Software Requirements Specification

for

Dexter

Prepared by

Vaibhav Bajpai, Rahul Burman, Nupur Dixit, Yadavendra

Galgotias College of Engineering and Technology

October 9, 2008

Dexter

making search Web 2.0 ready!

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the document is to describe the system requirements of the project 'Dexter'. he document contains sufficient detail in the functional system requirements so that a full-fledged design solution can be devised. The functional level requirements are specified using the Use Case Model approach.

The document would place borders around the problem, solidify ideas, and help break down the problem into its component parts in an orderly fashion.

The document would also serve as the parent document for testing and validation strategies that will be applied to the requirements for verification.

Dexter is a product which allows user to search the web. It differs from existing product in the market like google, yahoo by taking an alternative methodology to search by taking in account the human opinion.

1.2 Document Conventions

The terms included in single quotation mark signify terms having special meaning in context of 'Dexter' and their corresponding meaning have been explained in Glossary.

1.3 Intended Audience and Reading Suggestions

The intended audience of this document is people interested in understanding the basic requirement and functionality of 'Dexter'.

Dexter

making search Web 2.0 ready!

1.4 Product Scope

Objective - to blend social networking into the google search engine and encapsulate it into a Web 2.0 browsing environment.

Goals -

- The search results should be ordered in a way that they give preference to readers opinion over algorithmic or machine based approaches.
- The non submitted search results should show up in the same order as that by the page rank algorithm
- The search interface should be as simplistic as the google search engine homepage.
- The user should be able to separately 'submit' a url to the Dexter database.
- Any url ever 'voted up' or submitted should be logged in for future reference by the user.
- The user should be able to 'favorite' the voted results, and be able to tag them.
- User should be able to add 'friends'.
- The user should be able to follow the activity of his 'friends' and be able to 'shout'/'email' them of an interesting content.
- The privacy of the user profile should be maintained.
- User should be able to choose one of the two interfaces as default home ('Surf' /'Search')
- Mechanism should exist to weed out 'spam' and 'inappropriate' url and comment

Benefits -

- Dexter implements the social search paradigm where relevance of search results is determined by considering the interactions or contributions of users, in contrast to established algorithmic or machine-based approaches where relevance is determined by analyzing the text of each document or the link structure of the documents
- Dexter would enable users to vote the search results and comment on them, still preserving the current page rank algorithm
- Dexter would provide a full-fledged social bookmarking using tags on top of google search results and save 'em in the cloud.
- Dexter would also include another interface "Dexter Surf", that would "push" the top voted content in a specific category or a web service(like youtube) for a desired timeframe in the homepage itself, providing "information content" as soon as one logs in.
- Dexter would maintain profile pages and provide ability to make friends whose activities could be followed, thereby increasing collaboration
- Reduces impact of link spam on search result by relying less on link structure of web pages
- Increased relevance because each result has been selected by users
- leverage a network of trusted individuals by providing an indication of whether they thought a particular result was good or bad
- The introduction of 'human judgement' suggests that each web page has been viewed and endorsed by one or more people, and they have concluded it is relevant and worthy of being shared with others using human techniques that go beyond the computer's current ability to analyze a web page.
- Web pages are considered to be relevant from the reader's perspective, rather than the author who desires their content to be viewed, or the web master as they create links.
- As 'Dexter' would be constantly getting feedback it is potentially able to display results that are more current or in context with changing information

1.5 References

Software Requirements Specifications – Wikipedia

Dexter

making search Web 2.0 ready!

2. Overall Description

2.1 Product Perspective

Dexter is assumed to be a replacement for current conventional Web 1.0 search engines, that are based upon an algorithm that decides the value/order of a search result URL based upon a fixed criteria *(link recommendation as used in Google or keyword based search as used in Yahoo!)*.

These search engines do not involve the human element in deciding the effectiveness of a search result URL which seems inadequate in today's Web 2.0 scenario.

The philosophy behind “Dexter” is that humans are the best judge of a pages relevance and its their opinion which is given a higher priority over algorithmic approaches.

2.2 Product Functions

- The user should be able to search the web without having to leave 'Dexter'
- The user should be able to search without being logged in, though should not be allowed to vote or comment.
- The user should be able to separately submit a url to the Dexter database.
- The user should be able to bookmark the voted results, and be able to tag them.
- The user should be able to see its recent activity as in comments and votes.
- The user should be able to follow the activity of its friends and be able to shout/email them of an interesting content.
- User should be able to choose one of the two interfaces as default home ('Surf' / 'Classic')
- User should be able to add dexter friends from google contacts.
- User can view the article that have been endorsed maximum no of times in various 'category', 'channel' or 'timeframe'
- Categories and channel can be added/removed dynamically.
- Any url/comment can be reported by the user after the user has 'voted it down' and moderators would take decision on the reported url and comment.

2.3 User Classes and Characteristics

Users

Dexter treats every “user” as equal invariant of one's technical expertise, educational level, or experience to reflect transparency in the sanctity of the votes.

Moderators

will have the privilege to

- delete a submitted url based on profanity of content
- delete a submitted comment based on profanity of content
- change the category of submitted url based on reports received.

Administrators

- have privilege to add/remove categories and order them as would appear on the web-page.
- have privilege to add/remove channel and order them as would appear on the web-page.
- have privilege to ban user
- have privilege to add/remove IM service name offered to user to supply their IM details
- have privilege to add/remove a visibility control level
- Assign moderator privilege to users.

2.4 Operating Environment

Client-Side Requirements

Hardware Platform -

Dexter, being a web-application only restricts the hardware requirements for the client to the capability of a machine being able to connect to the internet or the LAN in which the web application is deployed to

Operating System -

Dexter, being a web-application would allow the client side interface to run on all Operating Systems that have an inbuilt network stack incorporated into the kernel.

Software Platform -

Browser – IE 6+, Firefox 2+, Safari, Opera, Chrome

Server-Side Requirements

Hardware Platform -

- at-least 256 MB RAM
- 10 GB HDD space

Operating System -

- 32-bit Linux/Windows Operating System
- 2 GB Page File/Swap

Software Platforms -

- Java SE 6
- Java EE 5 compatible Application Server (Glass fish)
- Apache Web Server
- Struts and Hibernate libraries.

2.4.1 Design and Implementation Constraints

Paradigm Constraints

- As users can directly add results to a social search engine there is a risk that some users could insert search spam directly into the search engine. Elimination or prevention of this spam would require the ability to detect the validity of a users' contribution, such as whether it agrees with other trusted users.
- There are so many unique searches conducted that most searches, while valid, are performed very infrequently. A search engine that relied on users filling in all the searches would be at a disadvantage to one that used machines to crawl and index the entire web.
- The application would not be much useful to users until it acquires a reasonable user base.

Hardware/Software Limitations

- Oracle 10XE used in the project is not available for Mac OSX
- Java SE 6 is not available for 32-bit hardware running Mac OSX

Software Constraints

- Explicit usage of Free and/or Open Source Software
- Explicit Avoidance of Bloated Packages (Netbeans)

Design Constraints:

- No Explicit Style code in HTML
- No Flash Content
- No Frames
- No Tables for Layout Design

Coding Constraints:

- No Business Logic in Presentation Layer
- Separation of Form Beans and Data Beans
- Security Layer in Authentication
- Resolving the Object-Relational Mismatch
- Explicit HTML Coding using HTML Editor than using RAD tools

2.5 User Documentation

- Java Docs for the source.
- About/FAQ in the Web-application website.
- Video Screen-cast explaining the features/usage.

2.6 Assumptions and Dependencies

- Dexter being an overlay on top of the search engine, assumes the 24x7 functioning of the search engine in itself.

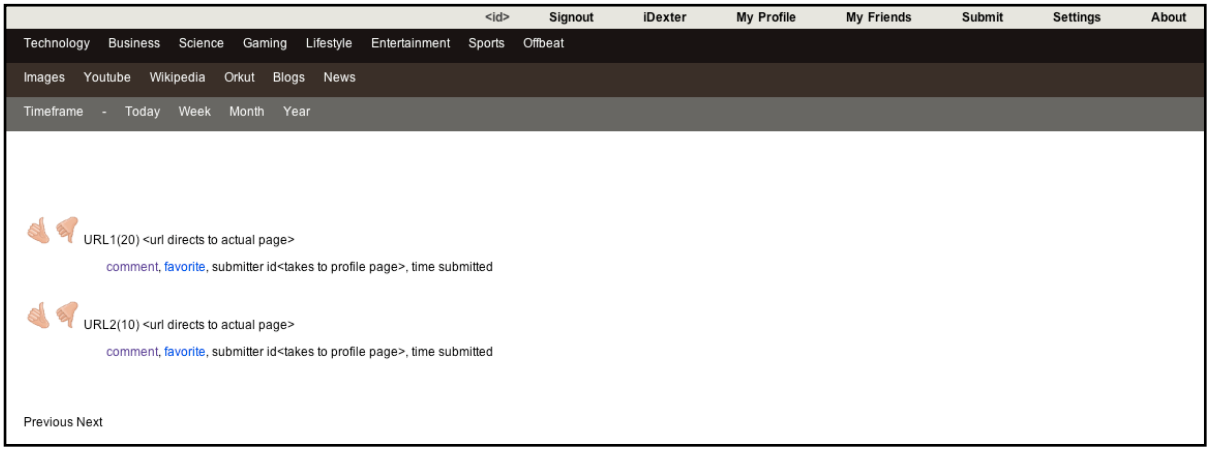
- Dexter uses Google AJAX Web Search API to retrieve the search results pushed against a query, and assumes the API would remain open to public.

- Dexter uses Google Account Authentication API to retrieve contact details of a user, and assumes the API would remain open to public.

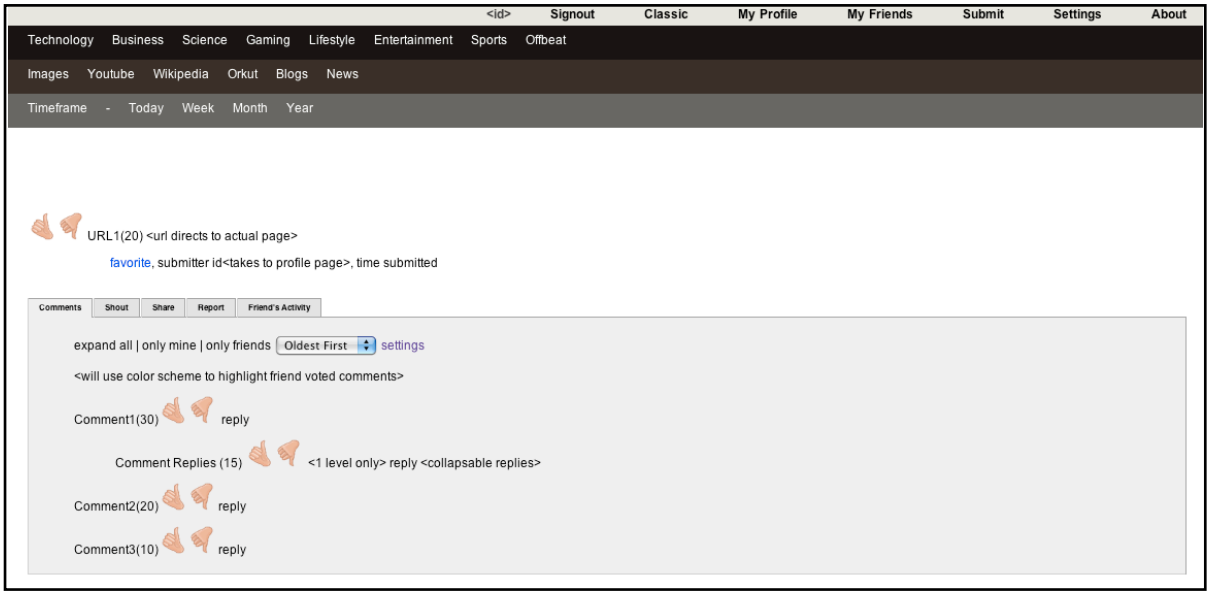
3. External Interface Requirements

3.1 User Interfaces

Search Results Page – Higher voted results on top.



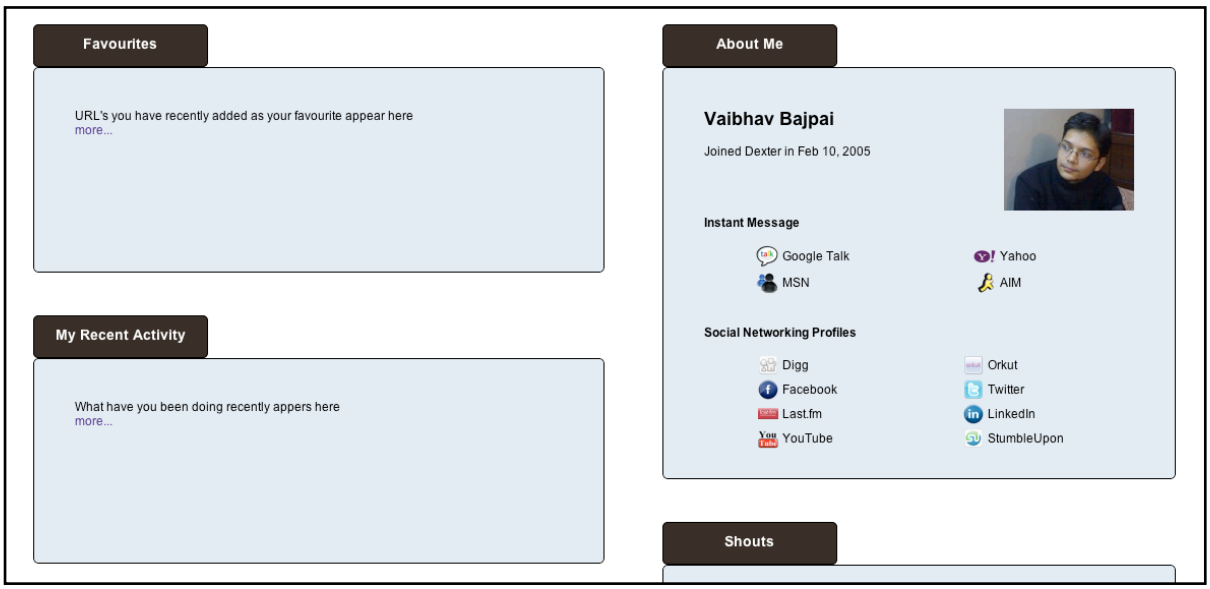
Action specific to a particular URL



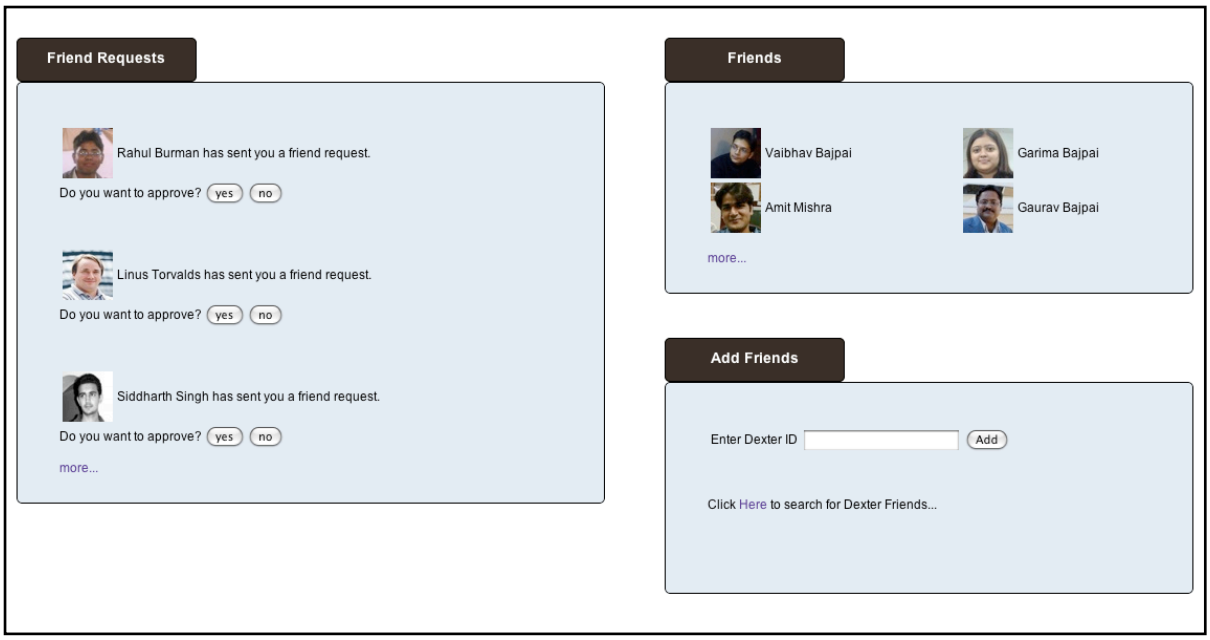
Dexter

making search Web 2.0 ready!

User Profile Page



User's Friends Page



3.2 Hardware Interfaces

Communication Protocols

- HTTP 1.1
- SSL for Login Authentication.

Minimum Number of Machines

- one or more - Database Storage
- one - Application Server Deployment (Server)
- one or more - Client

3.3 Software Interfaces

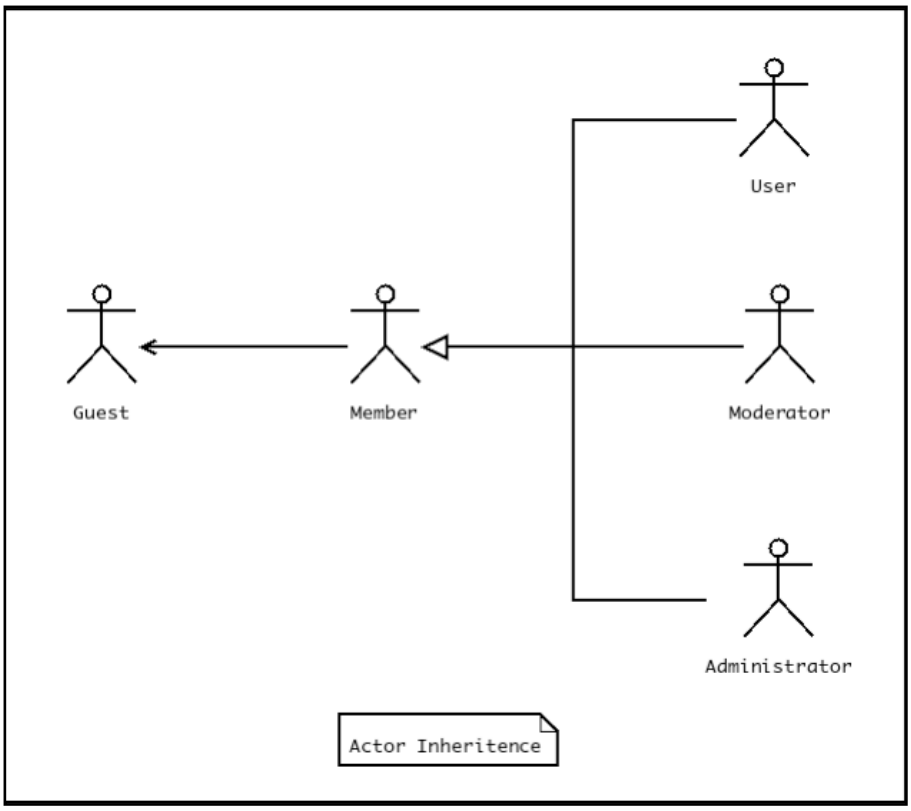
Database will be distributed across machines, the recordset would be retrieved by the application server deployed at another machine, where the the result sets would map to EJB modules inside the EJB container which will be pushed to the Web Container for the servlets to work upon and build presentation level java beans for the JSP to invoke and display the results as standard HTML page, which would be sent to the client upon a GET/POST request.

3.4 Communications Interfaces

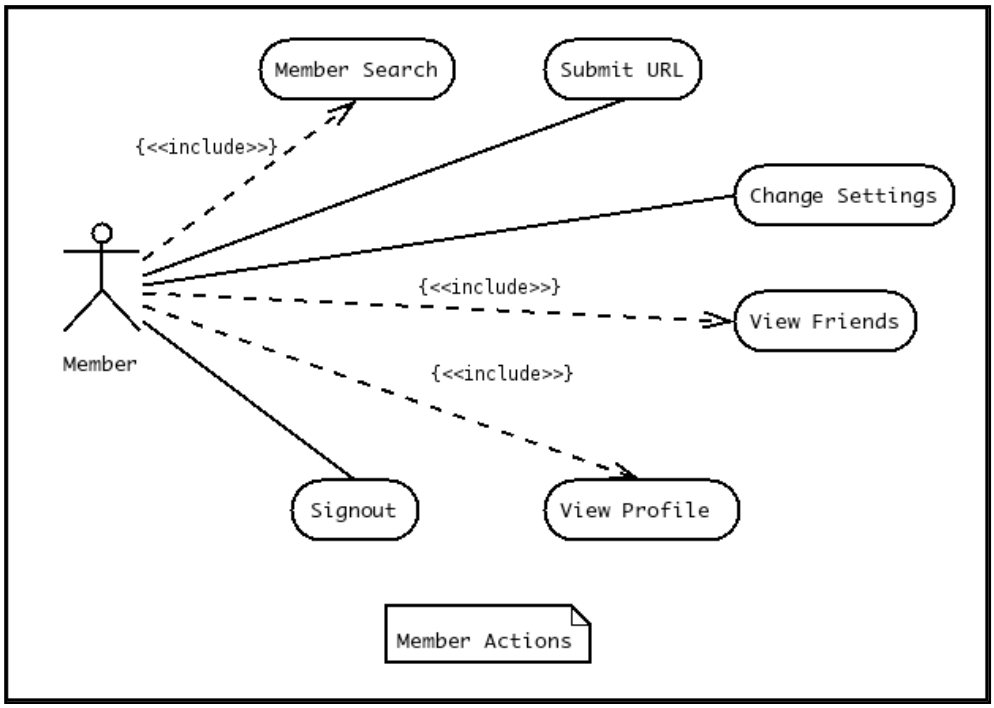
- Web-browser – the web-app would be accessed by the client using a web browser
- Protocols – HTTP 1.1, SMTP for sending email to friends
- Security – SSL for login authentication
- Data Transfer – the server response would be gzipped for efficiency.

4. System Features

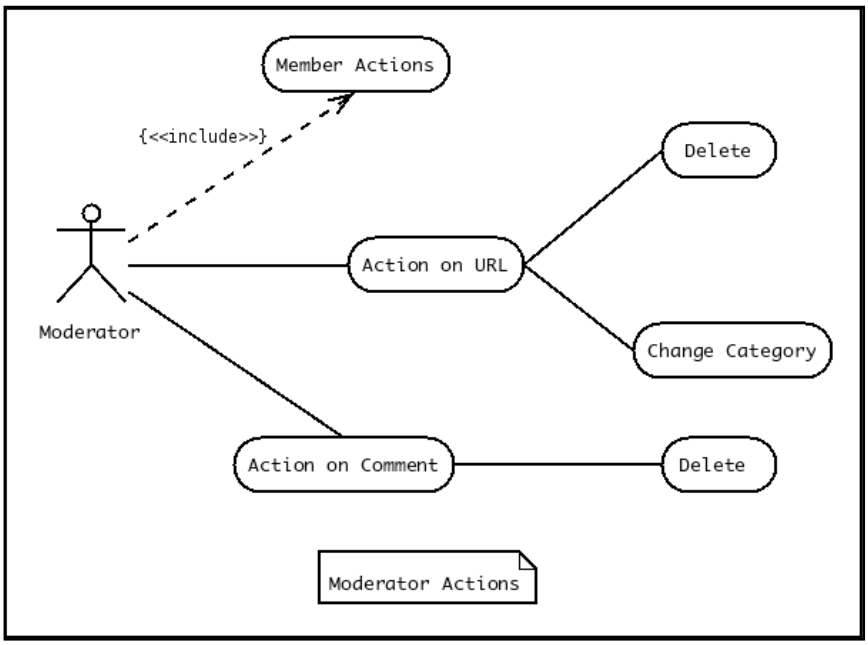
4.1 Actor Inheritance



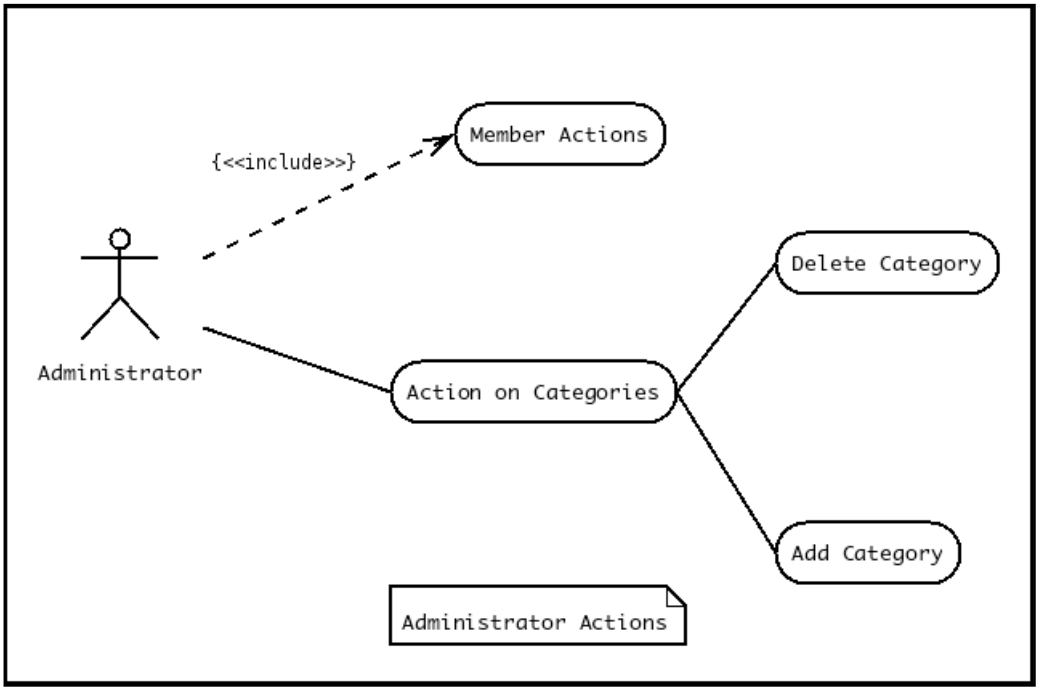
4.2 Member Actions (Feature 1)



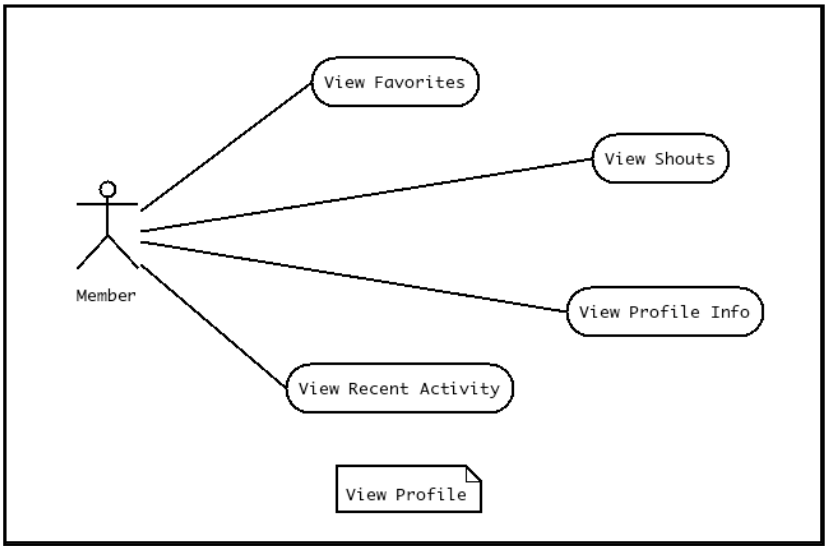
4.3 Moderator Actions (Feature 2)



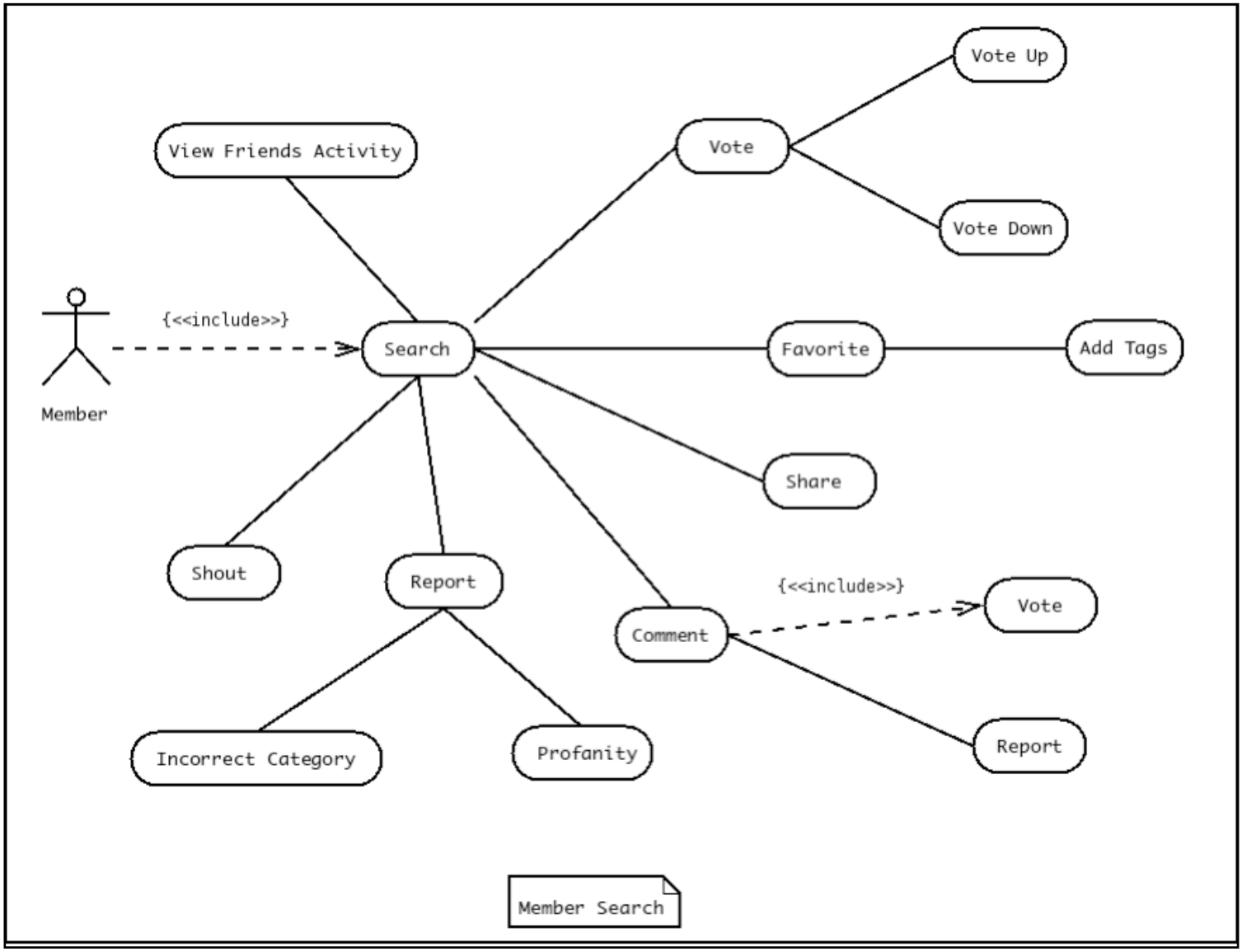
4.4 Administrator Actions (Feature 3)



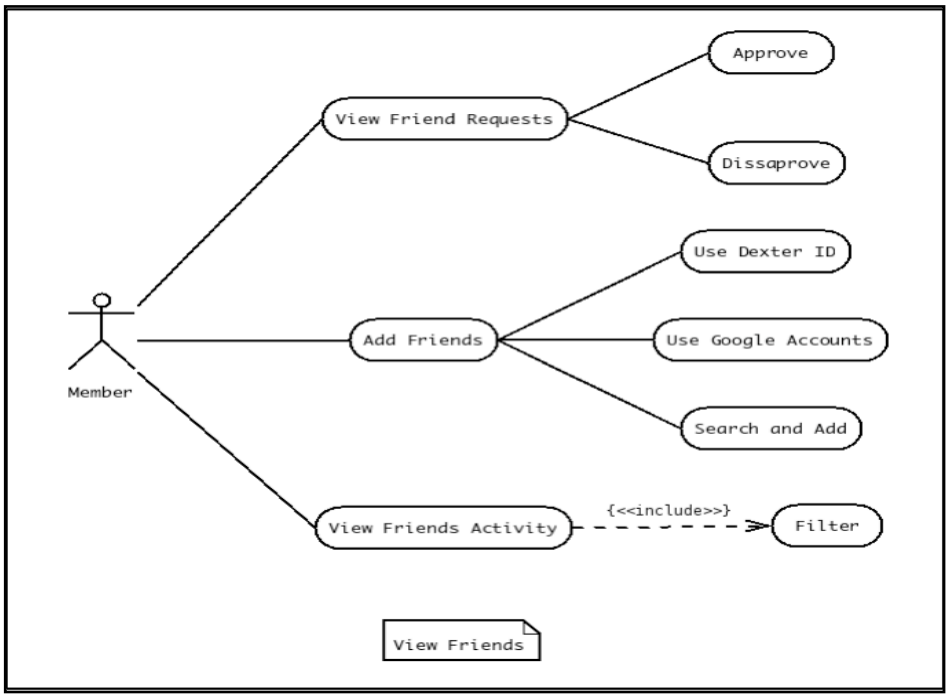
4.5 View Profile (Feature 5)



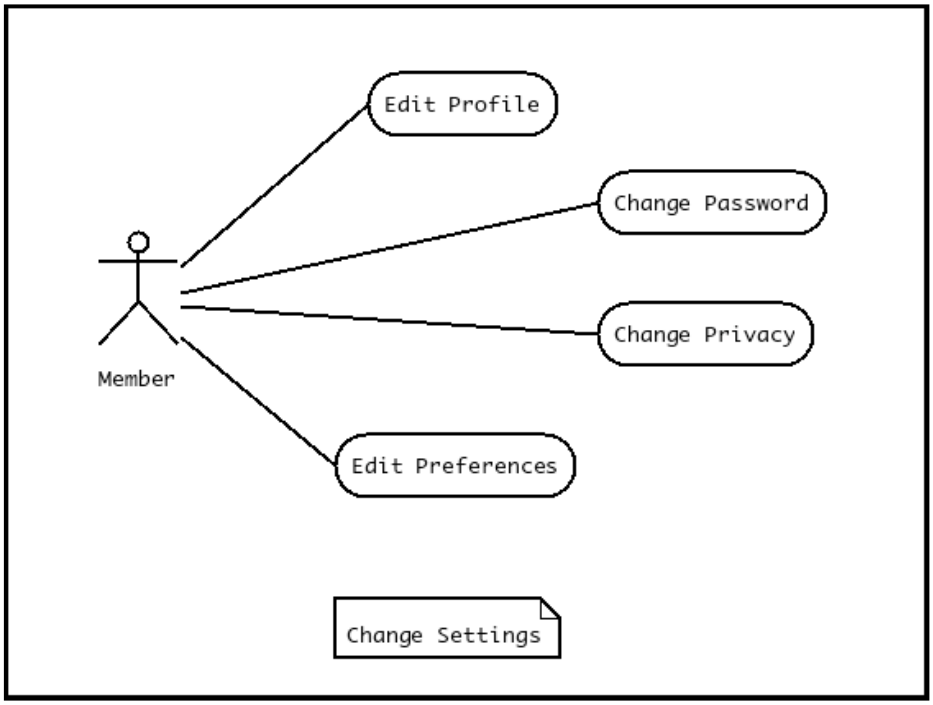
4.6 Member Search (Feature 4)



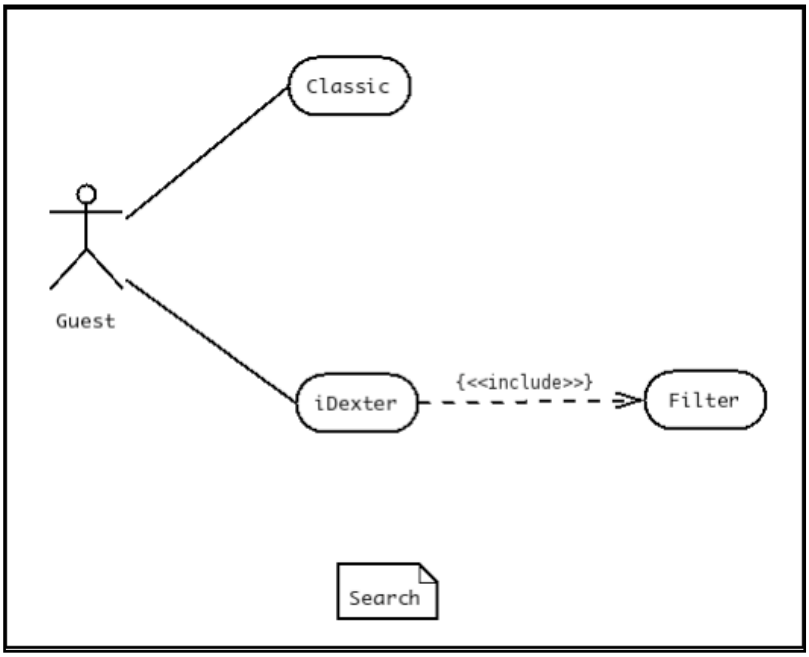
4.7 View Friends (Feature 6)



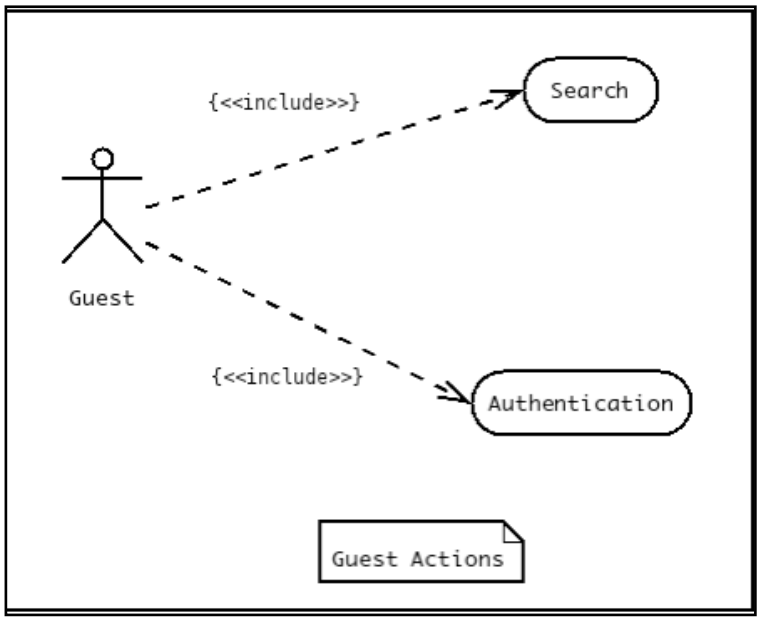
4.8 Settings (Feature 7)



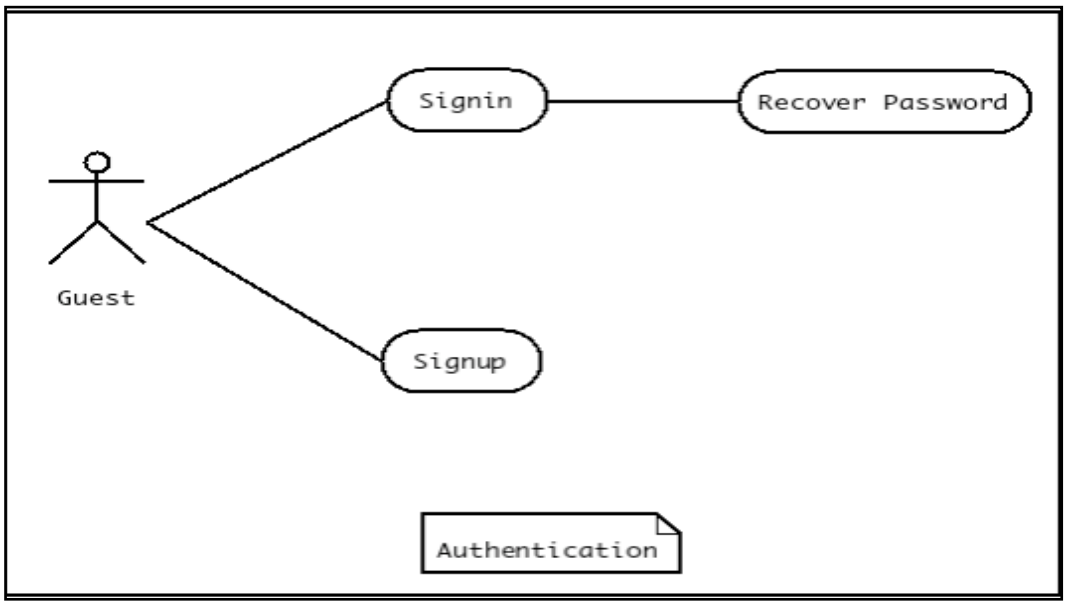
4.9 Search (Feature 8)



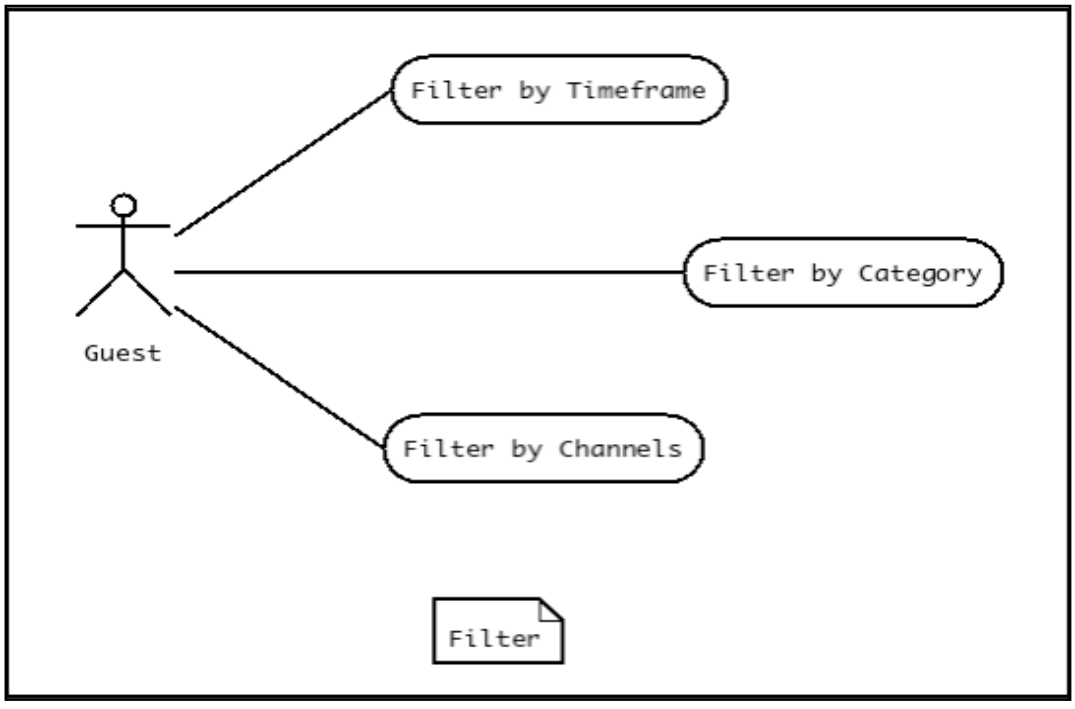
4.10 Guest Actions (Feature 9)



4.11 Authentication (Feature 10)



4.12 Filter (Feature 11)



5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The google search result retrieval and display should be as fast as possible.

5.2 Safety Requirements

- The search pattern as being logged by the web-application would reveal the user-characteristics and one's activity over the web.

5.3 Security Requirements

- The registration information should traverse the cloud in an encrypted form to prevent eavesdropping (need to use SSL)

- User needs to have options to set the privacy level for registration information and its activities in the form of comments or votes.

5.4 Software Quality Attributes

- **Adaptability** – as dexter directly uses the google search result API, it would automatically adapt to the changing order of search results which have not yet been submitted.

- **Flexibility** – the search results are flexible as they are user ordered.

- **Maintainability** – Moderators/Administrators have the work to maintain the content in the Dexter database

- **Reusability** – Dexter being open-source could be used by other projects who would wish to add to its functionality.

- **Usability** – Its even usable in cases when user does not know what to search for, and can use the iDexter interface to get started.

Dexter

making search Web 2.0 ready!

Appendix A: Glossary

Dexter - A Java EE Web Application based upon Struts and Hibernate Framework building an overlay on top of conventional Web 1.0 search engines to provide human collaboration and participation possibility

Dexter Surf - One of the two possible environments of interaction with the web-app. Surf provides content suggestion based upon popularity of voted search results in a particular timeframe, category and channel.

Dexter Search - One of the two possible environments of interaction with web-app. Search restricts to the classic search scenario waiting for the user to push through a keyboard for relevant response. Search being simplistic to Surf is considerably faster and recommended for normal search scenarios.

Channel - Filter the search results based on a particular website (Wikipedia, Orkut) or a media type (images/videos)

Category - Filter the search results based on field of interest (technology, Fun, Offbeat)

Timeframe - Filter the search results falling in a particular week,month,year

Votes - A method for the user to participate and help in providing the apparent usefulness of a search url by voting up or down. Votes eventually affect the positioning of the search url the next time a search for the same keyword is done.

URL Submission - A possibility to submit an arbitrary URL a user stumbles upon to when browsing and would want the Dexter database to have, being unsure of which keyword the search engine would show the URL against.

Shouts - Messages sent from one dexter user to another. (not Instant Message)

Favorite - Bookmark a search result for later.

Tag - keywords stored against a favorite for effective retrieval.

Report Comment - comments could be reported on profanity to the moderator for possible deletion.

Report URL - Search result URL's could be reported on profanity or incorrect category submission to the moderator for possible actions.