9-2008	Milind Amit
003	Detail Design of project	Study of Web editor software, Object oriented analysis, Study of modules	1-10-2008	14-10-2008	Milind Amit Aarti Manasi
004	Design of interface	Main Editor GUI, Dashboard, WYSIWYG editor Menus	16-1-2009	4-2-2009	Milind Amit
005	Coding	Coding for user interface events, Toolbars, Code conversion module, Study of CRUD operations in Silverlight, Implementation of database and login and sign up page	4-2-2009	17-3-2009	Milind Amit Aarti Manasi
006	Rich interactive elements added	Designing interactive navigation page, Falling star background, Image carousel	17-3-2009	22-3-2009	Milind Amit Aarti Manasi
007	Testing	Development of test cases Validation and verification	18-3-2009	24-3-2009	Aarti Manasi
008	Deployment of application on IIS server	Publishing project on server	31-3-2009	31-3-2009	Milind Amit Aarti Manasi
009	Documentation	SRS, High level design, Test plan and test cases, Final Report	24-3-2009	31-3-2009	Milind Amit Aarti Manasi

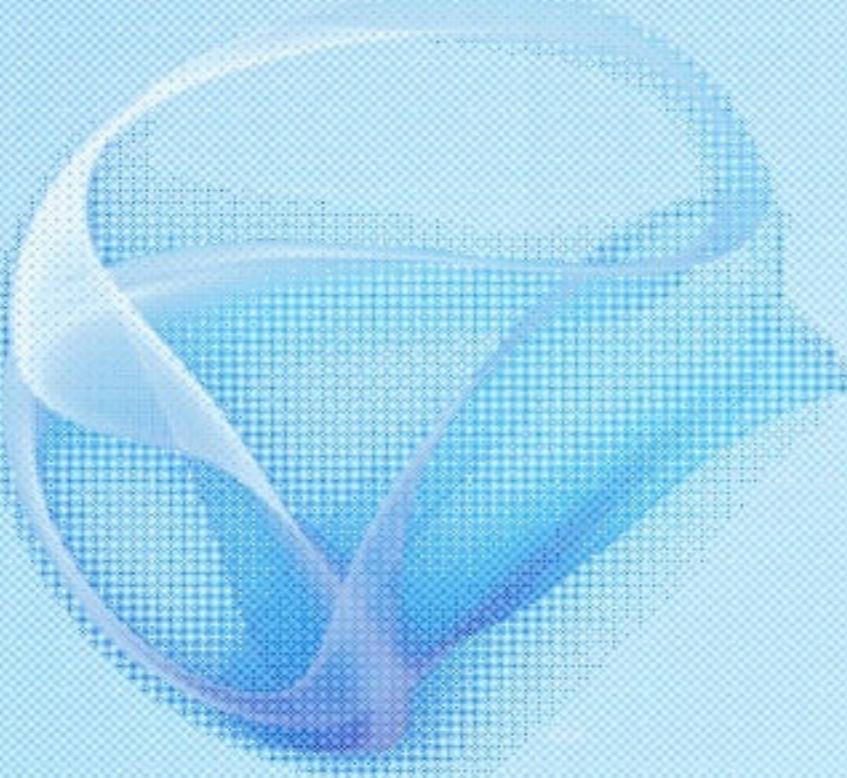
Table 2.2 Task Sheet

2.3 Project Management Approach

2.3.1 Project Roles and Responsibilities

Role	Responsibilities	Participant
Project Guides	<ul style="list-style-type: none"> • Ultimate decision-maker and tie-breaker. • Provide project oversight and guidance. • Review/approve project elements. • Getting the Team Off on the Right Foot 	Internal Guide Ms. Deipali Gore External Guide Mr. Pravin Kalel
Project Members	<ul style="list-style-type: none"> • Draft initial charter and project plan. • Update plan regularly. • Identify and resolve issues. • Identify and mitigate risks. 	Milind Gokhale Amit Gosavi Aarti Mavlingkar Manasi Sahasrabudhe
Steering Committee	<ul style="list-style-type: none"> • Approve scope changes • Provides direction to the Project • Help resolve issues and policy decisions • Update plan regularly. 	Ms. Deipali Gore Mr. Pravin Kalel Milind Gokhale Amit Gosavi Aarti Mavlingkar Manasi Sahasrabudhe
External Guide coordinator	<ul style="list-style-type: none"> • The first point of contact for the team members with external guide 	Milind Gokhale

Table 2.3 Project Roles and Responsibilities



3.

Requirement Analysis

3. Requirement Analysis

3.1 Introduction

3.1.1 Purpose

The purpose of WebyWeb is to allow the user to design web pages on internet on the fly.

Current techniques for web page development include software's like Dreamweaver [5], and Microsoft Expression Web [6] etc. But disadvantage with them is that they need installations on the respective machines. To overcome this drawback, we with our project "WebyWeb" using SAAS as a domain, aim at centralizing the process of web page creation. With this, the WebyWeb editor is centralized on server thereby eliminating the need to install the application. The user doesn't need to install patches whenever any new feature is added into the software. All the updatings are centralized for maintaining consistency and thus reducing the overhead for client to update the software individually. Payment will be done on the basis of package that will be subscribed for. The user won't have to pay expensively for the high end functionality which he/she might never use.

WebyWeb can be used by any novice user who has minimum technical knowledge about web page development. With highly user friendly and rich graphical environment, the experience of web page development will be enriching and easy. WebyWeb provides implementation of all the basic html tags. Insertion of form elements like buttons, checkbox, images, tables etc is provided. Easy navigation from design to code view check the code generated for the corresponding web design. The entire user's work session will be stored on server thus providing portability.

3.1.2 Intended Audience and Reading Suggestions

Intended audience of this document will be the developers of WebyWeb, project manager, marketing staff, user's testers, and documentation writers. It is recommended that the SRS document is read sequentially.

3.1.3 Project Scope

WebyWeb which is based on SaaS domain aims at providing the users to develop their own web page on the fly. It enhances the user work experience by providing aesthetic environment. The software WebyWeb will revolutionize the way web pages are designed. With this software, websites can be designed in cloud without the requirement of costly website designing softwares and installations. It will provide a user friendly dashboard which will serve for all user needs. All the HTML tags and scripts may be implemented. WebyWeb is a WYSIWYG editor, and will also give a code view of the page being designed.

3.2 Overall Description

3.2.1 Product Perspective

Web sites have become means of information sharing and interactivity. For developing any web page, many softwares are available like Dreamweaver etc. But disadvantage with them is that they need installation, payment is expensive and a user that need not use high end functionality has to pay complete cost of the package. Moreover, coding knowledge may be required to use these softwares.

By implementation of the WebyWeb project, we aim to change the scenario by allowing user to develop web pages on internet without having the need of installations of the softwares.

3.2.2 Product Features

- Application installation is not required; users access the application on the web (only plug-in installation may be required).
- Updates and upgrades to newer versions are automatic.
- Any computer with an Internet connection can become an access point to an application, no matter what operating system is installed.
- The risk of viral infection is greatly decreased when running an application on the web instead of an executable.

Key features:

1. WYSIWYG Presto-based editor.
2. Implementation of basic HTML tags and insertion of –
 - a. Tables
 - b. Images
 - c. HTML form elements (radio button, checkbox, button).
 - d. Rich Text formatting.
3. Maintaining user's work session along with the files created in the committed state.

3.2.3 User Classes and Characteristics

Amateur user: These are the users who are completely new to webpage development and have no knowledge of coding, but wish to design their own web pages.

WebyWeb has dual interface showing design view as well code view. Thus amateur users will not find it difficult to design their own web pages. Using design view they can make the website the way they want. On the other side expert users have an added advantage of knowledge about coding.

3.2.4 Operating Environment

WebyWeb will operate on a web browser. All it requires for running at client end is the Silverlight [7] plug-in installed on the client's web browser. It is cross browser (can be run on most web browsers including Internet explorer, Mozilla, etc.) and cross platform (windows vista/XP, Linux)

3.2.5 Design and Implementation Constraints

Minimum number of computers is at least 2 which will limit the deployment process if they are not available. Since Silverlight provides rich web based interactions, resolution of screen and graphics aspects of hardware are extremely important. Installations of Microsoft expression blend [8], visual studio 2008 [9], Silverlight are essentials which enable the developer to develop their application.

3.2.6 User Documentation

- User manuals

3.2.7 Assumptions and Dependencies

Assumptions:

1. Silverlight 1.1/2 plug-in installed.
2. The PC is connected to internet.
3. 512 GB RAM.
4. 1024 X 768 monitors resolution with 24 bit color.
5. Graphics processor that is DirectX® 9-capable.

3.3 System Features

3.3.1 WYSIWYG Presto-based editor (High Priority):

WYSIWYG means What You See Is What You Get. This means that the web page designer software has a user friendly interface. The user just has to imagine the look of the page and may design it just as designing on a piece of paper. Using tools provided by the software.

3.3.2 The Design View for webpage creation:

This feature displays web page being designed as it is. The coding details are hidden, making it possible for non-coders to create web pages and sites.

3.4 External Interface Requirements

3.4.1 User Interfaces

A dashboard is provided to the user for development of web pages. It includes a menu bar, tool box, properties window, status bar.



Figure 3.1: WebyWeb UI

3.4.2 Hardware Interfaces

TCP/IP:

It is a reliable connection oriented protocol that allows byte streaming originating on one machine to be delivered without any error on any other machine in the network. It fragments the incoming byte stream into discrete messages and parses each one on to the internet layer. At the destination, the receiving TCP process reassembles received messages into the output stream. TCP also handles the flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

3.4.3 Communications Interfaces

HTTP: Hyper text transfer protocol (HTTP) is a method used to transfer or convey information on the World Wide Web. Its original purpose was to provide the way to publish and retrieve HTML pages.

Development of HTTP was co-ordinated by the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) culminating in the publication of RFCs, most notably RFC 2616, which defines HTTP/1.1 the version of HTTP in common use today. HTTP is a request/response protocol between clients and servers. The originating client, such as web browser, spider or other end user tool, is referred to as a user agent. The destination server which stores or creates resources such as HTML files and images is called the origin server. In between the user agent and origin server, any is several intermediaries such as proxies, gateways, tunnels.

An HTTP client initiates a request by establishing Transmission Control Protocol (TCP) connection to a particular port on a remote host. An HTTP server listening on that port waits for the client to send a request message. Upon receiving request, the server sends back a status line, such as “HTTP/1.1 200 OK” and a message of its own, the body of which is perhaps the request file, an error message, or some other information. Resources to be accessed by HTTP are identified using Uniform Resource Identifiers (URIs) (or, more specifically, URLs). Using the HTTP: or HTTPS URI schemes.

3.5 Other Nonfunctional Requirements

3.5.1 Performance Requirements

The web server must be able to handle and support multiple instances of application. The time between request and reply should be less in case of online help. Minimum time should be taken by the application to display preview the web pages developed by the user. In case of power failure, the data should be stored in the state that was last saved by the user.

3.5.2 Safety Requirements

1. A firewall should be present for packet – filtering and other security issues.
2. A user can be able to view/update only his web pages.

3.5.3 Software Quality Attributes

There are various software quality attributes that are taken into consideration –

1. **Availability** – As WebyWeb is a web based service provided to the users, it will be available as long as server is up.
2. **Interoperability** – WebyWeb is interoperable on various operating systems, thus increasing the applications usability and flexibility.
3. **Maintainability** – WebyWeb is a SaaS based service. Hence, all the updates are centralized making the maintenance convenient.
4. **Usability** - The main purpose of developing this WebyWeb is to enable the users who wish to develop their web pages online and who aren't well versed with the technical aspects of coding on standalone applications like Dreamweaver, MS front page etc.
5. **Creative** – Users can apply their imagination in full extent thus enhancing their creative power without actually worrying much about the technical details.

3.6 Project Architecture

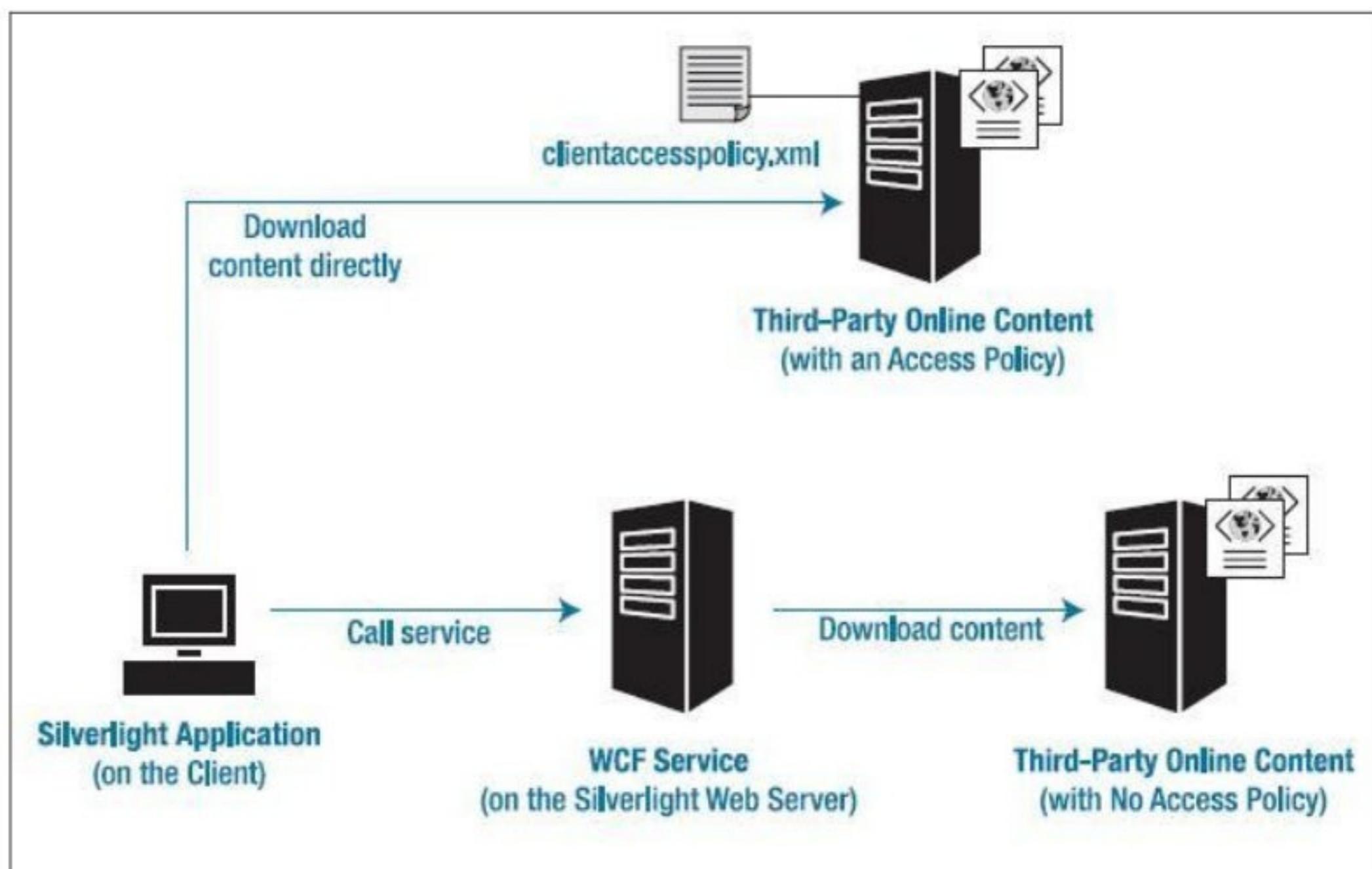


Figure 3.2 : WebyWeb architecture

WebyWeb has client server architecture. It is based on cloud computing. Architecture is divided into two parts namely the database and the main editor. The project resides on WebyWeb server.

In our system, there's a significant workload shift. Local computers no longer have to do all the heavy lifting when it comes to running applications. The network of computers that make up the cloud handles them instead. Hardware and software demands on the user's side decrease. The only thing the user's computer needs to be able to run is the WebyWeb's **interface software** that is in a xap [9] file which is loaded on client PC on fly as the user accesses the WebyWeb website.

WebyWeb is divided into two sections: the **Main Editor** and the **Database** as every cloud computing system is. They connect to each other through a network. The Main Editor is the side the client operates on. This includes the code conversion engine, Dashboard interface, User Workspace area. The database is the section which maintains the user information, files and log. Storage devices are required to keep all clients' information. Our system will make a copy of all clients' information and we will store it on other back up devices on daily basis.

3.7 Technology Used

3.7.1 Microsoft Silverlight

Silverlight is Microsoft's cross-browser, cross-platform browser plug-in that allows the creation of interactive web applications that employ high quality streaming media, vector graphics, images, and animation. Deployed as a plug-in for the major browsers on the Windows, Mac and Linux (supported by Novell) operating systems, web developers can craft interactive applications that have an identical user experience on the vast majority of web browsers deployed today. Silverlight addresses a disconnect that exists today in web development workflow where the design intent of graphics designers and interaction designers cannot be faithfully communicated to and crafted by the web developers. In Silverlight, this intent is created in design tools like Expression Design and Expression Blend and passed off to web developers in XML-based XAML data files.

The fidelity of the designers' ideas is kept as there is a clear separation between the design in XAML and the code in JavaScript. Silverlight is a browser plug-in that renders XAML and exposes a JavaScript programming model in 1.0, extending this with a .NET programming model and runtime in Silverlight 2. Since the Silverlight 2 runtime has full access to the HTML Document Object Model (DOM) and cross-domain data access abilities, it is possible for future Silverlight applications to implement analytics entirely in managed code or in JavaScript code that is invoked from the managed code.

3.8 Libraries Used

SR NO	NAME OF DLL	CLASS NAME	SUMMARY
1	System.Windows	FontWeights	Provides a set of predefined font weights as static property values.
2		TextAlignment	Specifies whether the text in the object is left-aligned, right-aligned, centered, or justified
3	System.Windows.Browser	HtmlEventArgs : System.EventArgs	Provides event details to event handlers.
4	System.Windows.Controls	OpenFileDialog	Represents an open file dialog box that enables a user to select one or more files. This class cannot be inherited.
5		StackPanel : System.Windows.Controls.Panel	Arranges child elements into a single line that can be oriented horizontally or vertically.
6	System.Xml	XmlNodeType	Specifies the type of node.
7	System.Configuration	DateTime	Represents an instant in time, typically expressed as a date and time of day.
8	System.Data	XmlReadMode	Specifies how to read XML data and a relational schema into a System.Data.DataSet.

Table 3.1: Libraries used



4.

Design

4. Design Document

4.1 Introduction

4.1.1 Background

Software as a service is a model of software deployment where an application is hosted as a service provided to customers across the Internet. By eliminating the need to install and run the application on the customer's own computer, **SaaS** alleviates the customer's burden of software maintenance, ongoing operation, and support. It reduces up-front expense of software purchases through on-demand pricing for the required services.

Usually softwares such as Macromedia Dreamweaver or Adobe GoLive are used for making web pages and websites.

1. But it requires installation of these softwares.
2. Thus space and resources of the host are not required. Only an internet connection can do the work.

Advantages over older systems:

1. We can create a web development tool in cloud which helps in creating web pages online on fly.
2. Centralized updates for software.
3. Portability of website development.
4. Global users who have basic knowledge of using computers and internet.

4.2 Architecture

4.2.1 Introduction

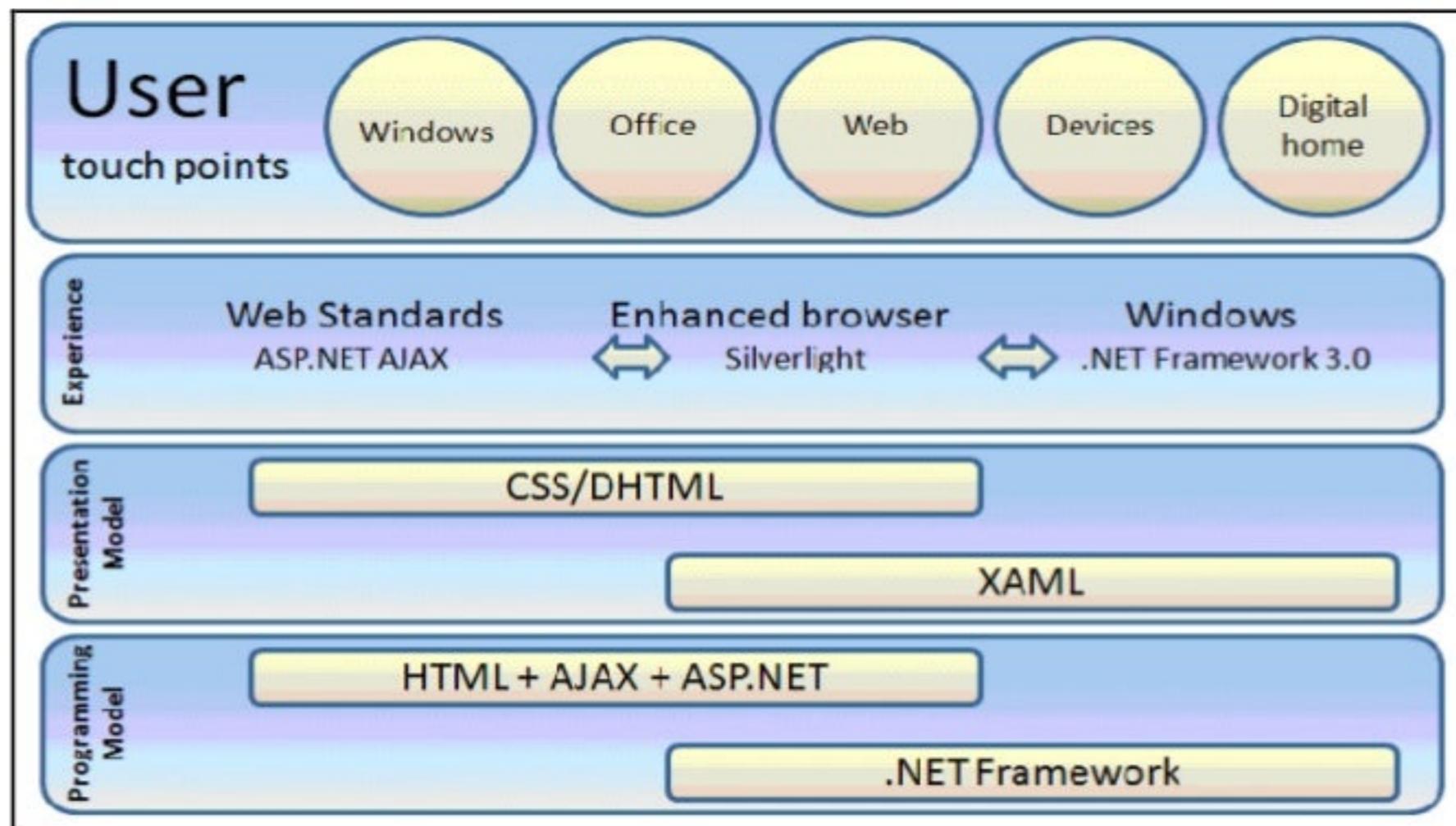


Figure 4.1: Silverlight Architecture

- Silverlight is the next step in evolving the potential user-experience richness in which application developers and designers can present to their clients. It does this by allowing designers to express their creativity and save their work in a format that will work directly on the Web. In the past, a designer would design a Web site and a user experience using tools that provide a rich output, but the developer would have to meet the constraints of the Web platform in being able to deliver them. In the Silverlight model, designers can build their desired user experience and express this as XAML. The XAML can then be incorporated directly by a developer into a Web page using the Silverlight runtime. Thus, the two can work more closely than ever before to provide a rich client user experience.
- It will have Web based GUI.
- It will have multiple numbers of instances on web server.

4.2.2 Data

4.2.2.1 File and Data Formats

Data at server side: XAML files, HTML files, Markup files, Source files, Project files, ASP.NET Server page.

When user selects to store pages designed on client side (i.e. on local HD) then the file is saved in XML format. This XML file may be opened in WebyWeb and the HTML code can be obtained and saved on server side.

4.3 Code

4.3.1 Introduction

WebyWeb has client server architecture. The GUI is made with Microsoft Blend 2 and the event handlers and codes are done in Microsoft Visual Studio 2008. This facilitates easy interface design along with coding which is reflected as any changes are made in design.

4.3.2 Modules

4.3.2.1 Module A: The Database

This module consists of user login and registration. It maintains the user's information and also user's files. It consists of two tables, one to include user's personal information and other to maintain user's files and projects. The input to database for personal information is given by the user in registration page. The files saved in editor are stored in the files table in database. The output is in the form of list of files made by the user so that he may choose to edit or delete files.

4.3.2.1.1 Internal Functions

- Navigation
 - It is used to navigate to the next page on user's login or on choosing to make a new project or edit current project.
 - Name of the function: Navigate
 - Return type: void

- Parameters: 1
 - Usercontrol newpage – It accepts the page to navigate to on call.
- Subscribe
 - This function is used to validate user's input in registration form and to accept the input and update it in the database.
 - Name of the function:
 - Return type: sup_btn_subscribe_Click
 - Parameters: 2
 - Object sender – It indicates the button clicked by the user.
 - RoutedEventArgs – It indicates the way call was made. I.e. on mouse over or mouse click etc.

4.3.2.2 Module B: The Main Editor

This module consists of the main editor IDE displayed to the user to make web pages. The input is in the form of various html elements inserted and text inserted by the user along with the formatting applied to the text. The output is in the form of HTML code generated by code generator. Finally user may save his work which is updated in the database.

4.3.2.2.1 Internal Functions

● Code Generator

- This method converts the xml code made from the design view rich text box of the user's document into the html code that may be viewed across any browser.
- Name of the function: ConvertRichTextToHTML
- Return type: string[]
- Parameters: 1
 - String RichTextXML – It accepts the xml format text from the design view and returns the converted html code in the form of string.

- **Update formatting text**

- This function continuously updates the formatting applied to the text in rich text box. Thus as the cursor is moved, this function is called to indicate the formatting applied. Thus the toolbar buttons act as both – formatting applier as well as indicator.
- Name of the function: `UpdateFormattingControls`
- Return type: `void`
- Parameters: No parameters

- **Adding styles dynamically**

- This function is used to insert new styles into the style list as the user applies new formatting to the text. It dynamically adds customs styles which may be later used for new text inserted in rich text box.
- Name of the function: `AddStyle`
- Return type: `void`
- Parameters: 1
 - String `styleID`: This indicates the `styleID` in the list of styles to indicate the custom style that may be applied.

4.3.3 Interfaces

Every module must expose one or more interfaces. If these interfaces are not trivial and clear from the module descriptions, they should be listed and described in an orderly fashion at this point. This description should detail the purpose of each interface and in what context it can be used. It should also mention what kind of modules or objects expose it and which ones will be using it. If using the interface requires obtaining a reference to it, there should be an explanation of how such a reference is to be obtained and what information will be needed in order to do so. Following the description of the interface, there should come a list of the interface functions, referring to the very same items mentioned above (see section 2.4.1.n Internal Functions).

It is advisable to add here a description of the scheme that will determine how the interfaces will evolve in future versions of the system in order to answer new requirements and modifications. If backward and forward compatibility is one of the requirements, explain how this scheme supports it.

4.4 Operation

4.4.1 User types

In broad perspective we can have 2 types of users for Weby Web:

1. Expert user: These are the users who are well acquainted with the web development and have worked with development softwares as well as have knowledge of coding.
2. Amateur user: These are the users who are completely new to webpage development and have no knowledge of coding, but wish to design their own web pages.

4.4.2 Scenarios

1. **Development Phase and design phase** - When a user logs on to the Weby Web, user will have an environment consisting of controls, menus, and tools for developing a basic HTML webpage.
2. **Preview Phase** - After the design phase, user previews his developed web page.

4.4.3 Installation

- No installation required at client side.
- Deployment of system on server.
- Resources needed for a successful installation:
 1. high speed connectivity
 2. Silverlight Plug in
 3. Web space (future scope).
- Centralized installation on the server.

4.4.4 Licensing

Weby Web is protected by a licensing scheme. This application is priced on a per-user basis or depends on the scheme user selects. License is validated on the basis of online query to a central server.

Purchasing/Sign-up- The purchase experience of a SaaS application is completed online. There is no third party or store between the client and the company.

4.4.5 Upgrades

- Upgradation is done at server side. No Upgradation is required at client side.
- Software will be installed on the server. All the data files are stored on the server. Unless the server crashes all the data will be secured.

Because the application runs on a central server a SaaS application can be constantly updated, ensuring the user they always dispose of the latest version. This way software can be easily innovated and users can see their feedback quickly incorporated in the software. When the user's feedback and developing the software go hand in hand, a developer can ensure the software will continue to improve. Therefore, SaaS development cycles should be periodic and “managed to no more than 90-180 days to enable incorporation of feedback and refinements”

4.4.6 Uninstall

Uninstallation of the software is performed on the server side. The owners of the WebyWeb are authorized for uninstallation. Customer's files will be left behind at server side so that the customer will have an opportunity to save it on their client machine.

4.5 Development

1. necessary resources – 3 PCs, Server, Web space
2. number of developers - 4,
3. Required skills
 - Knowledge about HTML
 - Networking protocols
 - Microsoft Visual studio 2008
 - Microsoft Expression Blend
 - Silverlight development
4. hardware requirements :
 - Intel® Pentium® or AMD processor, 1 GHz with MMX or equivalent
 - 512 MB of RAM
 - 350 MB available hard disk space

- Graphics processor that is DirectX® 9-capable
 - 1024 x 768 monitor resolution with 24-bit color
5. Environment and development tools required for the development process.
- Silverlight 1.0 SDK
 - MS visual studio 2008
 - MS Expression blend
 - .NET Framework 3.0 or higher
 - Microsoft windows XP with service pack 2 or Vista

4.6 Miscellaneous

4.6.1 Conformance with standards

WebyWeb which is developed is in accordance with the standards prescribed in MS visual studio 2008, Expression Blend and Silverlight. The system would conform to the W3C guidelines for HTML pages.

Link – <http://www.w3.org/MarkUp/>

4.6.2 Interoperability with other systems

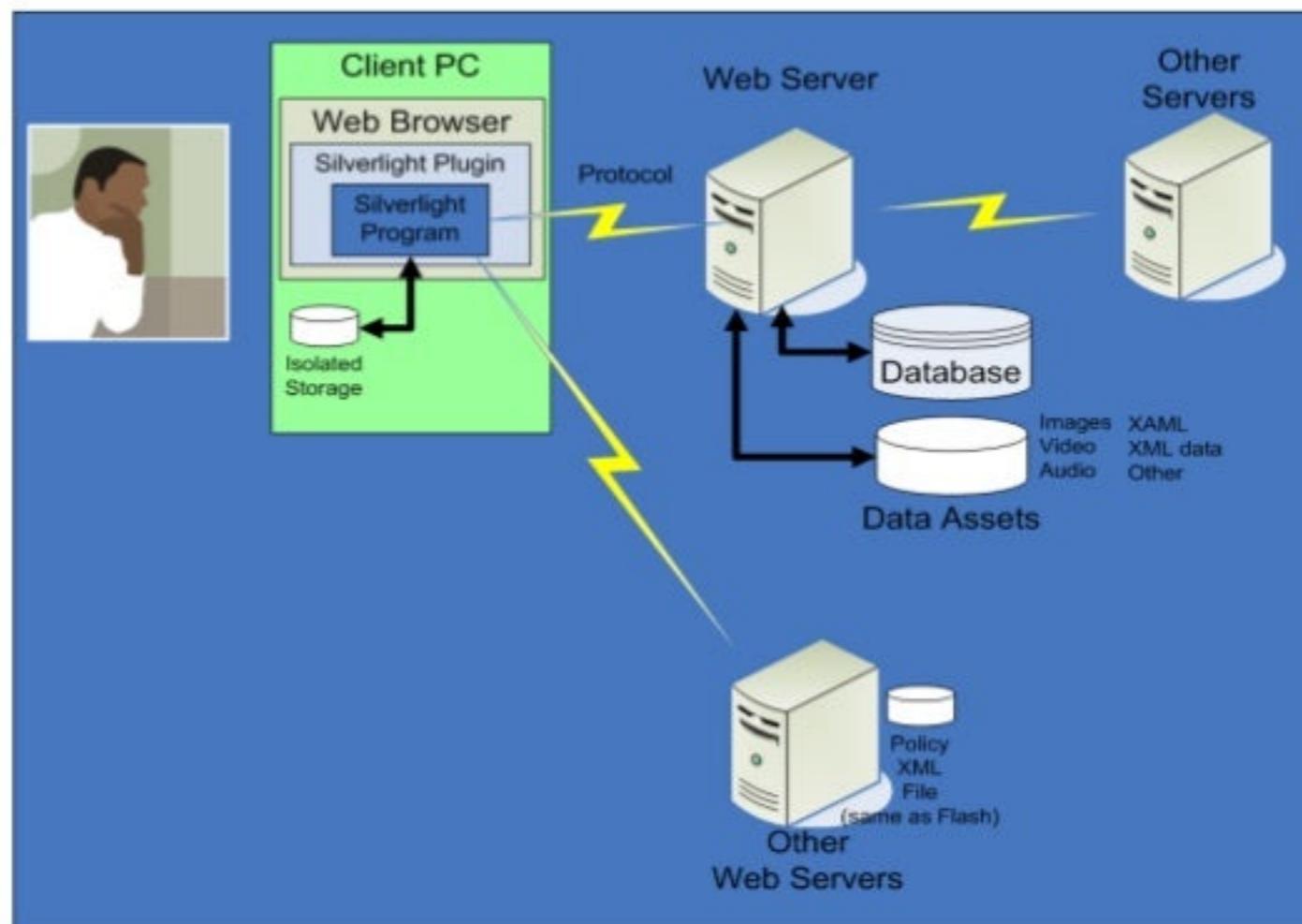


Figure 4.2: Interoperability with other systems

OS/browser	<u>Chrome</u>	<u>IE 6 SP1</u>	<u>Firefox/Mozilla</u>	<u>Safari</u>	<u>Opera</u>
Windows Vista/2008	N/A	N/A	1.0, 2.0	1.0, 2.0; via NPAPI	Unofficially
Windows XP/2003/Home Server	2.0	N/A	1.0, 2.0	1.0, 2.0; via NPAPI	Unofficially
Windows 2000	N/A	2.0	N/A	2.0; via NPAPI	Planned
Windows Mobile 6	N/A	1.0	N/A	N/A	N/A
Mac OS 10.4/10.5 PowerPC	N/A	N/A	1.0	1.0	Planned
Mac OS 10.4/10.5 Intel	N/A	N/A	1.0, 2.0	1.0, 2.0	Planned
Linux x86 (<u>Moonlight</u>)	N/A	N/A	Planned	N/A	Planned

Table 4.1 : Silverlight across various browsers

4.6.3 Security

Security is provided at the server side which will secure all the user's sessions.

4.6.4 Open Issues

Debugging, scalability, security are the open issues.

4.7 Analysis Models

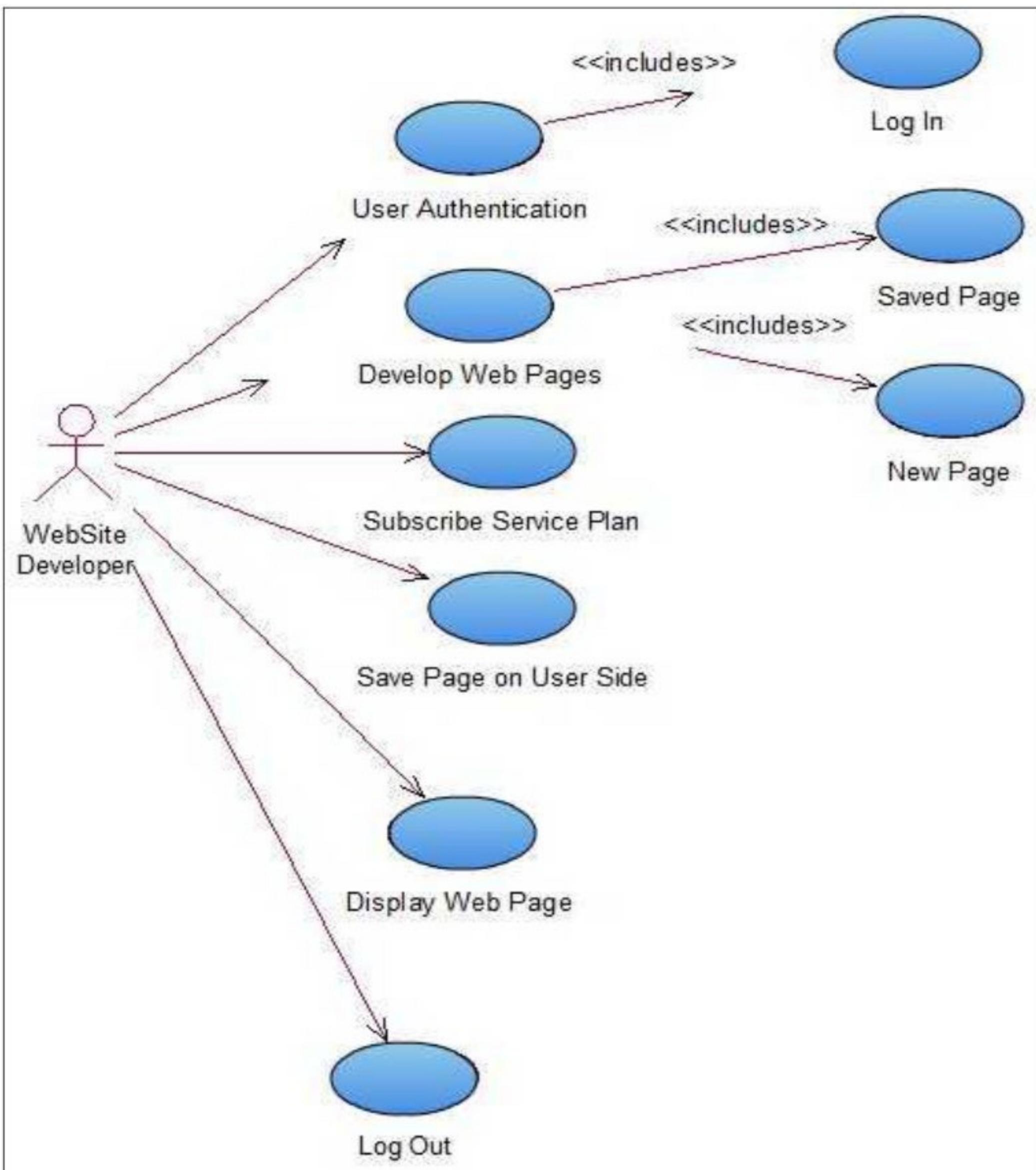


Figure 4.3: Use Case Diagram



Figure 4.4: Class Diagram

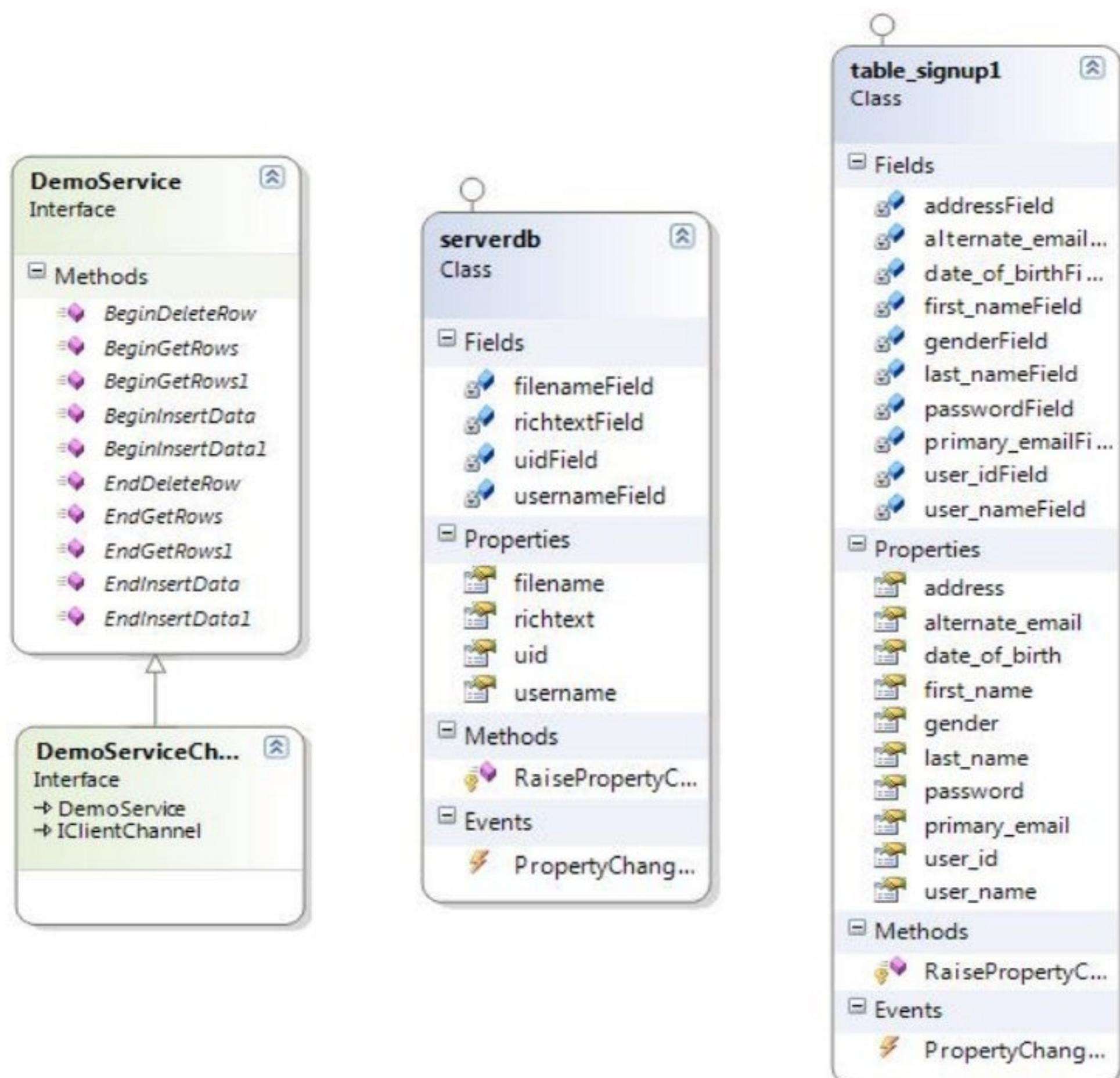


Figure 4.5: Class diagram for interfaces and classes

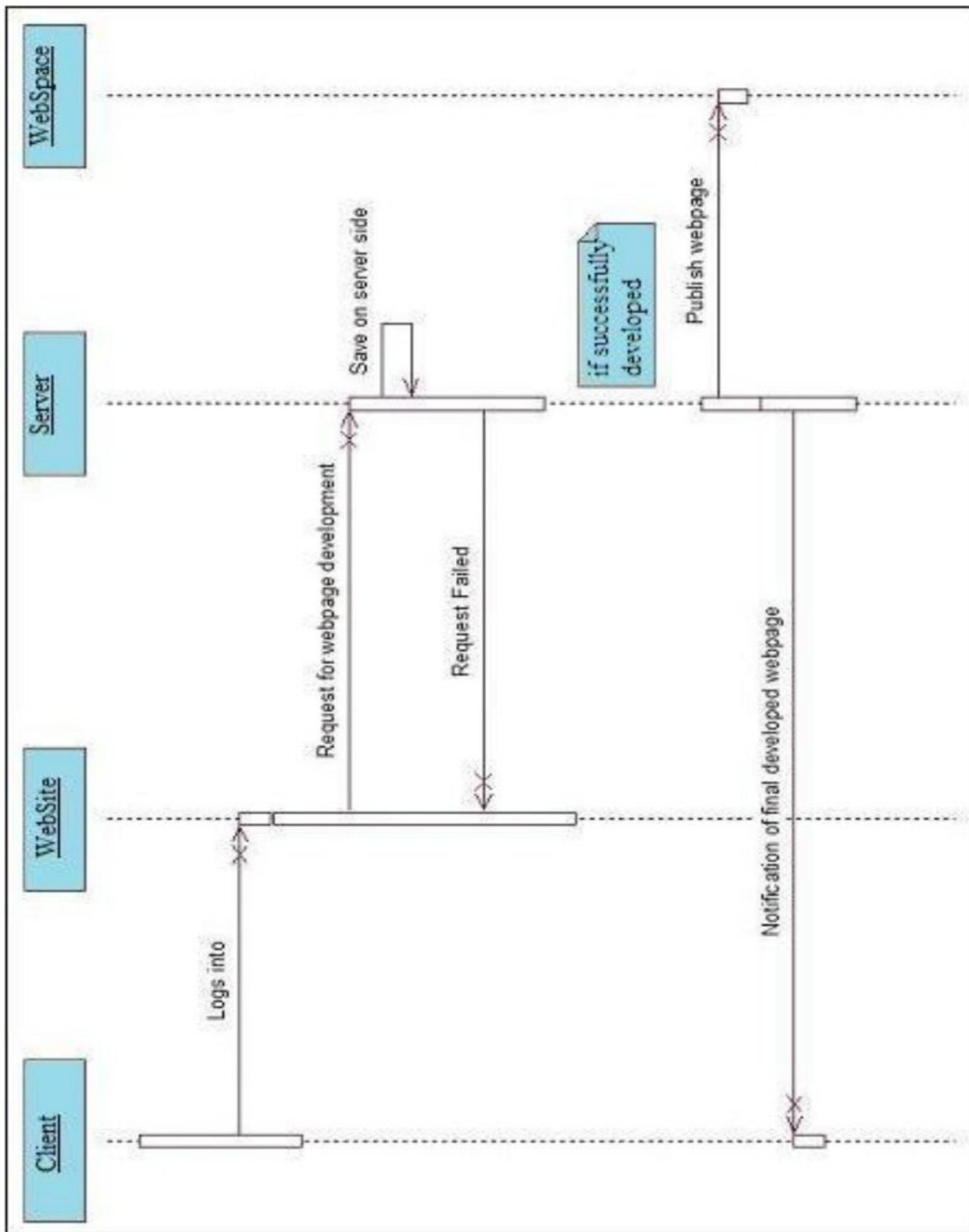


Figure 4.6: Sequence Diagram

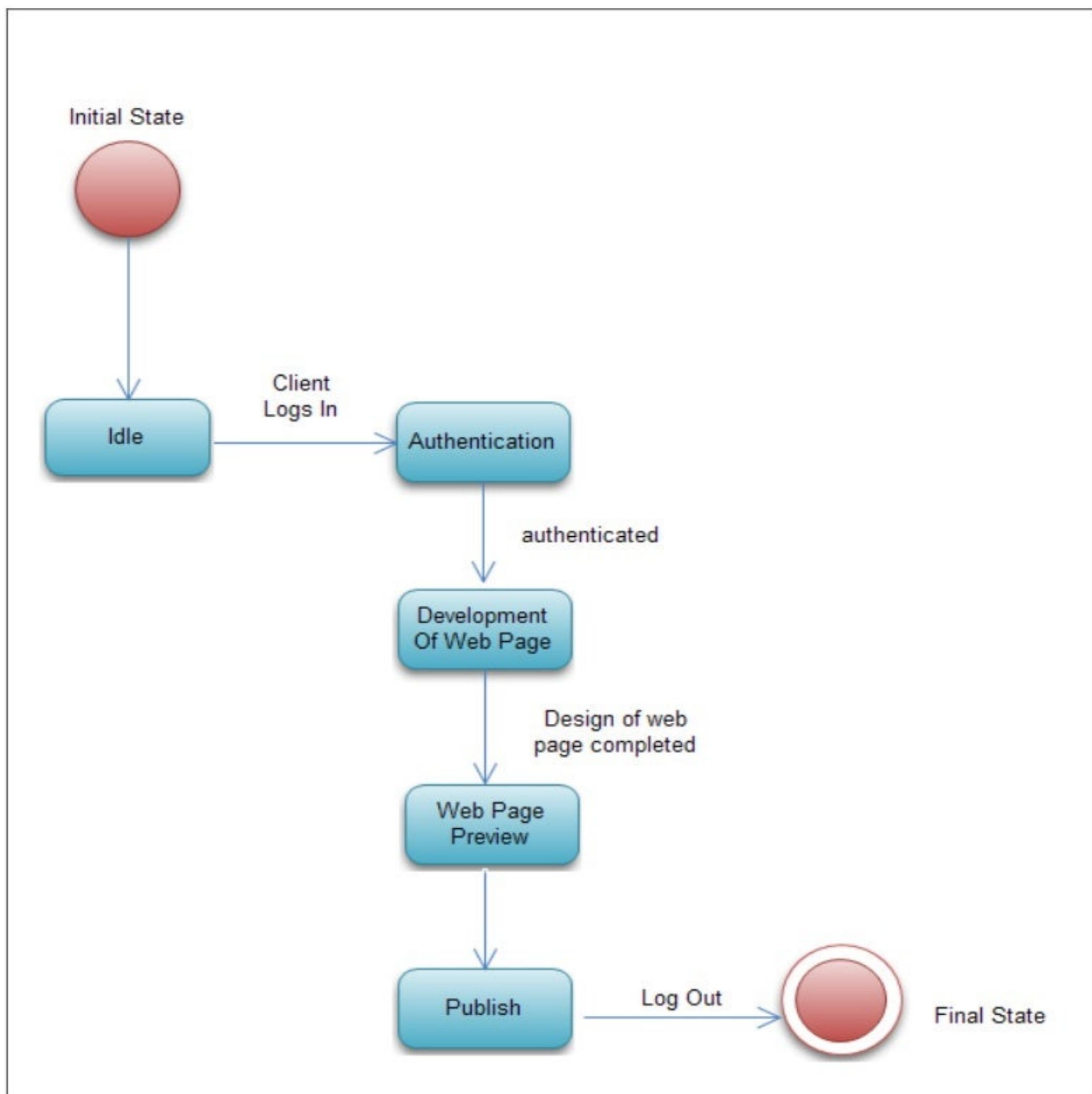
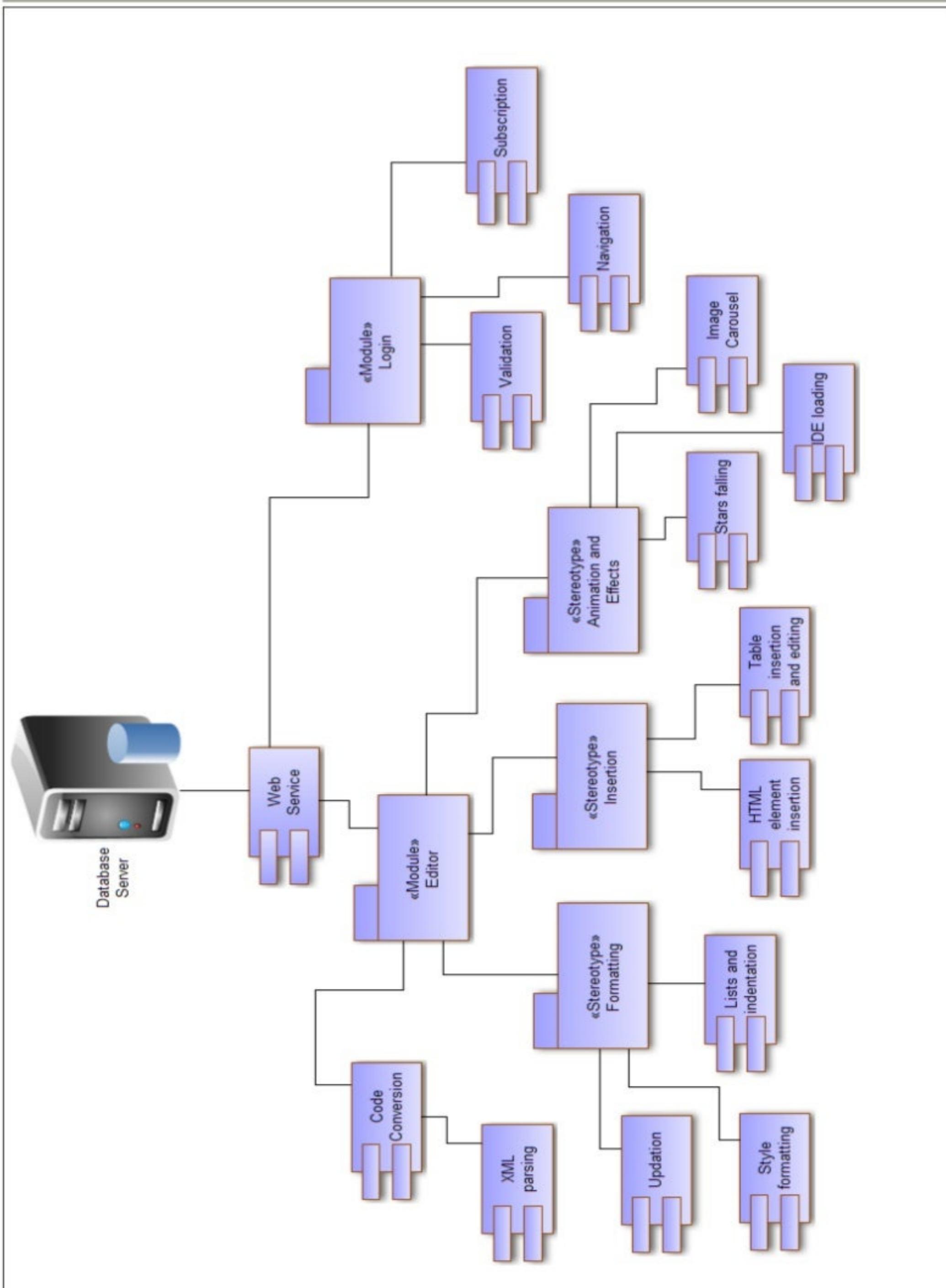
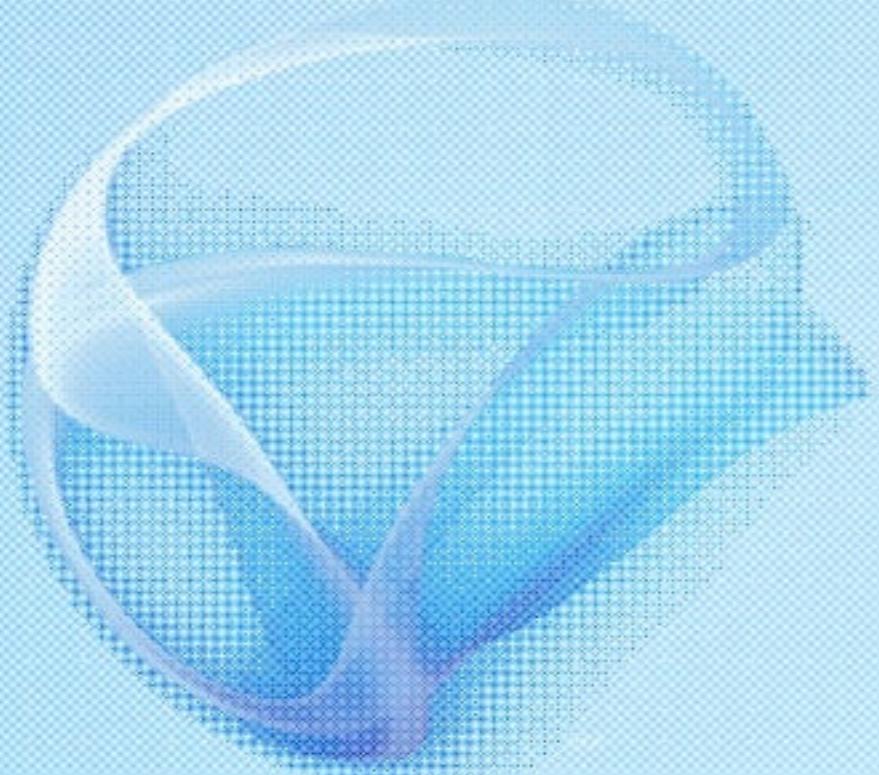


Figure 4.7: State Chart Diagram

**Figure 4.8: Component Diagram**



5. Testing

5. Testing

5.1 Test Plan

5.1.1 Test Plan Identifier:

- WebyWeb release 1.0 MTP 1.0

5.1.2 Introduction

In order to overcome the drawbacks of existing html based editors (installations, no centralized updations, unnecessary payment for features that are not going to be utilized etc), developers of WebyWeb felt the need to be able to offer its clients the opportunity to develop web pages online. This project's goal is provide a basic online web page development (which can be enhanced in the future) ASAP.

The initial release of the WebyWeb site and application will be known as WebyWeb release 1.0. The target audience will initially be for any novice user. Once the system is working successful, WebyWeb intends to start an aggressive marketing campaign to attract new clients for the online development of web pages through this application.

Specifically, testing will now consist of the following phases (listed chronologically):

- Unit and integration level – adherence to coding standards and successful communication between units.

5.1.3 Test Items

- Test items – Image, tables, radio button, check box, horizontal line, text field, label, hyperlink

The scope of this Testing activity will include:

- WebyWeb release 1.0 Web page development application and supporting infrastructure

The scope of this testing activity will not include:

- WebyWeb documentation e.g.: Requirements & Design Specifications or User, Operations & Installation Guides

5.1.4 Features to be tested

- Accessibility
- Coding standards
- Compatibility
- Functional
- Navigation-
 - Login Validation,
 - Registration form validation,
 - database connectivity,
- Scalability
- Security
- Usability

5.1.5 Features Not to Be Tested

It is the intent that all of the individual test cases contained in each test plan will be performed.

However, if time does not permit, some of the low priority test cases may be dropped.

5.1.6 Approach/Strategy

The philosophy of the testing is risk-based testing, i.e. each test case will be prioritized as, High, Medium, or Low priority and then scheduled accordingly (Highest first). Exceptions to this general rule might include instances where:

- A large number of low priority test cases can be executed using a small amount of resources
- A lower priority test is a pre-requisite of another higher priority test e.g. an expensive and high priority usability test might necessitate many of the inexpensive low priority navigational tests to have passed

The testing will use manual testing. The WebyWeb's source code will be frozen while being tested. Except for critical fixes that are blocking the testing efforts, changes will not be scheduled while a unit of code is being tested.

5.1.7 Item Pass/Fail Criteria

The entrance criteria's for each phase of testing must be met before the next phase can commence. Formal approval will be granted by the internal and external project guide.

The guides will retain the decision as to whether the total and/or criticality of any or all detected incidents/defects warrant the delay (or rework) of the WebyWeb release 1.0.

5.1.8 Suspension Criteria and Resumption Requirements

In general, testing will only stop if the Web site Under Test (**WUT**) becomes unavailable. If testing is suspended due to the Web site becoming unavailable, testing will be resumed once access to the Web site is reestablished. Certain individual test cases may be suspended, skipped or reduced if prerequisite tests have previously failed e.g. usability testing may be skipped if a significant number of Web page navigational tests fail.

5.1.9 Test Deliverables

The following documents will be generated as a result of these testing activities:

- Master test plan (MTP - this document)
- Deliverable documents: test plan
- Test input and output data (Test cases).

5.1.10 Test Environments

- Software Requirements-
 1. Silverlight 1.1/2 plug-in installed.
 2. The PC is connected to internet/ intranet.
- Hardware Requirements-
 1. 512 GB RAM.
 2. 1024 X 768 monitors resolution with 24 bit color.
 3. Graphics processor that is DirectX® 9-capable.

5.1.11 Staffing and Training Needs

- General development & testing techniques
- WebyWeb application development lifecycle methodology

5.1.12 Schedule

The following tentative schedule will hopefully be met:

- Test design (this document) is expected to be completed by the end of March.
- Test execution is expected to last no more than two weeks and to start immediately after the test plans have been approved and the Web application has been hosted

5.1.13 Risks and Contingencies

The following seeks to identify some of the more likely project risks and propose possible contingencies:

- Web site becomes unavailable – Testing will be delayed until this situation is rectified
- A large number of defects/incidents makes it functionally impossible to run all of the test cases
 - As many test cases as possible will be executed, The guides conjunction with developers will ultimately make the decision as to whether the number of defects/incidents warrants delaying the implementation of the production version.

- Not enough time to complete all test cases. If time cannot be extended, individual test cases will be skipped, starting with the lowest priority.

5.1.14 Approvals

The Internal and External guide must approve this plan.

WebyWeb release 1.0 MTP 1.0

UNIT & INTEGRATION TEST PLAN

5.1.15 Test Plan Identifier

WebyWeb release 1.0 UNTP 1.0

5.1.16 Introduction

This testing phase will use a number of testing techniques. The decision as to which technique(s) to use for any given unit of code will reside with the team leader responsible for signing-off on the Module.

5.1.17 Features and Functions to Test

- **Accessibility**

UIAC1 - Low

The colours used on this Web page are friendly to colour blind viewers

- **Compatibility**

UICO1 - Medium

Size of WebyWeb application resized automatically according to screen. Cross Browser support.

UICO2 – Medium

The content of the WebyWeb is clearly readable

- **Coding standards**

Each of the units of code that make up the module being tested (typically a single fully functional Web page) must be coded to all of the following coding standards, any deviations from the standard must be documented and approved

UICS1 – High

The code must pass the following syntax and design requirements:

- Each unit of code has been inherited or copied from the most appropriate object class or Template.
- HTML code must be coded to the W3C HTML 4.0 standard and validated via the W3C validation service.
- Error messages do not describe the internal workings of the program

UICS2 - Medium

Any Form used on the Web page meets the following requirements:

- If radio controls are used, a default is always selected
- The browser places the cursor on the most appropriate field/control when the Form is first viewed
- Using the browsers Tab key allows the client to tab through the input fields on the Form in a top to bottom, left to right order
- All data entry fields are checked for invalid data and an appropriate error message is displayed if the data is found to be invalid
- All validations are performed (and error messages displayed) in a top-down, left-to-right Fashion
- Using equivalence partitioning techniques, all data entry fields will be checked to ensure that they are able to accept valid values and that their error checking routines can handle invalid data appropriately

UICS3 - Low

Any Pop-up used on the Web page meets the following requirements:

- The pop-up follows WebyWeb GUI standard
- The pop-up is not too large for the parent window
- The pop-up's initial screen positioning is appropriate

- **Functional**

1. Cut, Copy, Paste, and Zoom.

2. Menu bar options,

3. Toolbar options-

- Insert -table, images, buttons, hyperlinks,
- Format – Applying Bold, Italics, left, right, center alignment, indent, out dent, ordered and unordered list, subscript, superscript, strikethrough, font size, font style, font color.
- Edit Table – Add rows & columns, delete row and column are the features in WebyWeb are to be tested.

- **Navigation**

UINA1 - High

All the links on the Web page will be checked to ensure that they meet the following specifications:

- The link should not be broken (unless the target has not yet been developed) and goes to the most appropriate location
- The link will not alter the browser's default link colours
- The link must have an associated "Title" link tag specified
- Internal links must use lowercase characters for the address

- **Security**

UISE1 - High

Input data received from the client must be parsed to make sure that it does not contain "out of bounds" or "buffer overflow" input data

UISE2 - High

Input data received from the client must be parsed to make sure that it does not contain inappropriate meta-character sequences e.g. &&

- **Usability**

UIUS1 - Low

Mandatory data entry fields may be flagged with a visual cue e.g. highlight in red

UIUS2 - Low

Related information is grouped together on the Web page to minimized eye movement

UIUS3 - Low

When viewed via the clients anticipated hardware/software the page fits without the need for a horizontal scroll bar

5.1.18 Features and Functions not to Test

Notable features and functions that will not be tested include: None

WebyWeb release 1.0 Test Plan Version 1.0

5.2 Test Procedure

All the tests are conducted using manual testing. Testing includes

5.2.1 Black Box Testing

Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional. Here, the valid and invalid input is selected and the correct output is determined. The testing is carried out for:-

- a. Authentication of user
- b. Generation of Web page using WebyWeb

5.2.2 Regression Testing

Regression testing seeks to uncover regression bugs. Regression testing is used not only for testing the correctness of a program, but it is also often used to track the quality of its output. The testing is carried out for:-

- Database
- Rich text box formatting (Bullets and numbering, Alignment)
- Code conversion Engine
- Menu and event handlers

5.2.3 Integration Testing

Integration testing, also known as integration and testing (I&T), is a software development process which program units are combined and tested as groups in multiple ways. The testing is carried out for:-

- Database integration into Main Editor
- Inclusion of tested and approved HTML editor into Main Editor
- Integration of Ultra Rich web elements with WebyWeb editor

5.3 Test Cases

5.3.1. “Sign In Module” Test Cases

- Test case for Sign In module with specific test data.

PROJECT : WEBYWEB MODULE : Sign In module	Document References : WEBYWEB-SRS- V1 TEST CASE NO :1 TEST DATE : 20-03-2009
FUNCTIONAL SPECIFICATION : User Authentication	
TEST OBJECTIVE : To check if the entered username and password are valid or invalid	
TEST DATA : USER NAME=“ Admin“ and PASSWORD =”WebyWeb “	

Test ID	Test Cases & User Action	Test Data	Expected Result	Actual Result	Comment
1	Blank Username & Valid Password Clicks On 'Sign In' Button	Username = "" & Password = "WebyWeb"	Display message "Username cannot be kept blank"	Valid message displayed	Working properly
2	Valid Username & Blank Password Clicks On ' Sign In' Button	Username = "Admin" & Password = ""	Display message "Password cannot be kept blank"	Valid message displayed	Working properly
3	Blank Username & Blank Password Clicks On ' Sign In' Button	Username = "" & Password = ""	Display message "Both fields are mandatory"	Valid message displayed	Working properly
4	Invalid Username & Valid Password Clicks On ' Sign In' Button	Username = "mcoe" & Password = " WebyWeb"	Display message "Invalid Username"	Valid message displayed	Working properly
5	Valid Username & Invalid Password Clicks On ' Sign In' Button	Username = "Admin" & Password = "webweby"	Display message "Invalid Password"	Valid message displayed	Working properly
6	Invalid Username & Invalid Password Clicks On ' Sign In' Button	Username = "mcoe" & Password = "passwd"	Display message "Both fields are invalid"	Valid message displayed	Working properly
7	Valid Username & Valid Password Clicks On 'Sign In' Button	Username = "Admin" & Password = "WebyWeb"	Go to the next page having users created project and details.	Proper form displayed	Working properly

Table 5.1: Test case for Sign In module

5.3.2 Test case for Sign Up module with specific test data.

PROJECT : WEBYWWEB MODULE : Sign Up module	Document References : WEBYWWEB - SRS- V1 TEST CASE NO :2 TEST DATE : 20-03-2009
FUNCTIONAL SPECIFICATION : User Registration	
TEST OBJECTIVE : To register user and check if the entered information is valid or not	
Precondition : Not a registered User, click on Sign Up button and registration form must be displayed	

Test Case ID	Test case name	Test case description	Test steps			Test status (P/F)	Test Priority
			Steps	Expected result	Actual Result		
Login01	Validate First Name and Last Name	To verify that First Name and Last Name must be of minimum 2 characters	Enter First Name and Last Name less than 2 characters	An Error Message “First Name and Last Name can not be less than 2 characters ” & error icon Must be displayed.	Error Message & icon Displayed	P	Medium
			Enter First Name and Last Name of 3 characters	No Error Icon	No Error Message Displayed	P	Medium
Pwd01	Validate Password	To verify that password should be between 4 to 12 characters	Enter password greater than 12 characters	An Error Message “password cannot be greater than 12 characters ” & error icon Must be displayed.	Error Message & icon Displayed	P	High
			Enter password of 10 characters	No Error Icon displayed (Valid password)	No Error Icon Displayed	P	High
			Enter password less than 4 characters	An Error Message “password cannot be less than 4 characters ” & error icon Must be displayed.	Error Message & icon displayed	P	High

Pwd02	Verify Password	To verify that password should be exactly same as entered in “Choose a Password”	Enter different password	An Error Message “password does not match” & error icon Must be displayed.	Error Message & icon Displayed	P	High
			Enter same password	No Error Icon displayed (Valid password)	No Error Icon Displayed	P	High
DateOfBirth01	Validate entered Date of Birth	To verify date of birth (DOB<=current date)	Select invalid date of birth	An Error Message “Invalid date of birth” & error icon Must be displayed.	Error Message & icon Displayed	P	High
			Select Valid date of birth	No Error Icon displayed (Valid DOB)	No Error Icon Displayed	P	High
EmailId 01	Validate primary email ID	To verify primary email ID does not start with special chars (@)	Enter primary email ID starting with @. E.g.-(@yahoo.com)	An Error Message “Primary email ID cannot start with this “ & error icon Must be displayed.	Error Message & icon Displayed	P	High
			Enter primary email ID starting with numbers. E.g.- (1a@yahoo.co m)	An Error Message “ 1 st Char of Primary email ID cannot start with this “ & error icon Must be displayed	Error Message & icon Displayed	P	High

			Enter valid primary email ID E.g.- (sam@yahoo.com)	No Error Icon displayed (Valid primary email Id)	No Error Icon Displayed	P	High
Subscription01	Validate Registration Form	To verify that all mandatory fields are filled properly	Mandatory fields are Incomplete	Error Message "Incomplete information" & error icons on corresponding fields must be displayed.	Error Message & icon Displayed	P	High
			Mandatory fields are completely filled	No Error Icon displayed (registration form filled completely)	No Error Icon Displayed	P	High

Table 5.2: Test case for Sign In module

5.3.3 Black Box Testing

Test Case Description	Test Case Id	Test Description	Expected Results	Actual Results	Test Status Pass/Fail	Severity
Code Generation	B001	Insertion of HTML elements	Selected element in rich text box with entered value	Element inserted in rich text box with entered value	PASS	High
		Rich text to HTML conversion	Converted HTML code in code view	HTML code generated	PASS	High
		Style Generation	Selected style applied	Custom style appears in style tag of HTML code	PASS	High
Login	B002	Registered User	Access to user's project list	Updated Project list displayed to user	PASS	Medium
		Unregistered user	Deny access	Access denied and appropriate message displayed	PASS	High
Sign Up	B003	Open a new account of user	Creation of user's account and entry into the database	User's account created & database update.	PASS	High

Table 5.3: Black Box testing

5.3.4 GUI Testing

Test case Description	Test case Id	Test Description	Expected Result	Actual Result	Test Status [Pass/Fail]	Severity
Aesthetic	GT01.1	All the input text field to be observed	All the input fields should be blank, with proper focus & tab indices	Input fields are blank with focus and tab index set	PASS	High
	GT01.2	Proper display of toolbar panes	On click, proper pane should be displayed	On click, proper pane displayed	PASS	High
	GT01.3	Tool tips should be observer	Appropriate tool tips for corresponding buttons.	Proper tool tips displayed on mouse over of buttons	PASS	Medium
	GT01.4	Table tools	All the table editing tools should be disabled when table is not in focus	No edit tools active in absence of focused table	PASS	Medium
Validation	GT02.1	Loading projects	Populating corresponding projects of the signed in user	Projects populated for the user	PASS	High
	GT02.2	Assignment of entered caption to HTML elements	Correct assignment of values to HTML form elements	Correct assignment of values to HTML form elements done	PASS	Medium
	Gt02.3	Full screen with scaling	In full screen mode, IDE should be scaled and editable	IDE is scaled but text editing is disabled	PARTIAL	High

Table 5.4: GUI testing

5.4 Test Oracle

An oracle is a mechanism for determining whether the program has passed or failed a test. Oracle acts as a reference guide for evaluating a product. A complete oracle would have three capabilities and would carry them out perfectly:

1. A Generator – To provide predicted or expected result for each test.
2. A Comparator – To compare predicted and obtained results
3. An Evaluator – To determine whether the comparison results are sufficiently close to be a pass.

5.4.1 Test Design

In oracle based testing we compare the behavior of the program under test to the behavior of the source we consider accurate.

I. WebyWeb is web editor software that was designed to compete with Macromedia Dreamweaver

1. Getting Started with Webpage development

In Dreamweaver, we need to install the software if it is not installed on the client computer which includes several steps.

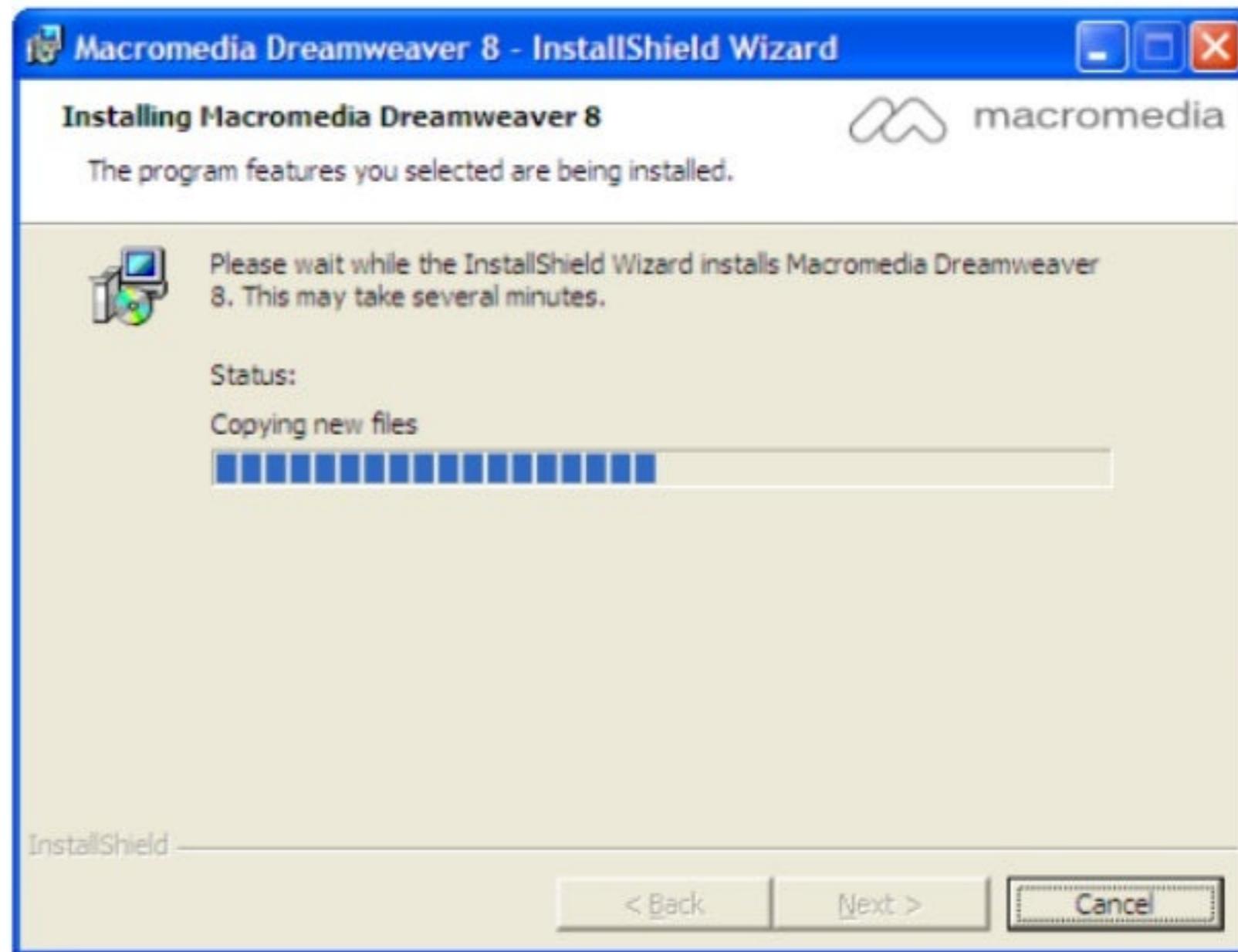


Figure 5.1 Dreamweaver Installation

With WebyWeb the user just needs to log on to the WebyWeb website and sign in.

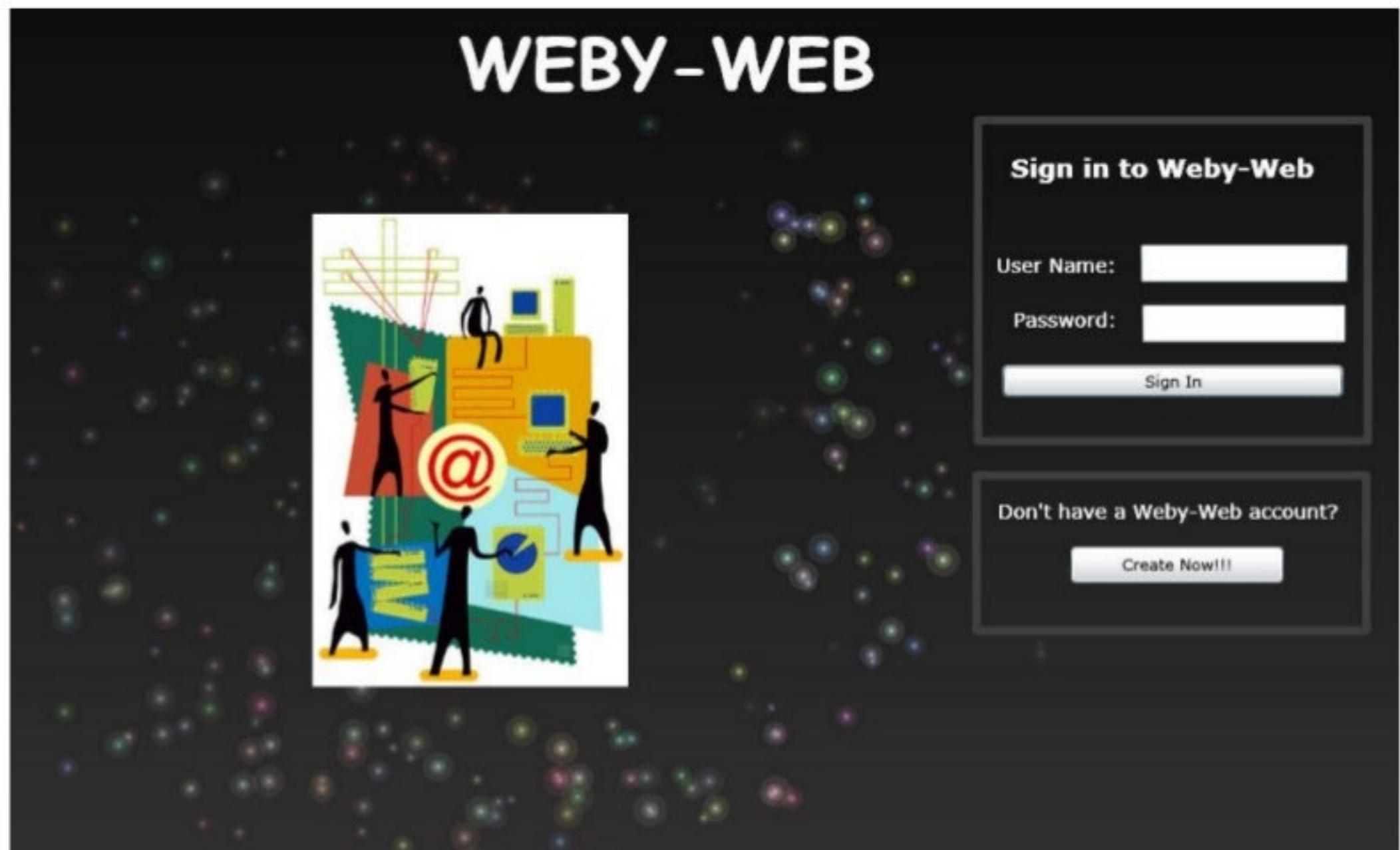


Figure 5.2 WebyWeb Homepage

2. Complexity of use

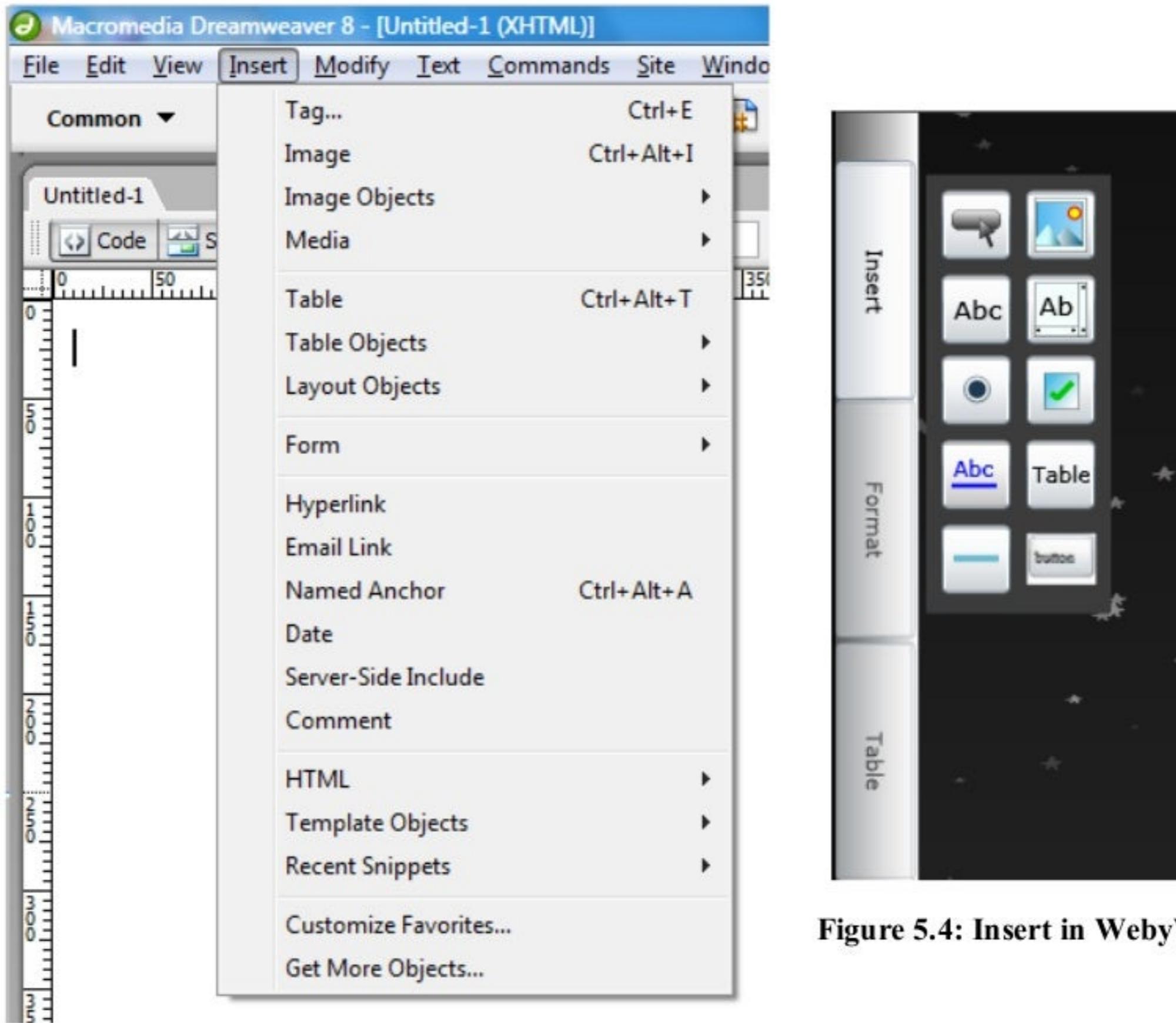


Figure 5.3: Insert in Dreamweaver

In Dreamweaver it is quite complex to insert available elements in web page. A novice user may find it difficult to insert elements.

WebyWeb has an easy to use dashboard interface which includes toolbars which make it easier for the user to insert html elements into the webpage.

Figure 5.4: Insert in WebyWeb

II. WebyWeb is web editor software that was designed to compete with Google Sites

1. Insertion of HTML Form elements like – Buttons, Radio buttons, checkboxes etc

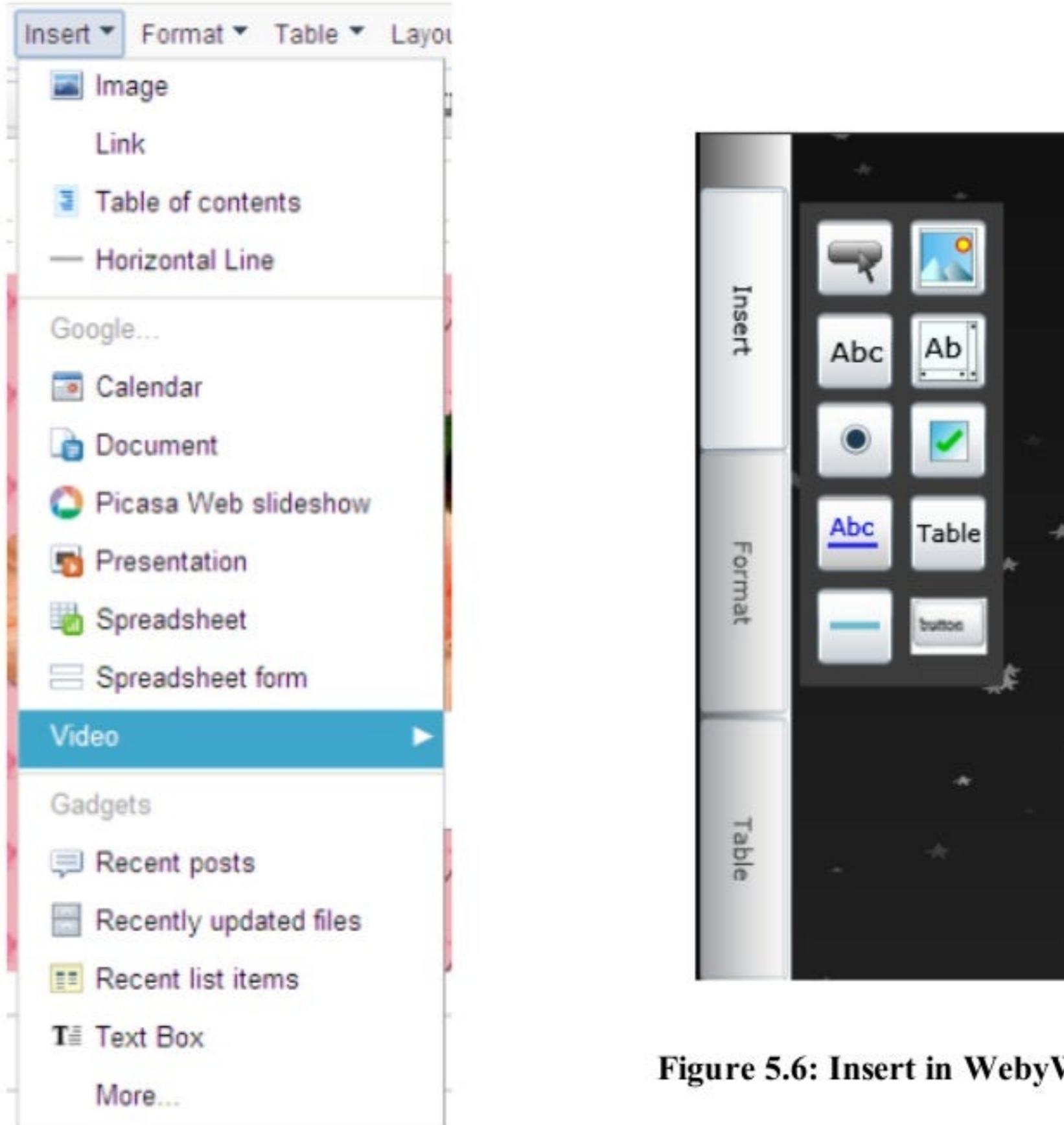


Figure 5.6: Insert in WebyWeb

Figure 5.5: Insert in Google

These elements insertion is not provided in Google sites. So designing simple forms is not possible in Google sites

Whereas these elements are provided in WebyWeb

Future Enhancements-

- **Debugger**

When a coder who has HTML knowledge designs a webpage, then sometimes it's easier for him to make changes in the code view and apply them rather than in design view. But in such cases it's difficult to detect errors and debug them. So a debugger will be helpful to easily design error free webpage even from the code view.

- **Syntax highlighting**

Syntax highlighting is a feature that displays text in code view in different colors and fonts according to the category of terms. This feature eases writing in a structured language such as a markup language like HTML as both structures and syntax errors are visually distinct. Highlighting does not affect the meaning of the text itself; it's made only for human readers/editors. Syntax highlighting is one strategy to improve the readability and context of the text.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML
2 <html>
3   <head>
4     <title>Example</title>
5     <link href="screen.css" rel="sty
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <ul id="nav">
12      <li>
13        <a href="one/">One</a>
14      </li>
15      <li>
16        <a href="two/">Two</a>
17      </li>
```

Figure 6.1: Syntax Highlighting

- **Memory and traffic management**

Giving facilities to the user to manage webspace as per the built website. Managing the visitors of the website and traffic log. Real-time visitor analytics, pagerank tracking, and more - all in one place.

- **Spell check**

Spell checking flags words in the document that may not be spelled correctly. This feature can help user avoid spelling mistakes in the content written and thus make a webpage containing correct content. A comprehensive spell checker enables website owners to spell check their website content during webpage creation much before they decide to publish it to their live website.

- **User 's site management**

This includes allowing user to view complete website from single editor. Thus user has better control over the website and addition, deletion and interlinking of pages. Moreover it gives user a better view in perspective of space required for the website.

- **Facility to design dynamic web pages**

Sometimes some dynamism in the webpage makes it more interactive. It includes support for ASP and Java Scripting which may be integrated in the webpage, making it dynamic. Some features like Custom URL, Dynamic style sheets, Widgets, conditional visibility may also be added.

Conclusion-

The purpose and objective of WebyWeb project is achieved. By providing extremely rich graphical user interface, web page designing is easy and in an aesthetic form. Flexibility in designing makes user explore their imagination and thus, even a novice user can dream and accomplish their wish of web page designing.

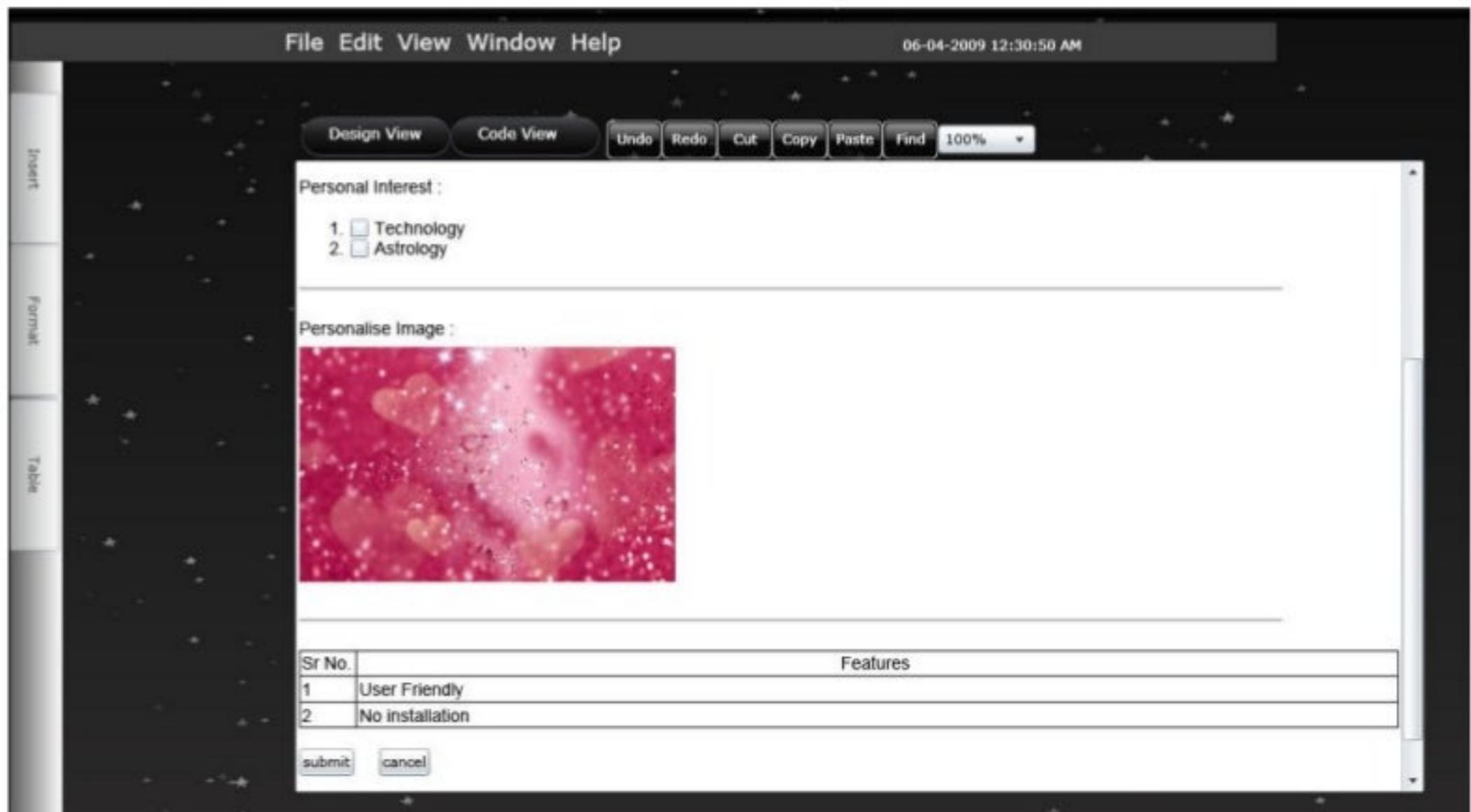


Figure 7.1 : Sample web page developed in WebyWeb

Appendix A: Glossary

SR.NO	ABBREVIATION	MEANING
1	SaaS	Software as a service
2	RIA	Rich internet application
3	WYSIWYG	What you see is what you get
4	W3C	World wide web consortium
5	IETF	Internet engineering task force
6	HTTP	Hyper text transfer protocol
7	HTTPS	HTTP Over Secured Socket layer
8	TCP	Transmission control protocol
9	Firewall	A logical barrier designed to prevent unauthorized or undesired communication between sections of a computer network
10	UDP	User datagram protocol
11	RFC	Request For Comments
12	POX	Plain Old XML

References

- [1] From Wikipedia, “HTML”,
<http://en.wikipedia.org/wiki/html>.
- [2] M.T. Hoogvliet, “SaaS Interface Design”, presented at Rotterdam University, 2008.
- [3] From Wikipedia, “On-demand Pricing”,
<http://en.wikipedia.org/wiki/On-demand>.
- [4] Christian Wenz, Essential Silverlight 2 Up-to-Date, O'Reilly, 2008.
- [5] <http://www.Adobe.com/Dreamweaver>
- [6] From Wikipedia, “Microsoft Expression Web”,
http://en.wikipedia.org/wiki/Microsoft_Expression_Web.
- [7] Jeff Scanlon, Accelerated Silverlight 2, Apress, 2008.
- [8] Brennan Williams, Microsoft Expression Blend UNLEASHED, SAMS, 2008.
- [9] Matthew MacDonald, Pro Silverlight 2 in C# 2008, Apress, 2008.