



WEB2WORKS – WEB 2.0 FRAMEWORK DEVELOPMENT

By

P- 2254

Arul Kumar P A

Registration Number: 71205621005

of

Kumaraguru College of Technology



Saravanampatti,

Coimbatore– 6.

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

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

of

MASTER OF COMPUTER APPLICATIONS


ANNA UNIVERSITY

CHENNAI 600 025

June 2008

BONAFIDE CERTIFICATE

Certified that this project report titled “**Web2Works – Web 2.0 Framework Development**” is the bonafide work of **Mr. Arul Kumar P A** (Registration Number:71205621005) who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.



SUPERVISOR

Mr.A.MuthuKumar M.Sc., M.C.A.,
Assistant Professor
Department of Computer Applications
Kumaraguru College of Technology
Coimabatore-6



HEAD OF THE DEPARTMENT

Dr.M.Gururajan M.Sc., Ph.D.,
Department of Computer Applications
Kumaraguru College of Technology
Coimbatore-6

Submitted to Project and Viva Examination held on 01.07.08



INTERNAL EXAMINER



EXTERNAL EXAMINER



April 24, 2008

PROJECT COMPLETION CERTIFICATE

This is to certify that **Mr. Arul Kumar P A** bearing trainee id no. 748274 student of final semester MCA from **Kumarguru College of Technology, Coimbatore** bearing Roll no: 05MCA05 (71205621005) has completed the project titled **Web2Works - Web 2.0 Framework Development Project (W2W)** at Wipro Technologies, EC4, T-17, First Floor, C wing, Bangalore – 560 100 under my guidance from December 10, 2007 to April 24, 2008 for academic requirements of his post graduation. We have observed that his work has been excellent and appreciate his sincere learning. He has performed the project with energy and enthusiasm.

I wish him all the best for all his future endeavors. This letter is issued based on his request.

Vijayanand Krishnan
Academy Head – e-Enabling

ABSTRACT

Web2Works is a project for creation of Wipro's Web 2.0 framework (W2W) from Wipro. The W2W framework enables you to drag-and-drop Web 2.0 components like Wikis, Chats, Blogs, RSS Feeds, Web Services and enterprise portlets to a central canvas, mash them up and synthesize a Web Application rapidly. It helps a semi-developer to compose new applications and services using a workbench of ready-made components (made available in a component explorer). The components can be saved as patterns and patterns can then be turned into deployable services. Web2Works brings you a layout similar to Eclipse, in the context of a browser. The user of Web2Works is typically a semi-technical person who wishes to concentrate on business logic to create a new application rapidly. The intent can also be personalization of an existing application or service. Web2Works is built on top of Ajax and Ajax-based Google Web Toolkit.

Web2.0 is the state of mind that desires to experience the web faster, better and cheaper. Web2Works is built on various recent technologies which are still buzz words in IT industry. To make the things from dreams to reality we made use of J2EE, AJAX, MySQL, GWT, RSS, JSON and REST.

The project is divided into multiple modules – core platform development, social networking integration, co-browsing capability development, telephony service integration, client-specific widget creation and domain-specific pattern implementation. All the above modules have distinct client-side and server-side sub-modules.

This is a framework development; we need to create readymade components for application. Here, we mostly concentrate on core platform and social networking integration.

ACKNOWLEDGEMENT

No significant achievement can be a solo performance especially when starting a project from ground up. Here I would like to acknowledge their precious co-operation and express my sincere gratitude to them. First of all, I thank the Almighty god, who blessed me the greatest sense and my parents for their kind prayers for completing this project work successfully.

I wish to express my deep unfathomable feeling of gratitude and indebtedness to **Dr. Joseph V. Thanikal, PhD, Principal – Kumaraguru College of Technology, Coimbatore** for the successful completion of the project work.

I am very gladly taking this opportunity to express a special word of thanks to **Dr. M. Gururajan M.Sc., PhD, Head of the Department, Kumaraguru College of Technology, Coimbatore** for encouraging me to do this work.

I am very much indebted to **Mr. A. Muthukumar M.Sc., M.C.A., Kumaraguru College of Technology, Coimbatore** for his complete assistance, guidance and support given to me throughout my project.

It is my pleasure, I am very grateful to **Mrs. R. Vanitha M.C.A., M.Phil., Kumaraguru College of Technology, Coimbatore** and my beloved internal guide. I express my sincere thanks to her for providing me with full freedom, constant encouragement, and valuable guidance.

I like to thank my external guide **Dr. Kingshuk Banerjee, Head, PCM COE, Wipro Technologies** for his valuable innovative ideas and constructive suggestions that all made my dreams come true.

I wish to thank **Mr. Vijay Anand Krishnan, E-enabling Academic head, Wipro Technologies** for his encouragement and providing this project.

I also like to thank the Staff Members of MCA, Members of Wipro PCM COE team and my beloved friends for their encouragement and valuable support.

TABLE OF CONTENTS

	ABSTRACT.....	iii
	LIST OF TABLES.....	vii
	LIST OF FIGURES.....	viii
I	INTRODUCTION.....	1
	1.1 Organization profile.....	1
	1.2 Wipro’s Center of Excellence.....	2
	1.3 Introduction to W2W framework.....	2
	1.4 Features of W2W.....	3
	1.5 Module description.....	4
	1.6 Scope.....	4
II	UNDERSTANDING REQUIREMENTS.....	5
	2.1 System study.....	5
	2.2 Architectural design.....	6
	2.3 System environment.....	8
	2.4 Programming environment.....	8
III	PROOF OF CONCEPTS.....	13
	3.1 PoC on GWT.....	13
	3.2 PoC on RSS.....	16
	3.3 PoC on JSON.....	17
	3.4 PoC on REST.....	19
IV	PLANNING ARCHITECTURE.....	23
	4.1 System flow diagram.....	23
	4.2 Usecase diagram.....	24
	4.3 Sequence diagram.....	25
	4.4 Activity diagram.....	27
	4.5 Class diagram.....	28
	4.6 Database design.....	31

V	PROJECT EXECUTION.....	35
	5.1 Module Description.....	35
	5.2 Module Execution.....	37
VI	PROJECT TESTING.....	38
	6.1 Purpose of Testing.....	38
	6.2 Unit Testing.....	38
	6.3 Integration Testing.....	38
	6.4 System Testing.....	39
	6.5 Acceptance Testing.....	39
	6.6 Test Cases.....	39
VII	PROJECT ROLLOUT.....	41
	7.1 Status.....	41
	7.2 Future enhancements.....	41
VIII	APPENDICES.....	42
	Appendix 1 Sample Screenshots.....	42
	Appendix 2 Abbreviations.....	56
IX	REFERENCES.....	57

LIST OF TABLES

Table No	Table Name	Page No.
3.1	HTTP header methods	19
4.1	UserDetails	32
4.2	PatternDetails	32
4.3	ServiceDetails	32
4.4	FeedDetails	33
4.5	TrackingDetails	33
4.6	CallDetails	34
4.7	SMSDetails	34
4.8	ConferenceDetails	34
6.1	Test cases	40

LIST OF FIGURES

Fig. No	Figure	Page No.
2.1	Architectural design of Web2Works	6
3.1	Structure of a JSON object	17
3.2	Structure of a JSON array	18
3.3	JSON value	18
3.4	Architecture of REST	20
4.1	Web2Works System flow diagram	23
4.2	Web2Works use case diagram	24
4.3	YouTube.com sequence diagram	25
4.4	BlogSpot.com sequence diagram	26
4.5	YouTube.com activity diagram	27
4.6	Web2Works interface diagram	28
4.7	Web2Works server side class diagram	29
4.8	Web2Works client side class diagram	30
4.9	Database Relationship diagram	31
A8.1	Login Page	42
A8.2	CNN RSS Feed Content	43
A8.3	Cnn.com RSS feed parsed	44
A8.4	Yahoo! Pipes JSON Feed	45
A8.5	Yahoo! Pipes JSON feed parsed	46
A8.6	YouTube Search result	47
A8.7	Blogger Login Page for blogspot.com	48
A8.8	Publishing a new post	49
A8.9	List of existing posts	50
A8.10	Posts in blogspot.com after publishing a post	51

A8.11	updating an existing post title	52
A8.12	updating existing post content	53
A8.13	My blog after update	54
A8.14	Delete a post	55

CHAPTER 1

INTRODUCTION

1.1 ORGANIZATIONAL PROFILE

Wipro Technologies is a global services provider delivering technology-driven business solutions that meet the strategic objectives of clients. Wipro has 40+ 'Centers of Excellence' that create solutions around specific needs of industries. Wipro delivers unmatched business value to customers through a combination of process excellence, quality frameworks and service delivery innovation.

Wipro is the World's first CMMi Level 5 certified software services company and the first outside USA to receive the IEEE Software Process Award. Wipro becomes the first Indian IT Service Provider to be awarded Gold-Level Status in Microsoft's Windows Embedded Partner Program. Wipro is the world's largest independent R&D Services Provider and Worlds 1st PCMM Level 5 software company. Wipro is one among the few companies in the world to be assessed at maturity level 5 for CMMI V1.2 across offshore and onsite development centers, 2007.

Wipro is the world's 1st IT Services Company to use Six Sigma and pioneers in applying Lean Manufacturing techniques to IT services. It is the largest independent R&D Services provider in the world. Over half billion revenue from R&D, among the top 3 offshore BPO services provider in the world, A strategic partner to five of the top ten most innovative companies in the world, only Indian company to be ranked among the top 10 global outsourcing providers in IAOP's 2006 Global Outsourcing 100 listing. World's first SEI CMM/CMMI Level 5 IT services company. Wipro is the first to get the BS15000 certification for its Global Command Centre. Functional RFID Enabled Concept Store and Global Data Synchronization

Laboratory BS7799 and ISO 9000 certified. Among the top 3 offshore BPO service providers in the world.

1.2 WIPRO'S CENTER OF EXCELLENCE

Wipro is a strategic partner to five of the top ten most innovative companies in the world. There are over 50 industry facing 'Centers of Excellence' 647 clients - 72000+ employees and 53 development centers across globe. Portal and Content Management is one of the leading vertical in Wipro which has a dedicated Center of Excellence for it. PCM-COE is a Research and Development division where future generation technologies are used to build applications. Web2Works is such an initiative that makes use of Web2.0

1.3 INTRODUCTION TO W2W FRAMEWORK

Web2Works is a unique platform for rapid synthesis of next generation web applications. It provides building blocks and patterns to build and deploy customized application services on-demand for a single client, multiple users, or for the entire enterprise using a visual drag-and-drop mash-up platform. In the collaborative world of Web 2.0, the end-users need access to a wide variety of information for running their operations. This information may be available across multiple applications, sources within the enterprise as well as outside. In addition each end-user may need different sets of data depending on the role or function that he or she has to perform.

In such a scenario, giving the Banking Customers or Business users the freedom to quickly build and deploy their own customized mash-ups of data from different sources would considerably increase the speed with which new mash-ups can be generated by the end-users themselves based on their specific requirements. How can end-users build and deploy their own client-side mash-ups with very little

Wipro has developed Web2Works, a business and technology framework, which helps in the rapid synthesis of client-side mash-ups to address this challenge. Wipro Web2Works provides an easy-to-use, drag-and-drop platform, wherein the end-users can easily build and deploy their own client-side mash-ups. Web2Works helps to adopt client-side mash-ups, a successful Web 2.0 functionality, into the client's enterprise.

Web2Works is customized to derive domain-specific solutions for the Telecom, Banking, Financial Services, Securities and Insurance industries.

1.4 FEATURES OF W2W

- (i) W2W is a business and technology framework built on concepts and principles of Web 2.0 that enables rapid synthesis of a Service Delivery Platform
- (ii) It is designed for telecom service providers (for other domains too) which helps them in creating more value added services
- (iii) Has strong commitment to open-source stack(one implementation flavor is based on Google Web toolkit and JBoss)
- (iv) The Framework supports concepts like social networking, collaboration, user generated content, blogging, podcasting, Wikis and multichannel feeds
- (v) Enables creation, orchestration, registration and deployment of new services, based on combination of pre-existing, registered services (e.g. tracking, gaming, billing)
- (vi) Offers client-side mash-up of existing services through browser-based, drag-and-drop

1.5 MODULE DESCRIPTION

In this collaborative world of Web 2.0, we are developing this web2works framework with several drag-and-drop and mash up components to develop and deploy their own customized data depending on their requirements. It is identified that Web2Works have the following modules:

- Core platform development
- Social networking integration

Core platform development is all about building the rich user interface in Web2.0 style and developing Web2Works components that is industry independent. This module helps users to develop Web 2.0 applications with just drag and drop of few components. So, much care is taken while developing this module to make it as user friendly as possible. This module includes developing visually pleasing interface, RSS syndication, JSON feeds and REST. It is called as the core platform because all other modules are mostly built on this and it decides how the user will interact with the system.

Social Networking is linking people each other in some way. Popular way of linking is blogging and YouTube video search. Blogging refers to personal online dairy and here we make use of blogspot.com API's. So, that user can still scribble in their online diary from Web2Works without visiting blogspot.com. User can also search and watch videos from Web2Works without visiting YouTube.com.

1.6 SCOPE

Scope of this project is to create a Web 2.0 framework that is browser based platform independent and helps the user to synthesis Web 2.0 applications rapidly. There will be predefined components that can be used to create applications. W2W can be used in any industry. However, right now the project is done with a focus on

CHAPTER 2

UNDERSTANDING REQUIREMENTS

2.1 SYSTEM STUDY

Web 2.0 is also known as “Semantic Web”. Web 2.0 is a state of mind. This concept is different from early web development because it moves away from static websites to dynamic WWW.

2.1.1 Objective

The objective of this project is to bring you a layout similar to Eclipse, in the context of a browser. The user of Web 2 Works is typically a semi-technical person who wishes to concentrate on business logic to create a new application rapidly. The intent can also be personalization of an existing application or service. Web2Works is built on top of Ajax and Ajax-based Google Web Toolkit. The rich technologies [AJAX, RSS, and GWT for rich interactive interfaces] provide such a wonderful things that make our dreams come true.

We have developed this framework for semi technical developer who can drag and drop the components from the components explorer and mash up them.

Then the developer can deploy that as a service or they can save as a pattern for future use. Basically JavaScript is a client side tool along with AJAX we can make the framework as more interactive as like a desktop application (means easy and faster). AJAX is very much useful to refresh and load the page contents dynamically. By the use of AJAX (XMLHTTP request) we can refresh a part of the web page without refreshing the whole. This can reduce the overhead. Thus we are achieving the faster.

Next is GWT, this is an open source provided by Google, whose purpose is to provide a conversion between java and java script.

2.1.2 Users of W2W

Earlier world less number of users created web pages for large number of users. But now all can publish web pages in the web. For that we need an advanced technology to manage, integrate and collaborate things and need to provide valid information. Web2Works can be used by any semi technical developers. To provide a valuable work web2.0 is used.

2.2 ARCHITECTURAL DESIGN

We have framed the architecture for Web2Works. The schematic diagram is show in the following figure.

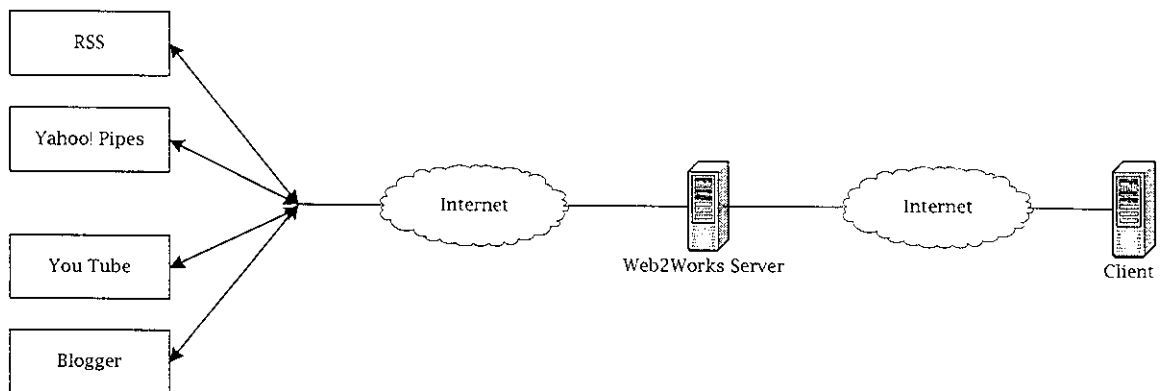


Fig. 2.1 Architectural design of Web2Works

Web2Works can be divided into two parts:

1. Client Side
2. Server Side

2.2.1 Client Side

Client side is basically an interactive user interface and an RPC mechanism to connect with server. The user interface and RPC is created using GWT. User interface can be divided into three parts:

- Component explorer
- Pattern and service explorer
- Work space

Apart from this we have tool bars for manipulating with services and patterns [save, delete, modify]

2.2.1.1 Component Explorer

Component explorer holds all the components which are ready to use for developing rapid applications. Gadgets, Bloggers, RSS, JSONS feeds, YouTube are few of the components available.

2.2.1.2 Patterns and Service Explorer

It is like a Windows explorer where the existing patterns and services are displayed in a tree view. User can drag and drop the existing components to modify it.

2.2.1.3 Workspace

This is the area where you can drag and drop the components for cooking the services or patterns.

2.2.2 Server Side

Server side plays a vital role where all the things happen. All requests from the client are made through RPC (Remote Procedure Call). Server communicates to the remote servers for the required data.

2.3 SYSTEM ENVIRONMENT

2.3.1 Hardware specification

Processor	:	Intel Pentium 2.66 GHz
Main Memory	:	2 GB
Secondary Memory	:	80 GB SATA
Monitor	:	Wipro 15" Flat display

2.3.2 Software specification

Operating System	:	Windows XP Pro
Back End	:	MySQL
Front End	:	JavaScript, GWT, Ajax
Programming Language	:	J2EE
Web Server	:	Tomcat
Browser	:	Firefox or IE 6+

2.3.3 Tools Used

IDE	:	Eclipse
-----	---	---------

2.4 PROGRAMMING ENVIRONMENT

2.4.1 J2EE

Platform independence-that is, the ability of a program to move easily from one computer system to another-is one of the most significant advantages that Java has over other programming languages, particularly if your software needs to run on many different platforms. If you're writing software for the World Wide Web, being able to run the same program on many different systems is crucial to that program's success.

Java is platform independent at both the source and the binary level. Of course, embedding program code in a web page creates special security requirements. As an Internet user accessing a page with embedded Java code, you need to be confident that it won't do anything that might interfere with the operation of your computer, or damage the data you have on your system. This implies that execution of the embedded code must be controlled in such a way that it will prevent accidental damage to your computer environment, as well as ensure that any Java code that was created with malicious intent is effectively inhibited. Java implicitly incorporates measures to minimize the possibility of such occurrences arising with a Java applet.

Java's support for the Internet and network-based applications generally doesn't end with applets. For example, Java Server Pages (JSP) provides a powerful means of building a server application that can dynamically create and download HTML pages to a client that are precisely customized for the specific request that is received. Of course, the pages that are generated by JSP can themselves contain Java applets. Apart from these Java provides many features which cannot be explained here.

2.4.2 Web 2.0

Web 2.0 is a network as 'platform', spanning all connected devices. The Key concepts behind Web 2.0 are Weblogs (Blog), Web feeds (RSS or Atom), Podcasting, Social Networking and Mashups. Web 2.0 takes good things from traditional web and has included lot of new things. Web 2.0 makes use of niche technologies and architectures like AJAX, RSS, JSON, Atom, RDF, SOA and REST. Web 2.0 can be further classified into RIA (Rich Internet Applications) and REA (Rich Enterprise Applications).

2.4.3 AJAX

AJAX stands for "Asynchronous JavaScript And XML". It is a web

platform technology usable on many different operating systems, computer architectures and web browsers as it is based on open standards such as JavaScript and XML. It uses XHTML and CSS, for making up and styling information. AJAX is commonly used for making web applications which are Rich Internet Applications (RIA). A classic example of AJAX is 'Google Maps'. Using AJAX technology in Web 2.0 applications results in the following significant benefits:

- Communication is faster, because less data is transmitted.
- The user stays on the same page, because less navigation from page to page is required
- The reloaded page does not flicker, because only small regions of the page get updated with each Ajax request.

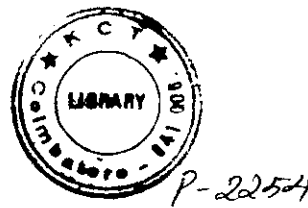
2.4.4 GWT

Google Web Toolkit (GWT) is an open source framework whose purpose is to compiling Java source code into JavaScript. GWT is an open source Java development framework that lets you escape the technologies that make writing AJAX applications so difficult and error prone. With GWT, you can develop and debug AJAX applications in the Java language using the Java development tools of your choice. When you deploy your application to production, the GWT compiler translates your Java application to browser-compliant JavaScript and HTML. GWT provides the following features and which makes easier to develop the applications.

- Dynamic, reusable User Interface Components,
- Really Simple RPC mechanism
- Browser History management and compatibility
- Completely open source.

The Advantages of Java to Java Script conversion are:

- Common JavaScript errors (like type mismatches) are easily caught at compile time rather than by users at runtime.
- Static type checking in the Java language boosts productivity while



- We can use all of our Java development IDE like eclipse

2.4.5 Tomcat

The Tomcat server is a Java based Web Application container that was created to run Servlets and Java Server Pages (JSP) in Web applications. Tomcat implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations.

2.4.6 MySQL

One of the most powerful SQL servers out there is called MySQL. It is a multithreaded, multi-user database management system. The Structured Query Language (SQL) is a very popular database language, and its standardization makes it quite easy to store, update and access data. One of the feature of MySQL Include: Handles large databases, in the area of 50,000,000+ records without memory leaks.

2.4.7 RSS 2.0

RSS stands for Really Simple Syndication. It is a family of 'Web feed' format used to publish frequently updated content. Earlier version was now called 'Rich Site Summary'. RSS makes it possible for people to keep up with their favorite web sites in an automated manner that's easier than checking them manually.

2.4.8 JSON

JavaScript Object Notation is the abbreviation for JSON. It is a simple, clean and light weight data-integration format. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages. It is built on two structures: A collection of name/value pairs and

2.4.9 REST

Representational state transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. REST strictly refers to a collection of network architecture principles which outline how resources are defined and addressed. RESTful web service can be used as an alternative to REST-less application.

2.4.10 Blogspot.com API

The Blogger data API allows client applications to view and update Blogger content in the form of Google data API feeds. Your client application can use the data API to create new blog posts, edit or delete existing posts, and query for posts that match particular criteria.

2.4.11 YouTube.com API

The YouTube APIs and Tools open up a full range of YouTube functionality for web pages, software and devices. Users of the API can upload new videos, edit video metadata such as comments and favorites and modify user settings. Your client application can request a list of videos that match specific search criteria, get access to the most popular YouTube videos or retrieve and maintain YouTube user information such as playlists, subscriptions or contacts.

CHAPTER 3

PROOF OF CONCEPTS

3.1 PoC ON GWT

Proofs are given for both server and client side of GWT, where in client side will have packaged JavaScript and to communicate with server side we have RPC mechanism. GWT is used to develop UI components [widgets] to make our application faster and smarter. We write all our code in Java where GWT converts the java code to corresponding JavaScript code. GWT has four following main components.

3.1.1 Java to JavaScript Compiler

Compiler converts the Java code into JavaScript code

3.1.2 Hosted Web Browser

The GWT Hosted Web Browser lets you run and execute your GWT applications in hosted mode, where your code runs as Java in the Java Virtual Machine without compiling to JavaScript. To accomplish this, the GWT browser embeds a special browser control (an Internet Explorer) with hooks into the JVM.

3.1.3 JRE Emulation Library

It contains most widely used packages of Java like Java.lang and Java.util packages [rest of packages of Java are not supported with this GWT].

3.1.4 Web UI Class Library

It contains set of custom interfaces and classes to create Widgets like buttons,

3.1.5 RPC Mechanism

GWT uses Remote Procedure Call [RPC], for interacting with server across a network and this RPC makes easy for the client and server to pass Java Object back and forth over HTTP. When used properly, RPCs give you the opportunity to move all of your UI logic to the client, resulting in greatly improved performance, reduced bandwidth, reduced web server load, and a pleasant user experience.

3.1.6 Service

The server-side code that gets invoked from the client is often referred to as a service, so the act of making a remote procedure call is sometimes referred to as invoking a service. To be clear, though, the term service in this context isn't the same as the more general "web service" concept. In particular, GWT services are not related to the Simple Object Access Protocol (SOAP).

In order to *create a service*, create a client side Java interface that extends the RemoteService tag interface.

```
public interface MyService extends RemoteService {
    public String myMethod(String s);
}
```

Any *implementation* of this service on the server-side must extend RemoteServiceServlet and implement this service interface.

```
public class MyServiceImpl extends RemoteServiceServlet implements
    MyService {
    public String myMethod(String s) {
        // Do something interesting
        return s;
    }
}
```

Asynchronous Interfaces are necessary to make a remote call from the client; we must create another interface, an asynchronous one, based on our original service interface.

```
interface MyServiceAsync {  
    public void myMethod(String s, AsyncCallback callback);  
}
```

The nature of asynchronous method calls requires the caller to pass in a callback object that can be notified when an asynchronous call completes, since by definition the caller cannot be blocked until the call completes. For the same reason, asynchronous methods do not have return types; they must always return void. After an asynchronous call is made, all communication back to the caller is via the passed-in callback object.

The relationship between a service interface and its asynchronous counterpart is straightforward:

- If a service interface is called `com.example.cal.client.SpellingService`, then the asynchronous interface must be called `com.example.cal.client.SpellingServiceAsync`. The asynchronous interface must be in the same package and have the same name, but with the suffix `Async`.
- For each method in your service interface,

```
public ReturnType methodName(ParamType1 param1, ParamType2 param2);
```

an asynchronous sibling method should be defined that looks like this:

```
public void methodName(ParamType1 param1, ParamType2 param2,  
    AsyncCallback callback);
```

3.2 PoC ON RSS

A Web Feed is a data format used for serving users with frequently updated content. RSS is a family of 'Web Feed' formats and makes it possible for people to keep up with their favorite web sites in an automated manner that's easier than checking them manually. RSS 2.0 is just an XML file with predefined XML tags.

An RSS 2.0 specification tells a list of channel elements that are required and optional. The required channel elements are title, link and description all other optional. The same applies for item element too. RSS is parsed in java using SAX parser where the starting and ending tag of each element is matched.

RSS 2.0 Format

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>Lift Off News</title>
    <link>http://liftoff.msfc.nasa.gov/</link>
    <description>Liftoff to Space Exploration.</description>
    <language>en-us</language>
    <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
    <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <generator>Weblog Editor 2.0</generator>
    <managingEditor>editor@example.com</managingEditor>
    <webMaster>webmaster@example.com</webMaster>

    <item>
      <title>Star City</title>
      <link>http:// nasa.gov/news/2003/news-starcity.asp</link>
      <description>
        How do Americans get ready?
      </description>
      <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
    </item>

    <item>
```

```

<link>http://liftoff.msfc.nasa.gov/</link>
<description>
    Sky watchers in Europe
</description>
<pubDate>Fri, 30 May 2003 11:06:42 GMT</pubDate>
</item>
</channel>
</rss>

```

3.3 PoC ON JSON

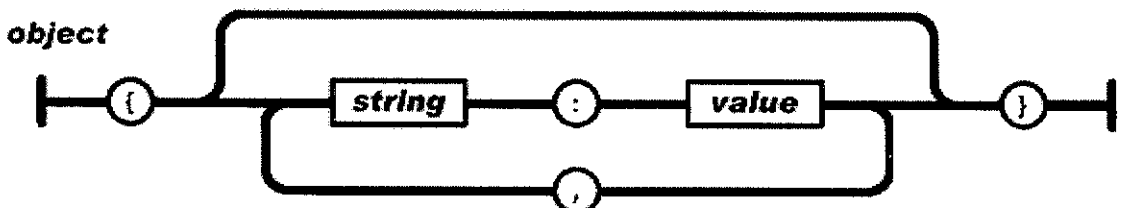
JSON stands for “JavaScript Object Notation” and it is a light weight data interchange format. JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, and dictionary.
- An ordered list of values. In most languages, this is realized as an array, vector, and list.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).



An *array* is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).

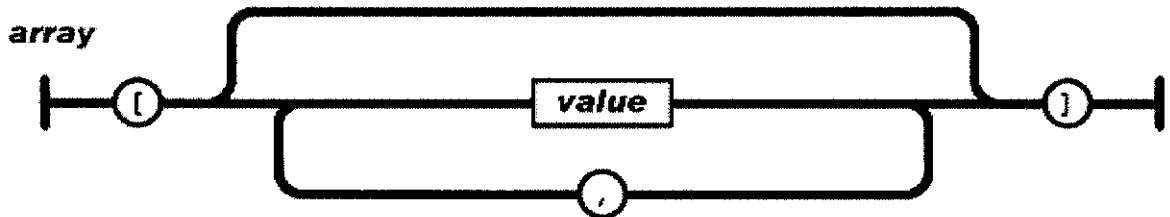


Fig. 3.2 Structure of a JSON array

A *value* can be a string in double quotes, or a number, or true or false or null, or an object or an array. These structures can be nested.

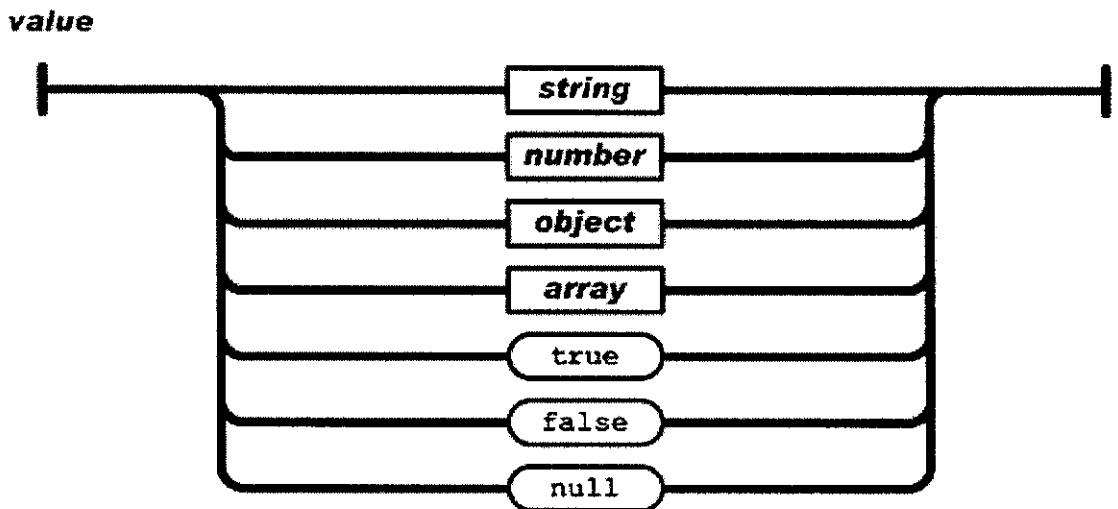


Fig. 3.3 JSON value

JSON Format

```
{
  "ObjectName1" :
  {
```

```

        "Name2" : "Value2"
    ],
    "ObjectName2" :
    [
        "Name" : "value",
        "Name2" : "Value2"
    ]
}

```

3.4 PoC ON REST

REST stands for Representational State Transfer. This POC talk about how RESTful web services can be implemented in Web2Works using Java. REST is implemented using Jersey 0.6EA API (A project in Java 1.5). REST helps us to create web services in a simpler way.

Java 1.5 (yet to be released) will provide support for REST by default. Jersey API used in this project is a 'Early Adopters' version which has got a long way to go for beta version.

GET and POST are the two most commonly used HTTP header methods. But, there are some more which are not much familiar to us. There are four HTTP methods available as per W3C. They are:

HTTP	CRUD
POST	Create, Update, Delete
GET	Read
PUT	Create, Update
DELETE	Delete

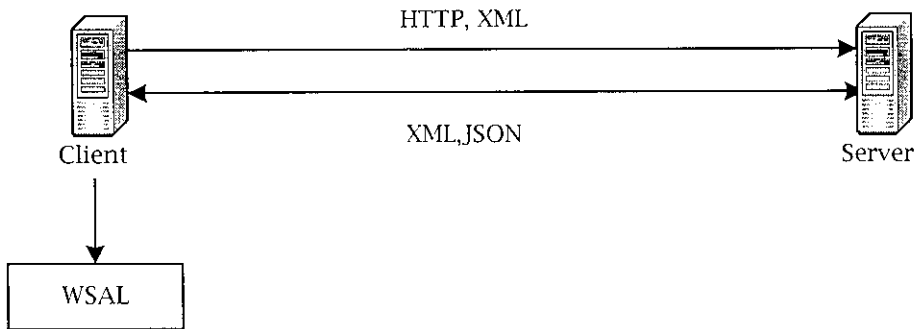


Fig. 3.4 Architecture of REST

3.4.1 Client

Client has WSAL file to make request to appropriate URL's. The request is made via HTTP protocol using XML.

3.4.2 Server

Server has a defined port number for RESTful Web service which is used to receive the request. Once the server receives the request it checks the HTTP header methods. Based on the header server acts accordingly whether to insert, update, delete or select. Response from the server is sent through XML or JSON format.

3.4.3 WSAL

WSAL (Web service application language) resides in client. It has defined URI to which the request should be made in order to get, add, modify or delete data.

Code:

```

// To start REST server

public class RESTServer
{
    HttpServer server;
    public void startServer()
  
```

```

handler=ContainerFactory.createContainer(Handler.class, RESTServlet
.class);

        try
        {
            server=HttpServer.create(new
InetSocketAddress(9998),0);
            server.createContext("/", handler);
            server.start();
            System.out.println("Server started. Port :
9998");
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }

    public void stopServer()
    {
        Server.stop();
    }
}

// Code to handle incoming request
@Path("/")
public class RESTServlet implements MessageBodyReader
{

    @GET
    @Produces("text/plain")
    @Path("item/{id}")
    public String getItemId(@UriParam("id") String itemid)
    {
        return itemid;
    }

    @DELETE
    @Produces("text/plain")
    @Path("item/{id}")
    public String deleteItem(@UriParam("id") String itemid)
    {
        return "Item deleted";
    }
}

//REST client implementation
class RESTClient
{
    public void access()
    {
        RESTServer ws=new RESTServer();
        ws.startServer();
    }
}

```



```
        try
        {
            URL url=new URL("http://localhost:9998/item/45");
            HttpURLConnection
urlConn=(HttpURLConnection)url.openConnection();
            urlConn.setRequestMethod("DELETE");
            urlConn.setDoOutput(true);
            urlConn.connect();

            input=new DataInputStream
(urlConn.getInputStream());
            while(null!=(str=input.readLine()))
            {
                outputStr = outputStr + str;
            }
            input.close();

            System.out.println("SERVER RESPONSE : " +
outputStr);
        }

        catch (Exception e)
        {
            e.printStackTrace();
        }

    }
}
```

CHAPTER 4

PLANNING ARCHITECTURE

To build Web2Works we made use of agile methodology for development. Because, it is an innovation project it has fluid requirements and developers were need to be flexible.

4.1 SYSTEM FLOW DIAGRAM

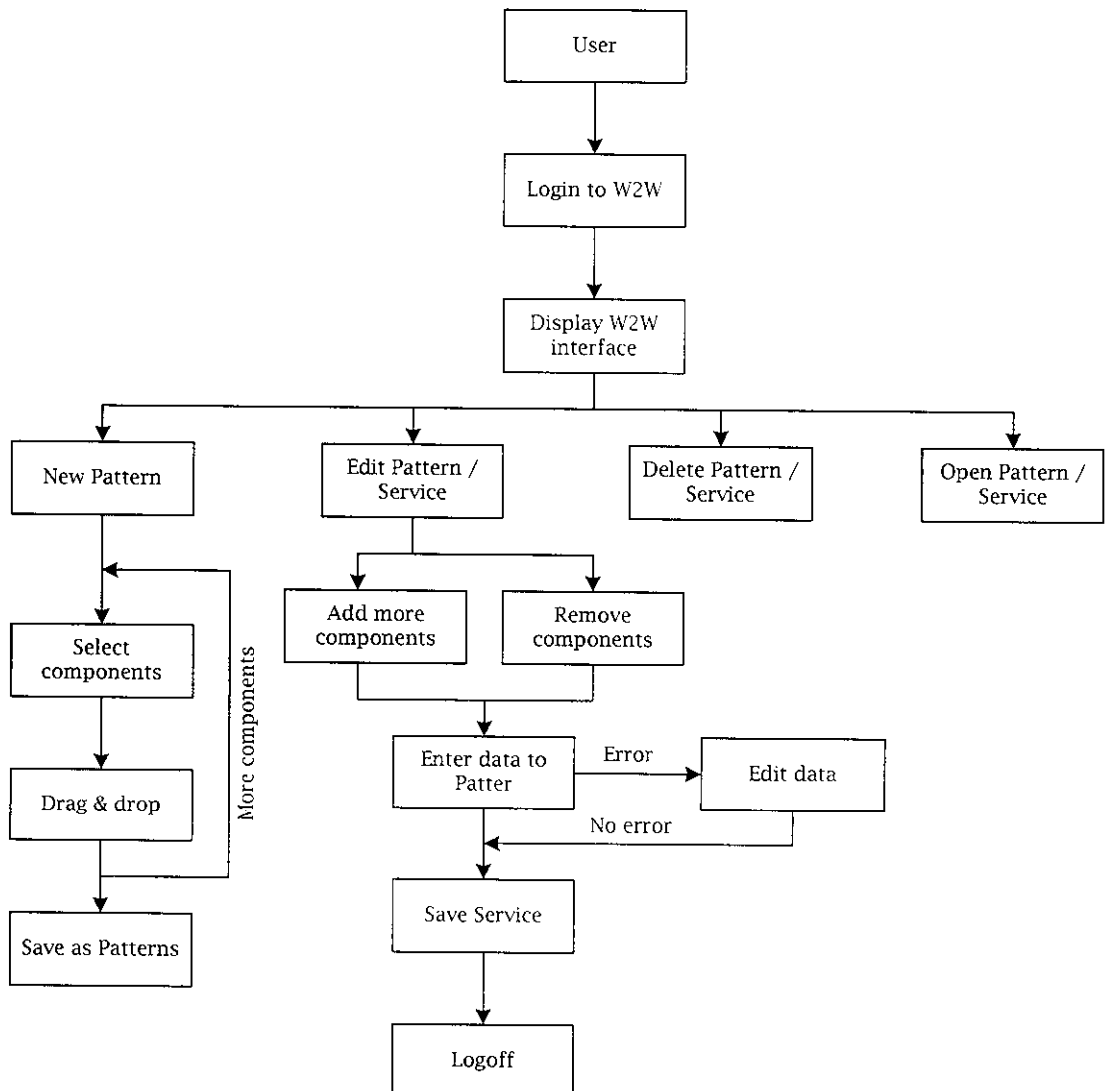


Fig 4.1 Web2Works System flow diagram

4.2 USECASE DIAGRAM

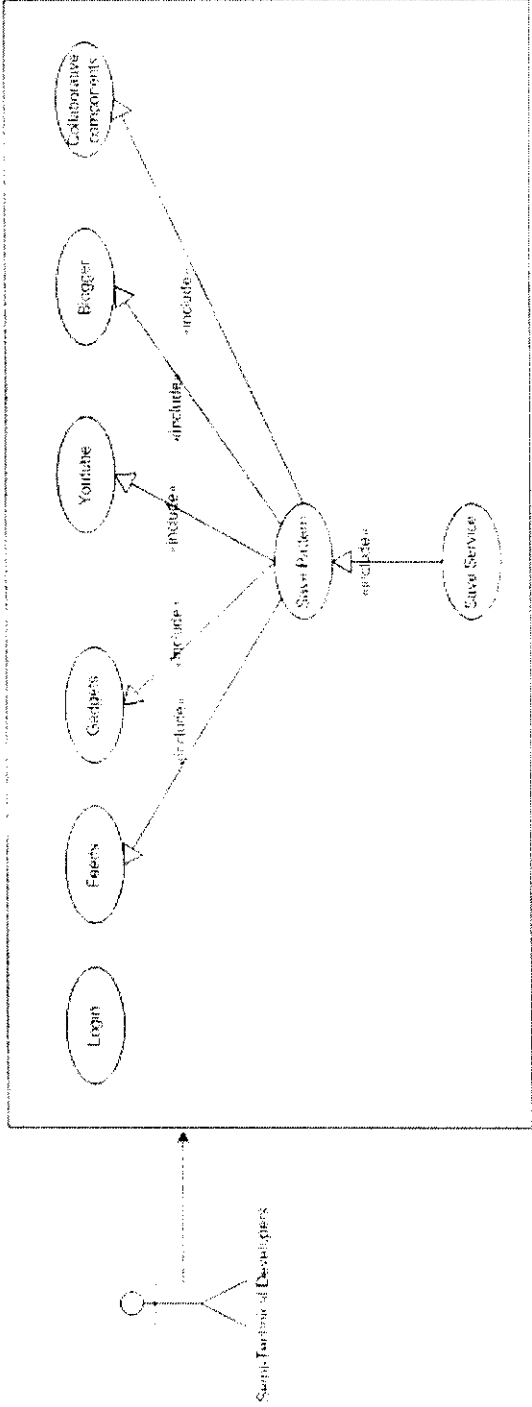


Fig 4.2 Web2Works use case diagram

4.3 SEQUENCE DIAGRAM

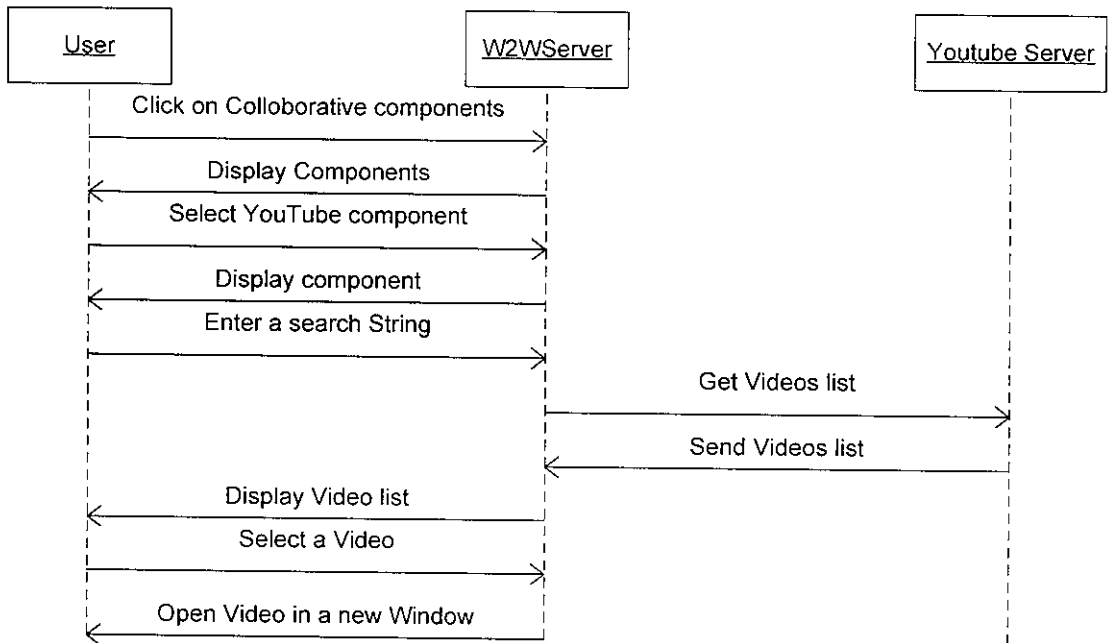


Fig 4.3 YouTube.com sequence diagram

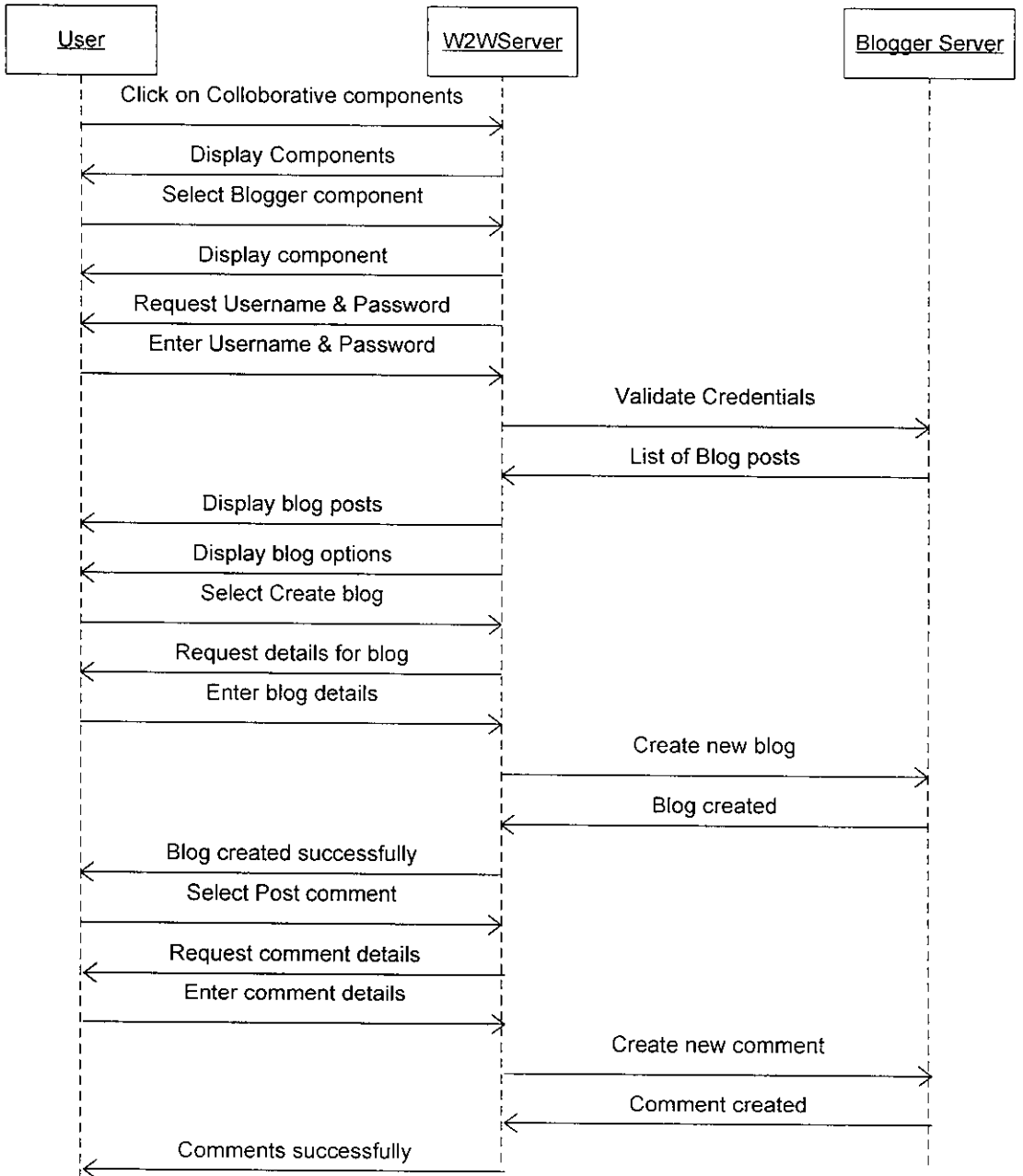


Fig 4.4 BlogSpot.com sequence diagram

4.4 ACTIVITY DIAGRAM

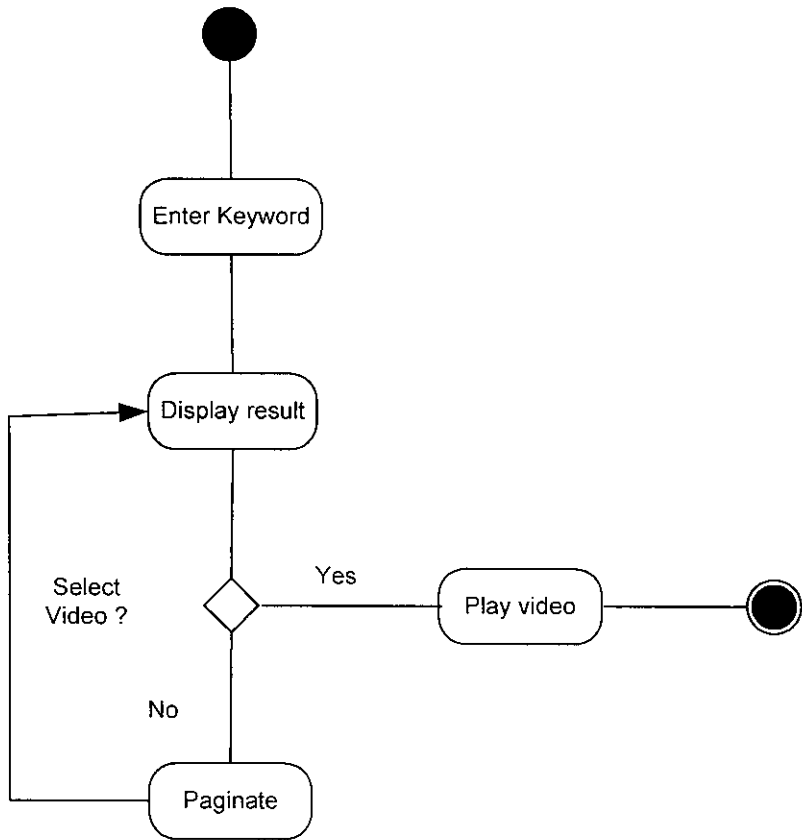


Fig 4.5 YouTube.com activity diagram

4.5 CLASS DIAGRAM

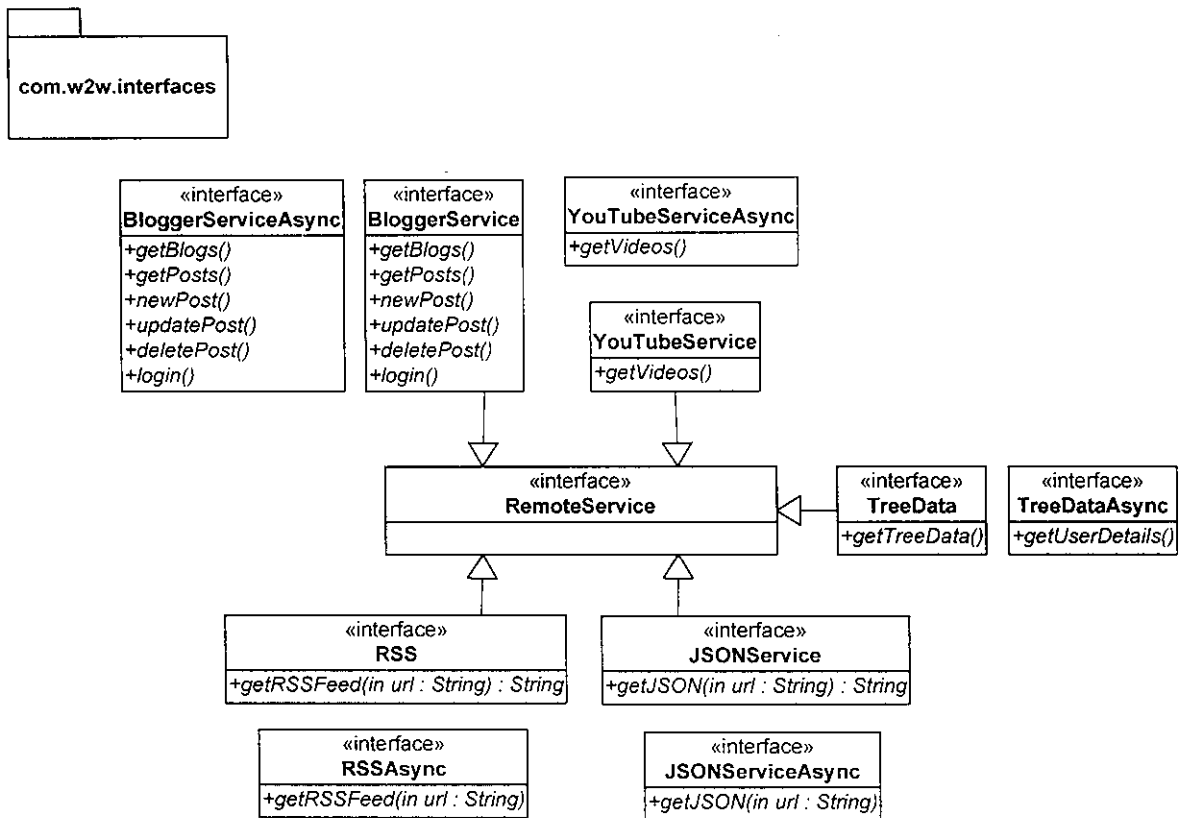


Fig 4.6 Web2Works interface diagram

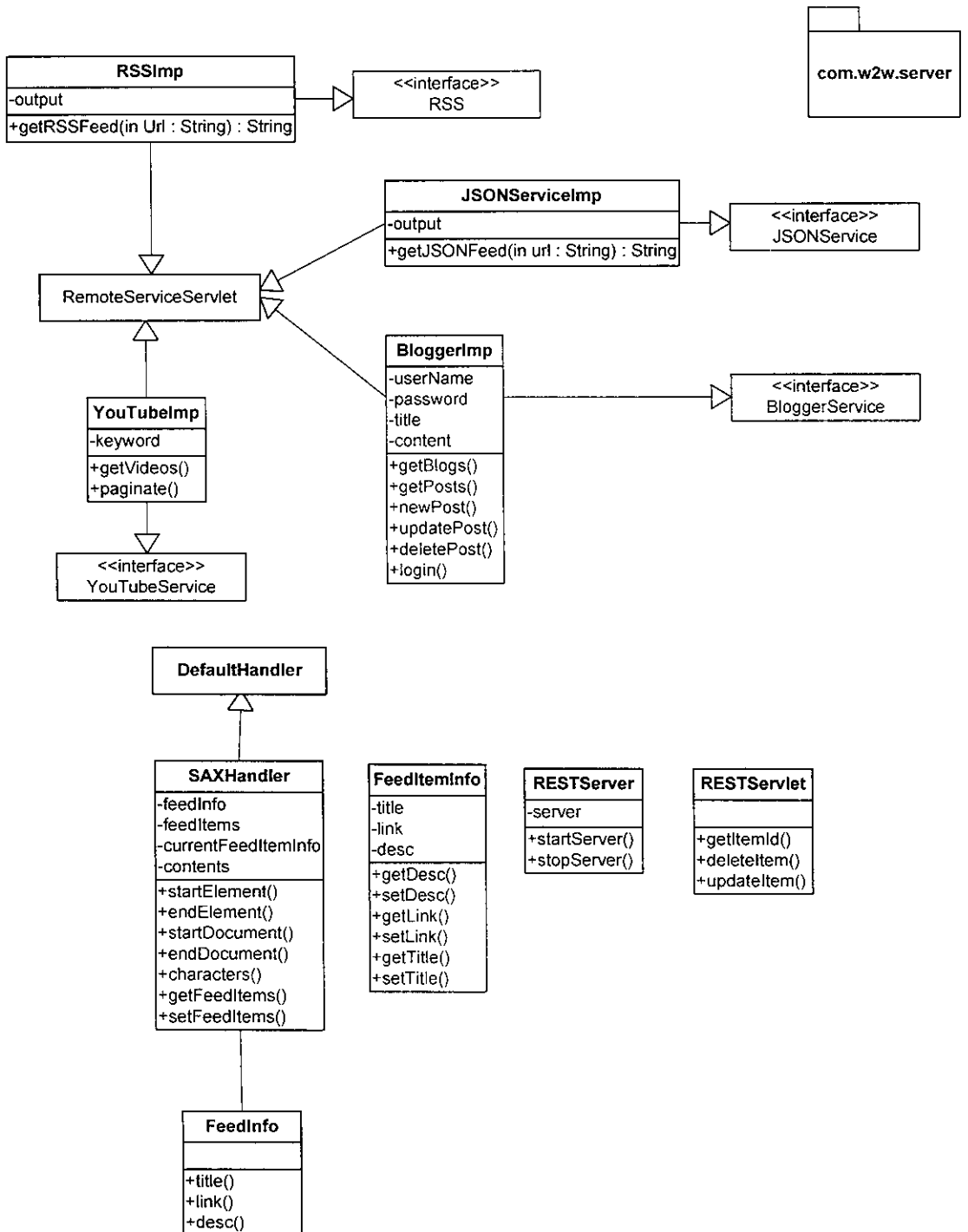


Fig 4.7 Web2Works server side class diagram

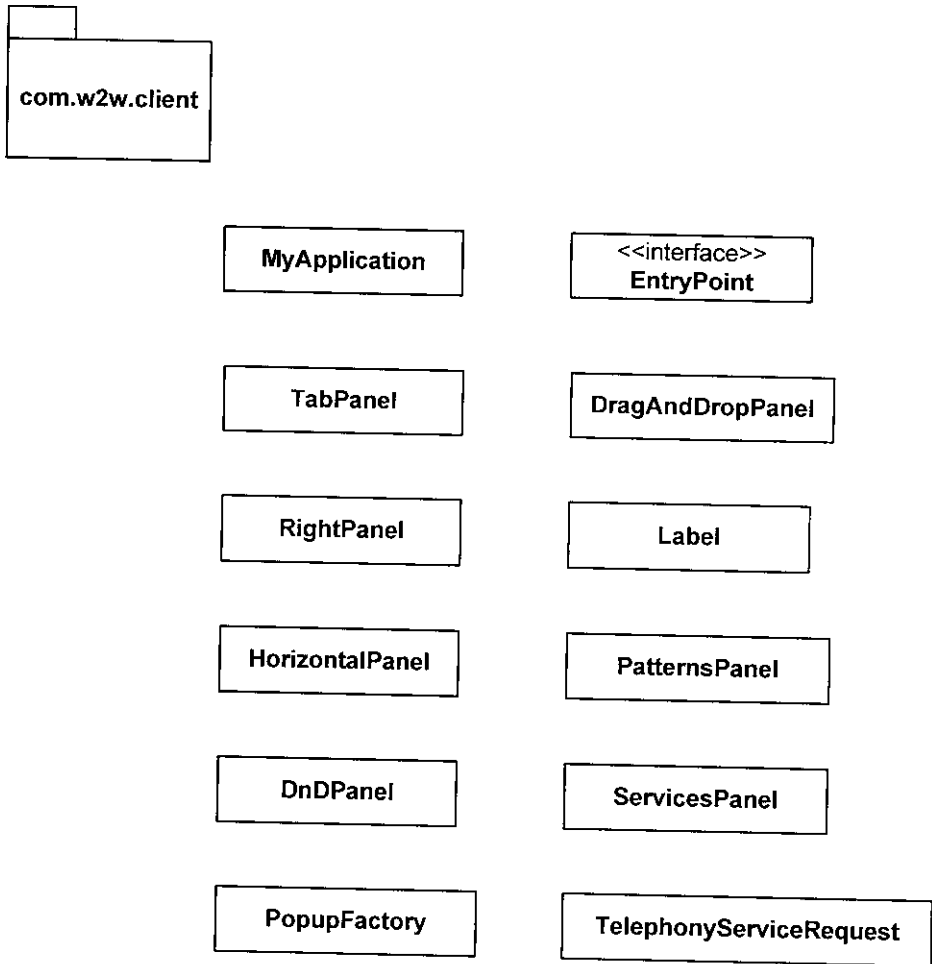


Fig 4.8 Web2Works client side class diagram

4.6 DATABASE DESIGN

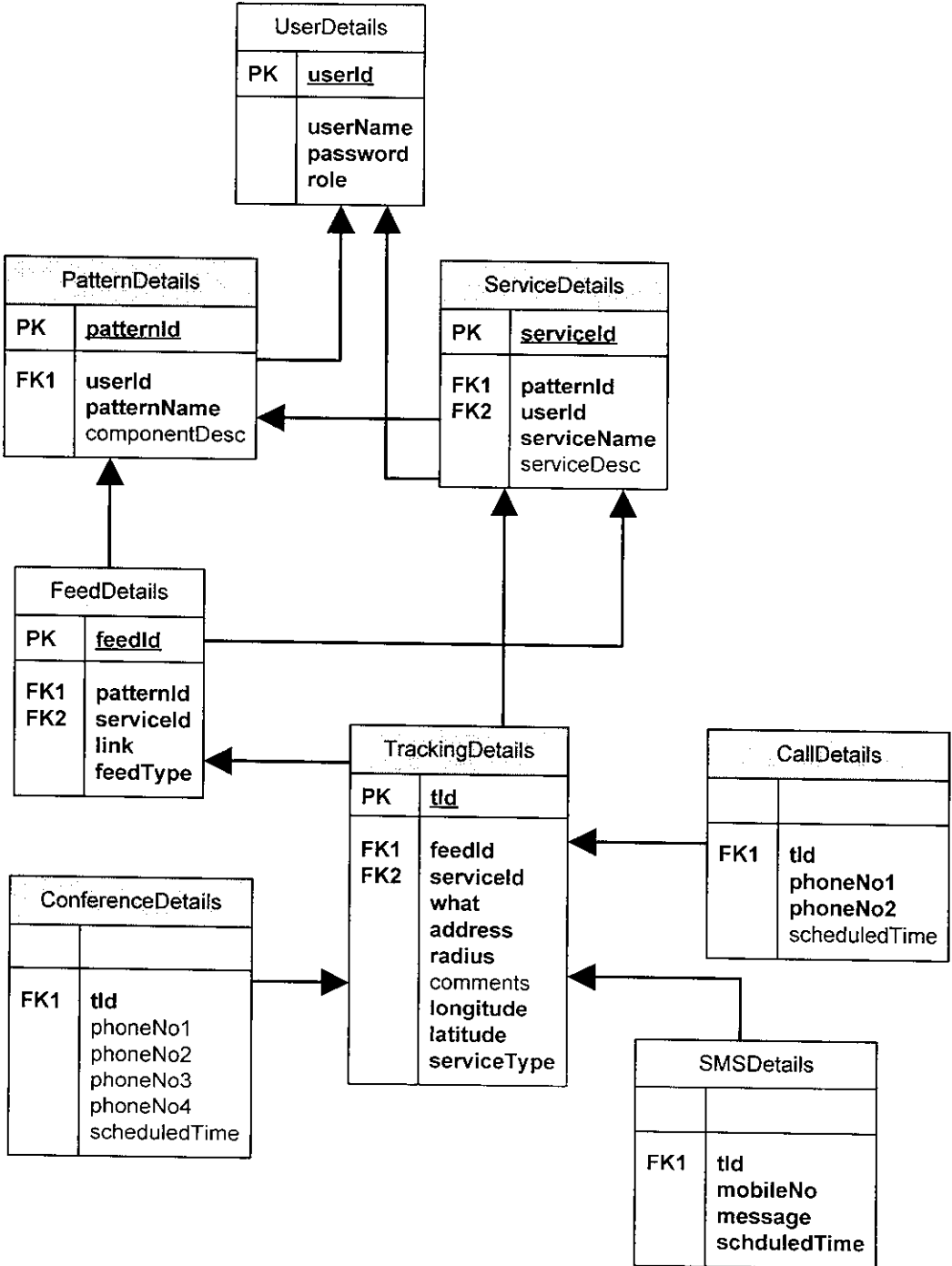


Fig 4.9 Database Relationship diagram

4.6.1 Table Structure

Table 4.1 UserDetails

This table contains registered user details.

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Primary key	Userid	Integer	User ID
	Username	Varchar(25)	User Name
	Password	Varchar(25)	Password
	Role	Char(1)	Role of the user

Table 4.2 PatternDetails

Details about patterns and its components are saved in this table.

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Primary key	Patternid	Integer	Pattern ID
Foreign key	Userid	Integer	User ID
	PatternName	Varchar(25)	Pattern Name
	ComponentDesc	Varchar(1000)	Pattern description

Table 4.3 ServiceDetails

This table Contains information about services.

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Primary key	Serviceid	Integer	Service ID
Foreign key	Patternid	Integer	Pattern ID
Foreign key	Userid	Integer	User ID
	ServiceName	Varchar(25)	Service Name
	ServiceDesc	Varchar(1000)	Service description

Table 4.4 FeedDetails

Details about RSS or JSON Feed are stored in this table

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Primary key	FeedId	Integer	Feed ID
Foreign key	Patternid	Integer	Pattern ID
Foreign key	Serviceid	Integer	Service ID
	Link	Varchar(250)	URL of a feed
	FeedType	Char(1)	Feed type (RSS/JSON)

Table 4.5 TrackingDetails

Details about tracking system

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Primary key	Tid	Integer	Tracking Service ID
	Serviceid	Integer	Service ID
	What	Integer	What to drag?
	Address	Varchar(25)	Address
	Radius	Varchar(10)	Map Information
	Comments	Varchar(100)	Tracking Comments
	Longitude	Varchar(20)	Map Information
	Latitude	Varchar(20)	Map Information
	ServiceType	Char(4)	Type

Table 4.6 CallDetails

Contains details about voice call

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Foreign key	Tid	Integer	Tracking Service ID
	PhoneNo1	Varchar(15)	Callee Phone Number
	PhoneNo2	Varchar(15)	Caller Phone Number
	ScheduledTime	DateTime	Schedule Time

Table 4.7 SMSDetails

Contains registered user details

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Foreign key	Tid	Integer	Tracking Service ID
	PhoneNo1	Varchar(15)	Phone Number
	Message	Varchar(20)	Message Content
	SchTime	DateTime	Schedule Time

Table 4.8 ConferenceDetails

This table has details about conference call service

CONSTRAINTS	FIELD NAME	TYPE	DESCRIPTION
Foreign key	Tid	Integer	Tracking Service ID
	PhoneNo1	Varchar(15)	Phone Number
	PhoneNo2	Varchar(15)	Phone Number
	PhoneNo3	Varchar(15)	Phone Number
	PhoneNo4	Varchar(15)	Phone Number
	ScheduledTime	DateTime	Schedule Time

CHAPTER 5

PROJECT EXECUTION

5.1 MODULE DESCRIPTION

Web2Works has several modules in it. Out of which I have done five modules. Here are the various modules and its description of what it does.

5.1.1 RSS Syndication

Component explorer has a 'RSS Feed' component under feeds tab. User drags and drops the RSS component in the mash up area. Immediately an RSS feeds window appears. User enters the URL of a feed in the provided text box and clicks on Go button. Behind the scene a RPC call is made to Web2Works server and getRSSFeed method is called. This method takes care of sending a request to the remote server, parsing the RSS feed and displaying it in an HTML format.

5.1.2 JSON feeds (Yahoo! Pipes)

JSON feeds are exactly same as RSS feeds. In component explorer there is a component name 'JSON Feed' under feeds tab. User drag and drop in the mash up area and enter a JSON feed URL. Here, Yahoo! Pipes URL. Once the user clicks on Go button an RPC call is made to Web2Works server to getJSONFeed method. Server contacts the Yahoo! Pipes server for data. Then, it parses the data received and converts it into an HTML format and displays it.

5.1.3 BlogSpot

'Blogger' component is available under Collaborative components tab in

BlogSpot.com. The credentials are taken to the server and verified. Second, User gets list of operation that he can perform. They are Blogs, Post Blog, Posts and Comment. When 'Blogs' button is clicked a call is made to getBlogs method to get the list of available blogs. User can create a new post by clicking on 'Post blog' button. Post Title and description are required to make a new post. To make this happen newPost method is used. Next, comes the 'Post' button which calls getPosts method to display the list of posts. User can then edit or delete a post as necessary. To delete a post deletePost method and to edit a post updatePost method is called.

5.1.4 YouTube

Under 'Collaborative Components' tab 'YouTube' component can be found where User drag and drop it and then enters a key word to search for in YouTube.com. Web2Works server makes a call to YouTube server and gets the search result. The result is then paginated and displayed in the window. User can click on a video to watch it without leaving the framework.

5.1.5 RESTful Web Service

Unlike other modules seen so far RESTful web service does not have any component in the component explorer. Server is started by calling startServer method in RESTServer class. It starts RESTful web service server by allocating a dedicated port number. Any request that comes to this port number will be handled by RESTServlet based on the HTTP request header.

5.1.6 Rich GUI

A rich GUI here means creating a drag and drop environment that will run in any browser. This is a challenging work that took many hours of hard work even to create small things. GUI for RSS, JSON, YouTube, Blogger components and few other components too are created.

5.2 MODULE EXECUTION

All the components were successfully integrated with Web2Works framework and it is executed with browsers like Firefox, IE7 and Google Web browser (in hosted mode). One of the applications of this framework is Child tracking for Telecom industry.

CHAPTER 6

PROJECT TESTING

6.1 PURPOSE OF TESTING

Testing is a process of executing the program with the explicit intention of finding errors, which makes the program fail. Hence a successful test is one, which finds an error. Test cases were devised with this purpose in mind. A test case is set of data that a system will process as normal input. Test results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required.

6.2 UNIT TESTING

Error free software can be accomplished only if it satisfies the smallest part. That is error free module. When developing the module as well as finishing the development we test to make sure that each module works without any error. The inputs are validated when accepting from the user. In this project all the modules have been tested individually and are given with test cases in predefined circumstances.

6.3 INTEGRATION TESTING

This testing is being done at design level i.e. in order to test the module interaction and behavior. So the developed software should behave as per written. Here GWT, Blogger, YouTube, REST and Feeds are integrated together successfully and found that they behave as expected. Behavior of all the modules continues to be fine even after the integration.

6.4 SYSTEM TESTING

At this point our Web2Works framework is become a proven technology that works for developing applications. This validation testing is done as per the idea and concepts and we came to know that all the behaviors are correct except the user interface. It should be more flexible and reliable as we expected. So other technologies like Flex and Adobe are being included to improve the things.

6.5 ACCEPTANCE TESTING

User acceptance testing is the key factor for the success of any system, so the developed framework is demonstrated for the three clients and their response was good. Since this framework is done under R&D, there are no direct clients involved in it. So we have sought for clients.

6.6 TEST CASES

Test cases show the list of test conducted along with the expected output, actual output and test status.

Table 6.1 Test cases

Test ID	Test condition	Expected output	Actual output	Remedy	Result
W2WT1	Use of empty values in required fields	Warning states that fields cannot be empty	Same as expected output	No	Pass
W2WT2	Use of Wrong data types	Should Display error:"Invalid data entered"	Same as expected output	No	Pass
W2WT3	Use of illegal user name, password	Should display error:"No such user exist"	Same as expected Output	No	Pass
W2WT4	Use of illegal characters	Should display error:"Special characters not accepted"	Same as expected output	No	Pass
W2WT5	Entering wrong URL	Should display error:"Invalid URL. Please check the URL"	Same as expected output	No	Pass
W2WT6	Invalid username or password for blogspot.com	Should display error:"Invalid credentials"	Same as expected output	No	Pass

CHAPTER 7

PROJECT ROLLOUT

7.1 STATUS

This is a Wipro's framework for Web 2.0 next generation web applications. This framework core parts are developed and released as a first release. There are few modifications are being done to look and feel, face book. That will be finished shortly and made available to clients.

7.2 FUTURE ENHANCEMENTS

In future, the core concept remains same. Along with some existing feature there are proposal to add more features like location based services in India [right now available in US and UK]. Then the look and feel of the web 2 works will be changed with the help of Adobe Flex rich user experience.

CHAPTER 8

APPENDICES

APPENDIX 1 : SAMPLE SCREENSHOTS

Description

Web2Works can be used only by authorized persons. This page requests the user to enter their credentials and check for the same whether it is valid or not.

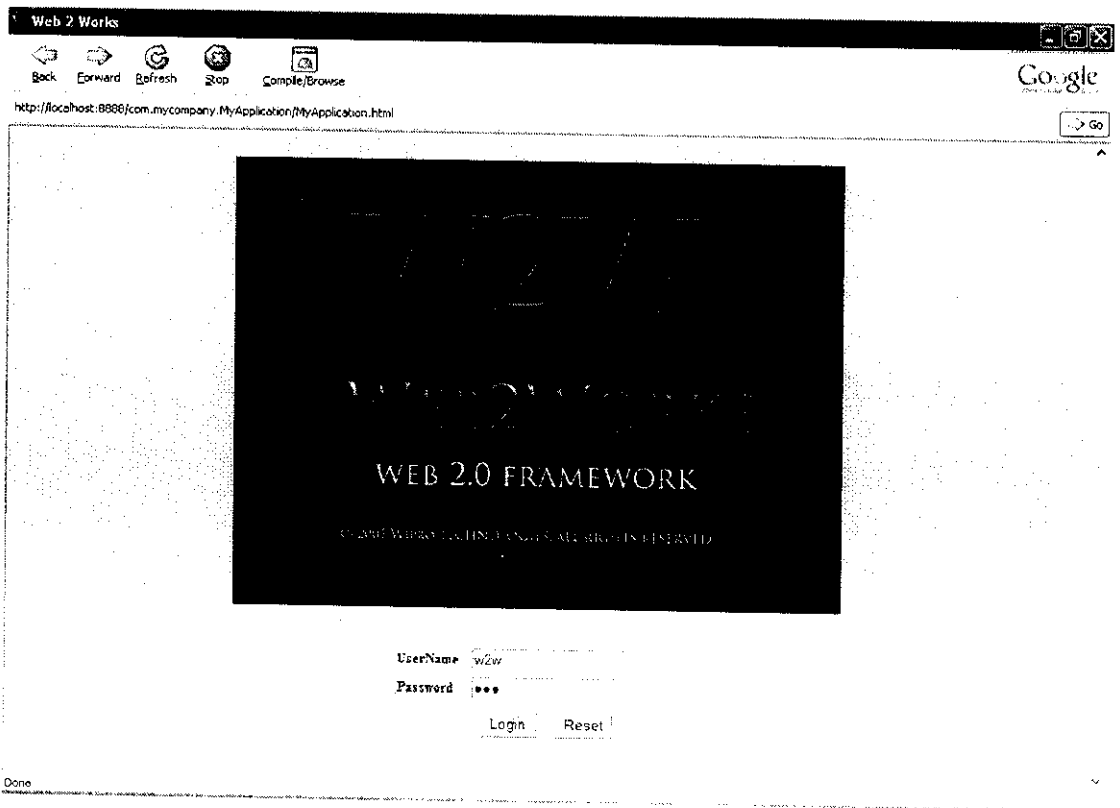


Fig. A8.1 Login Page

Description

This screen shows the content of RSS feed in cnn.com

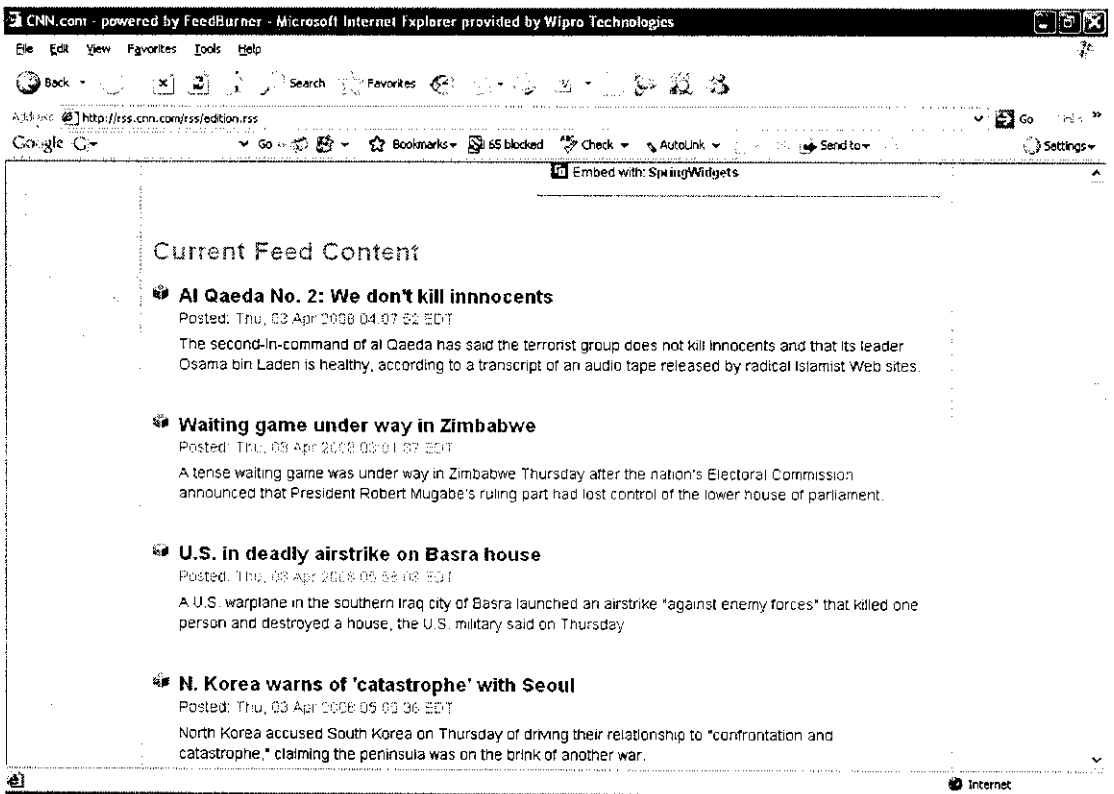


Fig. A8.2 CNN RSS Feed Content

Description

This screen shows parsed content of cnn.com RSS feed

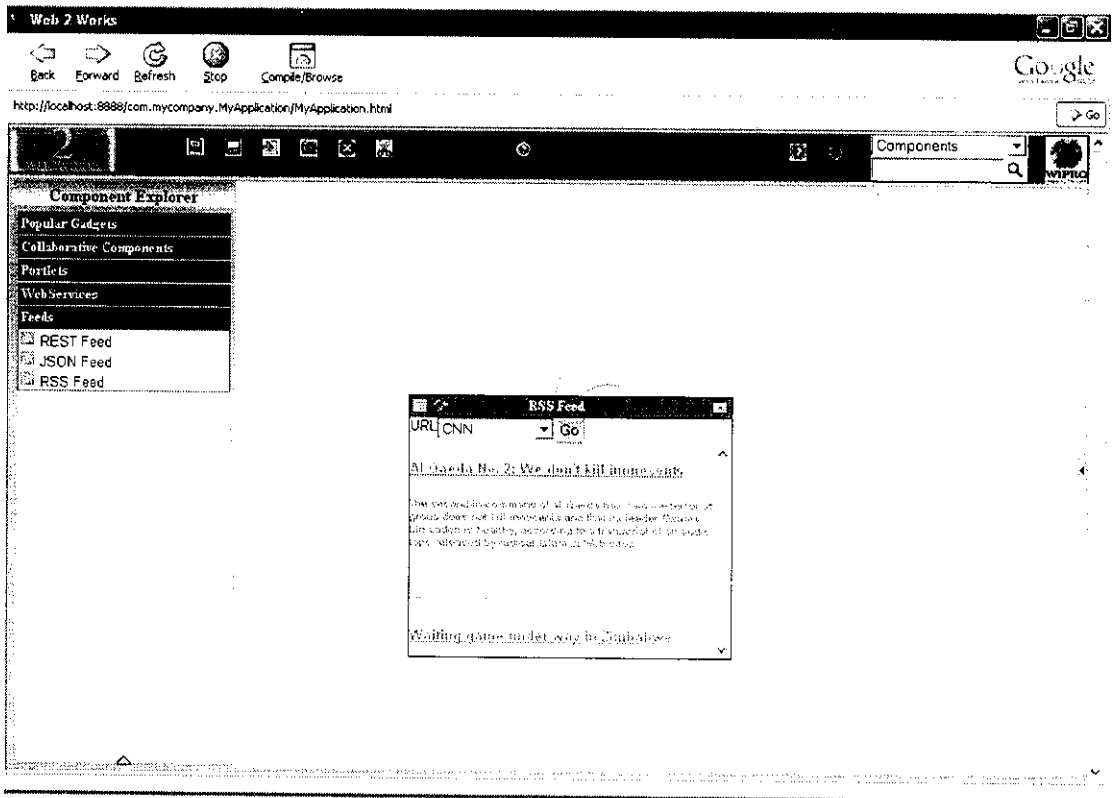


Fig. A8.3 Cnn.com RSS feed parsed

Description

This screen shows Yahoo! Pipes JSON feed

The screenshot shows a Mozilla Firefox browser window with the address bar displaying the URL: `http://pipes.yahoo.com/pipes/pipes.info?id=6B3W9E_Z36GQ55t1vC63w`. The page title is "Pipes: PCM-COE - Mozilla Firefox". The main content area displays the pipe name "PCM-COE" and a list of news items:

- 'Al-Qaida doesn't kill innocents'**
Al-Qaida's second-in-command Ayman al-Zawahiri described the United Nations as an enemy to Muslims and vowed attacks on Jews both inside and outside Israel.
- 'US image improves, but India ahead'**
US had its lowest positive ratings from Mexico, Argentina, Egypt, India and Russia, where fewer than 50% of those questioned in a poll had a good impression of the US.
- 200 Indians struggle to recover dues**
Indian workers, not been paid salaries for the last 9 months, are fighting an uphill battle to recover their dues from a Danish firm.
- 44 held for blocking bus in Karnataka**
The blockade was in retaliation to the protests by pro-Kannada groups in Karnataka opposing the Hoopeshadi Integrated Water Project.
- North Chabris out of news**
Address Aarti Chabris away from her seat of rumours and her flight as well.

The browser's status bar at the bottom shows "Done" and a page number "5".

Fig. A8.4 Yahoo! Pipes JSON Feed

Description

This screen shows the parsed JSON feed from Yahoo! Pipes

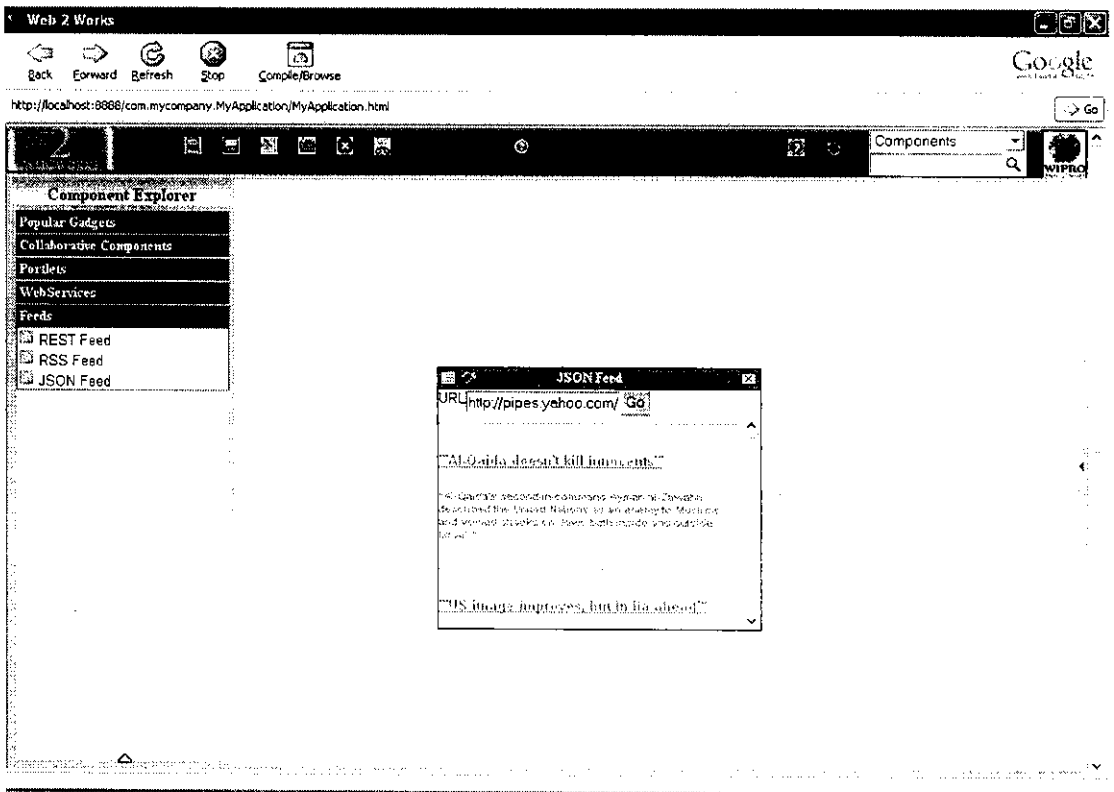


Fig. A8.5 Yahoo! Pipes JSON feed parsed

Description

YouTube search result is displayed in Web2Works

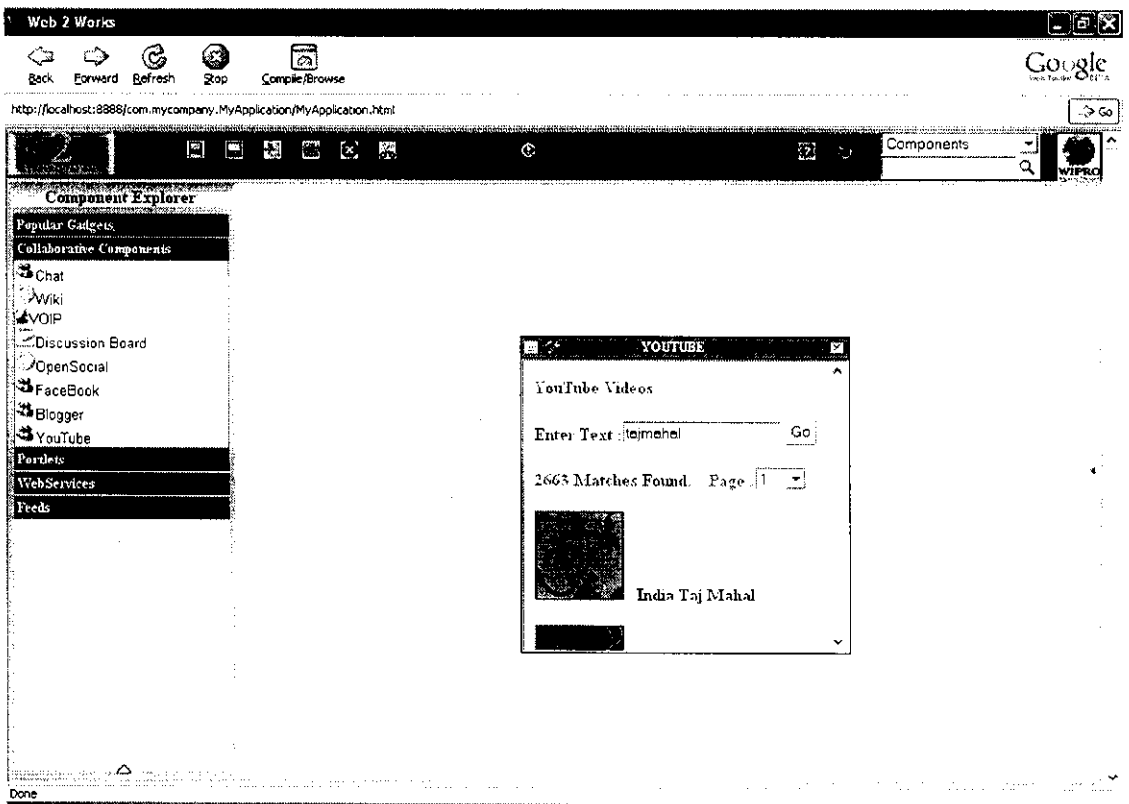


Fig. A8.6 YouTube Search result

Description

Blogger window displays login page for blogspot.com in Web2Works

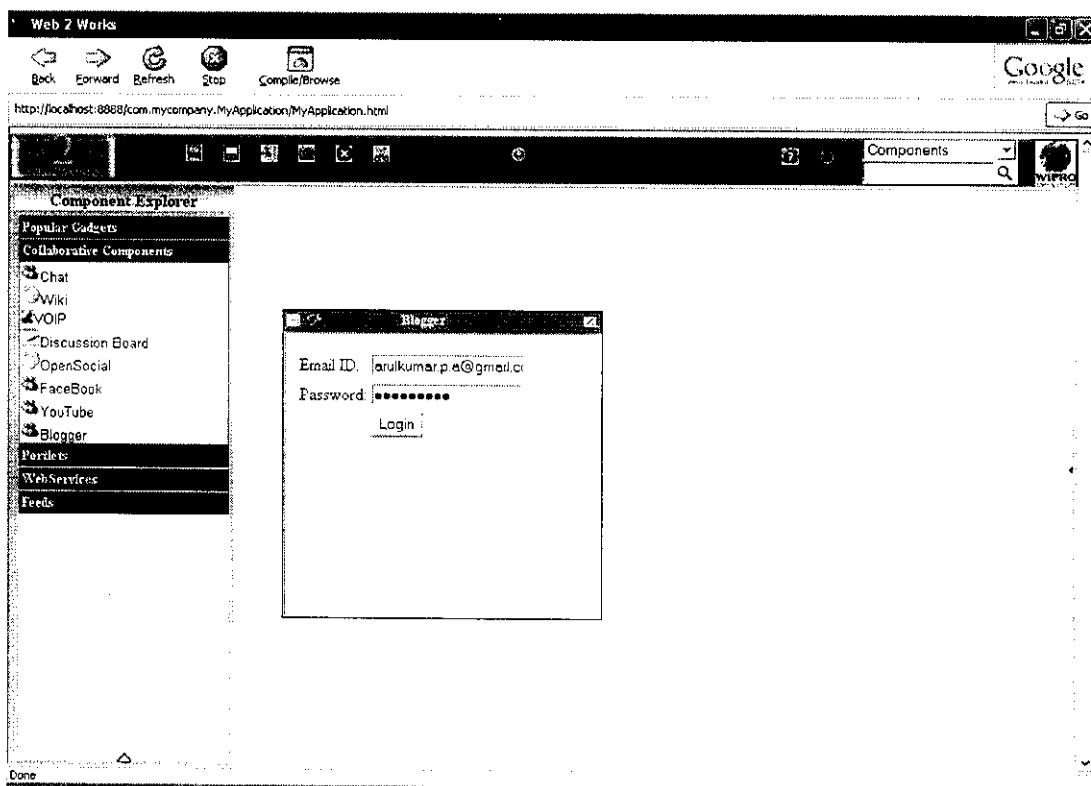


Fig. A8.7 Blogger Login Page for blogspot.com

Description

This screen shows that a new post is published in a blog.

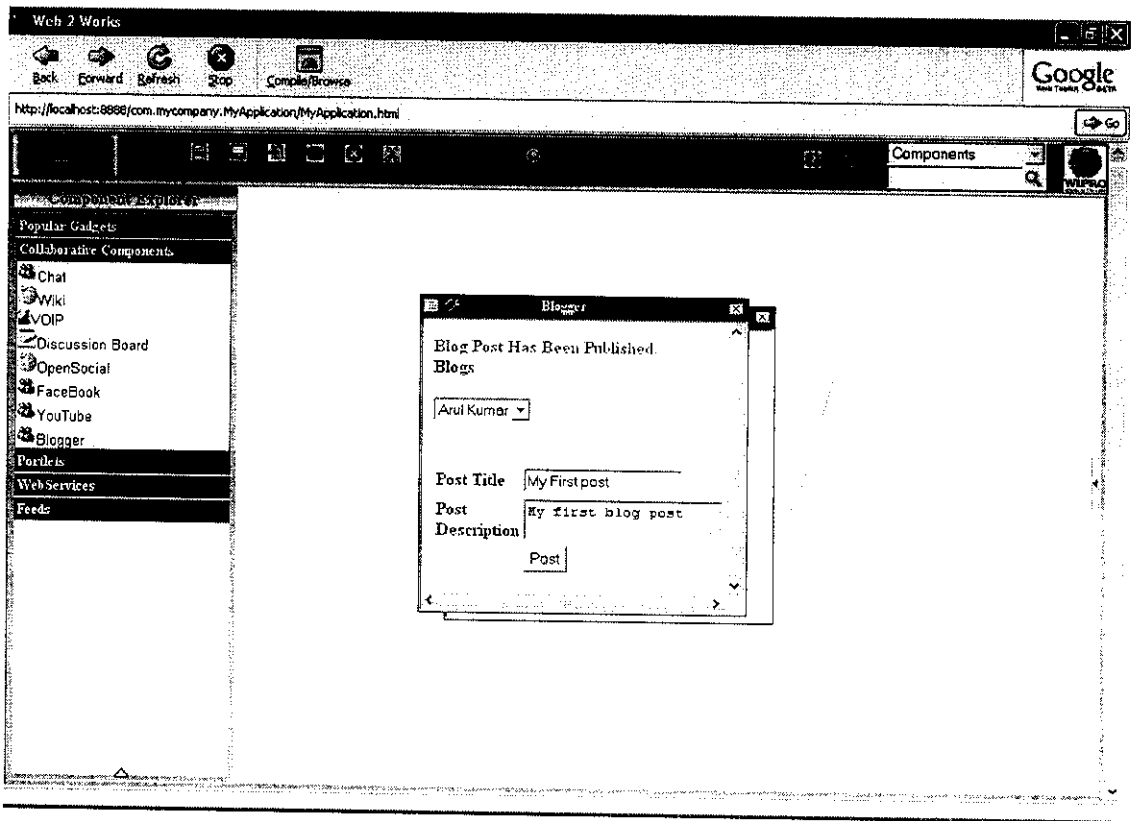


Fig. A8.8 Publishing a new post

Description

List of posts that are available in the blog are displayed.

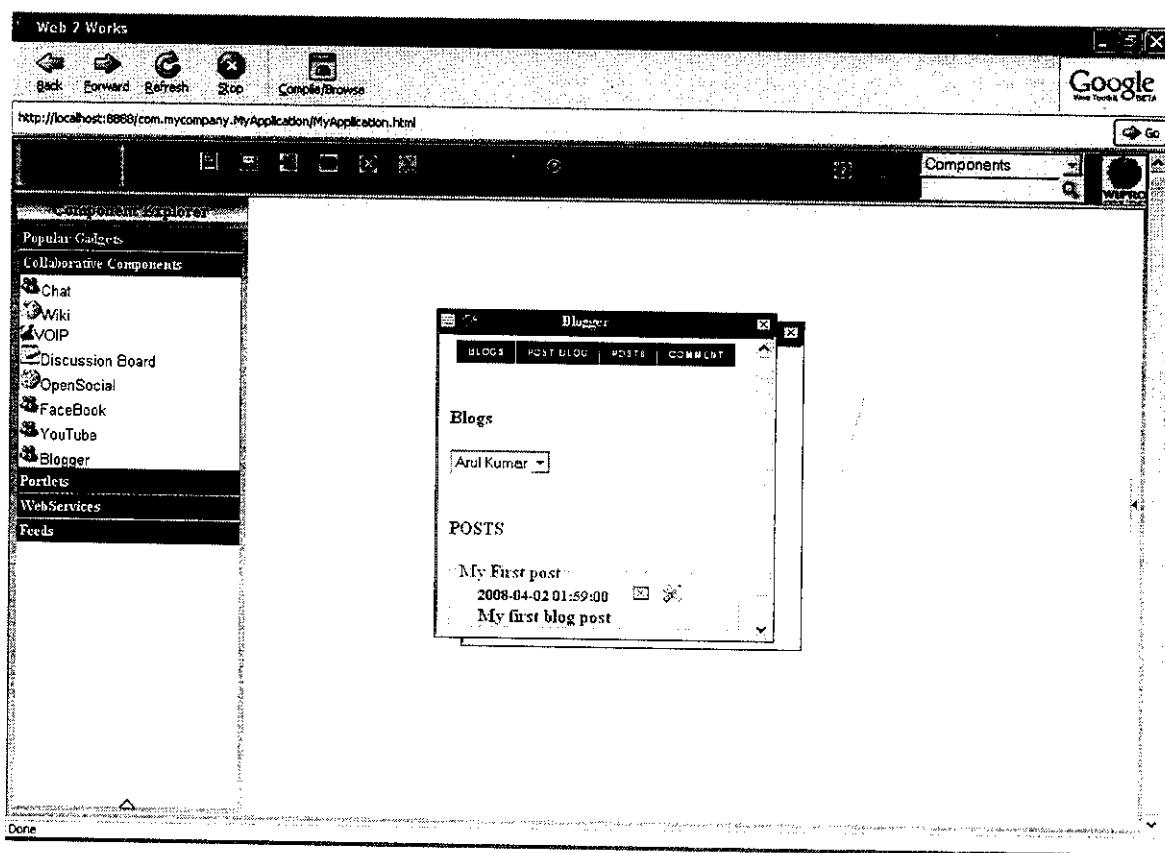


Fig. A8.9 List of existing posts

Description

Posts in arulkumarpa.blogspot.com after publish a new post.

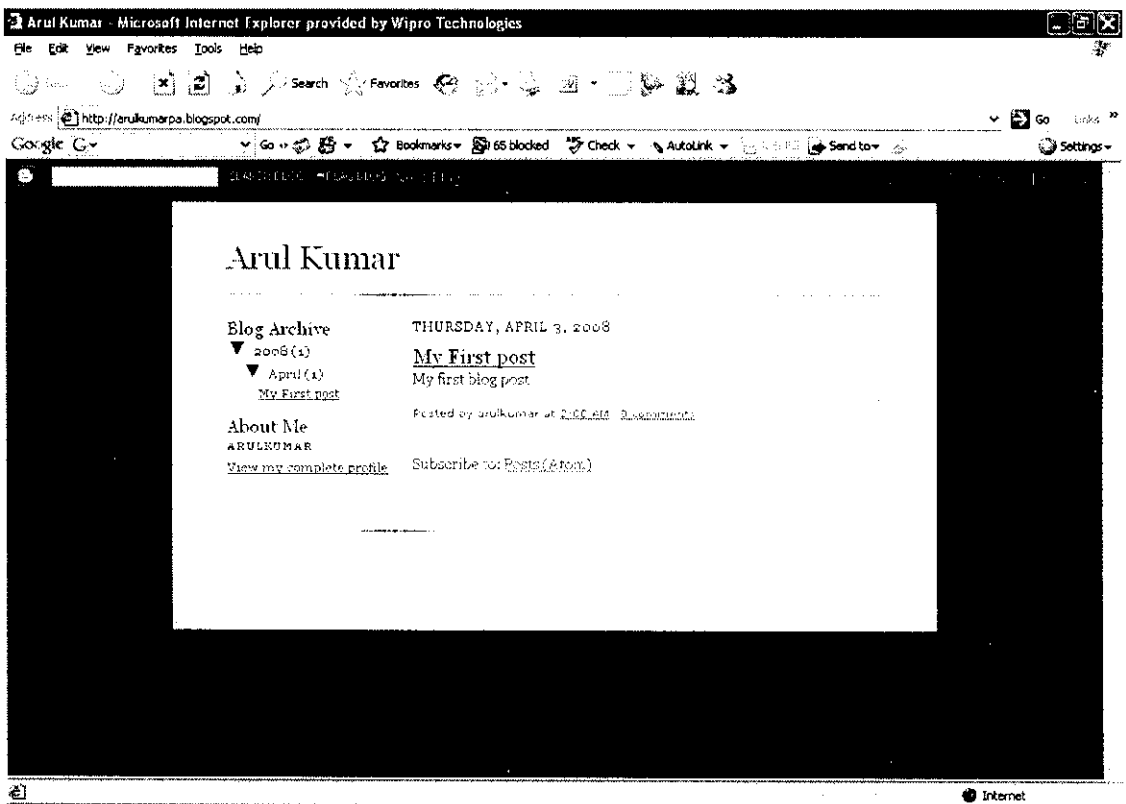


Fig. A8.10 Posts in blogspot.com after publishing a post

Description

Existing post is edited by clicking on an edit button. Web2Works It asks for a new title.

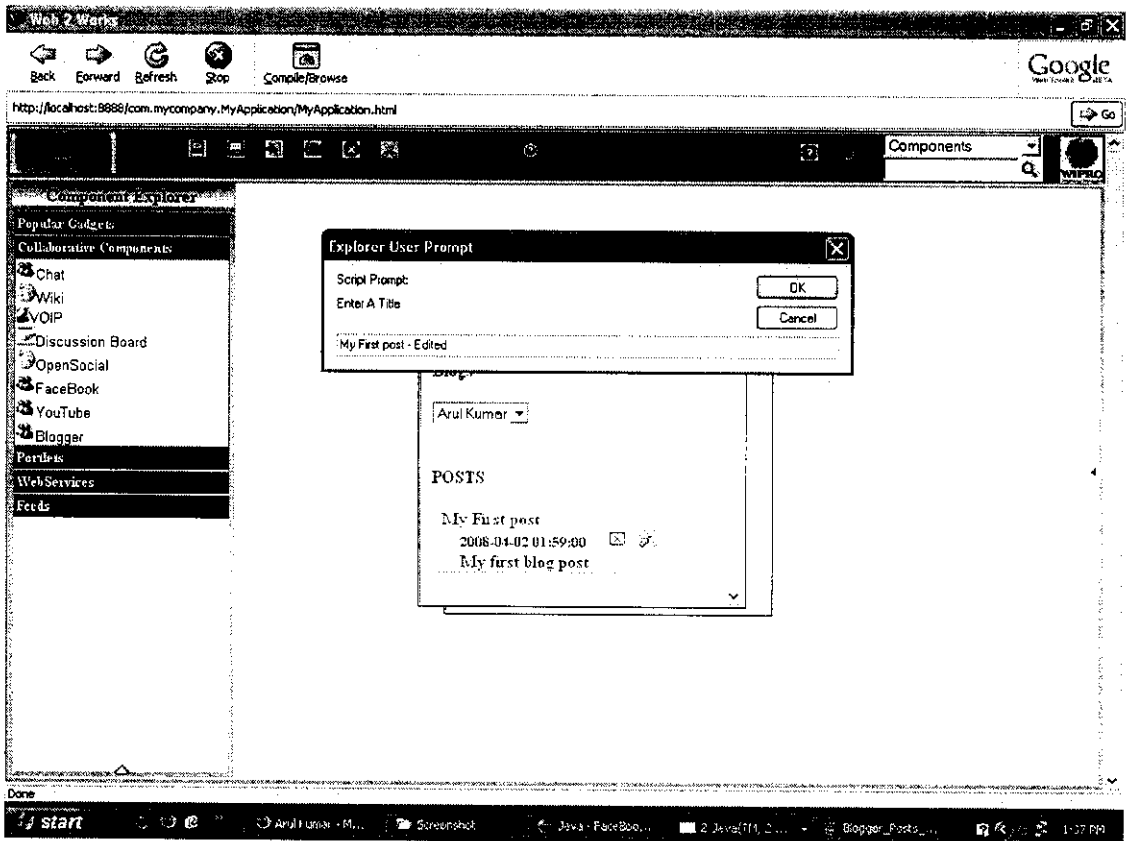


Fig. A8.11 updating an existing post title

Description

Web2Works asks for a new content to update.

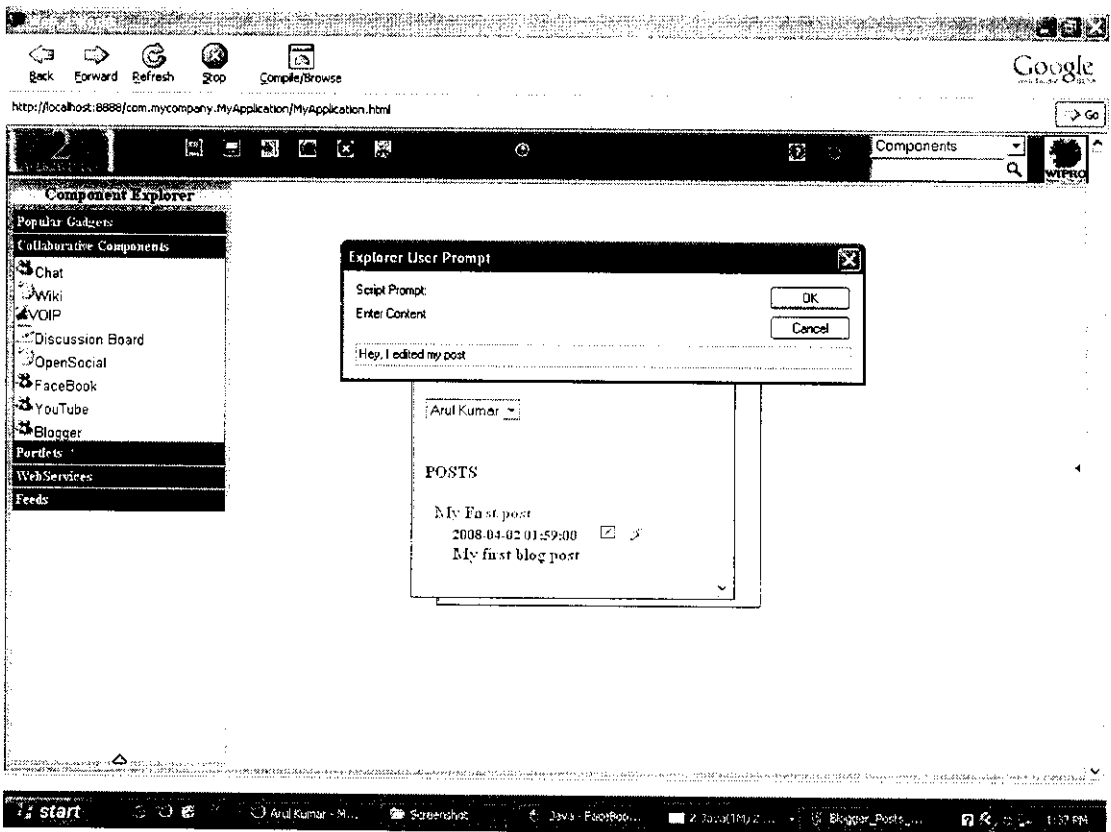


Fig. A8.12 updating existing post content

Description

View of my blog after updating

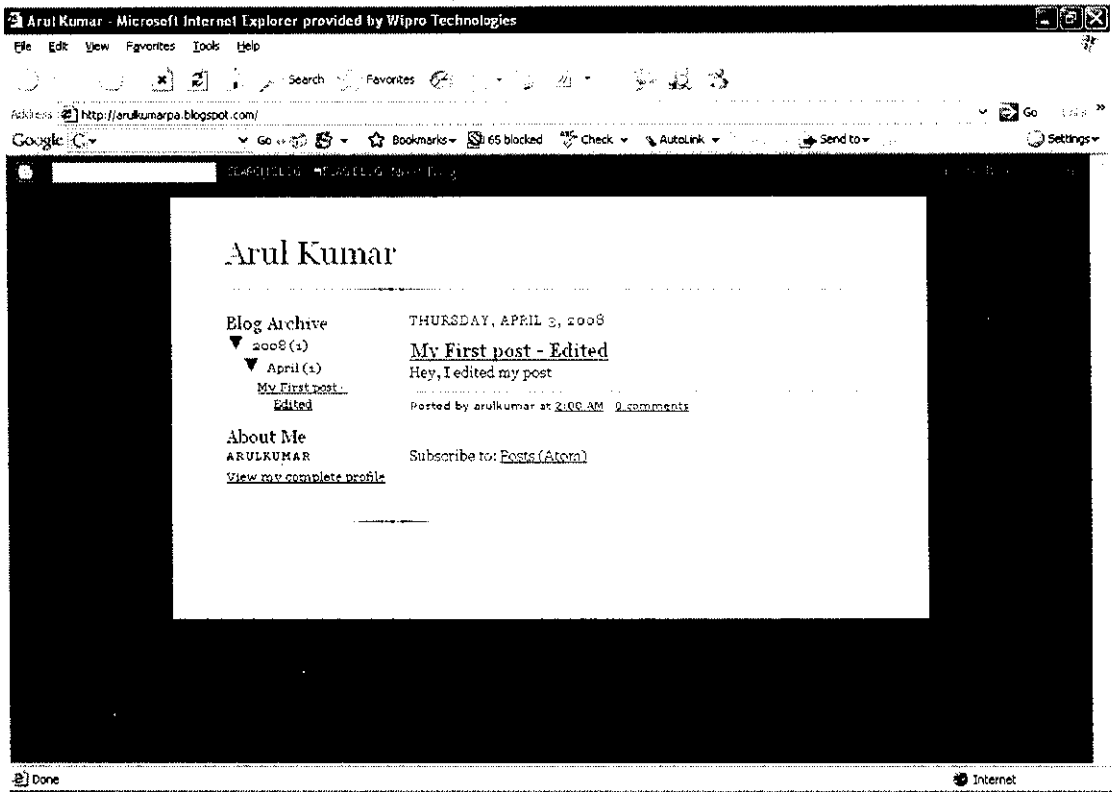


Fig. A8.13 My blog after update

Description

Deleting an existing post from Web2Works

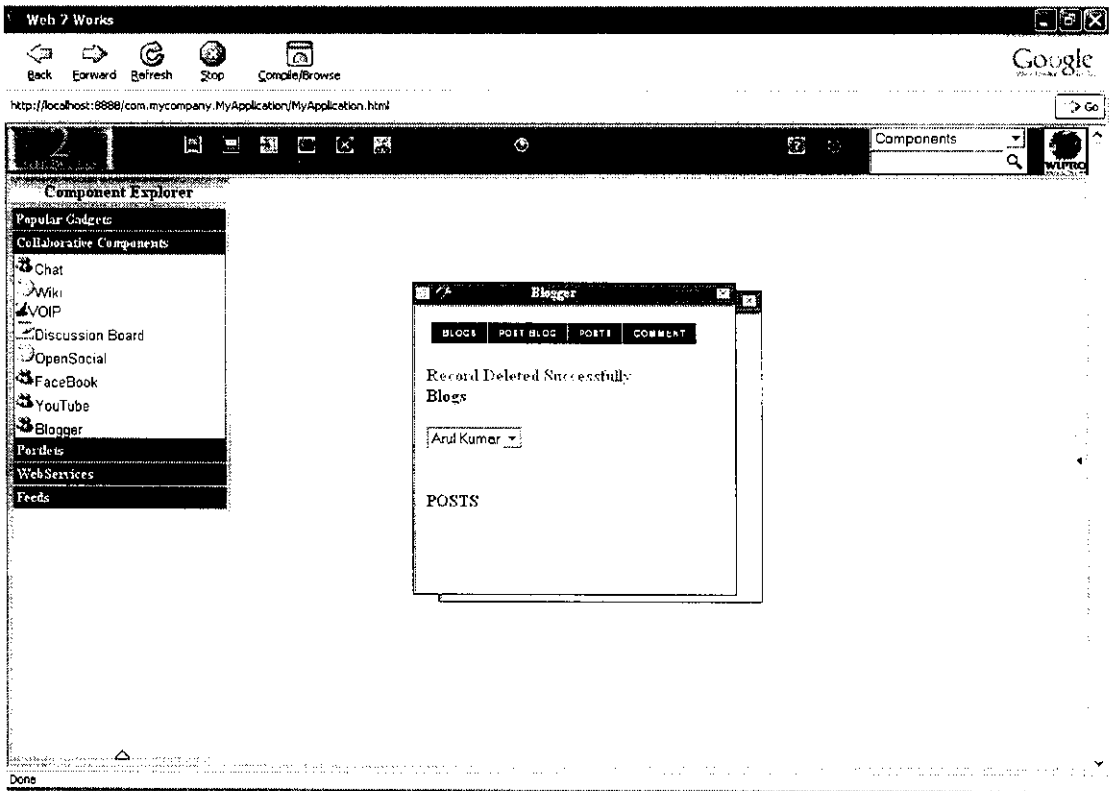


Fig. A8.14 Delete a post

APPENDIX 2 : ABBREVIATIONS

AJAX	Asynchronous Java script and XML
API	Application Programming Interface
BPO	Business Process outsourcing
CMM	Capability Maturity Model
CoE	Centre of Excellence
CSS	Cascading Style Sheet
EA	Early Adopters
GWT	Google Web Toolkit
IDE	Integrated Development Environment
IEEE	Institute of Electrics & Electronics Engineering
J2EE	Java 2 Enterprise Edition
JRE	Java Runtime
JSON	JavaScript Object Notation
PCM	Portal and Content Management
PCMM	People Capability Maturity Model
POC	Proof of Concepts
POT	Proof of Technologies
R&D	Research and development
REA	Eich Enterprise Applications
REST	Representational State Transfer
RIA	Rich Internet Applications
RPC	Remote Procedure Call
RSS	Really Simple Syndication
SaaS	Software-as-a-Service
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
VOIP	Voice Over Internet Protocol
W2W	Web 2 Works
WSAL	Web Service Application Language
WWW	World Wide Web
XHTML	Extensible Markup Language

CHAPTER 9

REFERENCES

1. Leonard Richardson and Samruby, "RESTful Web Services", O'reilly
2. Dan Pilone, "UML 2.0 in a nutshell", O'reilly
3. Joshua Eichorn, "Understanding AJAX using Java script to create rich internet applications"
4. <http://www.w3Schools.com>
5. <http://code.google.com/googlewebtoolkit/>
6. <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/>
7. <http://www.mysql.com/>
8. <http://www.rssboard.org/rss-specification>
9. <http://www.json.org/>
10. http://en.wikipedia.org/wiki/Representational_State_Transfer
11. <http://www.xfront.com/REST-Web-Services.html>
12. <http://code.google.com/apis/youtube/>
13. <http://code.blogger.com/>
14. <https://jersey.dev.java.net/servlets/ProjectDocumentList>