INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Nowadays, the web is playing a significant role in delivering information to user's fingertips. A number of techniques and tools like bookmarks, history tools, search engines, and contextual recall systems have been developed to support personal web revisitation. Among the common web behaviours, web revisitation is to re-find the previously viewed web pages, not only the page url, but also the page snapshot at the access timestamp. Getting back to previously viewed web pages is an uneasy task for users due to the large volume of personally accessed information on the web. Bookmarks constitute the popular re-finding support, users can search resource in web according to their preference. Several websites are displayed where user can bookmark their interested links.

Collaborative tagging is used to support tag resources on the web according to their personal preference. In collaborative tagging it increases the privacy threat because of the public availability of user generated data (tag). Tag suppression is a technology used for preventing privacy attackers from users' profile interest. While tagging user can suggest the tag name or can request the server to suggest tag name. We proposed resource recommendation and parental control. Resource recommendation provides relevant resource based on user interest. Parental control is used to block an unwanted general or specific category to the user in the group. This can be accessed only by a group owner. Facet block is the one which shows users, the categories available in the group.

1.2 SCOPE OF THE PROJECT

The scope of the project is collaborative tagging, to loosely classify resources based on end-user's feedback, expressed in the form of free-text labels (i.e., tags). Although collaborative tagging is mainly used to support tag-based resource discovery and browsing, it could also be exploited for other purposes. As an example, the tags collected by social bookmarking services can be exploited to enforce enhanced web access functionalities, like content

filtering and discovery, based on preferences specified by the end user. However, to achieve this enhanced use, the current architecture of collaborative tagging services must be extended by including a policy layer. The aim of this layer will be to enforce user preferences, intentionally denoting resources on the basis of the set of tags associated with them, and, possibly, other parameters concerning their trustworthiness (the percentage of users who have added a given tag, the social relationships and characteristics of those users, etc.).

1.3 OBJECTIVE OF PROJECT

The main aim of this project is to classify resources, protect user privacy while tagging and searching resources. Content filtering is a method used for parental control and facet block and it is based on preferences specified by end users.

1.4 EXISTING SYSTEM

The collection of end-users' private information stored by social services, is now recognized as a privacy threat it is worth noting that the public availability of user-generated data (as tags are) could be used to extract an accurate snapshot of users' interests or user profiles, containing sensitive information, such as health-related information, political preferences, salary or religion. Actually, the huge number of users using collaborative tagging services, and the fact that collaborative tagging is a service supported virtually by any social online application, increases the risk of cross referencing, thereby seriously compromising user privacy. Indeed, it could be possible to correlate the account of a user with other accounts they may have at different services, which would imply gaining far more precise information about the user profile.

1.5 PROPOSED SYSTEM

Proposed System protects user privacy to a certain extent, by dropping those tags that make a user profile show bias toward certain categories of interest. Tag suppression is a technique that has the purpose of preventing privacy attackers from profiling users' interests on the basis of the tags they specify. Tag perturbation consists of specific user tags with general tag categories.

Proposed system addresses two scenarios: resource recommendation and Parental control. In Resource recommendation, provides relevant resources based on user interest. Parental control concerns whenever a group user requests resource, group owner give privilege to access resources. We also provides Facet block for particular user.

LITERATURE SURVEY

CHAPTER 2
LITERATURE SURVEY

The purpose of the literature survey is to give the brief overview and also to establish complete information about the reference papers. The goal of literature survey is to completely specify the technical details related to the main project in a concise and unambiguous manner.

[2.1] Integrating Back, History and Bookmarks in Web Browsers

AUTHOR : Shaun Kaasten and Saul Greenberg, In HCL, pages vol.70, no.9 2001

DESCRIPTION :

Most Web browsers include Back, History and Bookmark facilities that simplify how people return to previously seen pages. While useful, these three facilities all operate on quite different underlying models, which undermines their usability. Our alternative revisitation system uses a single model of a recency-ordered history list to integrate Back, History and Bookmarks. Enhancements include: Back as a way to step through this list; implicit and explicit 'dog-ears' to mark pages on the list (replacing Bookmarks); searching or filtering the list through dynamic queries; and visual thumbnails to promote page recognition.

[2.2] The Impact of Bookmarks and Annotations on Refinding Information

AUTHOR : Ricardo Kawase, In CIDR, pages vol.28,no.2 ,pp.13-26,2003

DESCRIPTION:

Refinding information has been interwoven with web activity since its early beginning. Even though all common web browsers were equipped with a history list and bookmarks early enough to facilitate this need, most users typically use

search engines to refind information. However, both bookmarks and search based tools have significant limitations that impact their usability: the former are known to be hard to manage over the course of time, whereas the latter require the user to recall a specific combination of keywords or context. Most importantly, though, both are particularly inappropriate in cases where a piece of information is contained within an unstructured web page. In this paper, we present in-context annotation as a more efficient alternative to these methodologies. To verify this claim, we conducted a study in which we compare the performance of experienced users in all three approaches while revisiting specific pieces of information in the web after a long period of time. The outcomes suggest that in context annotation clearly outperforms both traditional strategies.

[2.3] Beyond the Usual Suspects: Context-Aware Revisitation Support

AUTHOR : Ricardo Kawase,George Papadakis, Eelco Herder and Wolfgang Nejdl, In HT,pages vol.10,no.2,pp. 27-36,2011

DESCRIPTION:

A considerable amount of our activities on the Web involves revisits to pages or sites. Reasons for revisiting include active monitoring of content, verification of information, regular use of online services, and reoccurring tasks. Browsers support for revisitation is mainly focused on frequently and recently visited pages. In this paper we present a dynamic browser toolbar that provides recommendations beyond these usual suspects, balancing diversity and relevance. The recommendation method used is a combination of ranking and propagation methods. Experimental outcomes show that this algorithm performs significantly better than the baseline method. Further experiments address the question whether it is more appropriate to recommend specific pages or rather (portal pages of) Web sites. We conducted two user studies with a dynamic

toolbar that relies on our recommendation algorithm. In this context, the outcomes confirm that users appreciate and use the contextual recommendations provided by the toolbar.

[2.4] Revisitation Patterns in World Wide Web Navigation

AUTHOR : Linda Tauscher and Saul Greenberg, Internet Journal of Human Computer studies, vol 11,no.7,pp.97-137,2000

DESCRIPTION:

We report on users' revisitation patterns to World Wide Web pages, and use these to lay an empirical foundation for the design of history mechanisms in web browsers. Through history, a user can return quickly to a previously visited page, possibly reducing the cognitive and physical overhead required to navigate to it from scratch. We analyzed 6 weeks of usage data collected from 23 users of a commercial browser. We found that 58% of an individual's pages are revisits, and that users continually add new web pages into their repertoire of visited pages. People tend to revisit pages just visited, access only a few pages frequently, browse in very small clusters of related pages, and generate only short sequences of repeated URL paths.

We compared different history mechanisms, and found that the stack-based prediction method prevalent in commercial browsers is inferior to the simpler approach of showing the last few recently visited URLs with duplicates removed. Other predictive approaches fare even better. Our results suggest new approaches to managing history in browsers.

[2.5] Utilizing Re-finding for Personalized Information Retrieval

AUTHOR : Sarah K Tyler, Jian wang yizhang, In WSDM vol.9,no.5,2010

DESCRIPTION:

Individuals often use search engines to return to web pages they have previously visited. This behaviour, called refinding, accounts for about 38% of all queries. While researchers have shown how re-finding differs from traditionally studied new-findings, research on how to predict and utilize re-finding is limited. In this paper we explore refinding for personalized search. We compared three machine learning algorithms (decision trees, Bayesian multinomial regression and support vector machines) to identify refindings. We then propose several re-ranking methods to utilize the prediction, including promoting predicted re-finding URLs and combining re-finding prediction with relevance estimation.

The experimental results demonstrate that using re-finding predictions can improve retrieval performance for personalized search.

[2.6] Contextual Web History: Using Visual and Contextual Cues to Improve Web Browser History

AUTHOR : Sungjoon Steve Won, Jing Jin, Jason I. Hong, In CHI pages

Vol. 26,pp.1457-1446, 2009

DESCRIPTION :

Modern web browsers offer history functionality, few people use it to revisit previously viewed web pages. In this paper, we present the design and evaluation of Contextual Web History (CWH), a novel browser history implementation which improves the visibility of the history feature and helps people find previously visited web pages. We present the results of a formative user study to understand what factors helped people in finding past web pages. From this, we developed CWH to be more visible to users, and supported search, browsing, thumbnails, and metadata. Combined, these relatively simple

features outperformed Mozilla Firefox 3's built-in browser history function, and greatly reduced the time and effort required to find and revisit a web page.

[2.7] FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the any organization as the project involves in major change of the current system. The adoption of this project would carried out with ease and in less time. For feasibility analysis, some understanding of the major requirements and the concepts of how the underlying technology works is essential. There are aspects in the feasibility study portion of the portion of the preliminary investigation:

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

2.7.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system relies within the budget and this was achieved because most of the technologies used for freely available. Only the customized products had to be purchased. So this can be easily affordable for people of different environment.

2.7.2 TECHNICAL FEASIBILITY

The technical issue is usually raised during the feasibility stage of investigation include the following:

Will the proposed system provide adequate response to inquiries, regardless of the number or location of the users?

Are the technical guarantees of accuracy reliability, ease of access and data security?

Can the system be upgraded if developed?

Does the necessary exist to do what is suggested?

Earlier no system existed to the needs of ‘parental control’. The current system developed is technical feasible. The proposed project requires less technical resources for execution as the existing SQL is used as database and tomcat server is used as it doesn’t require internet connectivity. Therefore it provides the technical guarantee of accuracy, reliability and security. The hardware and software requirement for this project are not many and are available as free as open source. The installation of these devices to does not need much human resource and easy installation.

2.7.3 OPERATIONAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. This system requires knowledge on using software and how it works for the administrator to monitor the system. The user must have basic knowledge on to search resources in the web and if they likes it they can bookmark interested link. While tagging user can give own tag name or asks the server to suggest tags. The system proposed would be which the parental control, that is it can be use to block a specific tag or a specific category is made as heart of this proposed project such that minimal knowledge on the innovative technologies is needed.

SYSTEM

DESIGN

CHAPTER 3

SYSTEM DESIGN

3.1 Design of the Proposed System

We propose to create the group like in social media, in that group admin will register and it will be stored in a database. Then group admin will add n number of users in that group. The group members can search the query and can bookmark the tag. The other members in the group can search the tag in users bookmarks or other users bookmark. When the users search the tag, the tag recommend their own tags or in their own tags. In this content filteration process will be used. The blocked tags cannot be access to the specified users.

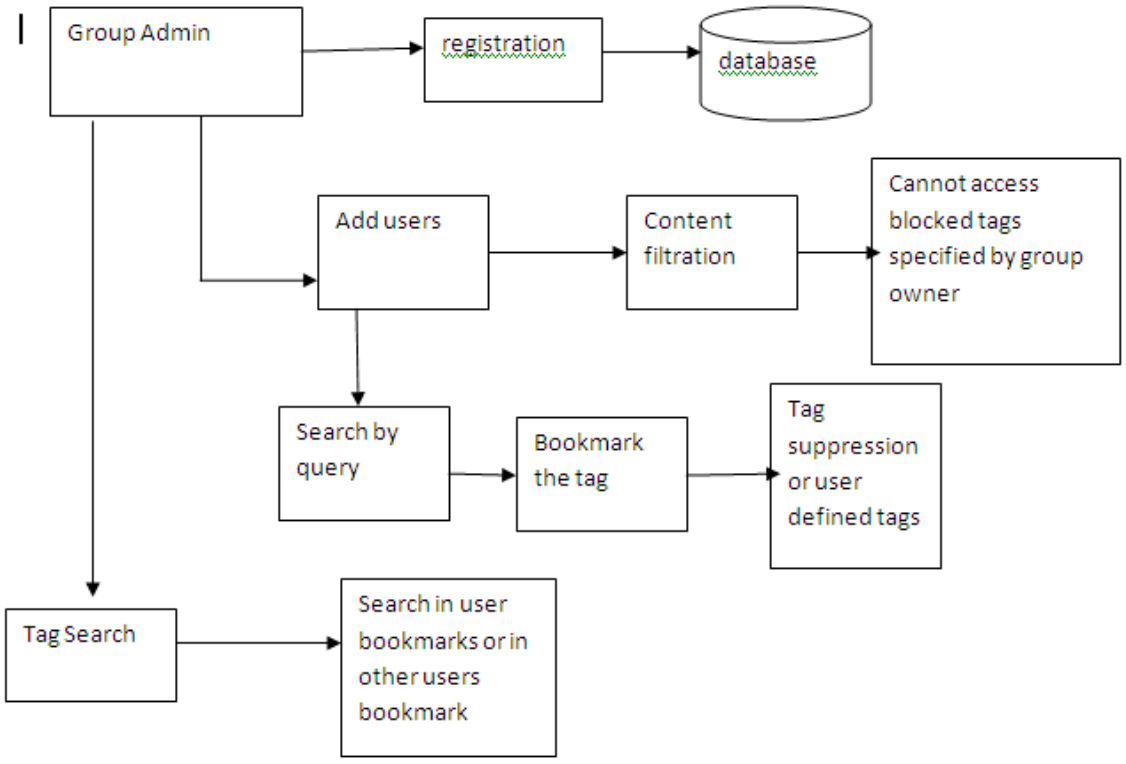

Fig 3.1 Proposed system Architecture

3.2 Data Flow Diagram for Proposed System

A data flow diagram **(**DFD**)** is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of the information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in a sequence or in parallel unlike a flowchart which also shows this information.

Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top-down approach to Systems Design. This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

DFD:
Level 0:

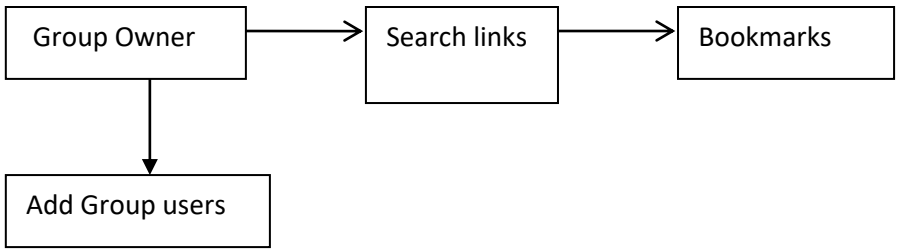

Fig 3.2 Data flow diagram level 0

Level1:

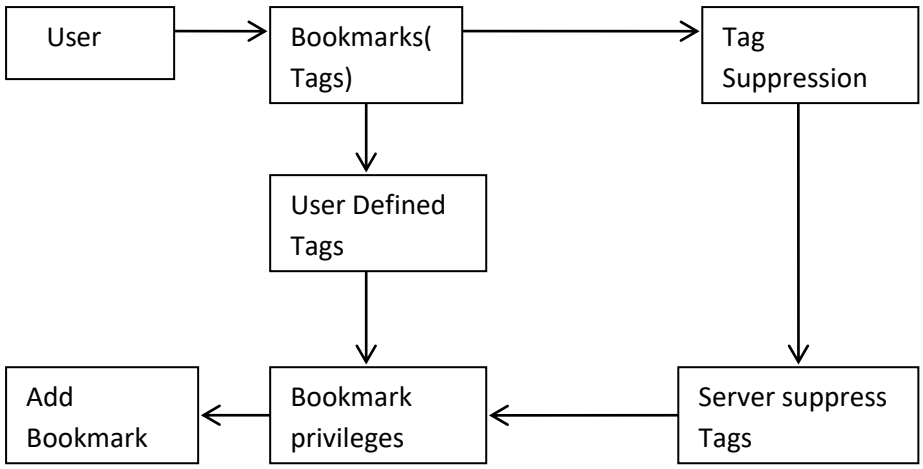

Fig 3.3 Data flow diagram level 1

Level2:

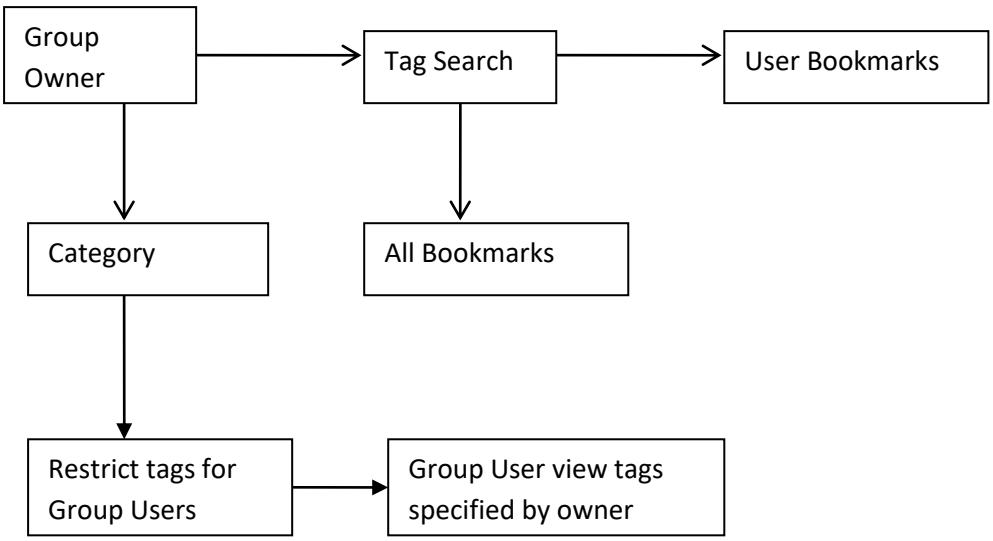

Fig 3.4 Data flow diagram level 2

3.3 UML Diagram

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development

process. The UML uses mostly graphical notations to express the design of software projects.

UML can be applied to diverse application domains (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component software development methods and for various implementation platforms (e.g., J2EE, .NET).

UML is a standard modelling language, not a software development process. UML Specification explained that process:

- Provides guidance as to the order of a team's activities
- Specifies what artifacts should be developed
- Directs the tasks of individual developers and the team as a whole
- Offers criteria for monitoring and measuring a project's products and activities

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

UML is intentionally process independent and could be applied in the context of different processes. Still, it is most suitable for use case driven,

iterative and incremental development processes. An example of such process is Rational Unified Process (RUP).

UML is not complete and it is not completely visual. Given some UML diagram, we can't be sure to understand depicted part or behaviour of the system from the diagram alone. Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams.

3.3.1 USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. they are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse

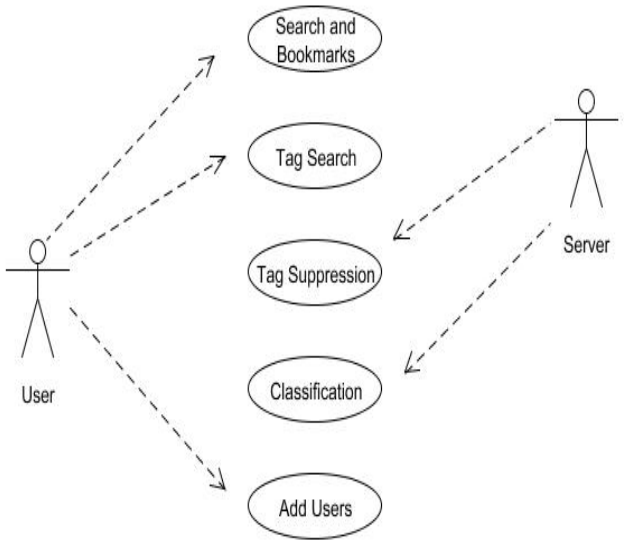

Fig 3.5 Use case Diagram

3.3.2 SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifacts for dynamic modeling, which focuses on identifying the behaviour within your system.

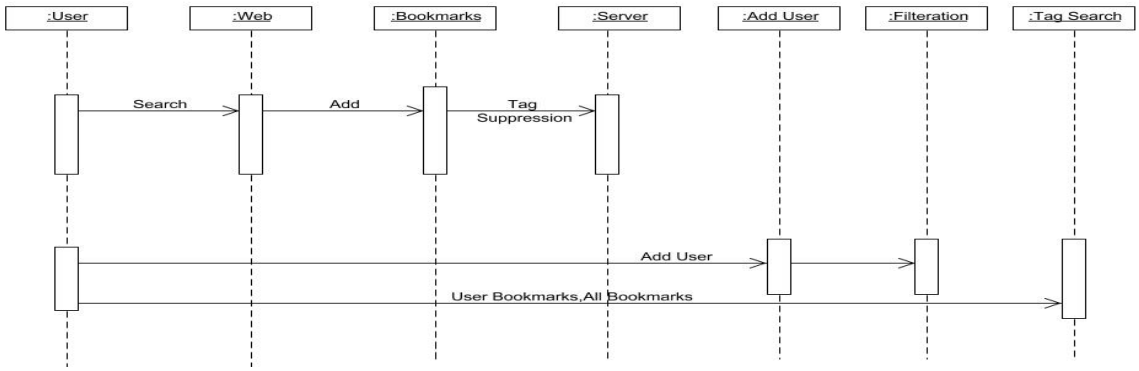

Fig 3.6 Sequence Diagram

3.3.3 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information. The class diagram is the main building block of object-oriented modelling. It can also be used for data modelling. The classes in the class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

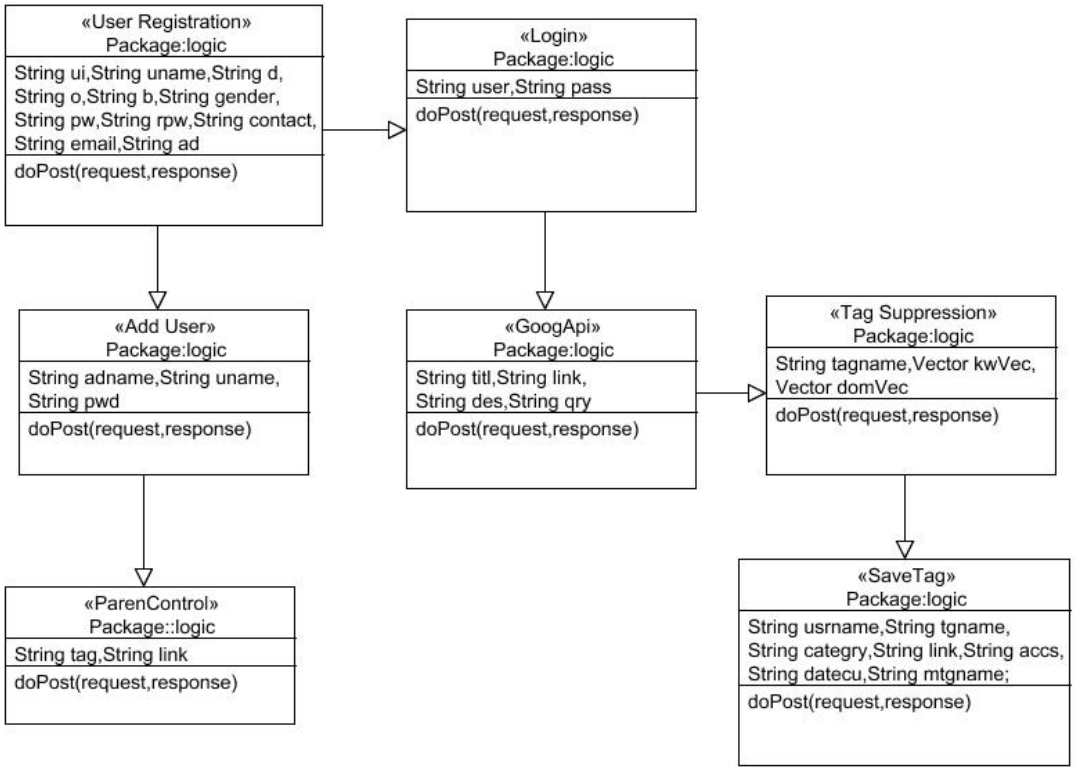

Fig 3.7 Class Diagram

3.3.4 COLLABORATION DIAGRAM

Another type of interaction diagram is the collaboration diagram. A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchange among the objects within the collaboration to achieve a desired outcome. It resembles a flow chart that potrays the roles, functionally and behaviour of individual objects as well as the overall operation of the system in real time. As the number of objects and messages grows, a collaboration diagram can become difficult to read. Several vendors offer software for creating and editing collaboration diagrams. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

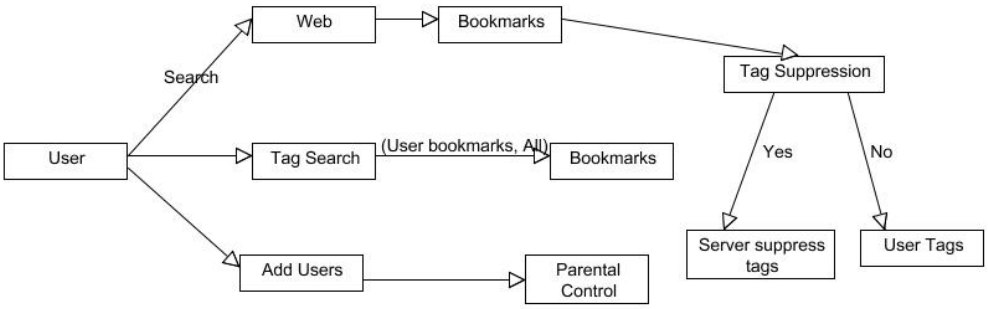

Fig 3.8 Collaboration Diagram

3.3.5 ACTIVITY DIAGRAM

The activity diagram is the graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between. It is used to model a large activity's sequential workflow by focussing on action sequences and respective action initiating conditions. The state of an activity relates to the performance of each workflow step. It is represented by shapes that are connected by arrows. Black circles represent an initial workflow state. A circled black circle indicates an end state. Rounded rectangles represent performed actions, which are described by text inside each rectangle. A diamond shape is used to represent a decision, which is a key activity diagram concept. Upon activity completion, a transition (or set of sequential activities) must be selected from a set of alternative transitions for all use cases. Synchronization bars indicates the start or completion of concurrent activities are used to represent parallel subflows.

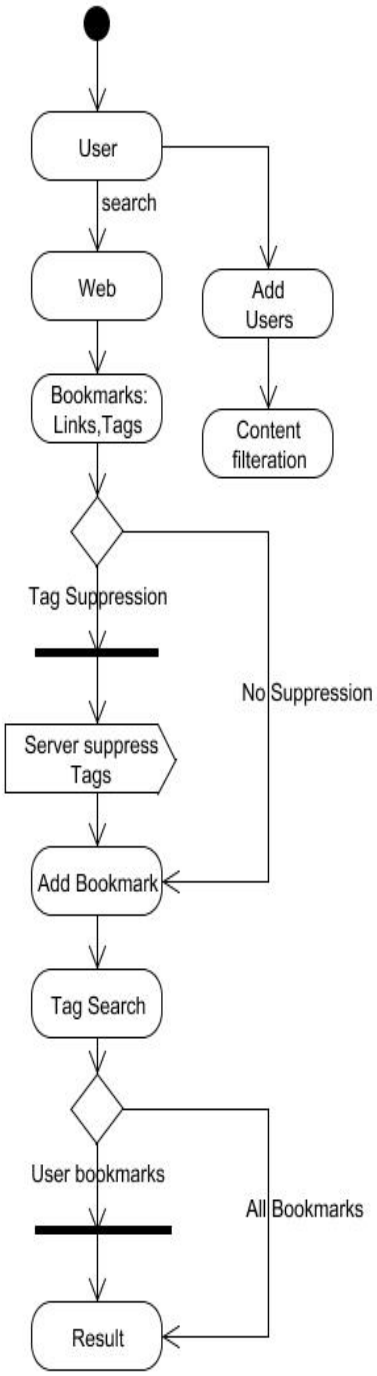

Fig 3.9 Activity Diagram

MODULE DESCRIPTIONS

CHAPTER 4

MODULE DESCRIPTIONS

4.1 MODULES

A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understandability, Modular continuity, Modular Protection.

The following are the modules of the project, which are planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

In the proposed system, there are five modules and they are described as follows:

- Add Group User
- Search and Bookmark
- Tag Suppression and Tag Recommendation
- Parental Control

4.2 MODULE DESCRIPTION

4.2.1 ADD GROUP USER

In this module, Group owner has to register their details. A group owner is nothing but an admin who adds other users. He has to register his details by giving user name and password. After successful registration, details are stored in database. When the group owner login, they can view they profile. Here Group owner can add users. The users who are added by the admin can also add other users. Group owner set username and password to all group users. Using this username and password, user can view group owner's profile, bookmarks etc. Group users register their details. At registration, group user has to provide username and password given by group owner. Group owner restricts users to view only specified contents.

4.2.2 SEARCH AND BOOKMARK

User can search resources in web according to their personal preferences. List of websites displayed where user can view their interested links. If the user likes the link, they can bookmark by giving tag for future tag search. Search can be done in two ways "online and offline" search. Online search is a process of interactively searching for and retrieving requested information via computer from databases via online. Offline search is a computer software that downloads tha data or webpages, making them available when the computer is offline, not connected to the internet.

Bookmarking is to make a record of the address of a web page on a computer so that you can find it again easily. While book marking, user can give multiple tags. Username, tag name, link and other details are stored in database. Tags given by user will be classified and stored in database. User can give access privileges to bookmarks. If the bookmark is private, only the user can view. If the bookmark is public, other users can view their bookmarks.

4.2.3 TAG SUPPRESSION AND TAG RECOMMENDATION

For instance, recent work has proposed interesting approaches to tag recommendation and suppression. Tag suppression concerns the possibility of identifying the most probable tags to be associated with a non -tagged resource, whereas tag recommendation is meant to suggest to users the tags to be used to describe resources they are bookmarking. Tags are predicted based on resource's content and its similarity with already tagged resources.

Tag recommendation is enforced by computing tag-based user profiles, and by suggesting tags specified on a given resource by users having similar characteristics or interests

User likes a link in web and bookmarks that link. User tag the bookmark. While tagging, user can give own tag or ask server to suggest tags. Server provides suppressed tags where user can choose tag. In this way, user protects their privacy while tagging. All the book marking information will be stored in database. If the user searches a tag, they can search in their bookmarks or in all bookmarks. If the link has multiple tags, user searched tag and other tags for that links will be displayed. Recommending users for the past 1 week links and tags.

4.2.4 PARENTAL CONTROL

Group owner can add users for content filtration purpose. Web filter is a program that can screen an incoming Web page to determine whether some or all of it should not be displayed to the user. The filter checks the origin or content of a Web page against a set of rules provided by company or person who has installed the web filter.

Group owner enable a web filter for group users by granting them access only to contents specified by group owner. Group owner denote which

resources is unsafe. By checking the available tag categories, group owner blocks the tags for users. Group user can access the tags giving username and password. Group user has restrictions only in Tag Search. All the other services, group user can access. (Search and bookmark, Add bookmark). If any one of the facet is blocked to particular user, then they cannot access regarding that facet.

NATURAL LANGUAGE PROCESSING TECHNIQUE:

The study which focuses on the interaction between human language and computer is called Natural Language Processing, or NLP for short. It sits at the intersection of computer science, artificial intelligence and computation linguistics. NLP is a way for computers to analyze, understand and derive meaning from human language in a smart and useful way. By utilizing NLP developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity, recognition relationship extraction, sentiment analysis, speech recognition, and topic segmentation. Apart from common word processor operations that treat like a mere sequence of symbols, NLP considers the hierarchical structure of languages: several words make a phrase, several phrases make a sentence and ultimately sentences convey the idea

NATURAL LANGUAGE PROCESSING USED FOR:

SENTENCE SEGMENTATION, PART OF SPEECH TAGGING, AND PURSUING:

Natural language processing can be used to analyse the part of a sentence to better understand the grammatical construction of the sentence.

DEEP ANALYTICS:

Deep analytics involves the application of advanced data processing techniques in order to extract specific information from large data sets. It is particularly useful when dealing with precisely targeted or high complex queries. It is often used in the financial sector scientific community, and biomedical sector.

MACHINE TRANSLATION:

Natural Language Processing is increasingly being used for machine translation programs, in which one human language is automatically translated into another human language.

NAMED ENTITY EXTRACTION:

In data mining, a named entity definition is a phrase or word that clearly identifies one item from a set of other item that have similar attributes.

CO-REFERENCE RESOLUTION:

In a chunk of text, co-reference resolution can be used to determine which words are used to refer to the same object.

AUTOMATIC SUMMARIZATION:

Natural language processing can be used to produce a readable summary from large number of chunks of text. For example one might do automatic summarization to produce a short summary of a dense academic article.

ADVANTAGES OF NATURAL LANGUAGE PROCESSING:

- The benefits of Natural language processing are innumerable.
- Natural language processing can be leveraged by companies to improve the efficiency of documentation processes, improve accuracy of documentation.
- NLP includes fast processing, user friendliness, inferring solutions which may or may not be created previously and the most important is an ability to converse in the language you like.

DISADVANTAGES OF NATURAL LANGUAGE PROCESSING:

- Requires clarification dialogue.
- May requires more keystrokes.
- Must not show context

REQUIREMENT SPECIFICATION

CHAPTER 5

REQUIREMENT SPECIFICATION

5.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

Hard Disk	:	40GB and above
RAM	:	2GB and above
Processor	:	P IV and above
Internet Dongle	:	1

5.2 SOFTWARE REQUIREMENTS

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

- Windows 7(32 bit) and above
- JDK 1.6
- Tomcat 6.0
- MySQL5.0
- WordNet 2.1

5.2.1OVERVIEW OF THE MYSQL DATABASE MANAGEMENT SYSTEM

What is MySQL?

- MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software. MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded in 1995 by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.
- The MySQL® software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. MySQL is a registered trademark of MySQL AB.

The MySQL software is Dual Licensed. Users can choose to use the MySQL software as an Open Source product under the terms of the GNU General Public License or can purchase a standard commercial license from MySQL AB.MySQL databases are relational.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that

with a well-designed database, the application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on the programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist.

MySQL Architecture

Three layer model:

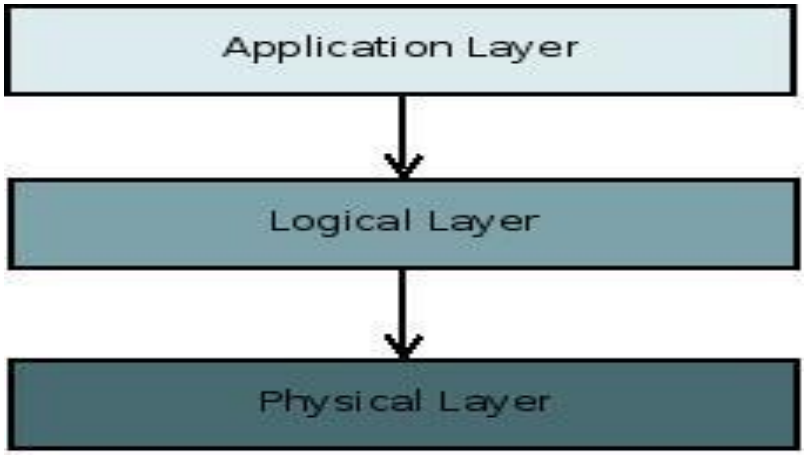

Fig 5.1 MySQL Architecture

- The application layer contains common network services for connection handling, authentication and security. This layer is where different clients interact with MySQL these clients can written in different API's:.NET, Java, C, C++, PHP, Python, Ruby, Tcl, Eiffel, etc...
- The Logical Layer is where the MySQL intelligence resides, it includes functionality for query parsing, analysis, caching and all built-in functions (math, date...). This layer also provides functionality common across the storage engines.

- The Physical Layer is responsible for storing and retrieving all data stored in “MySQL”. Associated with this layer are the storage engines, which MySQL interacts with very basic standard API's.

Storage engines:

MySQL supports several storage engines that act as handlers for different table types. MySQL storage engines include both those that handle transaction-safe tables and those that handle non-transaction-safe tables.

MySQL Cluster Overview

MySQL Cluster is a technology that enables clustering of in-memory databases in a shared-nothing system. The shared-nothing architecture enables the system to work with very inexpensive hardware, and with a minimum of specific requirements for hardware or software.

MySQL Cluster is designed not to have any single point of failure. In a shared-nothing system, each component is expected to have its own memory and disk, and the use of shared storage mechanisms such as network shares, network file systems, and SANs is not recommended or supported.

MySQL Cluster integrates the standard MySQL server with an in-memory clustered storage engine called NDB (which stands for “Network Data Base”). In our documentation, the term NDB refers to the part of the setup that is specific to the storage engine, whereas “MySQL Cluster” refers to the combination of one or more MySQL servers with the NDB storage engine.

MySQL Cluster consists of a set of computers, known as hosts, each running one or more processes. These processes, known as nodes, may include MySQL servers (for access to NDB data), data nodes (for storage of the data), one or more management servers, and possibly other specialized data access programs. The relationship of these components in a MySQL Cluster is shown here.

5.2.2 INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialled in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of main, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv[] of C), and is declared as a static method.

To output text from the program, we execute the println method of System. Out, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things :

- Programming language
- Platform

THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-

only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

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

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries **(packages)** of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

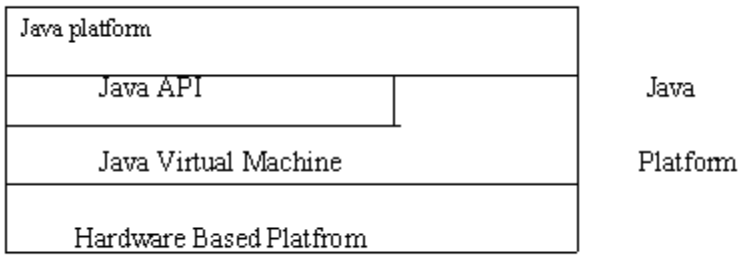

Fig 5.2 Java Platform

5.2.3 INTRODUCTION TO APPACHE TOM CAT

Tomcat, developed by Apache, is a standard reference implementation for Java servlets and JSP. It can be used standalone as a web server or be plugged into a web server like Apache, Netscape Enterprise Server, or Microsoft Internet

Information Server. To start Tomcat, it has to be first set the JAVA_HOME environment variable to the JDK home directory using the following command. The JDK home directory is where your JDK is stored. On my computer, it is c:\Program Files\jdk1.6.0_24.

The apache tomcat server is an open source, java-based web application container that was created to servlets and java server page (JSP) web applications. It was created under the apache-Jakarta subproject; however, due to its popularity, it is now hosted as separate apache project, where it is supported and enhanced by a group of volunteers from the open source java community.

The Tomcat Manager Web Application

The Tomcat manager web application is packaged with the tomcat server. It is installed in the context path of manager and provides the basic functionality to manage web applications running in the tomcat server from any web browser. Some of the provided functionality includes the ability to install, start, stop, remove, and report on web applications. Covers the details of the tomat manager web application.

The Server

The first container element referenced in this snippet is the <Server> element. It represents the entire Catalina servlet engine and is used as a top-level element for a single Tomcat instance. The <Server> element may contain one or more <Service> containers.

The Service

The next container element is the <Service> element, which holds a collection of one or more <Connector> elements that share a single <Engine> element. N-number of <Service> elements may be nested inside a single <Server> element.

The Connector

The next type of element is the <Connector> element, which defines the class that does the actual Handling requests and responses to and from a calling client application.

The Engine

The third container element is the <Engine> element. Each defined <Service> can have only one <Engine> element and this single <Engine> component handles all requests received by all of the defined <Connector> components defined by a parent service.

The Host

The <Host> element defines the virtual hosts that are contained in each instance of a Catalina <Engine>. Each <Host> can be a parent to one or more web applications, with each being represented by a <Context> component.

The Context

The <Context> element is the most commonly used container in a Tomcat instance. Each <Context> element represents an individual web application that is running within a defined <Host>. There is no limit to the number of contexts that can be defined within a <Host>.

Java Web Applications

The main function of the Tomcat server is to act as a container for Java web applications. Therefore, before we can begin our Tomcat-specific discussion, a brief introduction as to exactly what web applications are is in order. The concept of a web application was introduced with the release of the Java servlets specification. According to this specification, “a web application is a collection of servlets, html pages, classes, and other resources that can be bundled and run on multiple containers from multiple vendors.” What this really means is that a web application is a container that can hold any combination of the following list of objects:

- Servlets
- Java Server Pages (JSPs)
- Utility classes
- Static documents, including HTML, images, JavaScript libraries, cascading style sheets, and so on
- Client-side classes
- Meta-information describing the web application

One of the main characteristics of a web application is its relationship to the Servlets Context. Each web application has one and only one Servlets Context. This relationship is controlled by the servlets container and guarantees that no two web applications will clash when accessing objects in the Servlets Context.

5.2.4 WORDNET 2.1

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provide short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus. While it is accessible to human users via a web browser, its primary use is automatic text analysis and artificial intelligence applications.

It was created in Cognitive science laboratory of Princeton University under the direction of Psychology Professor George Armitage Miller.

As of 2012 Word Net's latest online version is 3.1.Word Net includes the lexical categories nouns, verbs, adjectives and adverbs but ignores prepositions, determines and other function words.

IMPLEMENTATION

CHAPTER 6

IMPLEMENTATION

6.1 SAMPLE CODE

DB SERVICES

javadbconnection.java

```
package DBServices;
import java.sql.Connection;
import java.sql.DriverManager;
public class DBConnection
{
private Connection con = null;
private static DBConnection dbc;
DBConnection()
{
try{
Class.forName("com.mysql.jdbc.Driver").newInstance();
con = DriverManager.getConnection("jdbc:mysql://localhost/collaborativetag",
"root", "root");
}
catch(Exception e)
{
e.printStackTrace();
}
}
```

```
public static DBConnection getInstance()
{
if(dbc==null)
{
 synchronized (DBConnection.class)
{
if(dbc==null)
dbc=new DBConnection();
}
}
return dbc;
}
 public Connection getConnection()
{
 return con;
}
}
```

javadbservices.java

```
package DBServices;
import java.sql.Connection;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

```
import java.util.TreeMap;
import java.util.List;
import java.util.ArrayList;
public class DBServices
{
TreeMap<String,Integer> tm=new TreeMap<String,Integer>();
DBConnection dbc = DBConnection.getInstance();
Connection con = dbc.getConnection();
ResultSet rs;
int incr=0;
    //function to retrieve username and password from regis table
public boolean testLogin(String uname,String pwd
{
boolean check=false;
try {
Statement st = con.createStatement();
rs = st.executeQuery("select username,pwd from register where
username='"+uname+"' and pwd='"+pwd+"'");
check=rs.next();
} catch (SQLException e)
{
e.printStackTrace();
}
return check;
}
```

```
public void insertRegis(String ui,String uname,String dob,String gender,String
pw,String rpw,String contact,String email,String ad)

{

 Try

{

Statement st = con.createStatement();

String qu = ("insert into register
values('"+ui+"','"+uname+"','"+dob+"','"+gender+"','"+pw+"','"+rpw+"','"+conta
ct+"','"+email+"','"+ad+"')");

st.executeUpdate(qu);

}

catch (SQLException e)

{

e.printStackTrace();

}

}

public boolean updateRegis(String ui,String uname,String dob,String
gender,String pw,String rpw,String contact,String email,String ad)

{

boolean check=false;

 Statement st;

try {

st = con.createStatement();

String qu="update register set
userid='"+ui+"',username='"+uname+"',dob='"+dob+"',gender='"+gender+"',pw
d='"+pw+"',rpw='"+rpw+"',contact='"+contact+"',email='"+email+"',address='"
+ad+"' where userid='"+ui+"'";
```

```
st.executeUpdate(qu);
check=true;
} catch (SQLException e)
{
// TODO Auto-generated catch block
e.printStackTrace();
}
return check;
}
public String selectRegis(String user,String pass)
{
Stringui="",uname="",dob="",gend="",pw="",rpw="",cont="",ema="",address=
"",all="";
Statement st;
try
{
st = con.createStatement();
ResultSet rs = st.executeQuery("select * from register where
username='"+user+"' and pwd='"+pass+"'");
while(rs.next())
{
ui=rs.getString("userid");
 uname=rs.getString("username");
dob=rs.getString("dob");
gend=rs.getString("gender");
```

```
pw=rs.getString("pwd");

rpw=rs.getString("rpw");

cont=rs.getString("contact");

ema=rs.getString("email");

address=rs.getString("address");

System.out.println("address"+address);

}

Stringall=ui+"!"+uname+"!"+dob+"!"+gend+"!"+pw+"!"+rpw+"!"+cont+"!"+e
ma+"!"+address;

} catch (SQLException e)

{

e.printStackTrace();

}

return all;

}

public String displayRegis()

{

Stringui="",uname="",dob="",gend="",pw="",rpw="",cont="",ema="",address=
"",all="";

Statement st;

try {

st = con.createStatement();

ResultSet rs = st.executeQuery("select * from register");

while(rs.next())

{
```

```
ui=rs.getString("uid");

uname=rs.getString("uname");

dob=rs.getString("dob");

gend=rs.getString("gender");

pw=rs.getString("pw");

rpw=rs.getString("repw");

cont=rs.getString("contact");

ema=rs.getString("email");

address=rs.getString("addr");

}

Stringall=ui+"!"+uname+"!"+dob+"!"+gend+"!"+pw+"!"+rpw+"!"+cont+"!"+e
ma+"!"+address;

} catch (SQLException e)

{

e.printStackTrace();

}

return all;

}

public void insertTag(String uname,String tag,String categor,String link,String
accs,java.sql.Date dtec)

{

System.out.println("DBService"+dtec);

try{

Statement st = con.createStatement();
```

```
String qu = ("insert into tag
values('"+uname+"','"+tag+"','"+categor+"','"+link+"','"+accs+"','"+dtec+"')");

st.executeUpdate(qu);

}

catch (SQLException e)

{

e.printStackTrace();

}}

public void selectTag(String tag,String categor)

{

Statement st;

try {

st = con.createStatement();

ResultSet rs = st.executeQuery("select tagn,category from tag");

while(rs.next()){

}

}catch (SQLException e)

{

e.printStackTrace();

}

}

public String selectLink(String tag)

{

Statement st;

String lnk="";
```

```
System.out.println("tag in selectlink"+tag);
 try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select link from tag where tagn='"+tag+"'");
while(rs.next())
{
lnk=(String)rs.getString("link");
}
}catch (SQLException e)
{
e.printStackTrace();
}
return lnk;
}
public List selectUsrs(String adm)
{
Statement st;
String lnk="";
List usrs= new ArrayList();
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select usrname from adduser where
admin='"+adm+"'");
while(rs.next())
{
```

```
                                        //lnk=(String)rs.getString("usrname");
usrs.add(rs.getString("usrname"));
 }
}catch (SQLException e)
{
e.printStackTrace();
}
return usrs;
}
public List selectCategry(String adm)
{
Statement st;
List ctgy= new ArrayList()
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select distinct category from tag where
username='"+adm+"'");
while(rs.next())
{
        //lnk=(String)rs.getString("usrname");
ctgy.add(rs.getString("category"));
}
}catch (SQLException e)
{
e.printStackTrace();
```

```
 }
return ctgy;
}
public String selectTrendLink(java.sql.Date bdat,java.sql.Date cdat)
{
Statement st;
String lnk="",llink="";
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select link from tag where bdate>='"+bdat+"' and bdate<='"+cdat+"' and access='Public'");
while(rs.next())
{
lnk=(String)rs.getString("link");
llink=llink+"*"+lnk;
}
}catch (SQLException e)
{
                                    // TODO Auto-generated catch block
 e.printStackTrace();
}
return llink;
}
public List selectLatestTag(java.sql.Date bdat,java.sql.Date cdat)
{
```

```
Statement st;
List tagna= new ArrayList();
List lin= new ArrayList();
List res = new ArrayList();
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select tagn,link from tag where
bdate>='"+bdat+"' and bdate<='"+cdat+"' and access='Public'");
while(rs.next())
{
tagna.add(rs.getString("tagn"));
lin.add(rs.getString("link"));
}
res.addAll(tagna);
res.add("*");
res.addAll(lin);
}catch (SQLException e)
{
e.printStackTrace();
 }
return res;
}
public List selectTrendingTag(java.sql.Date bdat,java.sql.Date cdat)
{
Statement st;
```

```
List tagna= new ArrayList();

List res = new ArrayList()

try {

st = con.createStatement();

ResultSet rs = st.executeQuery("select distinct query from search where
cdate>="'+bdat+"' and cdate<="'+cdat+"'");

while(rs.next())

{

tagna.add(rs.getString("query"));

}

res.addAll(tagna);

}catch (SQLException e)

{

e.printStackTrace();

}

return res;

}

public void insertUser(String adname,String uname,String pwd)

{

try{

Statement st = con.createStatement();

String qu = ("insert into adduser
values("'+adname+"',"'+uname+"',"'+pwd+"')");

st.executeUpdate(qu);

}
```

```
catch (SQLException e)
{
e.printStackTrace();
}
}
public void insertFile(String uname,String uurl)
{
try{
Statement st = con.createStatement();
String qu = ("insert into upload values('"+uname+"','"+uurl+"')");
st.executeUpdate(qu);
}
catch (SQLException e)
{
e.printStackTrace();
}
}
public List searchTag(String un,String acc)
{
List li = new ArrayList();
List tagna= new ArrayList();
List res = new ArrayList();
Statement st;
```

```
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select link,tagn from tag where
username='"+un+"' and access='"+acc+"'");
while(rs.next())
{
li.add(rs.getString("link"));
tagna.add(rs.getString("tagn"));
}
res.addAll(li);
res.add("*");
res.addAll(tagna);
}
catch (SQLException e)
{
e.printStackTrace();
}
return res;
}
public List searchAll(String una)
{
List li = new ArrayList();
List tagna= new ArrayList();
List res = new ArrayList();
Statement st;
```

```
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select link,tagn from tag where
username='"+una+"' or access='Public'");
while(rs.next())
{
li.add(rs.getString("link"));
tagna.add(rs.getString("tagn"));
}
res.addAll(li);
res.add("*");
res.addAll(tagna);
//System.out.println("res"+res);
}
catch (SQLException e)
{
e.printStackTrace();
}
return res;
}
public void insertUserTags(String uname,String categor,String tag)
{
try{
Statement st = con.createStatement();
```

```
String qu = ("insert into filteration
values('"+uname+"','"+categor+"','"+tag+"')");
st.executeUpdate(qu);
}
catch (SQLException e)
{
e.printStackTrace();
}
}
public void insertSearchQuery(String qry,java.sql.Date dtec)
{
try{
Statement st = con.createStatement();
String qu = ("insert into search values('"+qry+"','"+dtec+"')");
st.executeUpdate(qu);
}
catch (SQLException e)
{
                    // TODO Auto-generated catch block
e.printStackTrace();
}
}
public List category()
{
```

```
List res = new ArrayList();
Statement st;
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select DISTINCT category from tag");
while(rs.next())
{
res.add(rs.getString("category"));
}
//System.out.println("res"+res);
} catch (SQLException e)
{
e.printStackTrace();
}
return res;
}
public String selectTag(String tag)
{
Statement st;
String tg="",ctg="";
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select tagn from tag where category='"+tag+"'
and access='Public'");
while(rs.next())
```

```
{
tg=(String)rs.getString("tagn");
ctg=ctg+"*"+tg;
}
}catch (SQLException e)
{
e.printStackTrace();
}
System.out.println("tag= "+ctg);
return ctg;
}
public List showTag(String catg)
{
Statement st;
String tg,lnk;
List tagnls=new ArrayList();
List lnkls=new ArrayList();
List res=new ArrayList()
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select tagn,link from tag where
category="'+catg+"' and access='Public'");
while(rs.next())
{
tagnls.add(rs.getString("tagn"));
```

```
lnkls.add(rs.getString("link"));
}
res.addAll(tagnls);
res.add("*");
res.addAll(lnkls);
}catch (SQLException e)
{
e.printStackTrace();
}
return res;
}
public String selectFile(String name)
{
Statement st;
String url="";
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("select ufileurl from upload where
ufilename='"+name+"'");
while(rs.next())
{
url=(String)rs.getString("ufileurl");
System.out.println("url"+url);
}
```

```
}catch (SQLException e)
{
e.printStackTrace();
}
return url;
}
}
```

LOGIC:

Javaadduser.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.http.servlet.HttpsServlet;
import DBServices.DBServices;
public class AddUser extends HttpsServlet {
        /**
         * Constructor of the object.
```

```
        */
String adname,child,uname,pwd;
public HttpSession sessi;
public AddUser() {
super();
}
        /**
         * Destruction of the servlet. <br>
         */
public void destroy() {
super.destroy(); // Just puts "destroy" string in log
                // Put your code here
}
        /**
         * The doPost method of the servlet. <br>
         *
         * This method is called when a form has its tag value method equals to
post.
         *
         * @param request the request send by the client to the server
         * @param response the response send by the server to the client
         * @throws ServletException if an error occurred
         * @throws IOException if an error occurred
         */
public void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException {
response.setContentType("text/html");
sessi=request.getSession();
adname=(String)sessi.getAttribute("username");
uname=request.getParameter("usern");
pwd=request.getParameter("passw");
try
{
        //insert user details to register table in db
DBServices dbs=new DBServices();
dbs.insertUser(adname,uname, pwd);
String s = "User added successfully";
request.setAttribute("msg1", s);
RequestDispatcher req = request.getRequestDispatcher("Profile.jsp");
req.forward(request, response);
}
catch(Exception pce)
{
pce.printStackTrace();
}
}
        /**
         * Initialization of the servlet. <br>
         *
```

```
        * @throws ServletException if an error occurs
        */
public void init() throws ServletException {
                // Put your code here
}
}
```

javalogin.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.http.servlet.HttpsServlet;
import DBServices.DBServices;
public class Login extends HttpsServlet {
String user,pass;
public HttpSession sessi;
        /**
         * Constructor of the object.
         */
```

```
public Login()
{
super();
}

        /**
         * Destruction of the servlet. <br>
         */
public void destroy() {
super.destroy(); // Just puts "destroy" string in log
                // Put your code here
}
        /**
         * The doPost method of the servlet. <br>
         *
         * This method is called when a form has its tag value method equals to post.
         *
         * @param request the request send by the client to the server
         * @param response the response send by the server to the client
         * @throws ServletException if an error occurred
         * @throws IOException if an error occurred
         */
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
```

```
{
response.setContentType("text/html");
user = request.getParameter("user");
 pass = request.getParameter("pass");
  try
  {
DBServices db=new DBServices();
boolean check=db.testLogin(user, pass);
if(check)
{
DBServices dbs=new DBServices();
String dbval=dbs.selectRegis(user, pass);
String[] parts = dbval.split("!");
//System.out.println("parts"+parts[0]);
sessi=request.getSession();
sessi.setAttribute("userid",parts[0] );
sessi.setAttribute("username",parts[1] );
sessi.setAttribute("birthd",parts[2]);
sessi.setAttribute("gend",parts[3]);
sessi.setAttribute("pwd",parts[4]);
sessi.setAttribute("rpwd",parts[5]);
sessi.setAttribute("contact",parts[6]);
sessi.setAttribute("mail",parts[7]);
sessi.setAttribute("address",parts[8]);
```

```
RequestDispatcher req1 = request.getRequestDispatcher("Profile.jsp");
 req1.forward(request, response);
 }
 else
 {
System.out.println("Invalid user");
String s = "Invalid username or password";
request.setAttribute("inf", s);
RequestDispatcher req = request.getRequestDispatcher("index.jsp");
 req.forward(request, response);
 }
 }
 catch(Exception x)
  {
   System.out.println(x);
}
}
        /**
         * Initialization of the servlet. <br>
         *
         * @throws ServletException if an error occurs
         */
public void init() throws ServletException {
                // Put your code here
```

```
}
}
```

javaregister.java

```
package logic;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.http.servlet.HttpsServlet;
import DBServices.DBServices;
public class Register extends HttpsServlet
{
String ui,uname,d,o,b,gender,pw,rpw,contact,email,ad;
        /**
         * Constructor of the object.
         */
public Register()
{
super();
}
        /**
```

```
        * Destruction of the servlet. <br>
        */
public void destroy() {
super.destroy(); // Just puts "destroy" string in log
                // Put your code here
}
                /**
        * The doPost method of the servlet. <br>
        *
        * This method is called when a form has its tag value method equals to
post.
        *
        * @param request the request send by the client to the server
        * @param response the response send by the server to the client
        * @throws ServletException if an error occurred
        * @throws IOException if an error occurred
        */
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
                //Get user input from Register.jsp
ui = request.getParameter("ui");
uname = request.getParameter("uname");
d = request.getParameter("d");
o = request.getParameter("m");
```

```
b = request.getParameter("y");
gender = request.getParameter("gen");
pw = request.getParameter("pw");
rpw = request.getParameter("rpw");
contact = request.getParameter("con");
email = request.getParameter("e");
ad = request.getParameter("ad");
System.out.println("uname....."+uname);
System.out.println("date......"+d);
System.out.println("o....."+o);
System.out.println("b........."+b);
String dob = (d+"/"+o+"/"+b);
System.out.println("gender....."+gender);
System.out.println("pw....."+pw);
System.out.println("Contact...."+contact);
System.out.println("email...."+email);
System.out.println("ADD....."+ad);
try
{
                //insert user details to register table in db
DBServices dbs=new DBServices();
dbs.insertRegis(ui, uname, dob, gender, pw, rpw, contact, email, ad);
String s = "Your Registration Completed.........";
request.setAttribute("sta", s);
```

```
RequestDispatcher req = request.getRequestDispatcher("Register.jsp");
req.forward(request, response);
}
catch(Exception pce)
{
pce.printStackTrace();
}
}
        /**
         * Initialization of the servlet. <br>
         *
         * @throws ServletException if an error occurs
         */
public void init() throws ServletException {
                // Put your code here
}
}
```

6.2 SCREENSHOTS

6.2.1 HOME PAGE

The home page of the purchase portal is displayed in Fig. 6.1.

Fig 6.1 Screenshot for home page

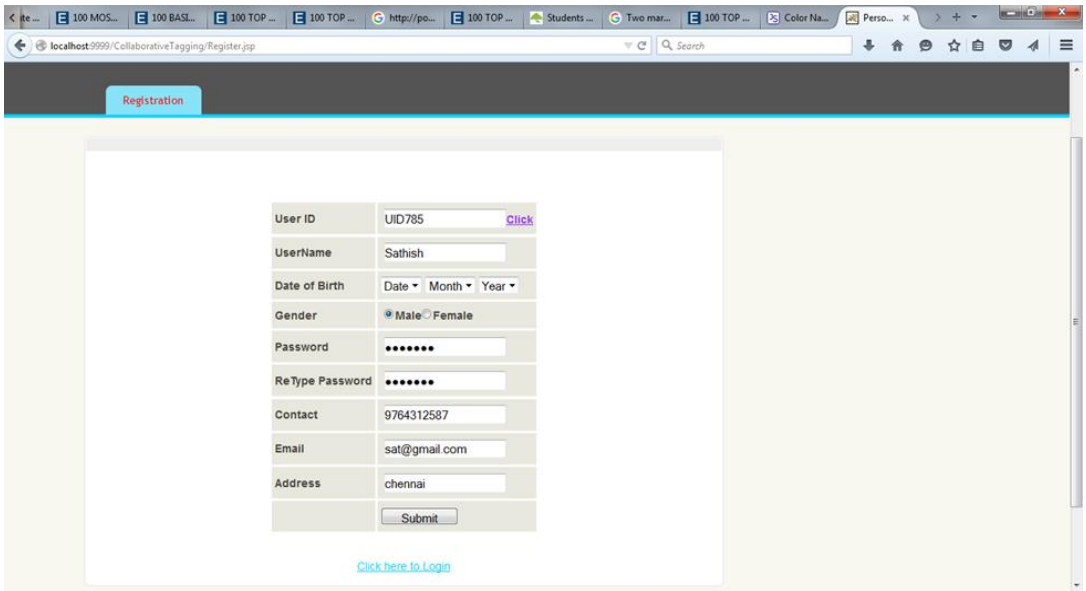

Fig 6.2 Screenshot for login page

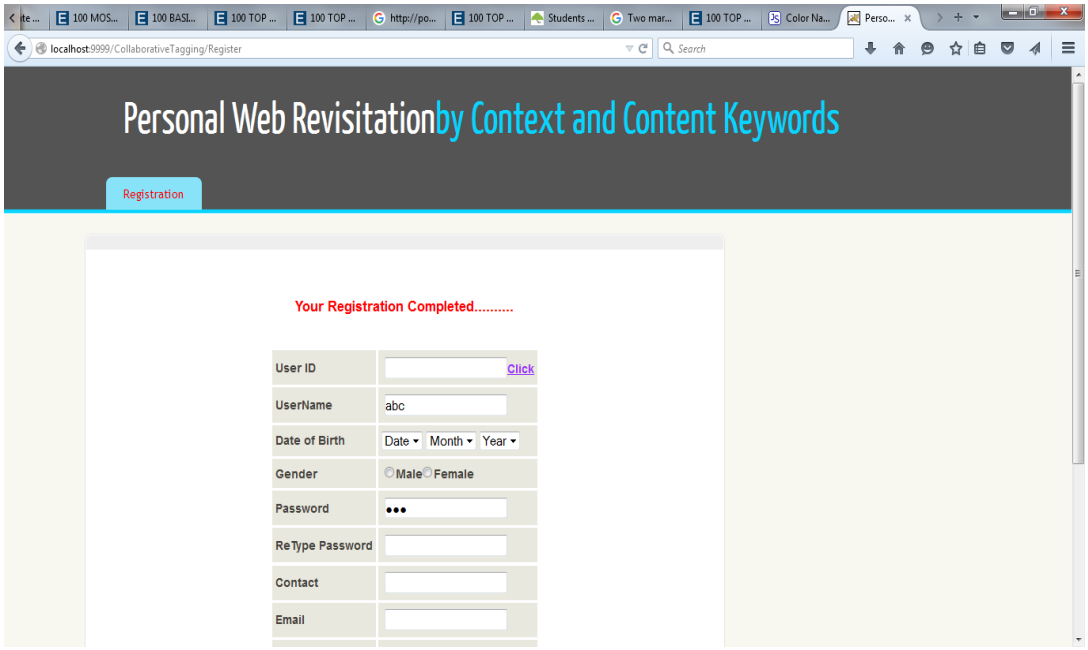

Fig 6.3 Screenshot for registration completed page

6.2 LOGIN ADMIN PROFILE

After your registration is completed group owner have to give their username and password to view their page. When the group owner login, they can view their profile.

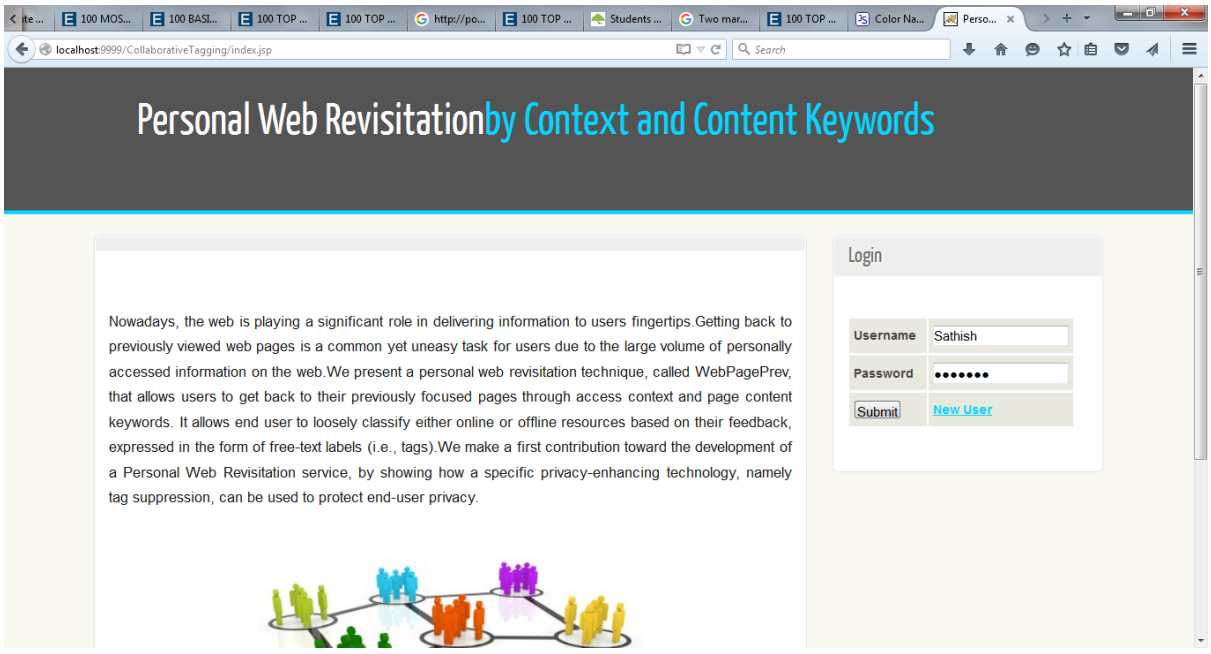

Fig 6.4 Screenshot for login page

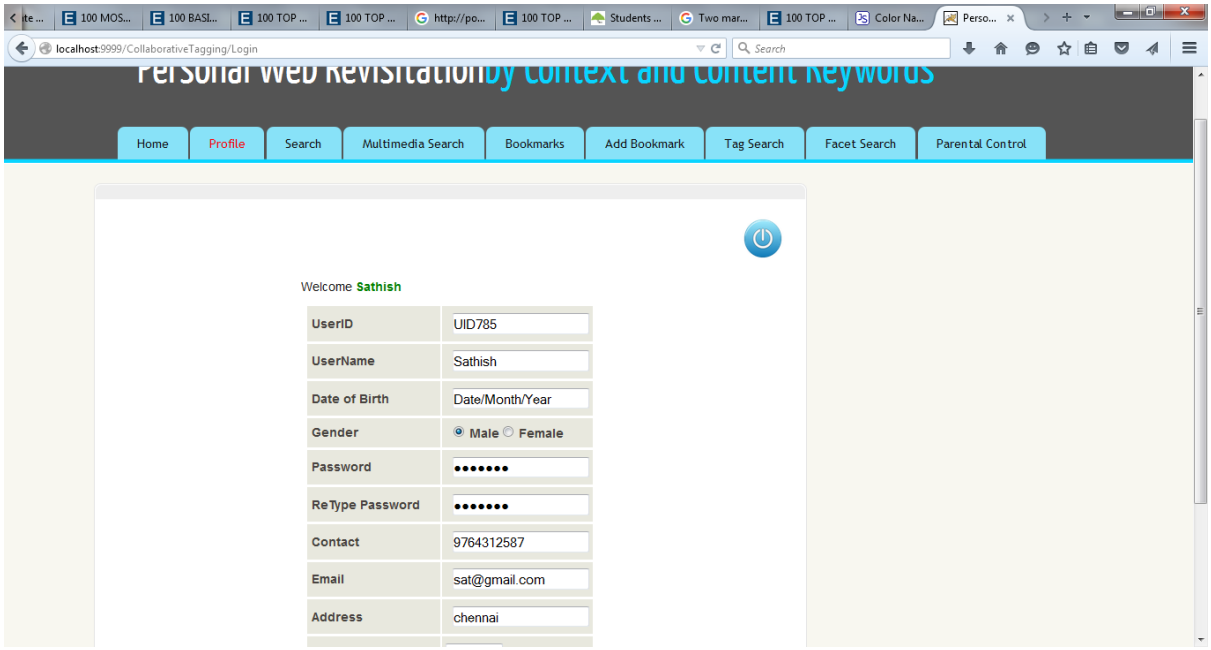

Fig 6.5 Screenshot for profile page

6.3 ADD USER

Here Group owner can add users. Group owner set username and password to all group users. Using this username and password, user can view group owner's profile, bookmarks etc.

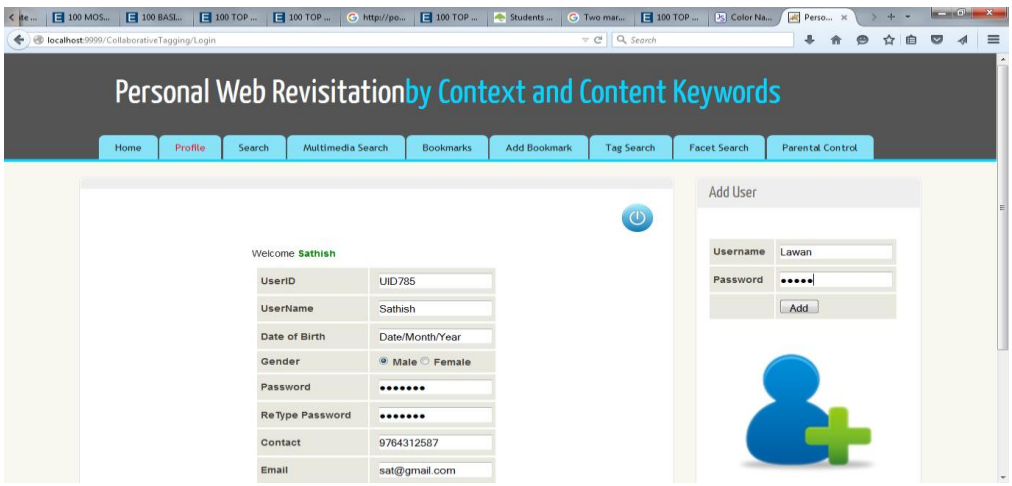

Fig 6.6 Screenshot for add user page

6.4 USER PROFILE

Group users register their details. At registration, group user has to provide username and password given by group owner. Group owner restricts users to view only specified contents.

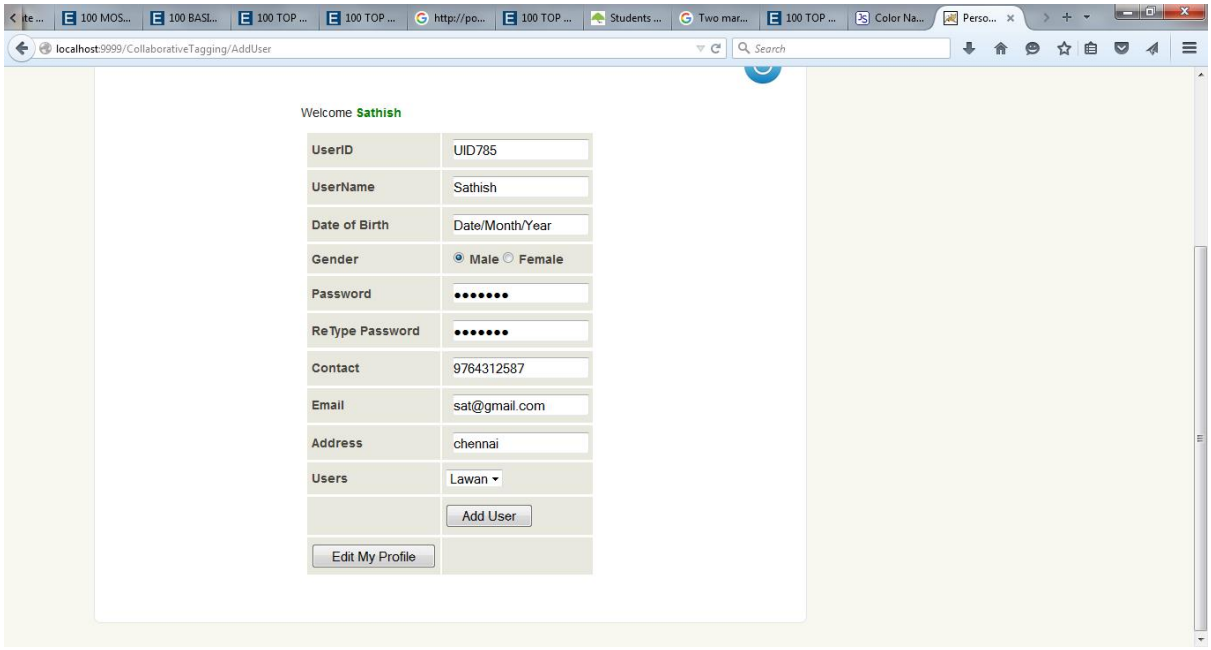

Fig 6.7 Screenshot for user profile page

6.5 ADD ANOTHER USER

Group owner can add multiple users in their profile. Users can also add another user in the group.

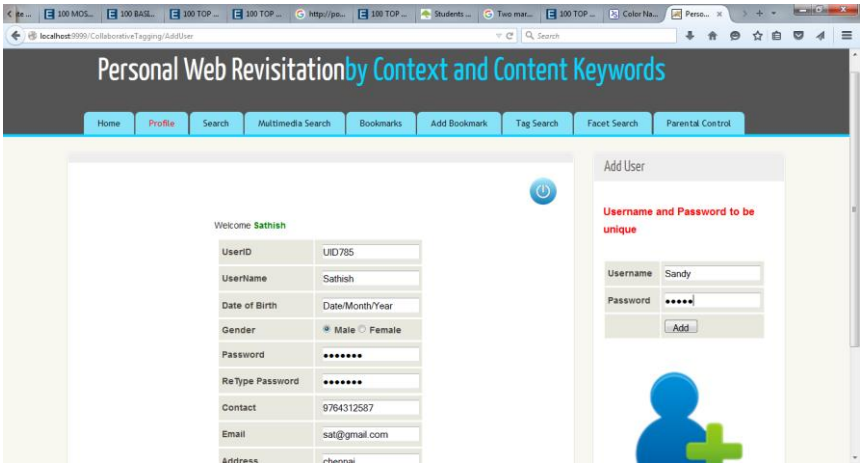

Fig 6.8 Screenshot for add user page

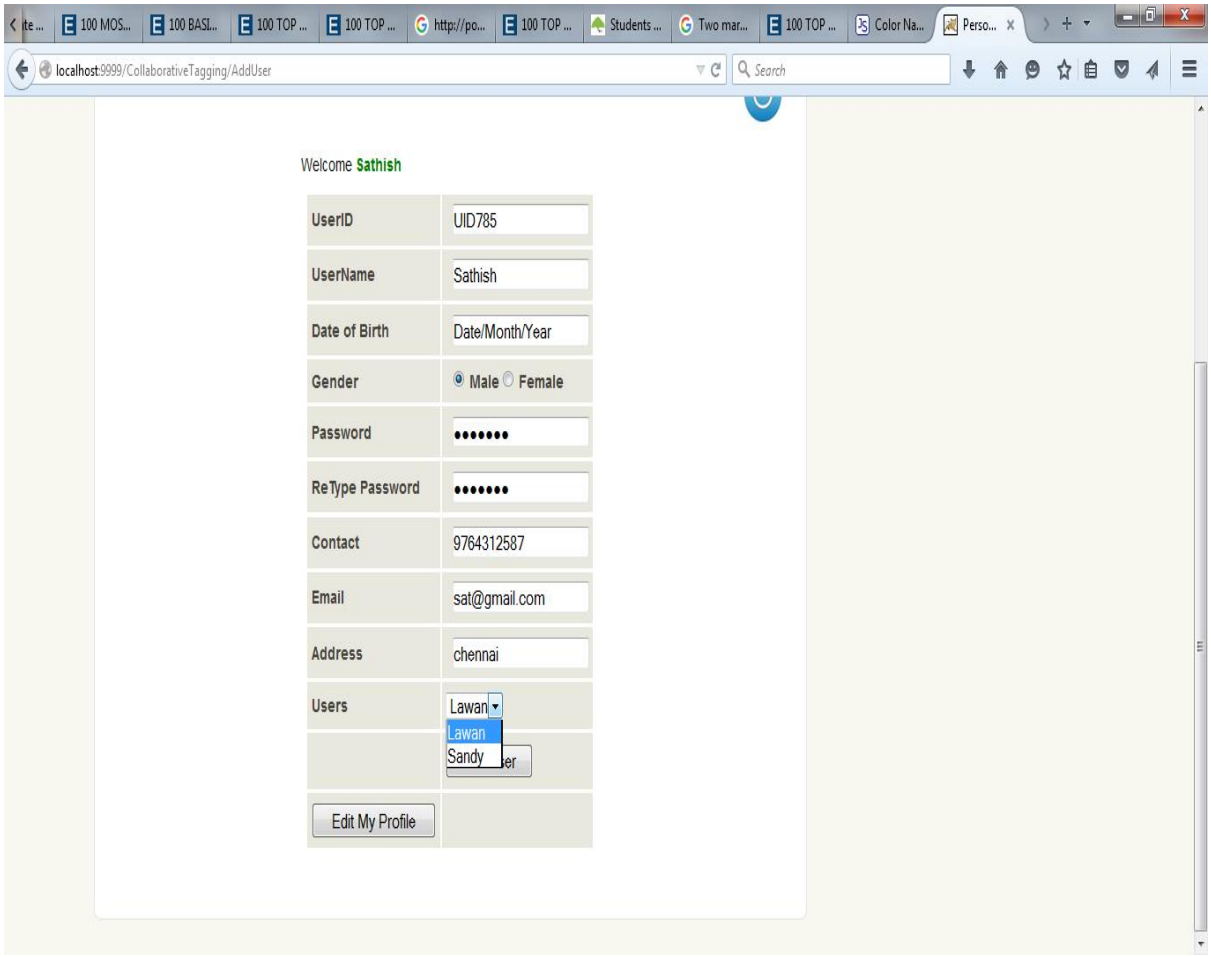

Fig 6.9 Screenshot for add another user page

6.6 SEARCH AND BOOKMARK

User can search resources in web according to their personal preferences. List of websites displayed where user can view his/her interested links. In search module provide a two categories of searching online searching and offline searching.

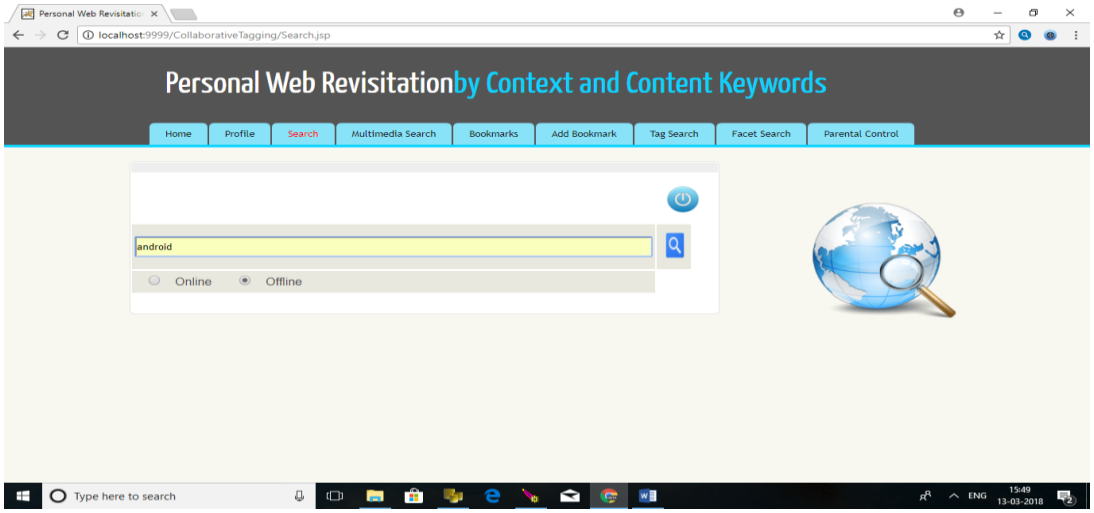

Fig 6.10 Screenshot for search page

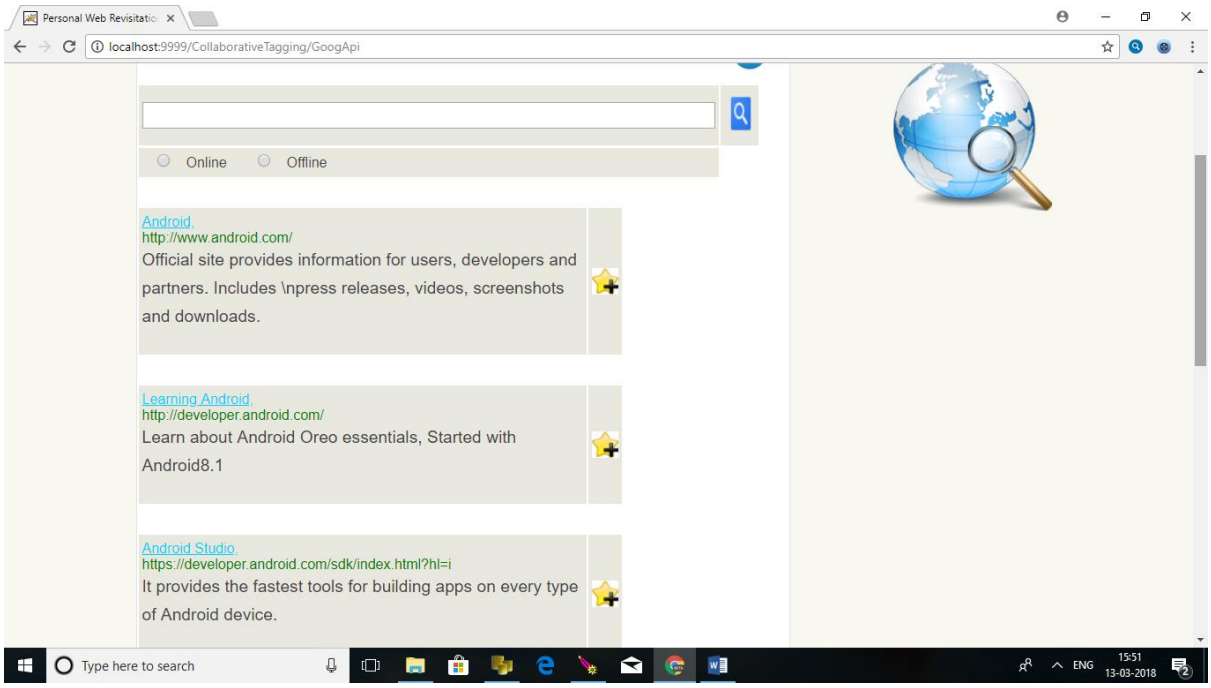

Fig 6.11 Screenshot for search and bookmark page

6.7 TAG SUPRESSION AND RECOMMENDATION

User likes a link in web and bookmarks that link. User tag the bookmark. While tagging, user can give own tag or ask server to suggest tags.

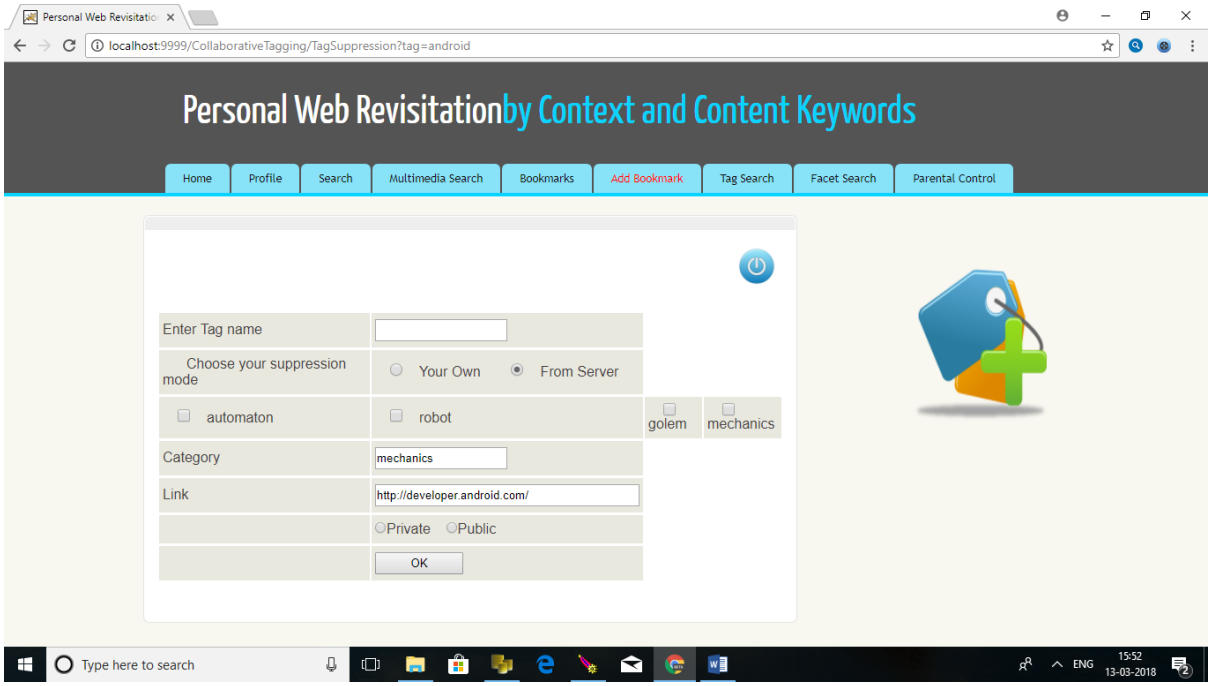

Fig 6.12 Screenshot for tag suppression page

User can give access privileges to bookmarks. If the bookmark is private, only the user can view. If the bookmark is public, other users can view their bookmarks.

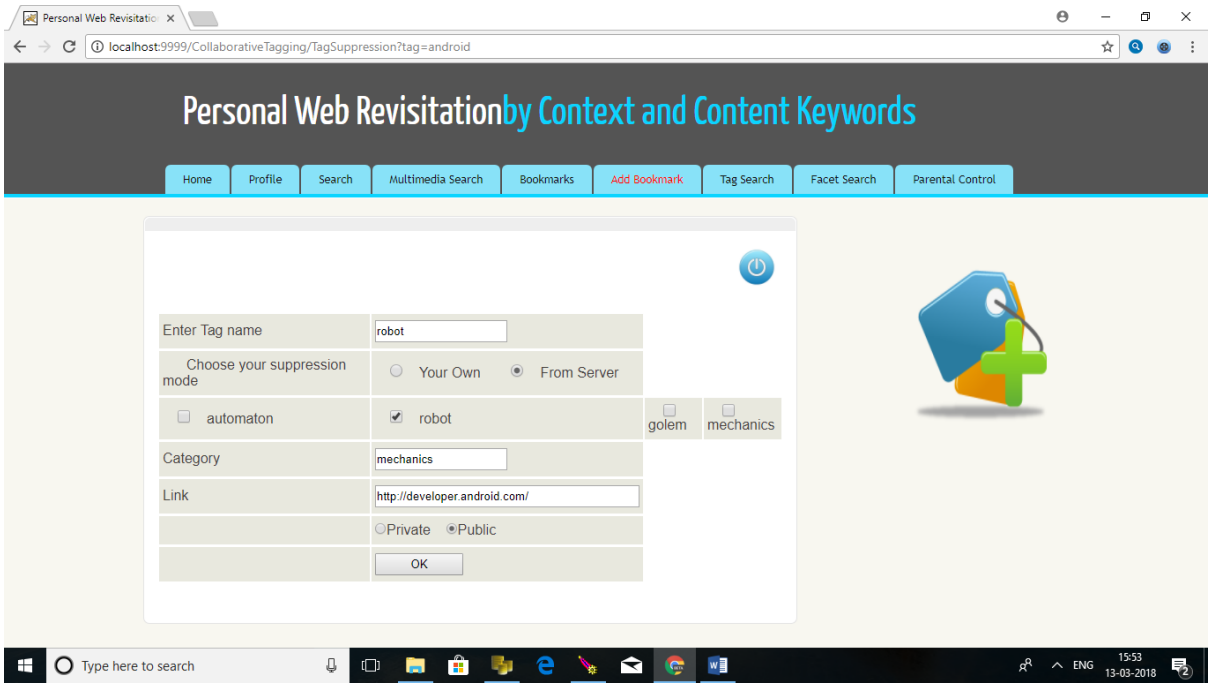

Fig 6.13 Screenshot for tag recommendation page

6.8 TAG SEARCH

All the book marking information will be stored in database. If the user searches a tag, he/she can search in their bookmarks or in all bookmarks. If the link has multiple tags, user searched tag and other tags for that links will be displayed.

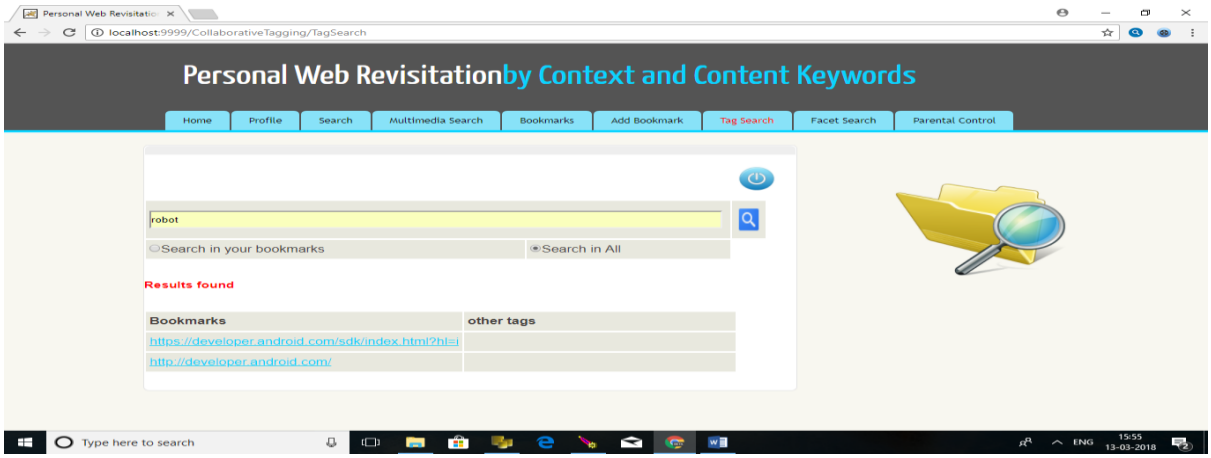

Fig 6.14 Screenshot for tag search page

6.9 FACET SEARCH

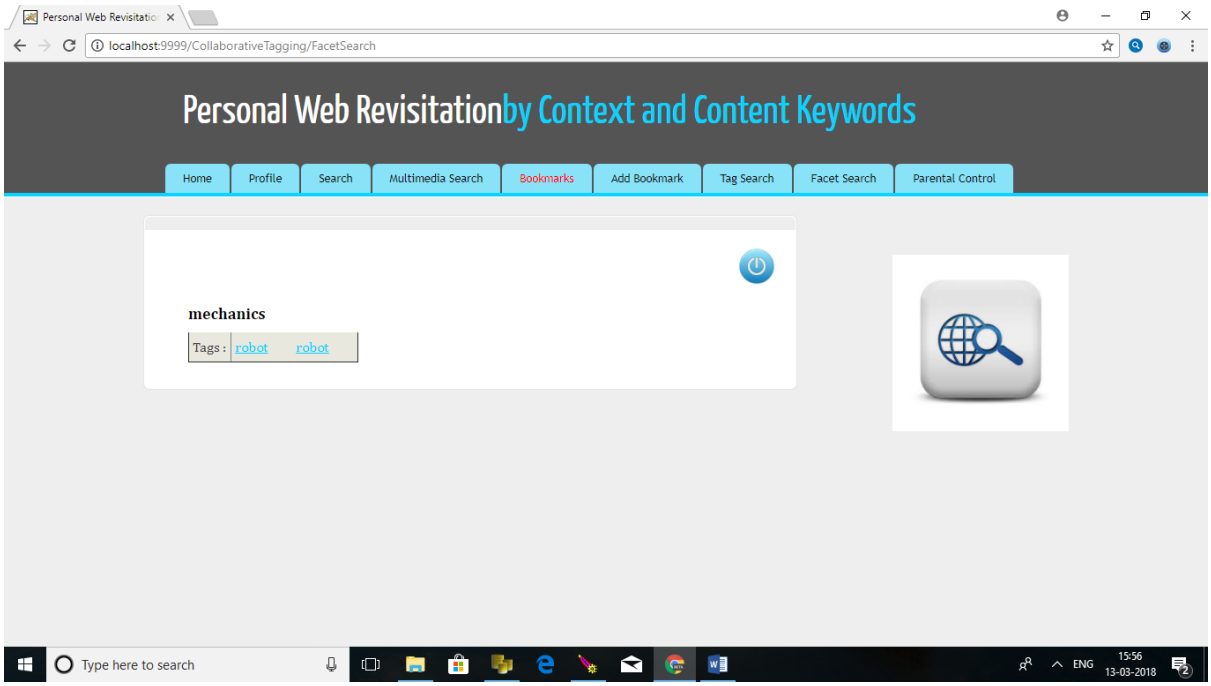

Fig 6.15 Screenshot for facet search page

6.10 PARENTAL CONTROL (BLOCKING)

Group owner denote which resources is un/safe. By checking the available tag categories, group owner blocks the tags for users.

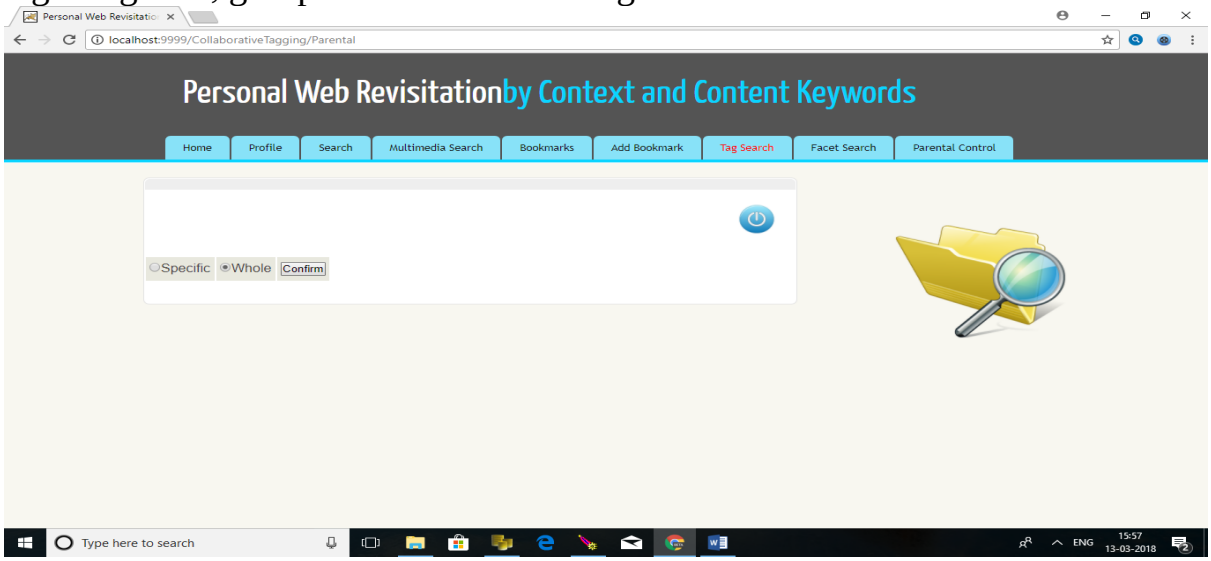

Fig 6.16 Screenshot for parental control page

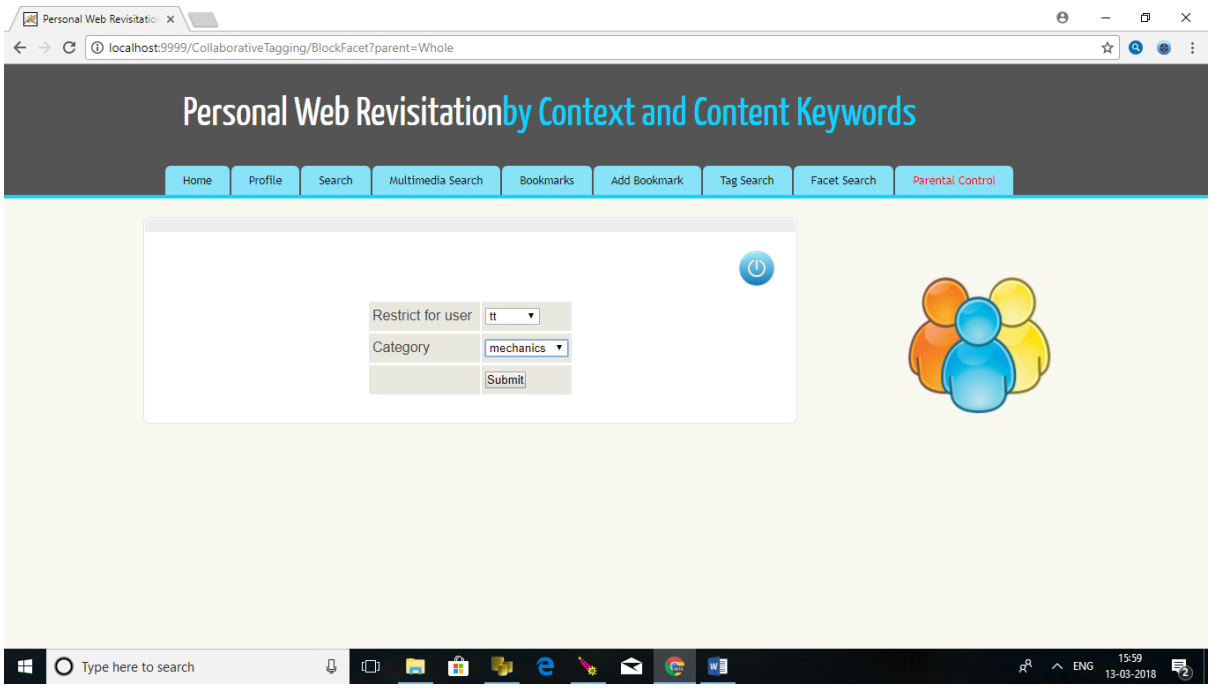

Fig 6.17 Screenshot for block page

- The category Mechanics will be blocked for the user tt.

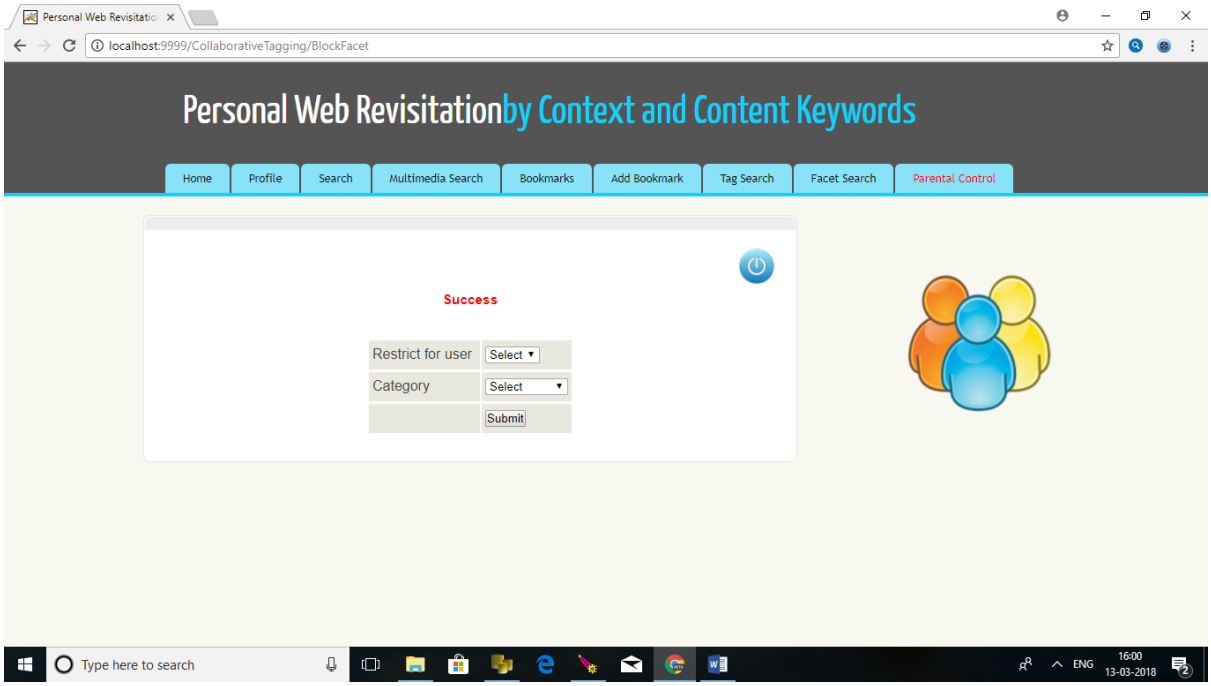

Fig 6.18 Screenshot for successful block page

- User login and view their profile.

Fig 6.19 Screenshot for new user login page

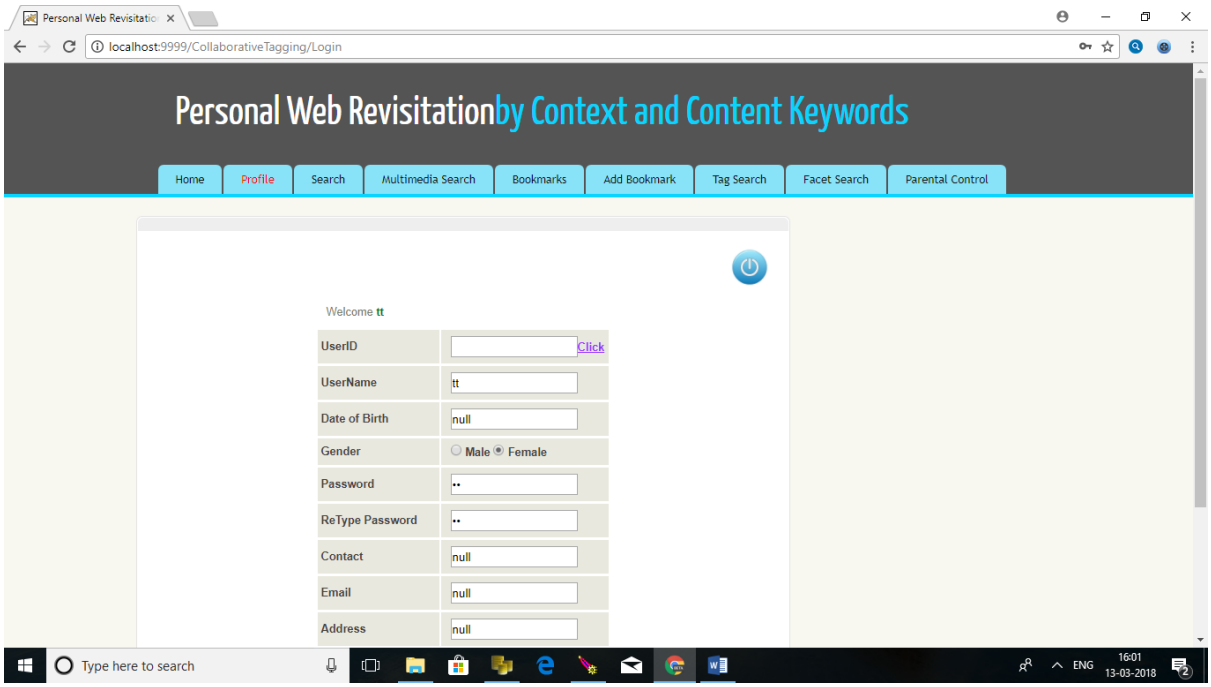

Fig 6.20 Screenshot for new user register page

- User search for a category Mechanics.This category did not shown for this user.

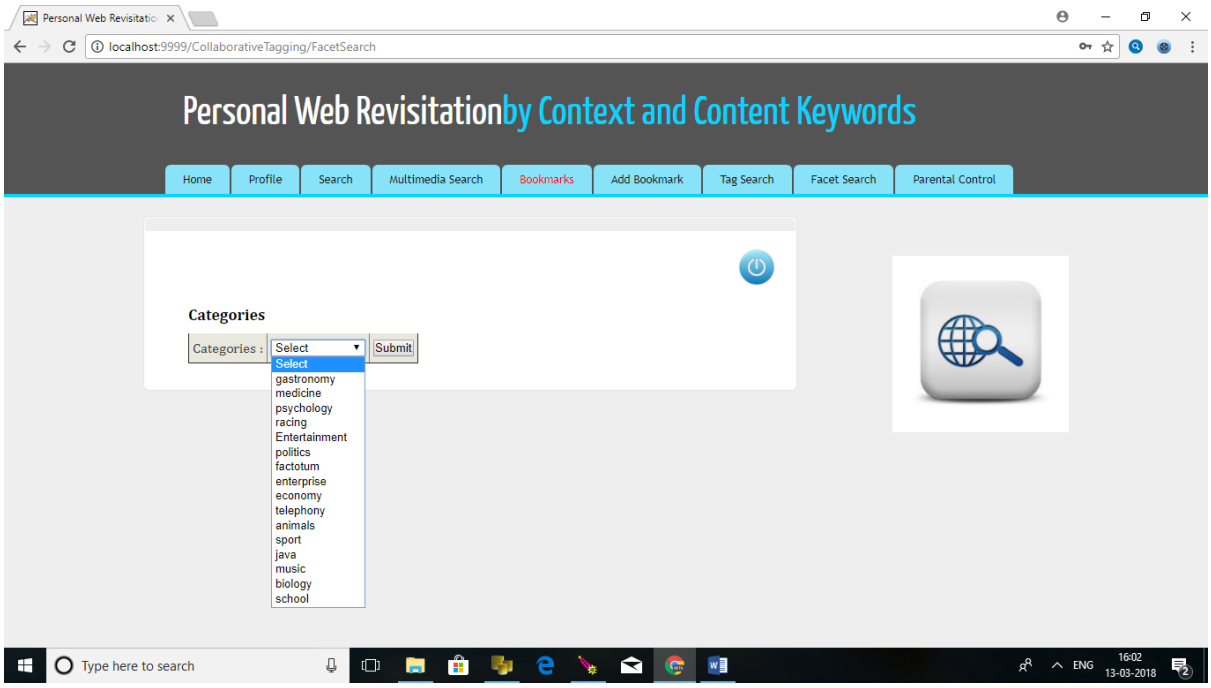

Fig 6.21 Screenshot for checking category

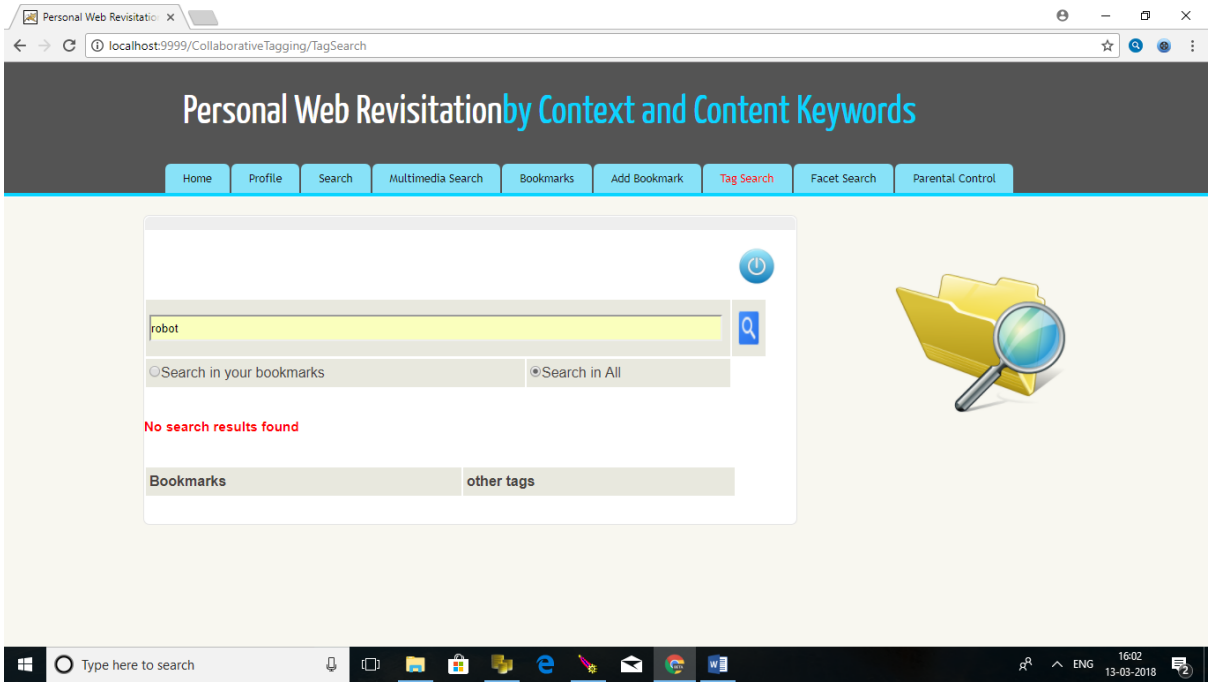

Fig 6.22 Screenshot for blocked category

6.3.1 TABLE FOR USER DETAILS

All the user details like their id, mail etc. are stored in a table in the database as shown in table 6.1.

Table 6.1 User details

User id	User name	Gender	Contact	Email	Address
1	abcd	Male	9098979695	abcd@gmail.com	Chennai
2	xyz	Female	9890979695	xyz@gmail.com	Vellore
3	hij	Male	8908786765	hij@gmail.com	Krishnagiri
4	mno	Female	7890654321	mno@gmail.com	Villupuram

6.3.2 TABLE FOR ADD USER DETAILS

Table 6.2 Add User details

User id	User name	Add Users	Private/Public Bookmark	Bookmarks Viewed by users
1	Abcd	Xyz	Private	abcd
2	Xyz	Hij	Public	abcd,xyz,hij,mno, pqr,stu
3	Hij	Mno	Public	abcd,xyz,hij,mno, pqr,stu
4	mno	Pqr,stu	private	mno

TESTING

AND

MAINTENANCE

CHAPTER 7

TESTING AND MAINTENANCE

7.1 TESTING

Implementation forms an important phase in the system development life cycle. It is a stage of the project work that transforms the design into a model. Testing was done to see if all the features provided in the modules are performing satisfactory and to ensure that the process of testing is as realistic as possible.

Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the program specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system is going to be implemented very soon.

Initially as a first step the executable form of the application is to be created and loaded in the common server machine which is accessible to all the users and the server is to be connected to a network. The final stage is to document the entire system which provides components and the operating procedures of the system.

The importance of software testing and its implementations with respect to software quality cannot be over emphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding.

Any product can be tested using either a black box testing or white box testing. Further testing can be implemented along the lines of code, integration and system testing.

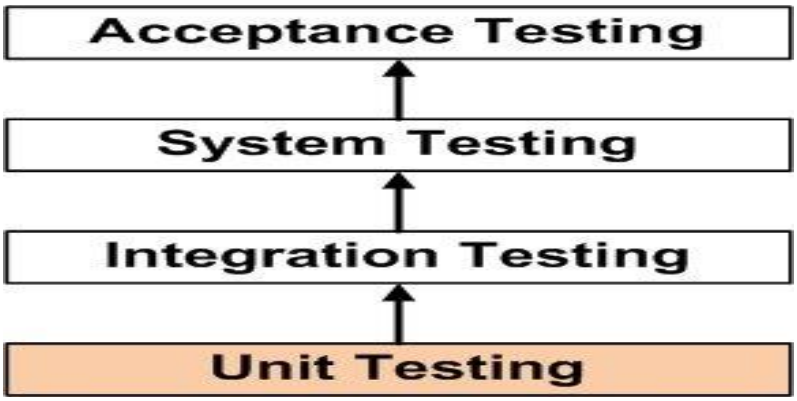

Figure 7.1 Levels of Testing

7.1.1 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that all gears mesh that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

7.1.2 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

7.1.3 UNIT TESTING

Unit testing focused on the smallest unit of software design. Using the procedural description as a guide, important control paths were tested to uncover errors within the boundary of the module. The module interface was tested to ensure proper informational flow in and out of the program unit under test. The local data structures were examined to ensure that the integrity of the temporarily stored data was maintained during all the steps of execution.

Unit Testing is a level of the software testing process where individual units or components of a software or system are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc.

In object-oriented programming, the smallest unit is a method, which may belong to a base or super class, abstract class or derived or child class. This is to be discouraged as there will probably be many individual units within that module.)Unit testing frameworks, drivers, stubs and mock or fake objects are used to assist in unit testing. Unit Testing is normally performed by software developers themselves or their peers. In rare cases it may also be performed by independent software testers.

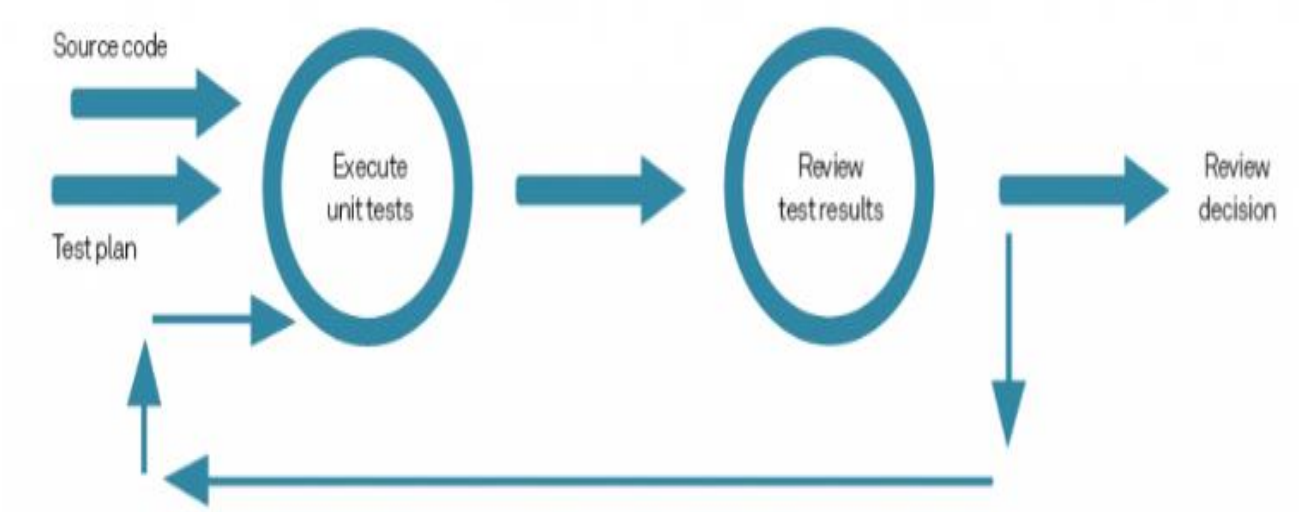

Figure 7.2 Unit testing

Tasks

- Unit Test Plan
  - Prepare
  - Review
  - Rework
  - Baseline
- Unit Test Cases/Scripts
  - Prepare
  - Review
  - Rework
  - Baseline
  - Unit Test
  - Perform

Benefits

- Unit testing increases confidence in changing/maintaining code. If good unit tests are written and if they are run every time any code is changed, the likelihood of any defects due to the change being promptly caught is very high.

- Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.
- Development is faster.
- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the cost (time, effort, destruction and humiliation) of a defect detected during acceptance testing or say when the software is live.
- Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span of several days, weeks or months need to be debugged.

Definition by ISTQB

- **Unit testing:** The testing of individual software components.

7.1.4 INTEGRATION TESTING

This is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interfacing. The objective was to consolidate all the unit tested modules and build a program structure that was dictated by design. Then tests were performed to ensure that all the modules interacted satisfactorily, while maintaining the integrity and accuracy of data during inter-module communication.

Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. The definition of a unit is debatable and it could mean any of the following:

1. The smallest testable part of a software
2. A ‘module’ which could consist of many of ’1′
3. A ‘component’ which could consist of many of ’2′

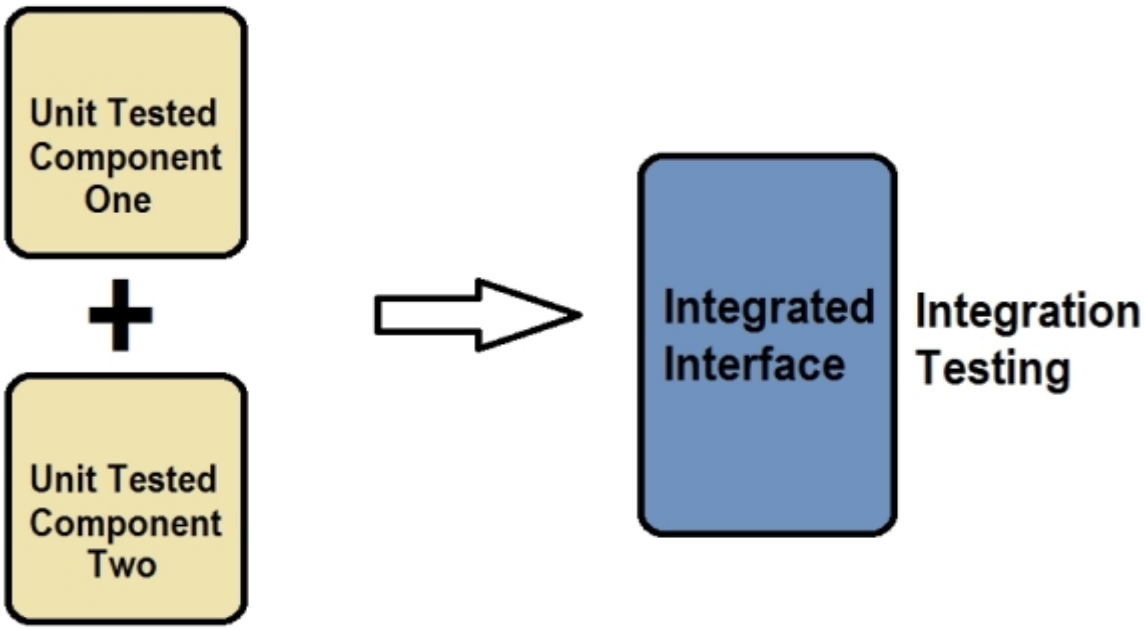

Figure 7.3 Integration Testing

Tasks

- Integration Test Plan
  - Prepare
  - Review
  - Rework
  - Baseline
- Integration Test Cases/Scripts
  - Prepare
  - Review
  - Rework
  - Baseline
- Integration Test
  - Perform

TEST DATA AND OUTPUT

FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

APPROACHES

- **Big bang** is an approach to Integration testing where all or most of the units are combined together and tested at one goes. This approach is taken when the testing team receives the entire software in a bundle. So, the difference between Big Bang Integration Testing and System Testing is that the former tests only the interactions between the units while the latter tests the entire system.
- **Top Down** is an approach to Integration Testing where top level units are tested first and lower level units are tested step by step after that. This approach is taken when top down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.
- **Bottom Up** is an approach to Integration Testing where bottom level units are tested first and upper level units step by step after that. This approach is taken when bottom up development approach is followed.

Test Drivers are needed to simulate higher level units which may not be available during the initial phases.

- **Sandwich or Hybrid** is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

Definition by ISTQB

- **Integration testing:** Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also c*omponent integration testing, system integration testing*.
- **Component integration testing:** Testing performed to expose defects in the interface and interaction between integrated components.
- **System integration testing:** Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

7.2 TEST CASES:

Table 7.2 Test cases

Test Case ID	Test Item	Input Specification		Output Specification	Pass/Fail
		Action	Case Description		
TC_01	Home Page	OnLoad	On the time of load	Home page must have: 1.Login 2.Register	Pass
TC_02	Login Page	OnClick	When the user clicks on the UI element	Login page must have: 1.Username, 2. Password. 3. Submit button.	Pass
TC_03	Login Page	OnSuccess	On the success event of login	Redirect to the admin profile search page.	Pass
TC_04	Login Page	OnFailure	When the login is failure it should display a failure notice	A server response with “invalid data”	Pass
TC_05	Add User	OnClick	When the user clicks on the UI element	The User must be added to the group	Pass
TC_06	Add User	OnFailure	When submit is failure it should display a notice	A server response “not available”	Pass

TC_07	Search Bookmark	OnSuccess	When submit is Success "Result found"(offline search)	The link is added to the bookmark	Pass
TC_08	Search Bookmark	OnFailure	When submit is failure, it is not available in offline search.	A server response "not available".	Pass
TC_09	Tag suppression	OnSuccess	When the user request the server for tag name	Server suggested the tag name for a requested category.	Pass
TC_10	Tag suppression	OnFailure	When submit is failure it should display a failure notice	Server response with "server not respond".	Pass
TC_11	Parental Control	OnSuccess	Group admin has to block some content for a particular user.	When blocking is done. User wants to Search the blocked content,it shows the "result not found".	Pass
TC_12	Parental Control	OnFailure	When submit is failure it should display a failure notice	When blocking is undone. User wants to Search the blocked content,it shows the "Result found".	Pass

SAMPLE TEST CASES:

Login Page

Test case 1

Action: when the user login his invalid username and password

Output: invalid user message pops

Code:

```
package login;

import static org.junit.Assert.*;

import org.junit.Test;

public class Pro1 {

public void test() {

AdminLogin test=new AdminLogin ();

int out= test.AdminLogin();

assertEquals(c3,out);
```

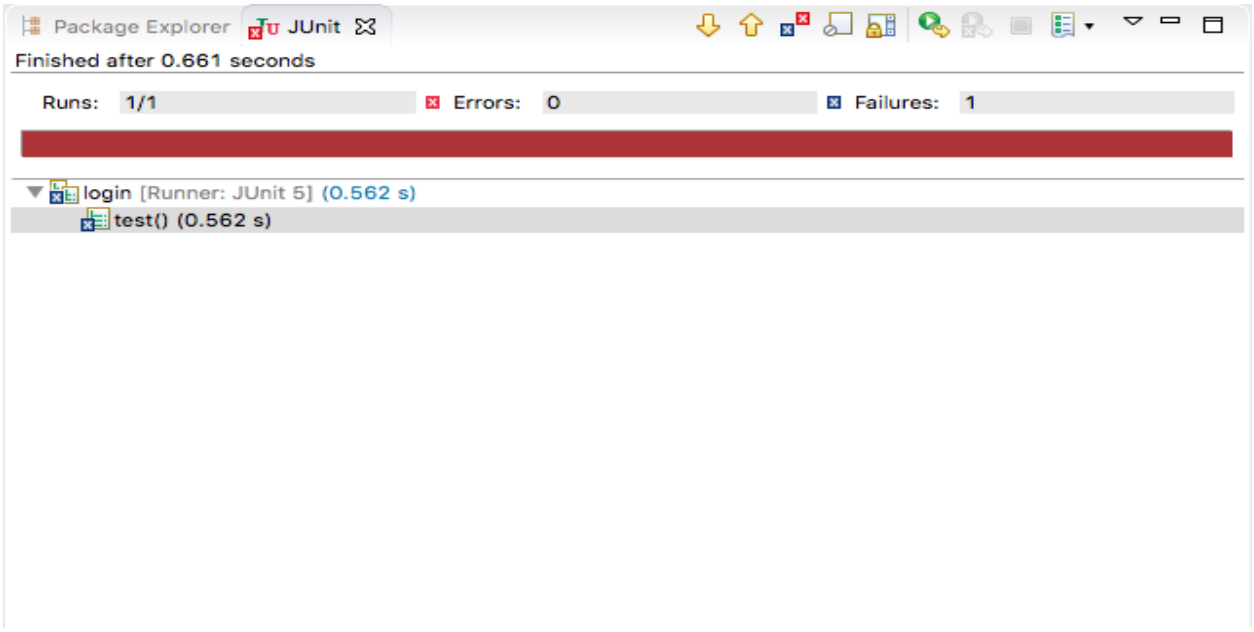

Fig 7.4 Screenshot for test case failure

Login Code:

```
package login;
import javax.swing.JOptionPane;
public class AdminLogin extends javax.swing.JFrame {
private Object JOptionpane;
int c;
public int AdminLogin() {
{
String username = new String(&amp;quot;raja&amp;quot;);
String password = new String(&amp;quot;y&amp;quot;);
if ((username.equals(&amp;quot;raja&amp;quot;)) || (password.equals(&amp;quot;y&amp;quot;)))
{
c=2;
}
else
{
c=3;
}
return c;
}
}
```

Test case 2

Login page

Action: when the user enters valid username and password

Output: login successful message pops

Code:

```
package login;

import static org.junit.Assert.*;

import org.junit.Test;

public class Pro1 {

public void test() {

AdminLogin test=new AdminLogin ();

int out= test.AdminLogin();

assertEquals(c3,out);
```

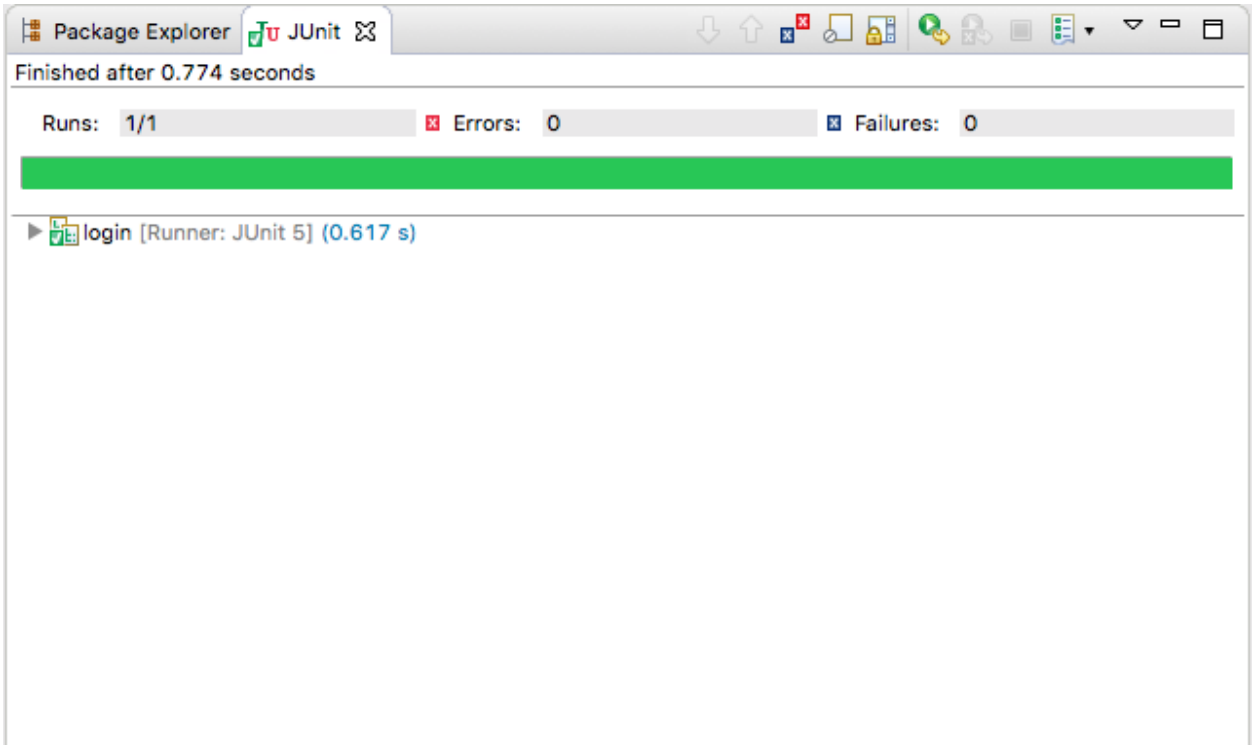

Fig 7.5 Screenshot for test case success

7.3 MAINTENANCE

After a software system has been verified, tested and implemented, it must continue to be maintained. Maintenance routines will vary depending on the type and complexity of the technology. Many software systems will come with a maintenance schedule or program recommended by the developer. Maintenance could be provided by the developer as part of the purchase agreement for the technology.

Systems will need to be maintained to ensure that they continue to perform to the level demonstrated during the system testing stage. If systems deteriorate, there is a risk that the systems will not perform to the required standard.

Ongoing monitoring or testing systems may be installed to ensure that maintenance needs are identified and met where necessary. Where systems are in long-term use, a system can be designed to monitor feedback from users and conduct any modifications or maintenance as needed. Where modifications to software are made as a result of system maintenance or upgrades, it may be necessary to instigate further rounds of system verification and testing to ensure that standards are still met by the modified system.

CONCLUSION AND FUTURE ENHANCEMENTS

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 CONCLUSION

In collaborative tagging, users tag resources on the web according to their personal preferences. Users therefore contribute to describe and classify those resources but this is inevitably at the expense of revealing their profile. This helps attacker to view users profile and collect user's information. Collaborative tagging is currently an extremely useful for online services. Although the collaborative tagging is mainly used to support tag-based resource discovery and browsing, it could be exploited for other purposes. One of these potential applications is the provision of web access functionalities such as content filtering and discovery.

8.2 FUTURE ENHANCEMENT

Two key scenarios, parental control and resource recommendation. Since we are not aware of similar experimental studies, we believe that what reported in this paper can be useful to evaluate further future development in the area. Future work includes the development of a full prototype for the experimented system and its testing and use in further scenarios.

REFERENCES

CHAPTER 9

REFRENCES

[1] A. Cockburn, S. Greenberg, S. Jones, B. Mckenzie, and M. Moyle. Improving web page revisitation: analysis, design and evaluation. IT & Society, 1(3):159–183, 2003.

[2] L. Tauscher and S. Greenberg. How people revisit web pages: empirical findings and implications for the design of history systems. International Journal of Human Computer Studies, 47(1):97– 137, 1997.

[3] J. Teevan, E. Adar, R. Jones, and M. Potts. Information re-retrieval: repeat queries in yahoo's logs. In SIGIR, pages 151–158, 2007.

[4] M. Mayer. Web history tools and revisitation support: a survey of existing approaches and directions. Foundations and Trends in HCI, 2(3):173–278, 2009.

[5] L. C. Wiggs, J. Weisberg, and A. Martin. Neural correlates of semantic and episodic memory retrieval. Neuropsychologia, pages 103–118, 1999.

[6] M. Lamming and M. Flynn. "forget-me-not": intimate computing in support of human memory. In FRIEND21 Intl. Symposium on Next Generation Human Interface, 1994.

[7] E. Tulving. What is episodic memory? Current Directions in Psychological Science, 2(3):67–70, 1993.

[8] C. E. Kulkarni, S. Raju, and R. Udupa. Memento: unifying content and context to aid webpage re-visitation. In UIST, pages 435–436, 2010.

[9] J. Hailpern, N. Jitkoff, A. Warr, K. Karahalios, R. Sesek, and N. Shkrob. Youpivot: improving recall with contextual search. In CHI, pages 1521–1530, 2011.

[10] T. Deng, L. Zhao, H. Wang, Q. Liu, and L. Feng. Refinder: a context-based information re-finding system. IEEE TKDE, 25(9):2119–2132, 2013.

[11] T. Deng, L. Zhao, and L. Feng. Enhancing web revisitation by contextual keywords. In ICWE, pages 323–337, 2013.

[12] H. Takano and T. Winograd. Dynamic bookmarks for the WWW. In HYPERTEXT, pages 297–298, 1998.

[13] S. Kaasten and S. Greenberg. Integrating back, history and bookmarks in web browsers. In HCI, pages 379–380, 2001.

[14] J. A. Gamez, J. L. Mateo, and J. M. Puerta. Improving revisitation browsers capability by using a dynamic bookmarks personal toolbar. In WISE, pages 643–652, 2007.

[15] R. Kawase, G. Papadakis, E. Herder, and W. Nejdl. Beyond the usual suspects: context-aware revisitation support. In HT, pages 27–36, 2011.

APPENDIX

CHAPTER 10

APPENDIX

(PUBLICATION DETAILS)

Paper Title	PRIVACY PRESERVING AND COLLABORATIVE TAGGING WITH FACET CONTROL
Authors	Mrs.S.Uma, G.Aswini, M.Jeyanthi, J.B.Shobana, G.Srisanthanalakshmi
Journal Name	IJRASET
Edition	Volume 6,issue III, March 2018
Month and Year	March 2018