Introduction

1 Introduction

1.1 Overview

The project entitled “Smart Blogging Application System” mainly facilitates subject matter experts to post their blogs on Cognizant’s wall. By this we can bring all subject matter experts under a single platform to discuss and bring out new ideas. This project can be used by any of the Cognizant’s employee.

Employee needs to have access to One-Cognizant portal. After getting the access the user can post the blogging contents and discuss on it. All the other users can also post their comments on the post.

1.2 Feasibility study

During analysis, we ascertained whether the proposed system will be financially viable to the sense of being able to meet the burden of servicing debt and whether the proposed system will satisfy the return expectations of those who provide the capital.

We have considered investment outlay, means of projecting, Projected profitability, Investment worthiness judged in terms of various criteria of merit and license cost. We will be able to deploy this project on One Cognizant portal with minimal cost.

1.2.1 Technical Feasibility

During analysis we found there was .Net, j2ee, SharePoint developers.

Evaluation parameters which we noticed during this analysis are: Ease of Development, Development Infrastructure, Company Existing Standards and Technology Directions.

Input Analysis: This is mainly concerned with identification; qualification and evaluation of project inputs. We have ensured right quality of inputs would be available at right time and right cost.

Output Analysis: This involves product specification in terms of physical features- colour, height, width, functional features as well as standards to be complied with ISO.

Considering all this we came to a conclusion that .Net is more suitable to develop this project.

1.2.2 Behavioural Feasibility

System must be good enough to react to any of the changes made i.e. it should facilitate change. The functional requirement we have analysed to design the system to map all the huge cases of requirement, the behavioural feasibility is analysed by considering availability, scalability and performance aspect of system as non-functional requirement.

1.2.3 Operational Feasibility

For system maintainability and sustainability, coding based standards are included in development. So that future enhancements can be done easily to the system.

1.3 Existing System

With new ideas and full market survey we are designing this leading software application

Limitations of the existing system

- Time consuming
- Manual Work
- Not user friendly
- Tough to handle the errors.

1.4 Proposed system (Problem definition)

The smart blogger application is a utility that enables Cognizant's associates to share their thoughts on various topics. This application will allow a user to voice his opinion on any topic, in the form of a blog post.

The user of this application can search for content relevant to any topic, written by any other user. This application will also generate tags automatically from a blog post and let the user pick keywords relevant to his post.

Overall this application will provide a clean and convenient way of sharing thoughts between associates.

With Social media conversations and number of people participating in the conversations growing exponentially, valuable information and expertise is lost in this information overload. Develop an application to summarize free text (social media conversations like Blogs, Posts and Comments) and come out with the keywords (nouns and adjectives). The key parts to the solution are:

Summarize: Given a free flowing text (English only) the algorithm should identify the keywords (nouns and adjectives).

Synonymize: Tag root words for each of the keywords identified.

Personalize: Build profiles/personas of associates by tying back the root words identifies back to the person making such posts

Literature Survey

2. Literature Survey

2.1 What is Blog?

- A blog is discussion or information site published on the World Wide Web and consisting of discussion entries typically in reverse chronological order. The emergence and growth of blog in the late 1990’s coincided with the advent of web publishing tools that facilitated the posting content by no technical users.

- A major are interactive allowing visitors leave comments and even message each other via GUI widgets on the blogs and it is this interactivity that distinguishes them from other static websites. In that sense, blogging can be a seen as a form of social networking Indeed, bloggers do not only produce contents to post on their blog but also build social relations with their renders and other bloggers

2.2 Why Blogging is Important

Internal blogs have many upsides:

- Broadcasting employees' knowledge and expertise.

- Creating a searchable, permanent archive of expertise. All the knowledge published on an internal blog stays there, and long after former employees has left the building.

- Promoting open discussion and collaboration. Employees may be more likely to speak up virtually than face-to-face.

- Connecting employees across department lines.

- Keeping staff up-to-date on information that's important, but not enough for a meeting.

- Creating accountability: Once you've blogged about something, it can't be denied later. Goodbye, email trails.

2.3 Survey on similar kind of system

Done Survey on

- bhratmane.com

- blogspot.com

By surveying the above sites got a border idea to develop Smart Blogger Application system. Review system was clearer and post creation methodologies are gone through with those sites

Hardware and Software Requirements

3. Hardware and Software Requirements

Hardware Requirement

Server Side:

- Operating System: Windows 9x/xp ,Windows ME
- Processor: Pentium 3.0 GHz or higher
- RAM: 256 Mb or more
- Hard Drive: 10 GB or more

Client side:

- Operating System: Windows 9x or above, MAC or UNIX.
- Processor: Pentium III or 2.0 GHz or higher.
- RAM: 256 Mb or more

Software Requirement

Client Side: .HTML, Web Browser, Windows XP/2000/Vista

Web Server: HTML, Windows XP/2000/Vista

Software Requirements Specification

4. Software Requirements Specification

4.1 Introduction

4.1.1 Purpose

The purpose of this document is to describe the software requirements that are to be satisfied by the implementation of the proposed application on 1C platform. This document includes the details of software users, process analysis consisting of the problem to be addressed by the application and the proposed solution for the same. The goal of this document is to systematically gather the needs and expectations and eventually design the complete solution for the application.

4.1.2 Document Conventions

Throughout this documentation, the following conventions have been used:-

- Font : Times New Roman
- Size 16 For Main Headings
- Size 14 For Sub Headings
- Size 12 For the Rest of the Document
- Words in Bold are important terms, and have been formatted to grab the attention of the reader.

4.1.3 Intended Audience and Reading Suggestions

- The problem owners who proposed the system.
- Data, Application and Technical architects who will work on the application development.
- Software testers
- End users of the application (The intended Audience for this document is the Cognizant associates who will benefit from the application.)

4.1.4 Product Scope

The smart blogger application is a utility that enables Cognizant's associates to share their thoughts on various topics. This application will allow a user to voice his opinion on any topic, in the form of a blog post. The user of this application can search for content relevant to any topic, written by any other user. This application will also generate tags automatically from a blog post and let the user pick keywords relevant to his post. Overall this application will provide a clean and convenient way of sharing thoughts between associates.

4.1.5 References

- IEEE Software Engineering Standards Committee,"IEEE Std 830-1998,IEEE Recommended Practice for Software Requirements Specifications", October 20,1998.

4.2 Overall Description

4.1.3 Product Perspective

The Smart Blogger Application is a new, self-contained product intended for use on the 1C Platform. While the Smart Blogger online application is the main focus of the project, there is also a server-side component which will be responsible for database and synchronization services. The scope of the project encompasses both server- and client-side functionalities, so both aspects are covered in detail within this document.

4.2.2 Product Functions

- Write a blog post
- Search for topics
- Search for authors
- Like posts and add comments
- Select tags
- Favorite articles
- View favorites

- View profiles
- Earn points
- Obtain Badges
- Suggest an article
- Edit an article
- Delete a comment
- Delete a favorite

4.2.3 User Classes and Characteristics

- End users of the application (The intended Audience for this document is the Cognizant associates who will benefit from the application.)
- Administrators will follow some constraints to change the database
- Security is provided using secured database (SQL)

4.2.4 Operating Environment

- **Compatibility on different operating systems**

  The software is developed for Microsoft Windows operating system. The application uses C# and .NET framework and MS SQL database server. The application is best used on these environments.

- **Compatibility on different platforms**

  The application does not operate at the data storage levels or the hardware levels. So the software does not depend on the difference pertaining to the underlying architecture. The one factor that may be affected is the performance which typically increases with the number of processor cores and the amount of hyper threading (Intel Processors).

4.2.5 Design and Implementation Constraints

Conformance to architecture standards – The application corresponds to the four level architecture standards. The four layer architecture has the following layers:

View layer:

- This is the layer where the physical window and widget objects live. It may also contain Controller classes as in classical MVC.
- Any new user interface widgets developed for this application are put in this layer.
- In most cases today this layer is completely generated by a window-builder tool.

Application Model layer:

- This layer mediates between the various user interface components on a GUI screen and translates the messages that they understand into messages understood by the objects in the domain model.
- It is responsible for the flow of the application and controls navigation from window to window.
- This layer is often partially generated by a window-builder and partially coded by the developer.

Domain Model layer:

- This is the layer where most objects found in an Object Oriented analysis and design will reside.
- Examples of the types of objects found in this layer may be Projects, Employees, Backlog, or whatever is appropriate to the problem domain.

Infrastructure layer:

- This is where the objects that represent connections to entities outside the application (specifically those outside the object world) reside.

- Examples of objects in this layer would include SQL Tables and Serial Ports

4.2.6 User Documentation

The product will include user manual. The user manual will include product overview, complete configuration of the used software, technical details and contact information which will include email address. The product will be compatible with any browser.

4.2.7 Assumptions and Dependencies

Full working of the application is dependent on the availability of Internet connection.

- **Assumptions**

In general it has been assumed that the user has complete knowledge of the system that means user is not a naïve user. Any data entered by him/her will be valid. To make the software as user friendly as possible but at the same time keeping in minds user requirements.

Server OS should be Windows XP/Higher.

Client PC should be Windows 9X/NT/WorkStation or Windows 2000with latest service pack.

4.3 External Interface Requirements

4.3.1 User Interfaces

Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. The software would be fully compatible with Google Chrome and Mozilla Firefox. No user would be able to access any part of the application without logging on to the system.

4.3.2 Hardware Interfaces

Server Side:

- Operating System: Windows 9x/xp ,Windows ME
- Processor: Pentium 3.0 GHz or higher
- RAM: 256 Mb or more
- Hard Drive: 10 GB or more

Client Side:

- Operating System: Windows 9x or above, MAC or UNIX.
- Processor: Pentium III or 2.0 GHz or higher.
- RAM: 256 Mb or more

4.3.3 Software Interfaces

Client Side: .HTML, Web Browser, Windows XP/2000/Vista

Web Server: HTML, Windows XP/2000/Vista

4.3.4 Communications Interfaces

The Customer must connect to the Internet to access the Appication:

- Dialup Modem of 52 kbps
- Broadband Internet
- Dialup or Broadband Connection with a Internet Provider.

4.4 System Features

These are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situation. In some cases, the functional requirements may also explicitly state what the system should not do.

4.4.1 Write a Blog Post

User clicks on the 'write a post' button and the application provides a text area where he can write his content and post it.

4.4.2 Search for Topics

User can query for any topic using keywords and can obtain a list of posts relevant to his query.

4.4.3 Search for Authors

User can view the authors contributing to specific fields and also view their posts.

4.4.4 Like Posts and Add Comments

The user can like a post and/or comment on it.

4.4.5 Select Tags

The application automatically generates tags based on the content of the post. These tags are displayed to the user so that he can add the tags relevant to his post and add more tags that the user feels is missing.

4.4.6 Favorite Articles

The user can favorite an article.

4.4.7 View Favorites

The user can also access the articles marked as a favorite by him directly from the home page by bypassing the search function.

4.4.8 View Profiles

A summary of a user's profile is accessible to any other user of the app. A user's profile contains the person’s name, date when he joined the app, points he has earned, the badges obtained by him , all his posts (as links) and so on.

4.4.9 Earn Points

A user is awarded points for every contribution he has made. For example, a user earns points for writing a post, getting likes, commenting so on. A user may be promoted to different levels according to the points he has earned.

4.4.10 Obtain Badges

A user can obtain different badges based on his activity. For example, he can get a badge for different types of activities such as

a) If the user writes many posts related to a topic and if he obtains a number of likes he may be awarded a badge accordingly.

b) If the user comments on a number of posts related to a certain topic.

c) If a user tags his posts, which results in an increase in the number of viewers, i.e., the tags are appropriate and relevant.

4.4.11 Suggest an Article

A user can recommend articles to other users and users can view the recommended articles.

4.4.12 Edit an Article

A user can edit his article at a later stage. The edited article will be displayed in all subsequent requests for the article.

4.4.13 Delete a Comment

A user can delete a comment he made on an article.

4.4.14 Delete a Favorite

An article can be un-favorite by a user. It will not be displayed in his home screen thereafter.

4.5 Other Nonfunctional Requirements

4.5.1 Performance Requirements

- **Response times** –The entire application must load for use within 500ms

- **Processing times** – The processing time for each of the functionality must be as less as possible.

- **Query and Reporting times** – The initial load time of the application must be within 500ms and the interleaving time for the application must be as less as possible.

4.5.2 Safety Requirements

No safety requirements have been identified.

4.5.3 Security Requirements

The system should not expose the details of the project and the team involved to any third party applications because it reveals software development life cycle strategies used by the organization to develop a project and the team members involved in it.

4.5.4 Software Quality Attributes

Capacity:

- **Throughput** – The application must be capable of handling requests from all the employees in the organization at the same time without any issues of performance and concurrency so that they are able to access the application in time.
- **Storage** – The amount of storage for the application depends up on the number of current projects and the number of employees in the organization currently using the application.
- **Year-on-year growth requirements** – Every year the number of employees of our company increases and the number of project the company undertakes increases. The application must be graded up to meet the increasing requirements.

Availability:

- **Hours of operation** –The application should be available 24X 7. The administrator, the scrum master and the team members must be able to use the application anywhere through the one Cognizant platform.
- **Locations of operation** – The application should be accessible for use anywhere through the internet and the one Cognizant platform. All the operations are carried out on the one Cognizant environment

Reliability:

- **Mean Time Between Failures** – The mean time between failures of the application is really high because the application is hosted on a reliable cloud and the application performs read and write only on storage.
- **Mean Time To Recovery** – Failure can occur if we are not able to provide an internet connection or the administrator has not given the required access privileges. The recovery time depends on the time taken to repair these facilities

Integrity:

- **Fault trapping (I/O)** - All software errors are captured by means of Exception handling algorithms of C# and .NET.
- **Data integrity** –The integrity of the data should be maintained. If a modification occurs all the components related by a constraint are also deleted along with the currently deleted value.

Usability:

- **Look and feel standards** –The application is developed with user interface and ease-of-use as the main focus. The user interface systematically guides the people using the application, with animations which will make the user feel comfortable and happy.
- **Internationalization / localization requirements** – The entire application uses English (US) standard. All the terms in this application are related to scrum agile practices and sprint poker estimation techniques. The application is targeted towards audiences who have knowledge about the project planning and estimation practices.

4.5.5 Business Rules

The basic fact here is the material for upload and download is totally dependent on the user's discretion and network has no responsibility for that matter. It only provides a medium of transport. It's up user's sense of morality in which manner he exploits the system.

4.6 Other Requirement

Other requirements are none.

Appendix A: Glossary

- **1C – One Cognizant Platform**

  One Cognizant Platform has been envisioned to provide next generation of information workplace to Cognizant associates, by addressing needs around user context, seamlessness, process centricity, multi-modality, personalization, rich interaction and social computing based on key features of Web2.0.

System Design Description

5. System Design Description

5.1 Introduction

5.1.2 Purpose

The purpose of this Document is to explain complete design details of "Smart Blogger Application" project. As IEEE standards Document indicates, the design report to show how the proposed software system will be structured in order to satisfy the requirements identified in Software Requirement Specification Document. In other words, it is aimed to translate software requirements defined in SRS document into a representation of software components, interfaces and data to be used later in implementation phase of the project. However, since every software design is open to changes and modifications, it is highly possible to make changes during implementation and thus update SRS and SDD documents accordingly.

5.1.3 Scope

Primarily, the system is built to facilitate the Cognizant's associates to share their thoughts so as to get all users under a single platform to discuss on the subject and voice his opinion on any topic, in the form of a blog post. The user of this application can search for content relevant to any topic, written by any other user. This application will also generate tags automatically from a blog post and let the user pick keywords relevant to his post. Overall this application will provide a clean and convenient way of sharing thoughts between associates.

5.1.4 References

IEEE.IEEE Std 830-1998 IEEE Recommended Practice for software requirement specifications.

5.1.5 Overview

The complete SDD will contain the general definition and features of the project, design constraints, the overall system architecture and data architecture. With the help of UML

Diagrams, design of the system and subsystem/modules will be explained visually in order to help the programmer to understand all information stated in the document correctly and easily.

5.1.6 Constraints

The overall application should fit into a viewable area of 965X525 px(Pixels).

5.2 System Overview

5.2.1 System Architecture

Architectural Design is a process of decomposing a large complex system into small subsystems. These subsystems are meant for providing some related services. The Architectural Design is basically a layout or a framework of the system for the subsystem control and communications.

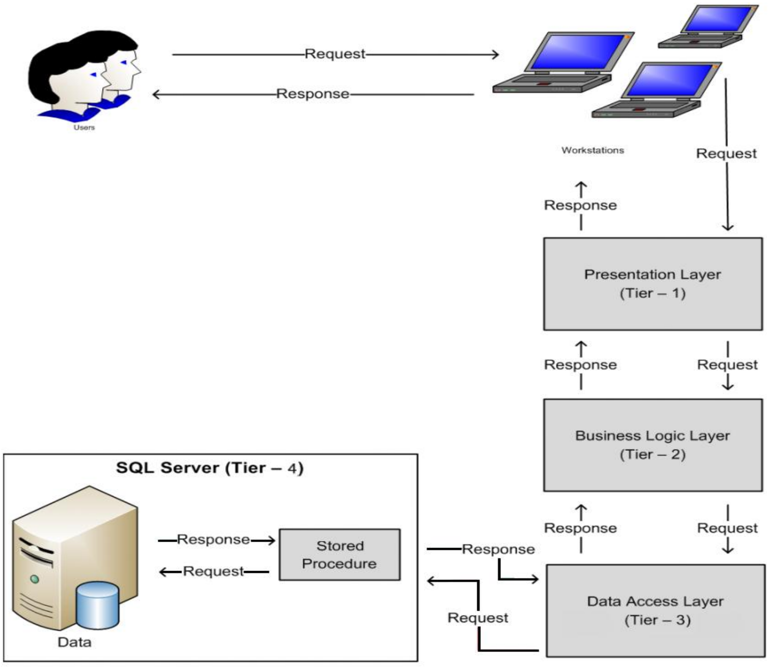


Figure: 5.1 -Tier Architecture

N-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application.

Architecture of Smart Blogger Application consists of 4 layers in which the User Interface, Business Logic, Data Access and Data Storage are developed and maintained as independent modules.

The 4-Tiers are:

1. **Tier – 1: Presentation Layer:** This layer is responsible for various activities between Users and application. All the user interface related logic resides at this layer. This layer does not have direct access to the database or Data Access Layer. So, all the data goes in or comes out to Presentation Layer will be through the Business Logic Layer only. *A reference of Business Logic Layer is added to this layer*.

2. **Tier – 2: Business Logic Layer:** Business Logic Layer performs all the Business Logics of the application. Business logic consists of two things viz., Core business logic and Data access logic. Data access logic is been segregated to different layer to provide more security and data encapsulation to the application. There is a separate class for each table of the database to provide better management, in this layer. This class has various methods. Business logic is applied to data coming from or going to Presentation Layer and Data Access Layer. *A reference of Data Access Layer is added to this layer*.

3. **Tier – 3: Data Access Layer:** This layer is a part of Business logic but is separated from core business logic. All the data related operations between application and database is performed over here. This layer has minimum manual coding.

4. **Tier – 4: Database:** This is core data and other objects to maintain and access it. For e.g. SQL server database. Core data is data in various tables and objects to maintain & access such data are various tables and Stored Procedures.

5.3 Data Dictionary

Table 5.3.1 tblUser

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
User	strUserId	Unique id of Associate	Varchar	Required	Primary Key
User	strLName	Last Name of Associate	Varchar		
User	strFName	First Name of Associate	Varchar		
User	strProfilePicture	Profile Picture	Varchar		
User	dCreatedAt	Time of Profile Creation	DateTime		
User	strDesignation	Designation of Associate	Varchar		
User	strEmailId	Email Id of Associate	Varchar		
User	dLastLogin	Time of Last Login	DateTime		
User	iPoints	Number of Points gained	Integer		
User	strDescription	Short Note about Associate	Varchar		
User	iRibbonId	Id of Ribbon achieved	Integer		Foreign Key

Table 5.3.2 tblPost

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Post	iPostId	Unique Id of the post	Integer	Required	Primary Key
Post	strTitle	Title of the post	Varchar		
Post	strContent	Article’s main body	Varchar		
Post	strPicture	Optional picture with a post	Varchar		
Post	strStatus	Status of the post	Varchar		
Post	dCreateTime	Time at which post was created	DateTime		
Post	dUpdateTime	Time at which post was updated(if applicable)	DateTime		

Post	strUserId	Id of the user writing the post	Varchar		Forign Key
Post	iViews	Number of users who have viewed the post till now	Integer		

Table 5.3.3 tblTag

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Tag	iTagId	Umique id of a tag	Integer	Required	Primary Key
Tag	strTagName	Name of the tag	Varchar		
Tag	iFrequency	Number of times this tag has been used	Integer		
Tag	strDescription	Description for the tag	Varchar		
Tag	strWords	Frequently appearing words with the tag	Varchar		

Table 5.3.4 tblComment

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Comment	iCommentId	Umique id of a comment	Integer	Required	Primary Key
Comment	dCreateTime	Time at which comment was written	DateTime		
Comment	strContent	Body of the comment	Varchar		

Comment	iUserId	Id of the user making the comment	Integer		Foreign Key
Comment	iPostId	Id of the post on which the comment was made	Integer		Foreign Key

Table 5.3.5 tblbadge

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Badge	iBadgeId	Umique id of the Badge	Integer	Required	Primary Key
Badge	strBadgeName	Name of the Badge	Varchar		
Badge	strIcon	Badge icon	Varchar		
Badge	strDescription	Description for the Badge	Varchar		

Table 5.3.6 tblRibbon

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Ribbon	iRibbonId	Umique id of a Ribbon	Integer	Required	Primary Key
Ribbon	iThreshold	Points needed to get the Ribbon	Integer		
Ribbon	strPicture	Picture of the Ribbon	Varchar		

Table 5.3.7 tblNotification

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Notification	iNId	Umique id of a Notification	Integer	Required	Primary Key
Notification	iRecipientId	Id of Recipient	Integer		Foreign Key
Notification	iSenderId	Id of Sender	Integer		Foreign Key
Notification	strActivityType	Type of Activity that generated the Notification	Varchar		
Notification	dTimeOfActivity	Time at which activity took place	DateTime		
Notification	strIsUnread		Varchar		
Notification	strConfig		Varchar		

Table 5.3.8 tblPostTag

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Badge	iBadgeId	Umique id of the Badge	Integer	Required	Primary Key
Badge	strBadgeName	Name of the Badge	Varchar		
Badge	strIcon	Badge icon	Varchar		
Badge	strDescription	Description for the Badge	Varchar		

Table 5.3.9 tblPostTag

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Post’s Tag	iPTId	Unique id in the table	Integer	Required	Primary Key
Post’s Tag	iPostId	Unique Id of a post	Integer		
Post’s Tag	iTagid	Unique Id of a tag	Integer		
Post’s Tag	iByUser	Differentiates between the tags given by the user with the tags generated by the system	Integer		

Table 5.3.10 tblUserTag

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
User’s Tag	iUTId	Unique id of the table	Integer	Required	Primary Key
User’s Tag	strUserId	Unique id of a user	Varchar		Foreign Key
User’s Tag	iTagId	Unique id of a tag	Integer		Foreign Key
User’s Tag	iUserTagFreq	Number of posts in which user has used this tag	Integer		

Table 5.3.11 tblUserBadge

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
User's Badge	iUBId	Unique id for the table	Integer	Required	Primary Key
User's Badge	iBadgeId	Unique id of a badge	Integer		Foreign Key
User's Badge	strUserId	Unique id of a user	Varchar		Foreign Key

Table 5.3.12 tblFavorite

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Favorite	iFavId	Unique id for the table	Integer	Required	Primary Key
Favorite	strUserId	Unique id of a user	Varchar		Foreign Key
Favrorite	ipostId	Unique id of a post	Integer		Foreign Key
Favorite	dFavDate	Date on which the post was favorite	Date		

Table 5.3.13 tblSuggestPost

Entity Name	Element Name	Definition	Type	Mandatory Entry	Constraints
Suggest Post	iSPId	Unique id for the table	Integer	Required	Primary Key
Suggest Post	strUserId1	Id of the user making the suggestion	Varchar		Foreign Key
Suggest Post	strUserId2	Id of the user for whom the post is	Varchar		Foreign Key

		being suggested			
Suggest Post	iPostId	Unique id of the suggested post	Integer		Foreign Key

Table 5.3.14 tblLikes

Entity Name	**Element Name**	**Definition**	**Type**	**Mandatory Entry**	**Constraints**
Post's Likes	iLId	Unique id for the table	Integer	Required	Primary Key
Post's Likes	iPostId	Unique id of a post	Integer		Foreign Key
Post's Likes	iCount	Number of Likes	Integer		
Post's Likes	strUserId	Unique id of a user	Varchar		Foreign Key

Table 5.3.15 tblCLikes

Entity Name	**Element Name**	**Definition**	**Type**	**Mandatory Entry**	**Constraints**
Comment's Likes	iCLId	Unique id for the table	Integer	Required	Primary Key
Comment's Likes	iCommentId	Unique id of comment	Integer		Foreign key
Comment's Likes	strUserId	Unique Id of a user	Varchar		Foreign Key

5.4 Function Design

5.4.1 Use Case Diagram

a use case diagram is a graphical notation for summarizing actors and use cases. The first step in a typical development effort is to analyze the description of the system and produce a model of the system requirements. Use case Diagram depicts a static view of the system functions and their static relationship, both with external entities and with one another. It consists of system, actor and use case.

System: The system is depicted as a rectangle.

Actor: Each actor is shown as a stick figure.

Use Case: Each use case is shown as a solid bordered oval labeled with the name of the use case.

A use case is a set of scenarios that describes the interaction between a user and a system. A use case diagram displays the relationship among actors and use cases.

An actor represents a user or another system that will interact with the system being modeled. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

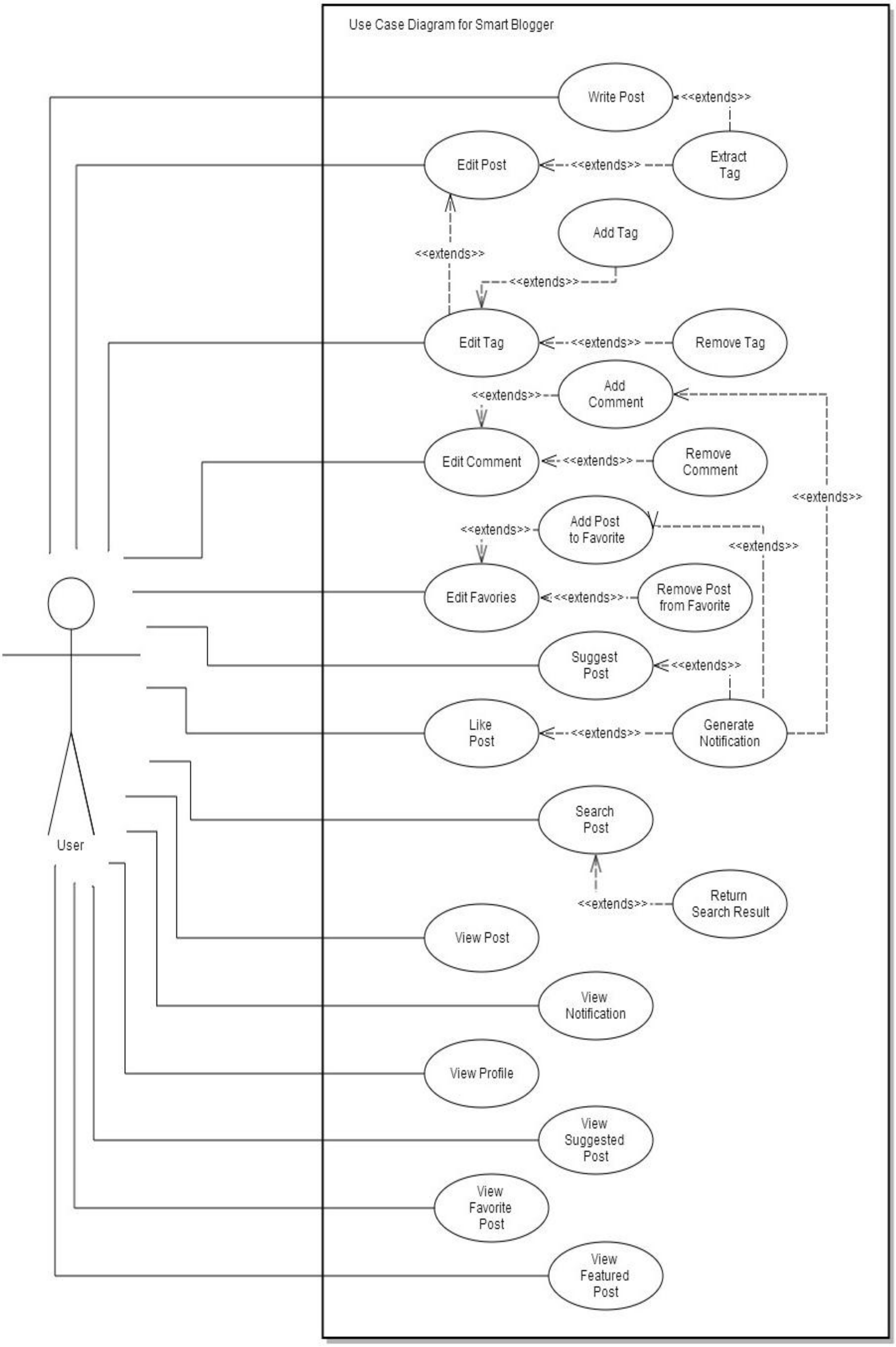


Figure 5.2: Use case Diagram for Smart Blogger Application

Actor Description

The possible actor here is the User that is an Associate of Cognizant.

User functionalities are:

- Write Post
- Edit Post
- Edit Tag
- Edit Comment
- Edit Favorites
- Suggest Post
- Like Post
- Search Post
- View Post
- View Notification
- View Profile
- View Suggested post
- View Favorite Post
- View Featured Post

Use Case Description

Write Post

Use Case	Write Post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects “Write Article” from the system. 2) The system provides a new post creation page to the user. 3) The user fills all the contents for the new post form. 4) The system extracts tags from the

	post, updates the database and awards points to the user.

Edit Post

Use Case	Edit Post
Primary Actor	User
Preconditions	User should have access to the Application and should be the author of the post to be edited.
Scenarios	1) The user opens a post from User Profile. 2) The user selects "Edit" from the system. 3) The system provides an edit post page to the user. 4) The user edits the contents of the post. 5) The system extracts tags from the post and updates the database.

Edit Tag

Use Case	Edit Tag
Primary Actor	User
Preconditions	User should have access to the Application and should be the author of the post whose tag is to be edited.
Scenarios	1) The user opens a post from User Profile. 2) The user selects "Edit" from the system. 3) The system provides an edit post page

	to the user. 4) The user selects “X” next to a tag. 5) The system removes the tag from the post and updates the database.
	1) The user opens a post from User Profile. 2) The user selects “Edit” from the system. 3) The system provides an edit post page to the user. 4) The user selects “+” next to tag list. 5) The system provides an add tag prompt to the user. 6) The user enters a new tag name. 7) The system adds the tag to the post and updates the database.

Edit Comment

Use Case	Edit Comment
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects a post. 2) The system provides a view post page to the user. 3) The user enters a comment for the post. 4) The system adds the comment to the post, award points to the user, generates a notification for the author of the post and updates the database.
Preconditions	User should have access to the Application and should be the author of the comment.

Scenarios	1) The user selects a post. 2) The system provides a view post page to the user. 3) The user selects “X” next to a comment for the post. 4) The system removes the comment from the post and updates the database.

Edit Favorites

Use Case	Edit Favorites
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects a post. 2) The system provides a view post page to the user. 3) The user selects the “add to Favorite” icon. 4) The system adds the post to user’s favorites, award points to the user, generates notification for the author of the post and updates the database.
Preconditions	User should have access to the Application and should have a post added to favorites.
Scenarios	1) The user selects a favorite post from favorites page. 2) The system provides a view post page to the user. 3) The user selects the “remove from Favorite” icon. 4) The system removes the post from user’s favorites and updates the

	database.

Suggest Post

Use Case	Suggest Post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects a post. 2) The system provides a view post page to the user. 3) The user selects the “Suggest” icon. 4) The system provides a prompt to enter recipient’s name to the user. 5) The user enters a name and selects a recipient. 6) The system generates a notification for the recipient, award points to the user and updates the database.

Like Post

Use Case	Like Post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects a post. 2) The system provides a view post page to the user. 3) The user selects the “Like” icon. 4) The system generates a notification for the author of the post, award points to the user and updates the database.

View Post

Use Case	View Post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects a post. 2) The system provides a view post page to the user.

Search Post

Use Case	Search Post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user enters a search text in the Search Box and clicks on the Search icon. 2) The system provides a Search Result page to the user.

View Notification

Use Case	View Notification
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects the Notification tab. 2) The system provides a notification page to the user.

View Profile

Use Case	View Profile
Primary Actor	User

Preconditions	User should have access to the Application.
Scenarios	1) The user selects the My Profile tab. 2) The system provides a User Profile page to the user.

View Suggested Post

Use Case	View Suggested post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects the Suggestions tab. 2) The system provides a Suggestions page to the user.

View Favorite Post

Use Case	View Favorite Post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects the Favorites tab. 2) The system provides a Favorites page to the user.

View Featured Post

Use Case	View Featured Post
Primary Actor	User
Preconditions	User should have access to the Application.
Scenarios	1) The user selects the Featured tab. 2) The system provides a Featured page to the user.

5.4.2 Behavioral Design

5.4.2.1Activity Diagrams

Activity diagrams show the flow of activities through the system. Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities. A fork is used when multiple activities are occurring at the same time. Rounded rectangles represent activities, Diamonds represent decisions, a black circle represents the start (initial state) of the workflow, and an encircled black circle represents the end (final state). Arrows run from the start towards the end and represent the order in which activities happen.

Activity Diagram for Write Post

Figure 5.3 shows an activity diagram for Write Post. The user must select the Write Article tab to write a post. A new post creation page is displayed where the user can enter the fields and submit the post. Thereafter, the system extracts the tags from the post, awards points to the user and updates the database.

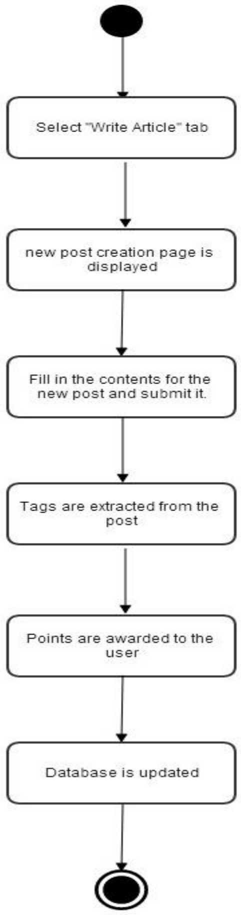

Select "Write Article" tab
new post creation page is displayed
Fill in the contents for the new post and submit it.
Tags are extracted from the post
Points are awarded to the user
Database is updated

Figure 5.3: Activity Diagram for Write Post

Activity Diagram for Edit Post

Figure 5.3 shows an activity diagram for Edit Post. To edit a post user opens a post from his/her user profile and selects Edit. Edit post page is displayed where user can edit the post and submit the changes. Once the post is submitted tags are extracted from the post and database is updated.

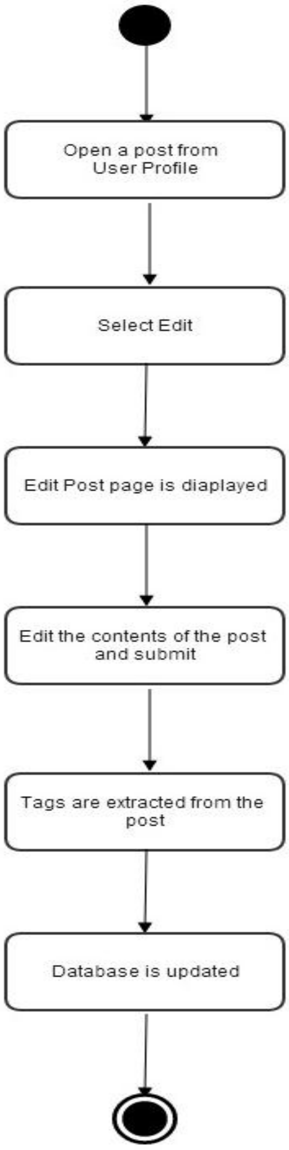

Open a post from User Profile
Select Edit
Edit Post page is diaplayed
Edit the contents of the post and submit
Tags are extracted from the post
Database is updated

Figure 5.4: Activity Diagram for Edit Post

Activity Diagram for Edit Tag

Figure 5.5 shows an activity diagram for Edit Tag. To edit a tag for a post a user must open a post from the user profile and click on edit. Edit post page is displayed where user can click “X” next to a tag to delete it from the post or click on “+” next to the tag list to add a tag. When user clicks on “+” add tag prompt is displayed where the user can enter a new tag and click on add. Once the tags are edited the database is updated by the system.

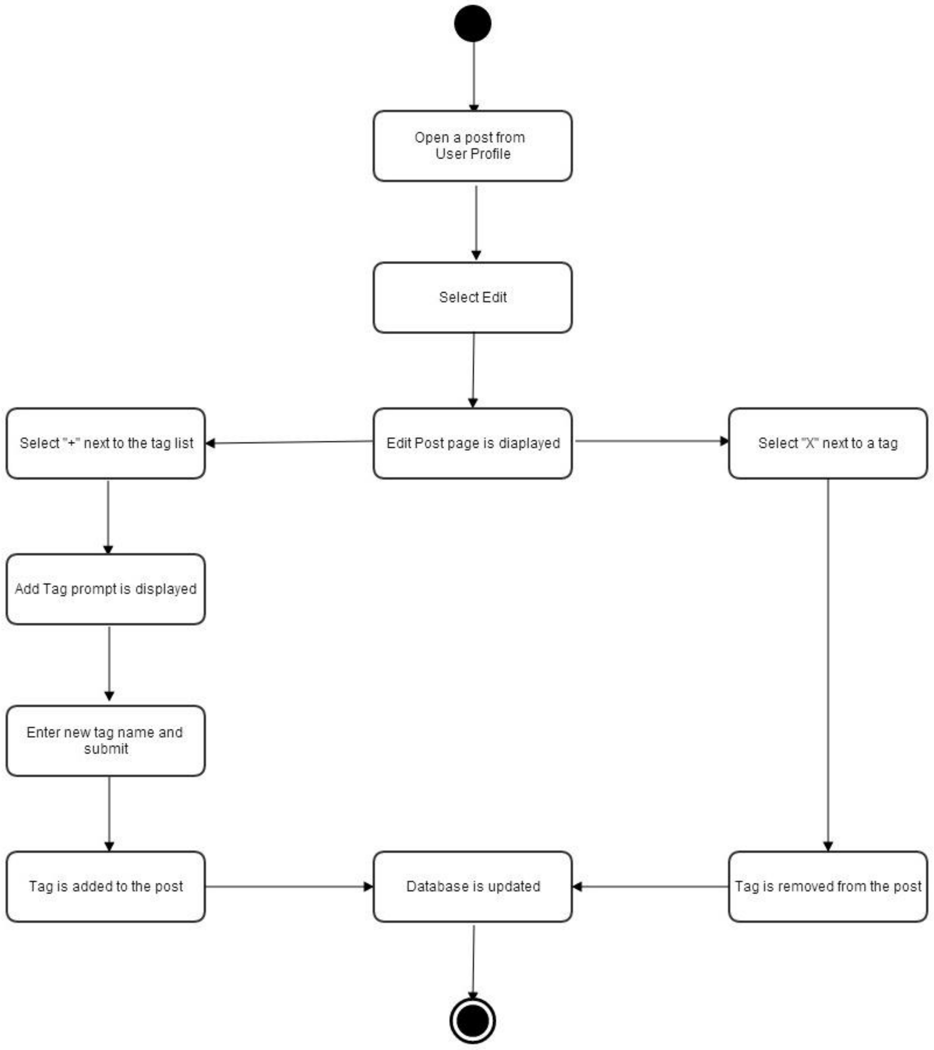

Open a post from User Profile
Select Edit
Select "+" next to the tag list
Edit Post page is diaplayed
Select "X" next to a tag
Add Tag prompt is displayed
Enter new tag name and submit
Tag is added to the post
Database is updated
Tag is removed from the post

Figure 5.5: Activity Diagram for Edit Tag

Activity Diagram for Edit Comment

Figure 5.6 sows an activity diagram for Edit comment. To edit a comment for a post a user must open a post. View post page is displayed where if the user is the author of the comment he/she can click “X” next to a comment to delete it from the post. To add a comment user can enter the comment in the comment box and click on submit. The system then adds the comment to the post, awards points to the user and generates notification for the author of the post. Once the comments are edited the database is updated by the system.

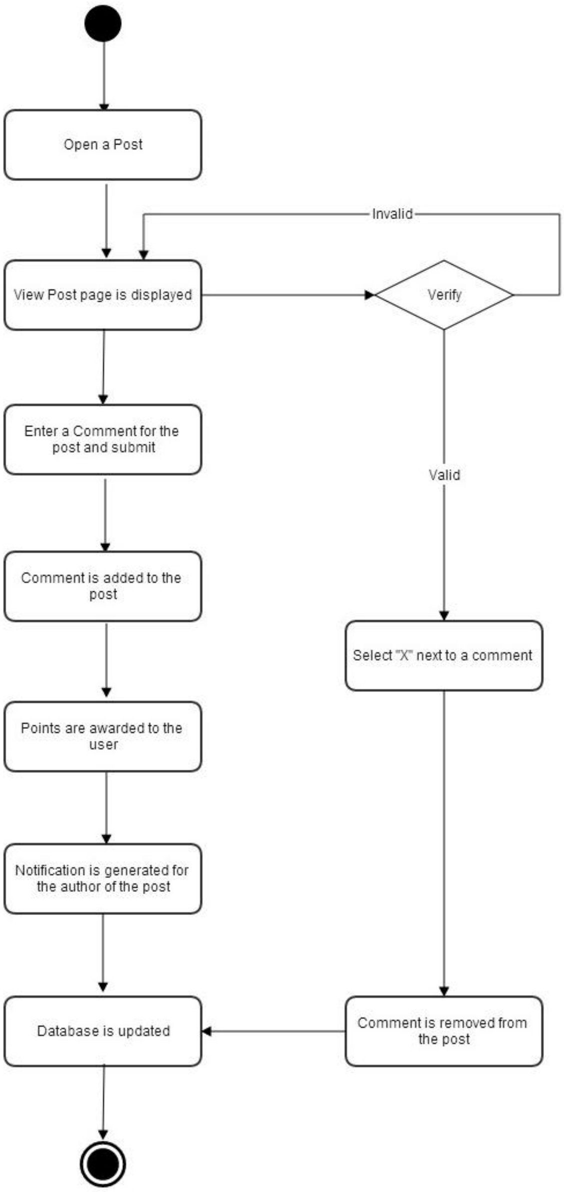

Open a Post
Invalid
View Post page is displayed
Verify
Enter a Comment for the post and submit
Valid
Comment is added to the post
Select "X" next to a comment
Points are awarded to the user
Notification is generated for the author of the post
Database is updated
Comment is removed from the post

Figure 5.6: Activity Diagram for Edit Comment

Activity Diagram for Edit Favorites

Figure 5.7 shows the activity diagram for Edit favorites. When a user opens a post View post page is displayed where user can click on “Add to Favorite” icon to add a post to his/her favorites. The system adds the post to the user’s favorites, awards points to the user and sends a notification to the author of the post. If a user wants to remove a post from his/her favorites he/she must click on “remove from Favorite” icon on the favorite post, the system then removes the post from their favorites list. Once the editing is over the system updates the database.

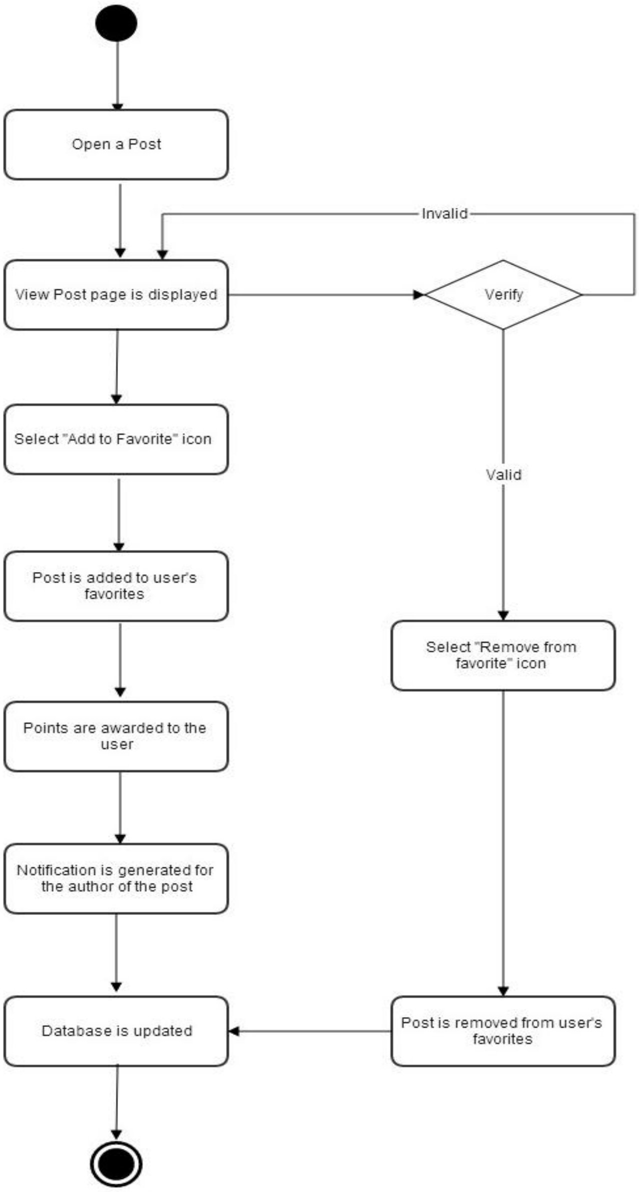
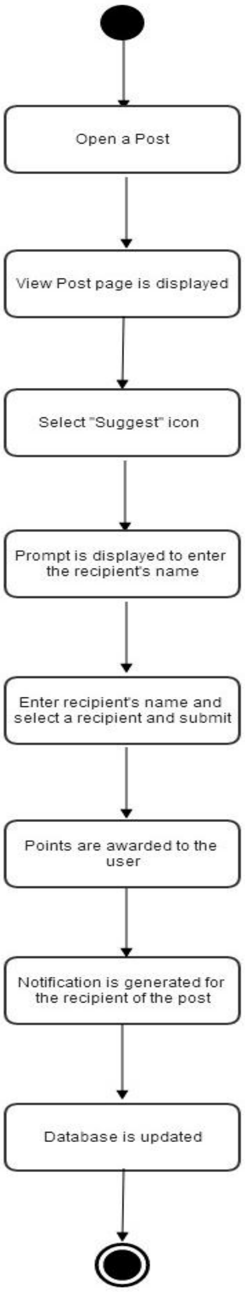

Open a Post
Invalid
View Post page is displayed
Verify
Select "Add to Favorite" icon
Valid
Post is added to user's favorites
Select "Remove from favorite" icon
Points are awarded to the user
Notification is generated for the author of the post
Database is updated
Post is removed from user's favorites

Figure 5.7: Activity Diagram for Edit Favorites

Activity Diagram for Suggest Post

Figure 5.8 shows the diagram for Suggest Post. To suggest a post a user must open a post and click on suggest icon. The system displays a prompt to the user to enter and select the name of another user to whom the post is to be suggested. The user selects and submits. The system then adds the points to the user profile, generates notification for the recipient of the suggestion and updates the database.


Open a Post
View Post page is displayed
Select "Suggest" icon
Prompt is displayed to enter the recipient's name
Enter recipient's name and select a recipient and submit
Points are awarded to the user
Notification is generated for the recipient of the post
Database is updated

igure 5.8: Activity Diagram for Suggest Post

Activity Diagram for Like Post

Figure 5.9 shows the activity diagram for Like Post. To like a a post user must open the post and click on the like icon. The system then awards points to the user, sends a notification to the author of the post and updates the database.

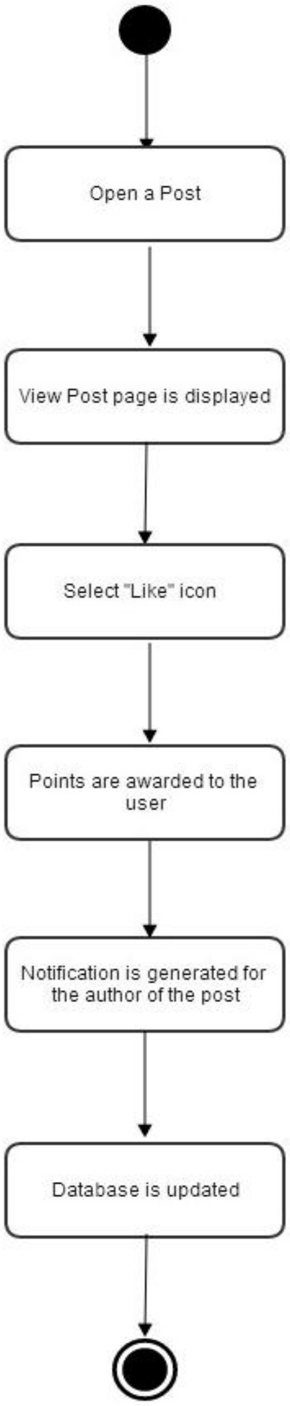

Open a Post
View Post page is displayed
Select "Like" icon
Points are awarded to the user
Notification is generated for the author of the post
Database is updated

Figure 5.9: Activity Diagram for Like Post

Activity Diagram for Search Post

Figure 5.10 shows an activity diagram for Search post. To search for a post the user must enter a search text in the search box and click on search icon. The system then returns the search results page with the results.

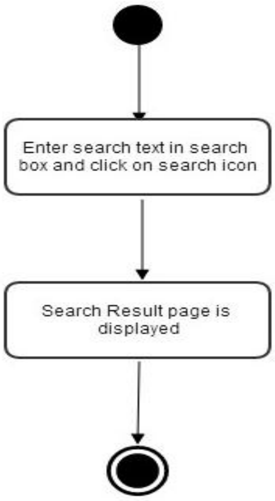

Enter search text in search box and click on search icon
Search Result page is displayed

Figure 5.10: Activity Diagram for Search Post

Activity Diagram for View Post

Figure 5.11 shows an activity diagram for View post. To view a post a user must click on a post title displayed. The system then returns a view post page with the full ppost.

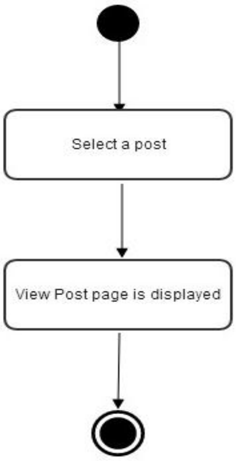


Figure 5.11: Activity Diagram for View Post

Activity Diagram for View Notification

Figure 5.12 shows an activity diagram for View Notification. To view the notifications user must click on Notification tab. When clicked system returns the Notification page.

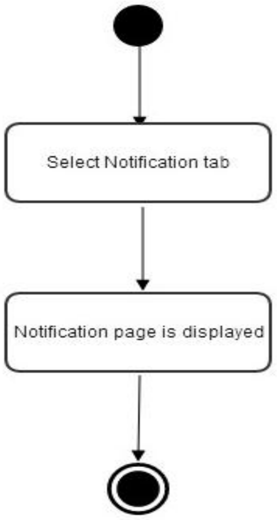


Figure 5.12: Activity Diagram for View Notification

Activity Diagram for View Profile

Figure 5.13 shows an activity diagram for View Profile. To view his/her profile user must click on My Profile tab. The system returns the user profile page.

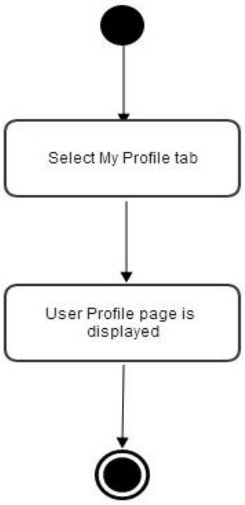


Figure 5.13: Activity Diagram for View Profile

Activity Diagram for View Suggested post

Figure 5.14 shows an activity diagram for View Suggested post. To view all suggestions made to the user and from other users, user must click on Suggestions tab. The system then returns a Suggestions page.

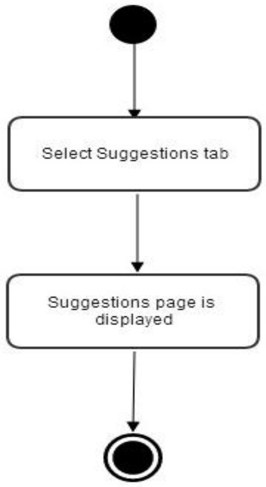


Figure 5.14: Activity Diagram for View Suggested post

Activity Diagram for View Favorite Post

Figure 5.15 shows the activity diagram for Favorite Post. To view the favorite articles user must click on the Favorite tab. The system then returns the Favorites page to the user.

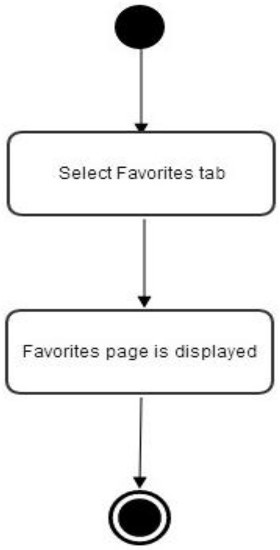


Figure 5.15: Activity Diagram for View Favorite Post

Activity Diagram for View Featured Post

Figure 5.16 shows the activity diagram for View featured Post. To view featured posts user must click on the featured tab. The system then returns the Featured page.

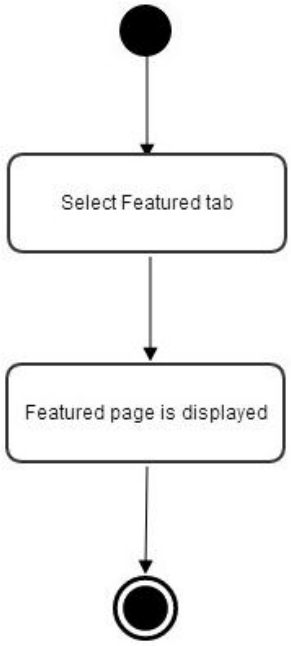


Figure 5.16: Activity Diagram for View Featured Post

5.4.2.3 Sequence Diagram

A Sequence Diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams.

Sequence Diagram for Write Post

Figure 5.17 shows a sequence diagram for Write Post. The user must select the Write Article tab to write a post. A new post creation page is displayed where the user can enter the fields and submit the post. Thereafter, the system extracts the tags from the post, awards points to the user and updates the database.

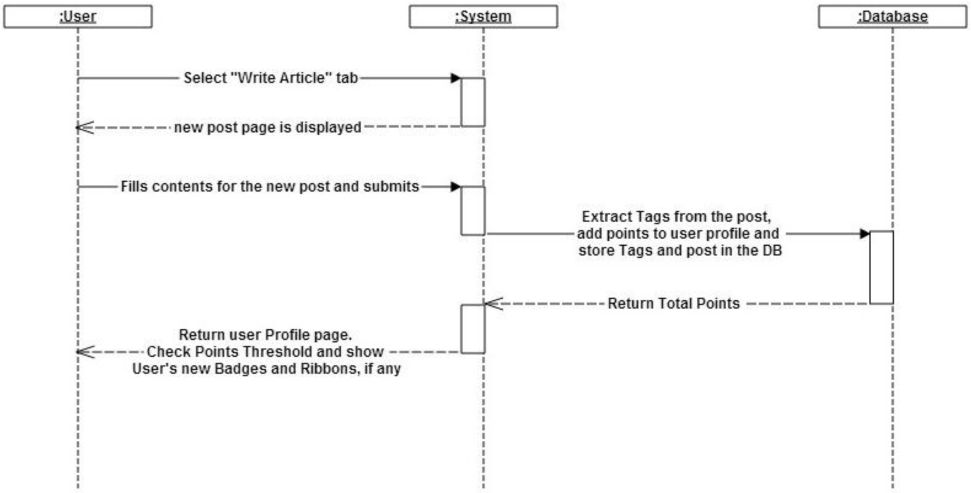

:User
:System
:Database
Select "Write Article" tab
new post page is displayed
Fills contents for the new post and submits
Extract Tags from the post, add points to user profile and store Tags and post in the DB
Return Total Points
Return user Profile page. Check Points Threshold and show User's new Badges and Ribbons, if any

Figure 5.17: Sequence Diagram for Write Post

Sequence Diagram for Edit Post

Figure 5.18 shows a sequence diagram for Edit Post. To edit a post user opens a post from his/her user profile and selects Edit. Edit post page is displayed where user can edit the post and submit the changes. Once the post is submitted tags are extracted from the post and database is updated.

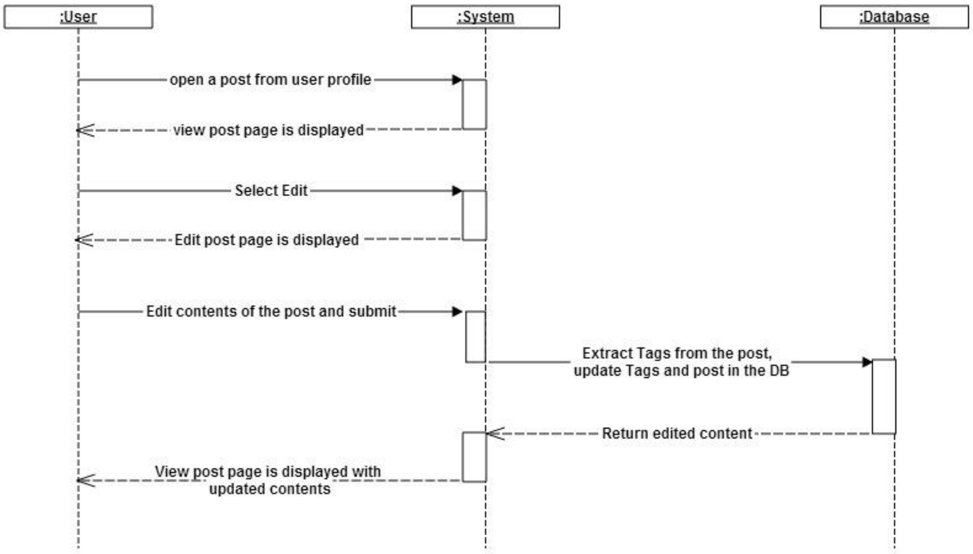

:User
:System
:Database
open a post from user profile
view post page is displayed
Select Edit
Edit post page is displayed
Edit contents of the post and submit
Extract Tags from the post, update Tags and post in the DB
Return edited content
View post page is displayed with updated contents

Figure 5.18: Sequence Diagram for Edit Post

Sequence Diagram for Edit Tag

Figure 5.19 shows a sequence diagram for Edit Tag. To edit a tag for a post a user must open a post from the user profile and click on edit. Edit post page is displayed where user can click “X” next to a tag to delete it from the post or click on “+” next to the tag list to add a tag. When user clicks on “+” add tag prompt is displayed where the user can enter a new tag and click on add. Once the tags are edited the database is updated by the system.

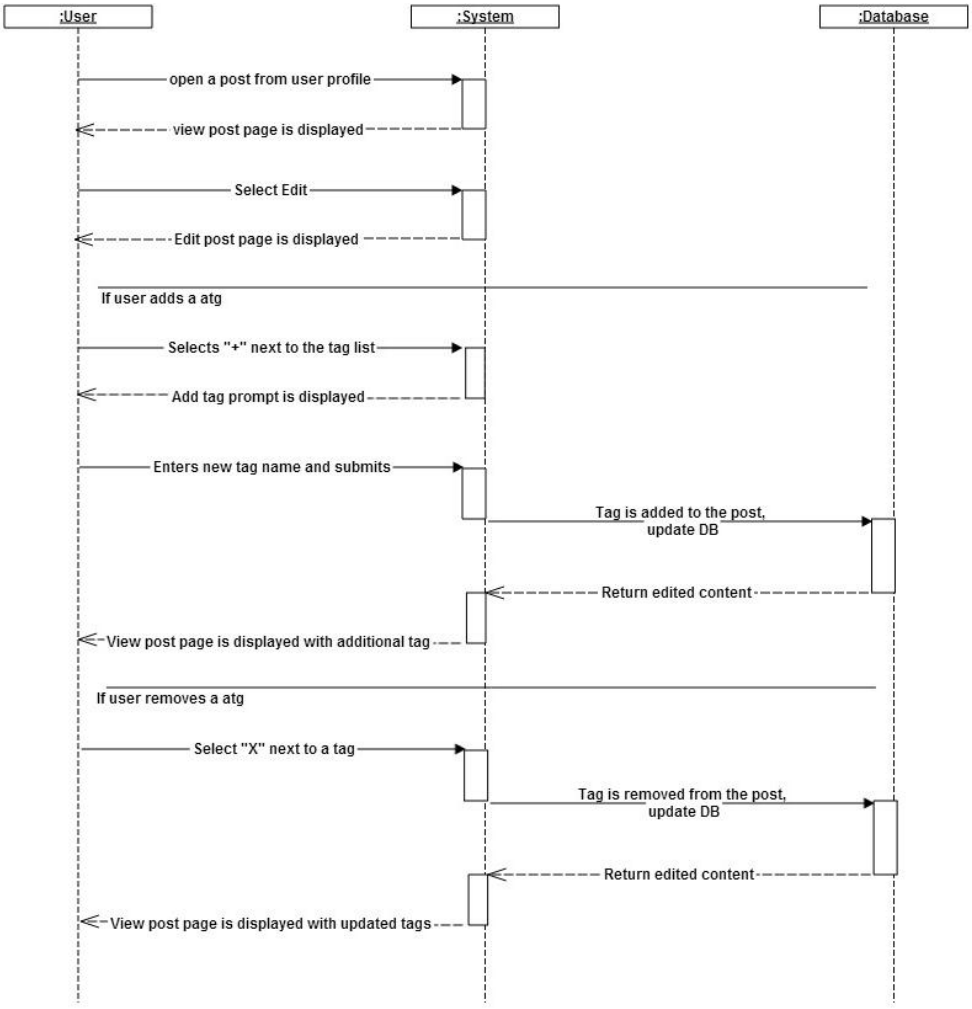

:User
:System
:Database
open a post from user profile
view post page is displayed
Select Edit
Edit post page is displayed
If user adds a atg
Selects "+" next to the tag list
Add tag prompt is displayed
Enters new tag name and submits
Tag is added to the post, update DB
Return edited content
View post page is displayed with additional tag
If user removes a atg
Select "X" next to a tag
Tag is removed from the post, update DB
Return edited content
View post page is displayed with updated tags

Figure 5.19: Sequence Diagram for Edit Tag

Sequence Diagram for Edit Comment

Figure 5.20 sows a sequence diagram for Edit comment. To edit a comment for a post a user must open a post. View post page is displayed where if the user is the author of the comment he/she can click "X" next to a comment to delete it from the post. To add a comment user can enter the comment in the comment box and click on submit. The system then adds the comment to the post, awards points to the user and generates notification for the author of the post. Once the comments are edited the database is updated by the system.

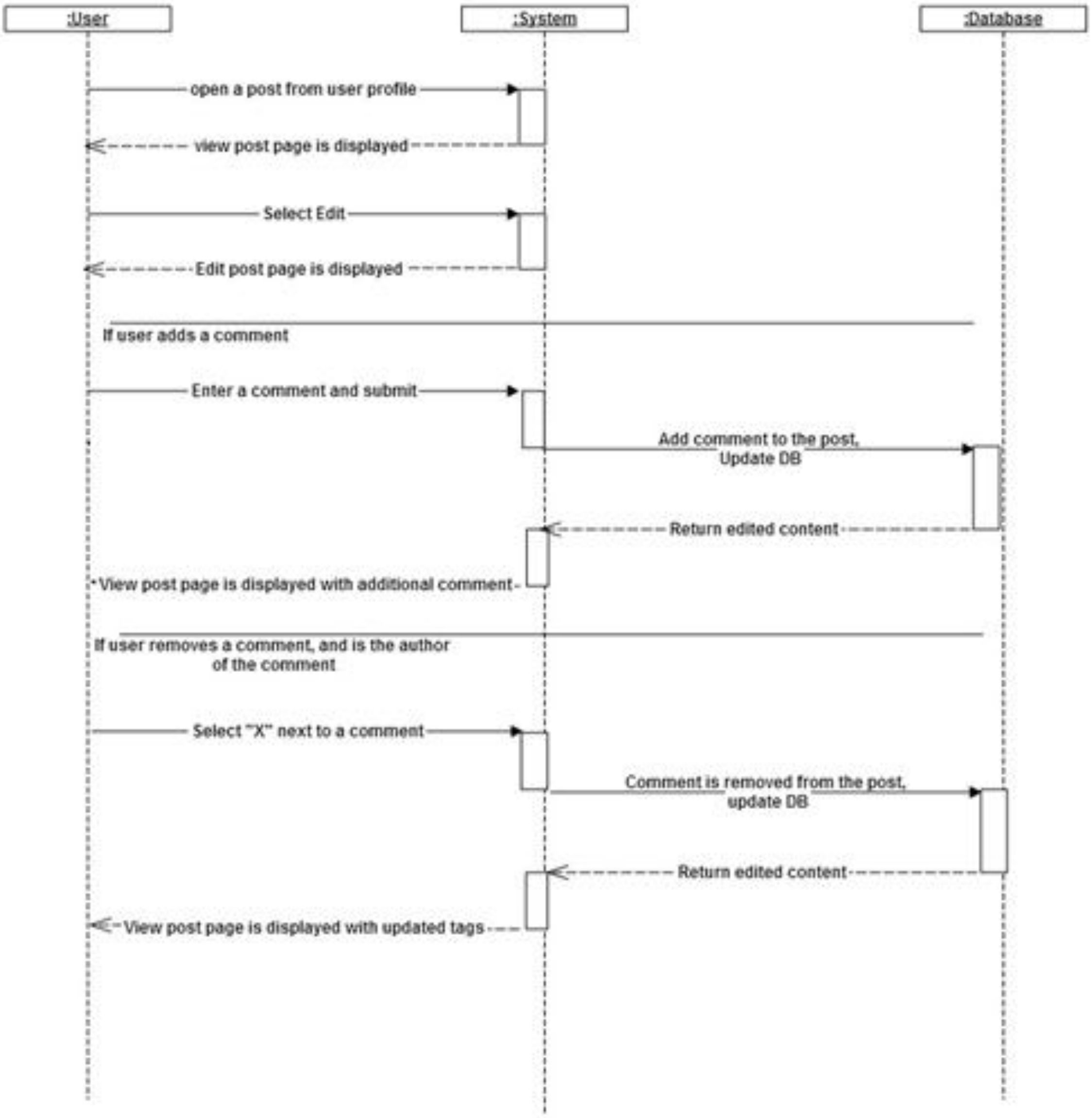

:User
:System
:Database
open a post from user profile
view post page is displayed
Select Edit
Edit post page is displayed
If user adds a comment
Enter a comment and submit
Add comment to the post, Update DB
Return edited content
View post page is displayed with additional comment
If user removes a comment, and is the author of the comment
Select "X" next to a comment
Comment is removed from the post, update DB
Return edited content
View post page is displayed with updated tags

Figure 5.20: Sequence Diagram for Edit Comment

Sequence Diagram for Edit Favorites

Figure 5.21 shows the sequence diagram for Edit favorites. When a user opens a post View post page is displayed where user can click on “Add to Favorite” icon to add a post to his/her favorites. The system adds the post to the user’s favorites, awards points to the user and sends a notification to the author of the post. If a user wants to remove a post from his/her favorites he/she must click on “remove from Favorite” icon on the favorite post, the system then removes the post from their favorites list. Once the editing is over the system updates the database.

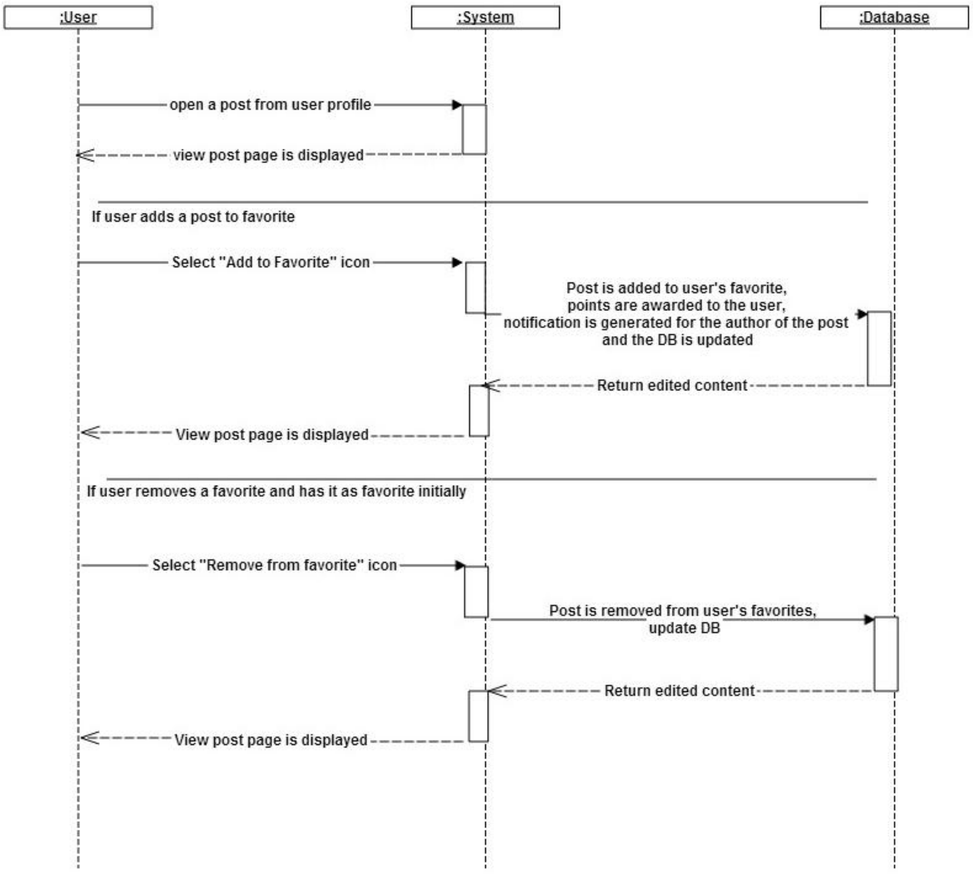

:User
:System
:Database
open a post from user profile
view post page is displayed
If user adds a post to favorite
Select "Add to Favorite" icon
Post is added to user's favorite, points are awarded to the user, notification is generated for the author of the post and the DB is updated
Return edited content
View post page is displayed
If user removes a favorite and has it as favorite initially
Select "Remove from favorite" icon
Post is removed from user's favorites, update DB
Return edited content
View post page is displayed

Figure 5.21: Sequence Diagram for Edit Favorites

Sequence Diagram for Suggest Post

Figure 5.22 shows the diagram for Suggest Post. To suggest a post a user must open a post and click on suggest icon. The system displays a prompt to the user to enter and select the name of another user to whom the post is to be suggested. The user selects and submits. The system then adds the points to the user profile, generates notification for the recipient of the suggestion and updates the database.

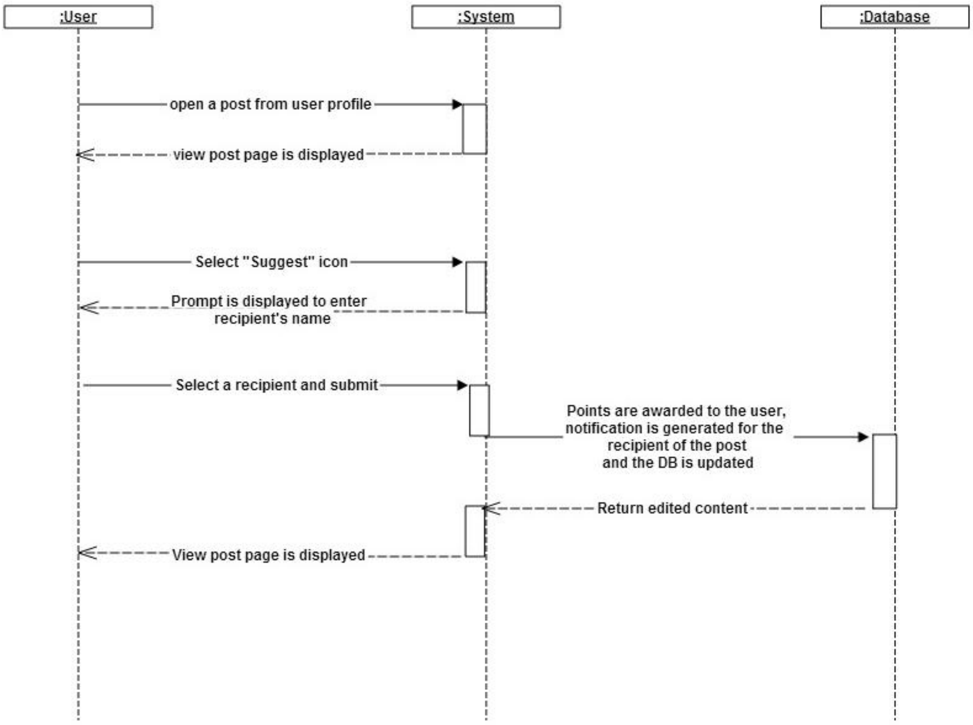

:User
:System
:Database
open a post from user profile
view post page is displayed
Select "Suggest" icon
Prompt is displayed to enter recipient's name
Select a recipient and submit
Points are awarded to the user, notification is generated for the recipient of the post and the DB is updated
Return edited content
View post page is displayed

Figure 5.22: Sequence Diagram for Suggest Post

Sequence Diagram for Like Post

Figure 5.23 shows the sequence diagram for Like Post. To like a a post user must open the post and click on the like icon. The system then awards points to the user, sends a notification to the author of the post and updates the database.

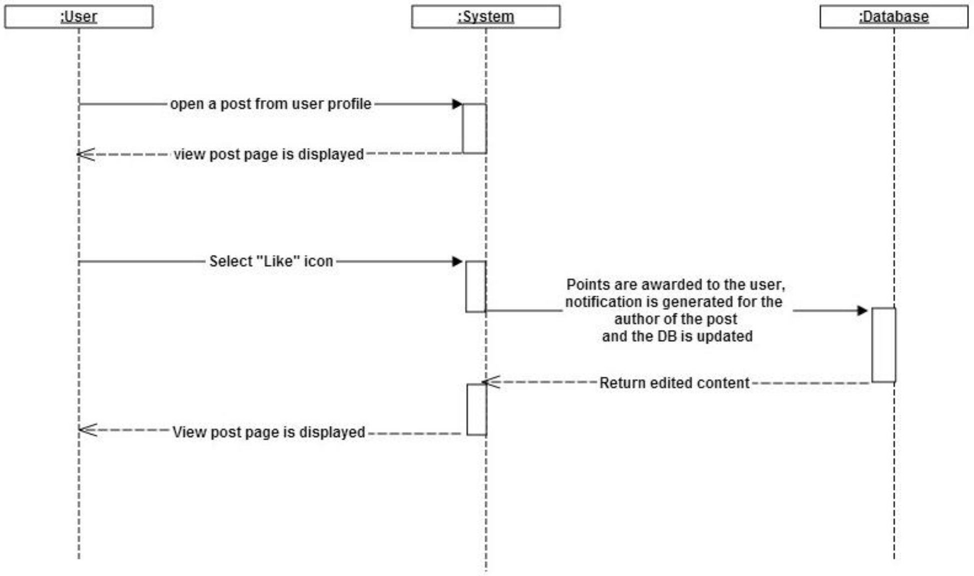

:User
:System
:Database
open a post from user profile
view post page is displayed
Select "Like" icon
Points are awarded to the user, notification is generated for the author of the post and the DB is updated
Return edited content
View post page is displayed

Figure 5.23: Sequence Diagram for Like Post

Sequence Diagram for Search Post

Figure 5.24 shows a sequence diagram for Search post. To search for a post the user must enter a search text in the search box and click on search icon. The system then returns the search results page with the results.

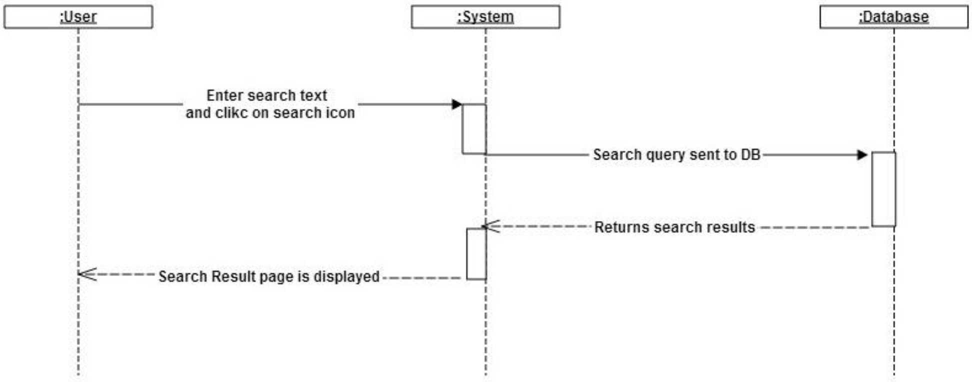

:User
:System
:Database
Enter search text and clikc on search icon
Search query sent to DB
Returns search results
Search Result page is displayed

Figure 5.24: Sequence Diagram for Search Post

Sequence Diagram for View Post

Figure 5.25 shows a sequence diagram for View post. To view a post a user must click on a post title displayed. The system then returns a view post page with the full ppost.

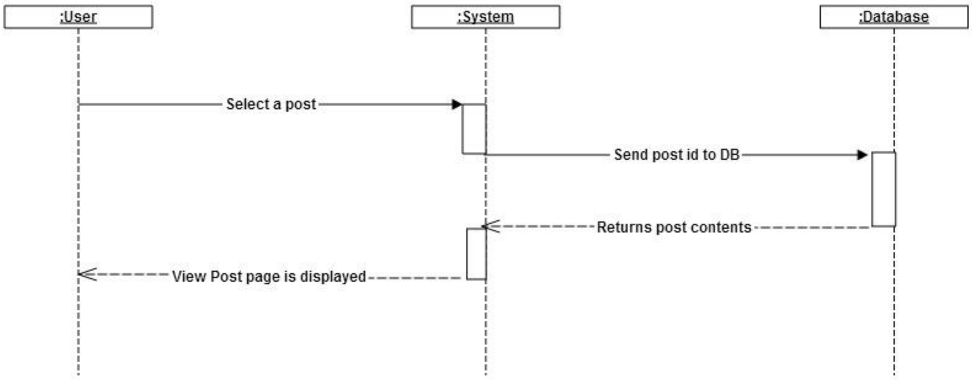


Figure 5.25: Sequence Diagram for View Post

Sequence Diagram for View Notification

Figure 5.26 shows a sequence diagram for View Notification. To view the notifications user must click on Notification tab. When clicked system returns the Notification page.

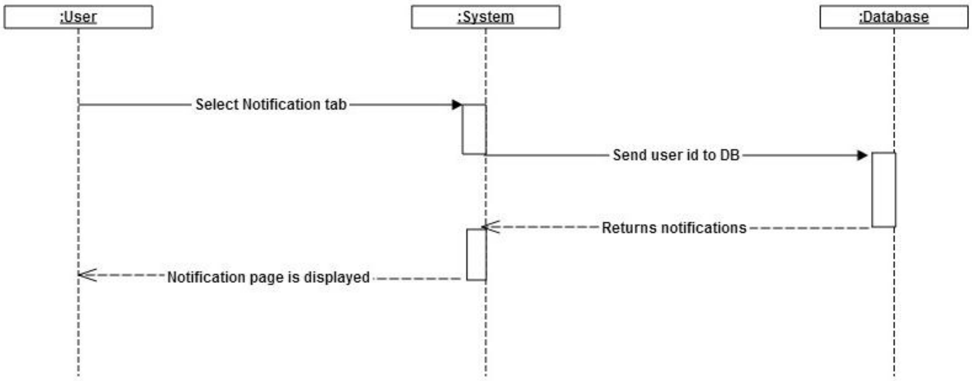


Figure 5.26: Sequence Diagram for View Notification

Sequence Diagram for View Profile

Figure 5.27 shows a sequence diagram for View Profile. To view his/her profile user must click on My Profile tab. The system returns the user profile page.

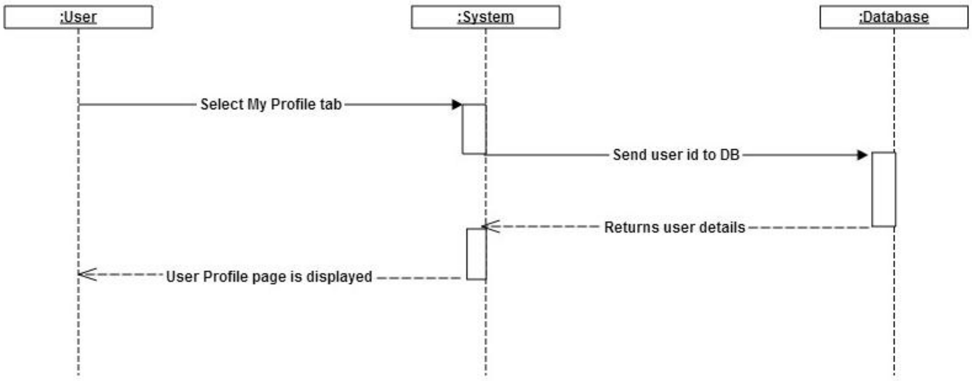


Figure 5.27: Sequence Diagram for View Profile

Sequence Diagram for View Suggested post

Figure 5.28 shows a sequence diagram for View Suggested post. To view all suggestions made to the user and from other users, user must click on Suggestions tab. The system then returns a Suggestions page.

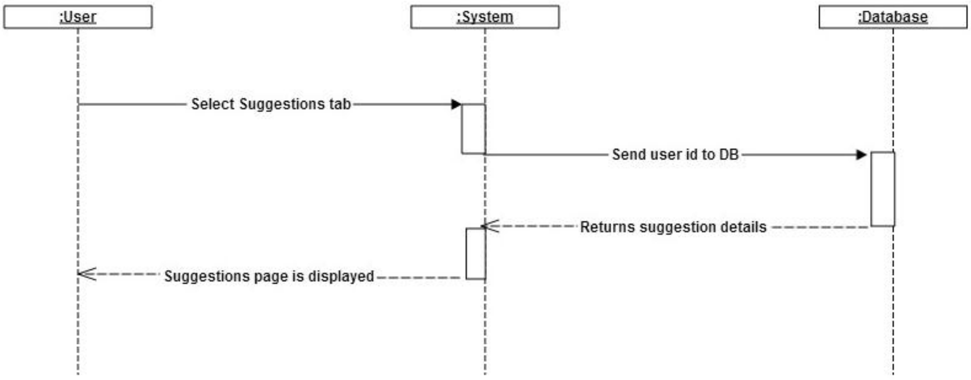


Figure 5.28: Sequence Diagram for View Suggested post

Sequence Diagram for View Favorite Post

Figure 5.29 shows the sequence diagram for Favorite Post. To view the favorite articles user must click on the Favorite tab. The system then returns the Favorites page to the user.

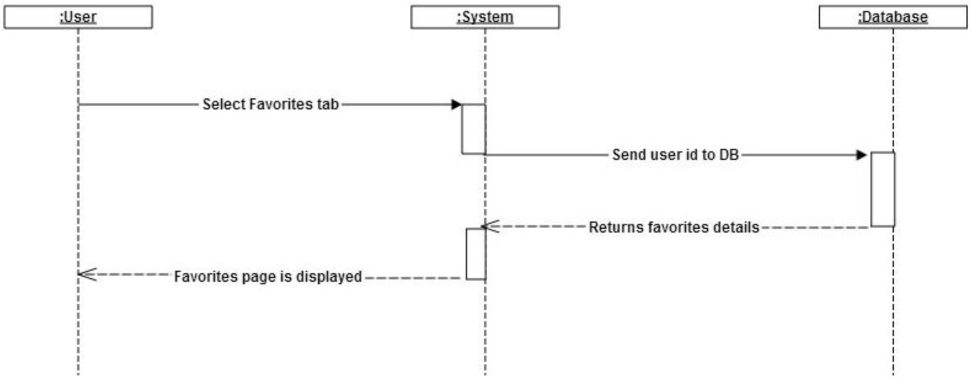


Figure 5.29: Sequence Diagram for View Favorite Post

Sequence Diagram for View Featured Post

Figure 5.30 shows the sequence diagram for View featured Post. To view featured posts user must click on the featured tab. The system then returns the Featured page.

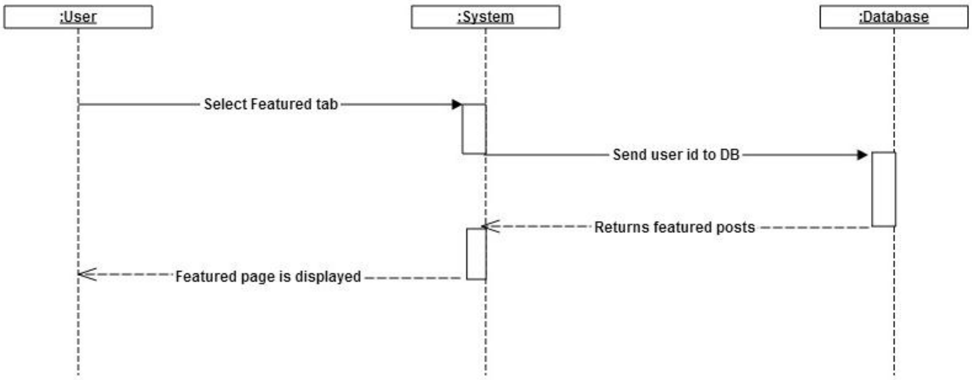


Figure 5.30: Sequence Diagram for View Featured Post

5.5 Data Design

Functional:

5.5.1 Class Diagram

A class diagram describes the classes that make up a system and the static relationships between them. Classes are defined in terms of their name, attributes (or data), and behaviors (or methods). The static relationships are association, aggregation, and inheritance.

Classes are typically modeled as rectangles with three sections: the top section for the name of the class, the middle section for the attributes of the class, and the bottom section for the methods of the class.

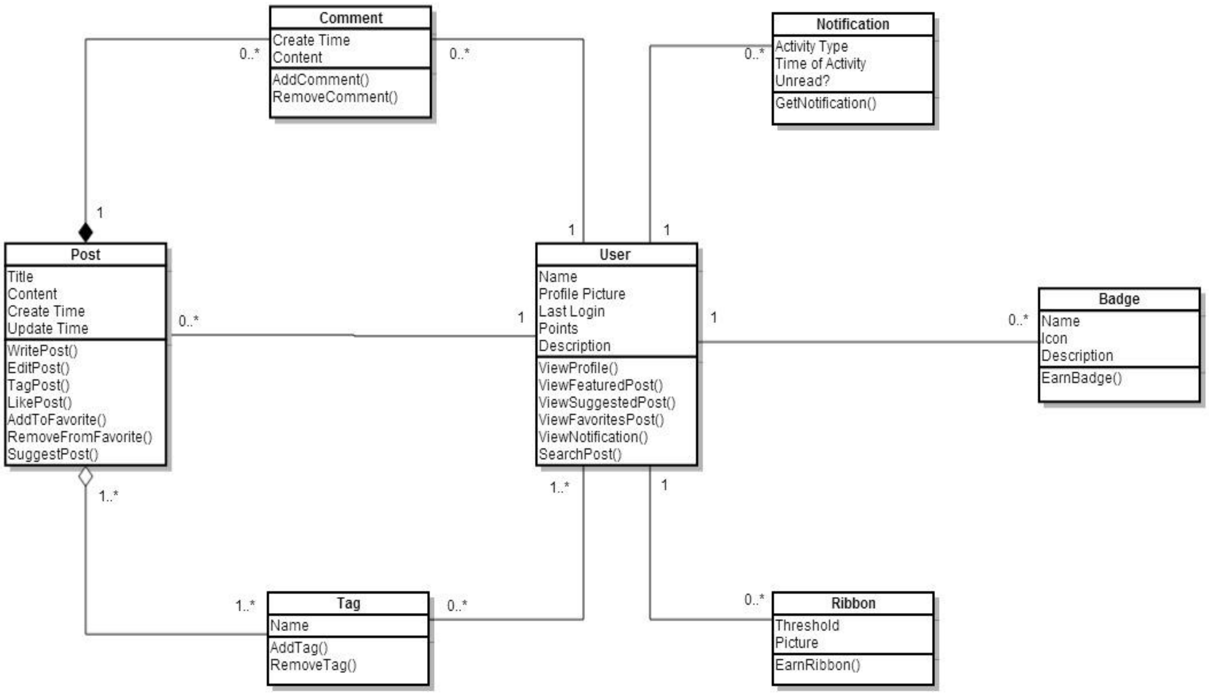


Figure 5.28: Class Diagram

5.5.2 Logical Data Model

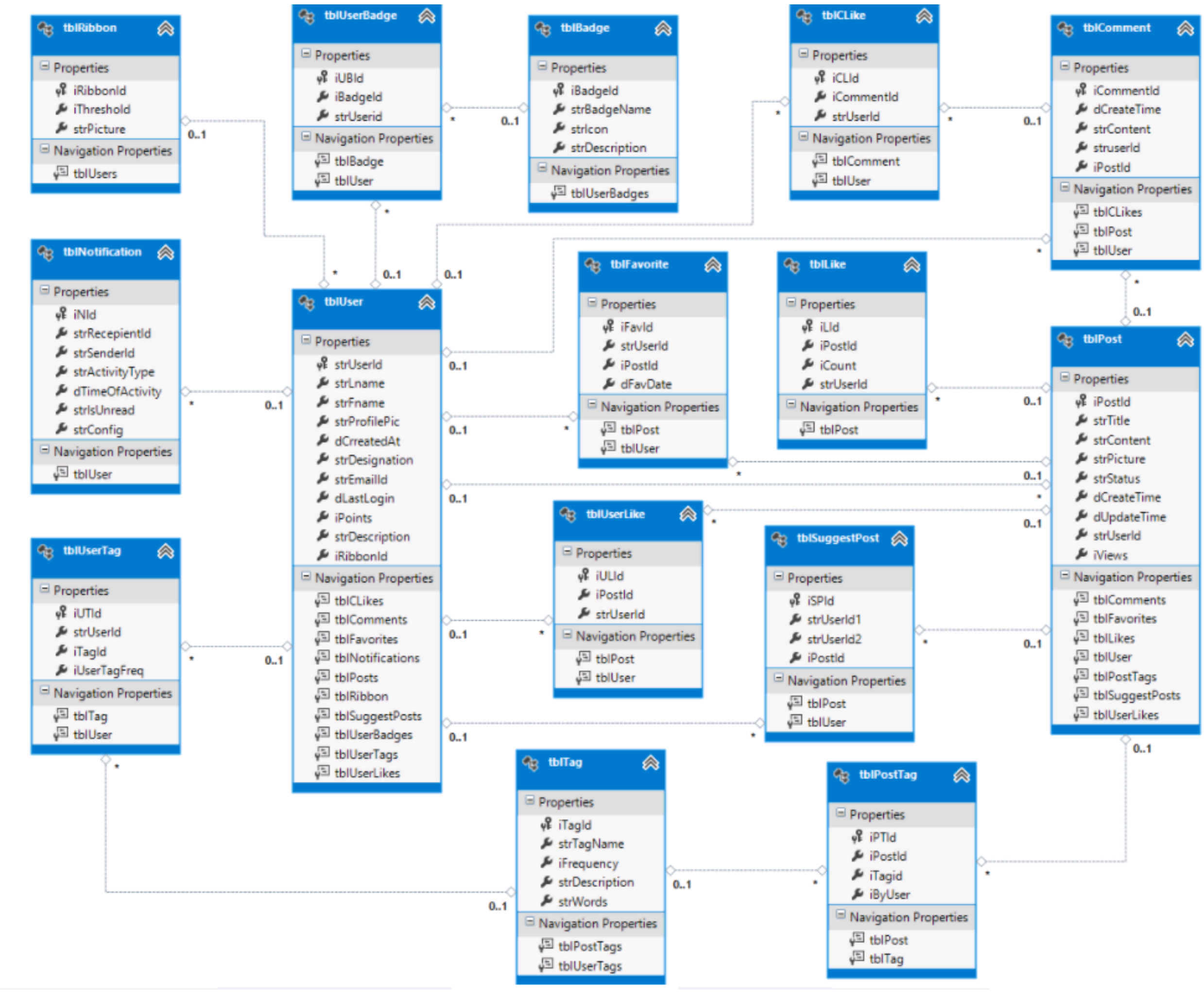


Figure 5.29: Database Design diagram

5.6 Human Interface Design

5.6.1 Overview of the User Interface

User interacts with the system through Graphical user Interface. The user selects input like new post, Registration, Enquiry, products, services, what we do etc.

Systems requirements associated with all of the user interfaces.

1. Ease of learning.

   - The system must be easy to learn for both novices and users with experience from similar systems.

2. Task efficiency.

   - The system must be efficient for the frequent user.

3. Ease of remembering.

   - The system must be easy to remember for the casual user.

4. Understandability.

   - The user must understand what the system does.

Implementation

6. Implementation

6.1 Write Post

Step 1: Start

Step 2: Open the Smart Blogger Application.

Step 3: Click on “Write Article” tab.

New post page is displayed.

Step 4: Fill in the fields to write a post.

Step 6: Click on “Post” button.

Step 7: Stop.

6.2 Extract Tags

Step 1: Start

Step 2: Preprocessing: Stem words by Porter algorithm and extract phrases based on the APriori algorithm. We extract phrases of up to 4 words with frequency more than 3 times. Discard stop words included in stop list used in the SMART system.

Step 3: Selection of frequent terms: Select the top frequent terms up to 30% of the number of running terms.

Step 4: Clustering frequent terms: Cluster a pair of terms whose Jensen-Shannon divergence is above the threshold (0.95 × log 2).

Step 5: Calculation of expected probability: Count the number of terms co-occurring to yield the expected probability.

Step 6: Calculation of $\chi 2$ value: For each term, count co-occurrence frequency. Count the total number of terms in the sentences. Calculate the $\chi 2$ value

Step 7: Output keywords: Show a given number of terms having the largest $\chi 2$ value.

Step 8: Stop

6.3 Edit Post

Step 1: Start.

Step 2: Click on “My Profile” tab.

Step 3: Open a post from User Posts.

Step 4: Click on “Edit”.

Edit Post page is displayed.

Step 5: Edit the contents of the post.

Step 6: Click on “Save”.

Step 7: Stop.

6.4 Edit Tag

Step 1: Start.

Step 2: Click on “My Profile” tab.

Step 3: Open a post from User Posts.

Step 4: Click on “Edit”.

```
          Edit Post page is displayed.
Step 5: If (add tag)
                A: Click on "+" next to the tag list.
                    Add new tag prompt displays.
                B: Enter a tag name and click on "Add".
                    The tag is added to the post.
        Else if (remove tag)
                A: Click on "X" next to a tag.
                     The tag is removed from the post.
Step 6: Stop
```

6.5 Edit Comment

```
Step 1: Start.
Step 2: Open a post.
Step 3: if (add comment)
                A: Enter a comment in comment box.
                B: Click on "Comment" button".
                  Comment is added to the post.
          Else if (remove comment)
                A: if (user is the author of the comment)
                        Click on "X" next to the comment.
                        The comment is removed from the post.
                    Else
                        Goto step 4.
Step 4: Stop.
```

6.6 Edit Favorites

```
Step 1: Start
Step 2: Open a post.
Step 3: If (add to favorite)
            If (post is not favorite)
                Click on “Add to Favorite” icon.
                Post is added to favorites.
            Else
                Goto step 4.
        Else if (remove from favorite)
            If (post is favorite)
                Click on “Remove from favorite” icon.
                Post is removed from favorites.
            Else
                Goto step 4.
Step 4: Stop.
```

6.7 Suggest Post

```
Step 1: Start
Step 2: Open a post.
Step 3: Click on “Suggest” icon.
        Prompt is displayed to enter recipient’s name.
Step 4: Select a recipient.
Step 5: click on “Suggest” button”.
        Post gets added to suggestions.
Step 6: Stop.
```

6.8 Earn Points

```
Step 1: Start.
Step 2: if (user writes a new post)
            Add 50 points to user’s profile.
Step 3: if (user writes a comment)
            Add 40 points to user’s profile.
Step 4: if (user suggests a post)
            Add 30 points to user’s profile.
Step 5: if (user favorites a post)
            Add 20 points to user’s profile.
```

```
Step 6: if (user likes a post)
            Add 10 points to user’s profile.
Step 7: Stop.
```

6.9 Generate Notification

```
Step 1: Start.
Step 2: if (user writes a comment)
            Generate a notification for the author of the post.
Step 4: if (user suggests a post)
            Generate a notification for the recipient.
Step 5: if (user favorites a post)
            Generate a notification for the author of the post.
Step 6: if (user likes a post)
            Generate a notification for the author of the post.
Step 7: Stop.
```

6.10 Award Badges and Ribbons

```
Step 1: Start.

Step 2: If (user earns points)

            A.  Check threshold for a new badge.
            B.  If (threshold reached)
                    Award Badge.
                    Generate Notification.
            C.  Check Threshold for a new ribbon.
            D.   If (threshold reached)
                    Award Ribbon.
                    Generate Notification.

Step 3: Stop.
```

Testing

7. Testing

7.1 Introduction

The aim of testing process is to identify all defects in a software product. Testing is any sequence aimed at evaluating the software for quality result it produces and the quality of result it handle. Testing is an operation to detect the differences between the approval result and actual result.

7.2 Test Case Design:

Testing is conducted in this project such that every function is fully tested and searched for errors .the testing process is performed by white Box Testing

7.2.1 White Box Testing

For This system white box test case is used by knowing all the internal working of the product. The internal working performs to specification and all internal components like connection inside the network have been adequately exercised

Inserted value for the variables collected through forms and tested all pages testing and hyperlink testing is done and ensured that everything is working properly

7.3 Testing Strategies

7.3.1 Unit Testing

Unit testing is used in this system for testing of different units or module.it is this system responsibility to think of advantages of doing unit testing before integration testing makes debugging easier. In this system unit testing has been exclusively done after each module.

7.3.2 Integration Testing

This testing process is used to test the module interface in order to ensure there are errors in the parameter passing when one module involves another module.

7.3.2 System Testing

The system Testing uses this testing methodology to meet the requirements system Testing are intended to prove that the system meets its objectives. As per the client requirements all the module are developed and ensured that they will meet all the requirements.

7.3.2 Acceptance Testing

This System uses this methodology to ensure that the user’s satisfaction and the whole product is user friendly. In our blogging Application System we have ensured to provide user friendly interface, which is ease of use, no complex training is required to use this system. With minimal knowledge one can use this system and make best out of it.it will meets the business and technical requirements that guided its design and development, works as expected and can be implemented with the same characteristics.

7.4 Test Cases

Test case #	**Test case Name**	**Test case Description**	**Inputs**	**Expected Output**	**Actual Output**	**Statu**
1	User_View_Post	Correct post should display	User clicks on a post title	Relevant post should open	Relevant post opens	Pass
2	User_Search_Post	Search should display correct results	1. User enter a search text 2. User click on search icon	All posts containing the search text should display	All posts containg the search text displays	Pass
3	User_Search_Author	Search should display relevant author to search text	1. User enter a search text 2. User click on search icon	All authors’ name who has written a post containing the search	All authors’ name who has written a post containing the search text displays	Pass

				text should display		
4	User_Make_Comment	A comment when posted should be displayed	User writes a comment on a post and submits	The comment should show up instantly in the comments section	ZThe comment shows up instantly in the comments section	Pass
5	Post_Like_Count	Number of likes should increase when a user likes a post	User clicks on like icon for a post	Number of likes should increase by 1	Number of likes increases by 1	Pass
6	Post_View_Count	Number of views should increase when user views a post	User opens a post to view	Number of views should increase by 1	Number of views increases by 1	Pass
7	Post_Favorite_Count	Number of times a post has been added to favorite should increase when user adds a post to favorite	User adds a post to favorite	Number of times the post has been added to favorite should increase by 1	Number of times the post has been added to favorite increase by 1	Pass
8	Post_Suggest_Count	Number of times the post has been	User suggests a post to	Number of times the post has	Number of times the post has	Pass

		suggested should increase when user suggest the post to another user	another user	been suggested should increase by 1	been suggested increase by 1	

Reports

8 Reports

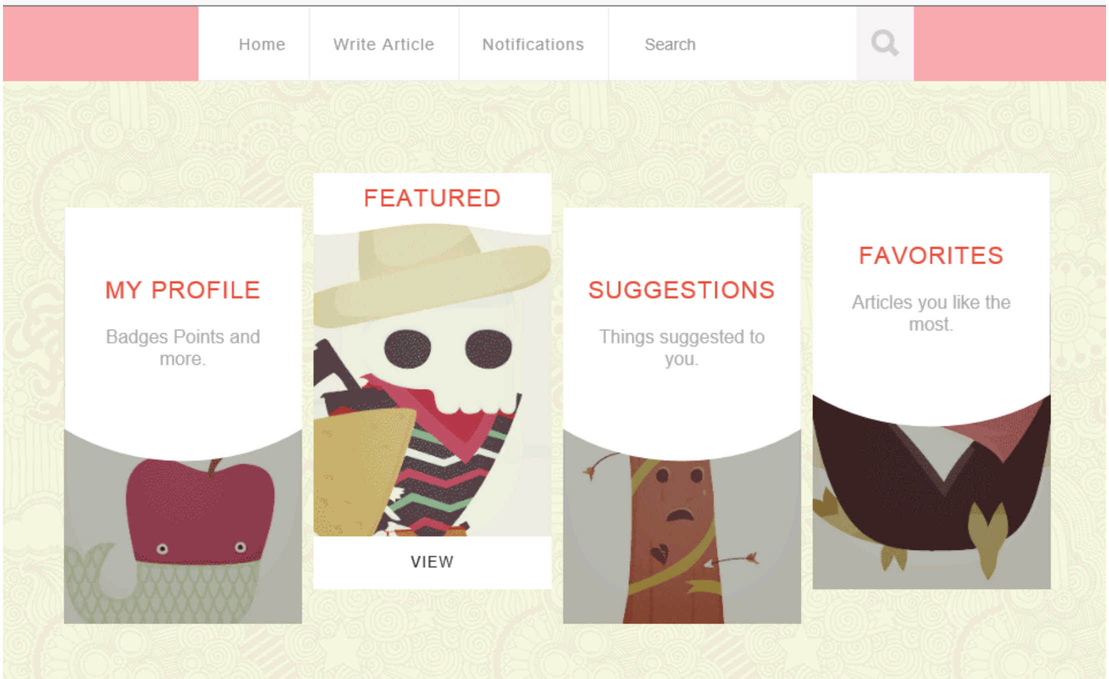
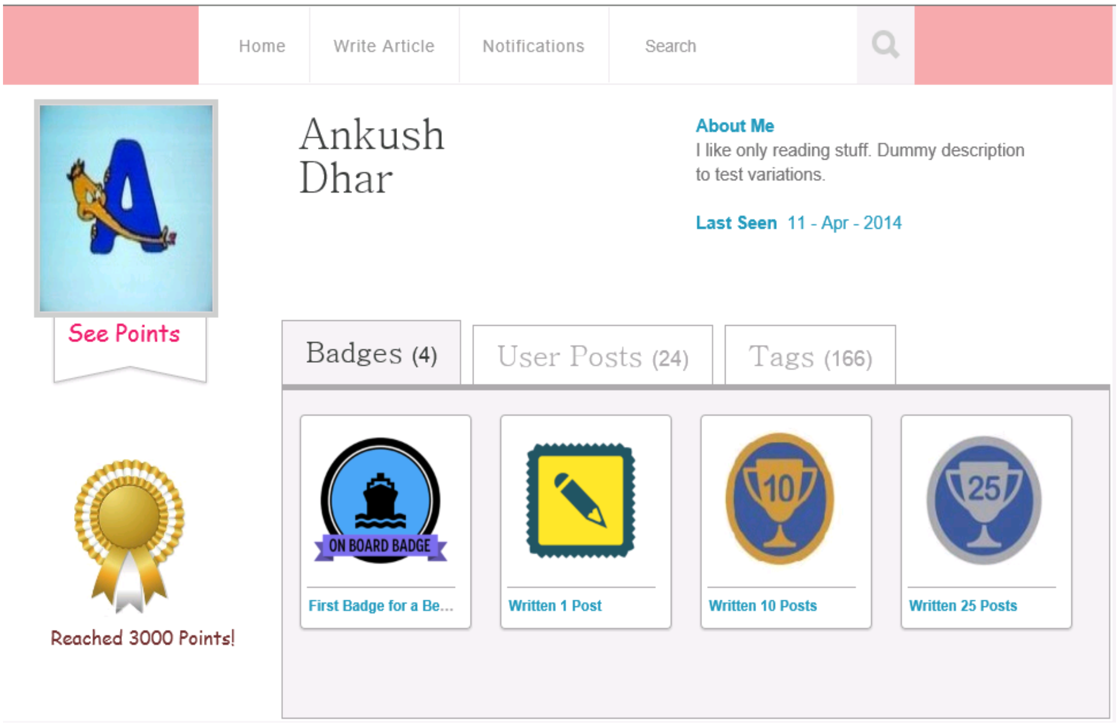


Figure 8.1: Home.aspx

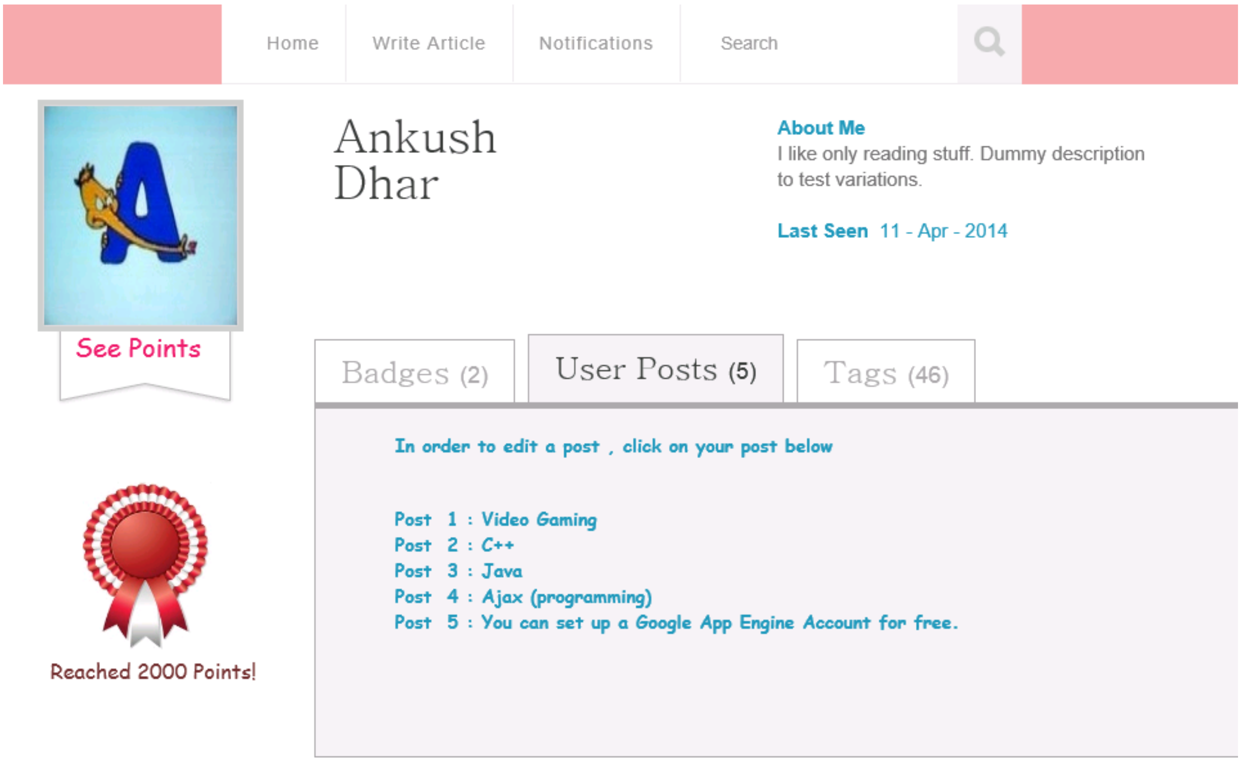


Ankush Dhar

About Me
I like only reading stuff. Dummy description to test variations.

Last Seen 11 - Apr - 2014

Figure 8.2: UserProfile.aspx



Ankush Dhar

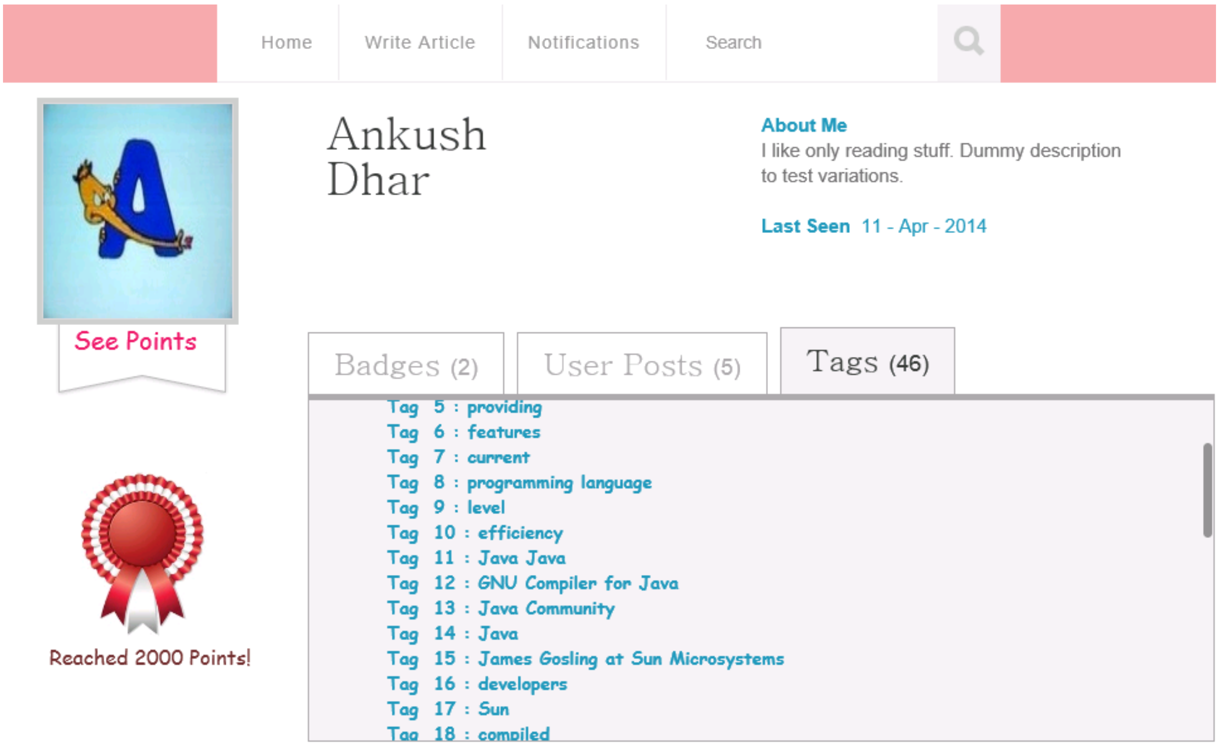
About Me
I like only reading stuff. Dummy description to test variations.

Last Seen 11 - Apr - 2014

In order to edit a post , click on your post below

Post 1 : Video Gaming
Post 2 : C++
Post 3 : Java
Post 4 : Ajax (programming)
Post 5 : You can set up a Google App Engine Account for free.

Figure 8.3: UserProfile.aspx



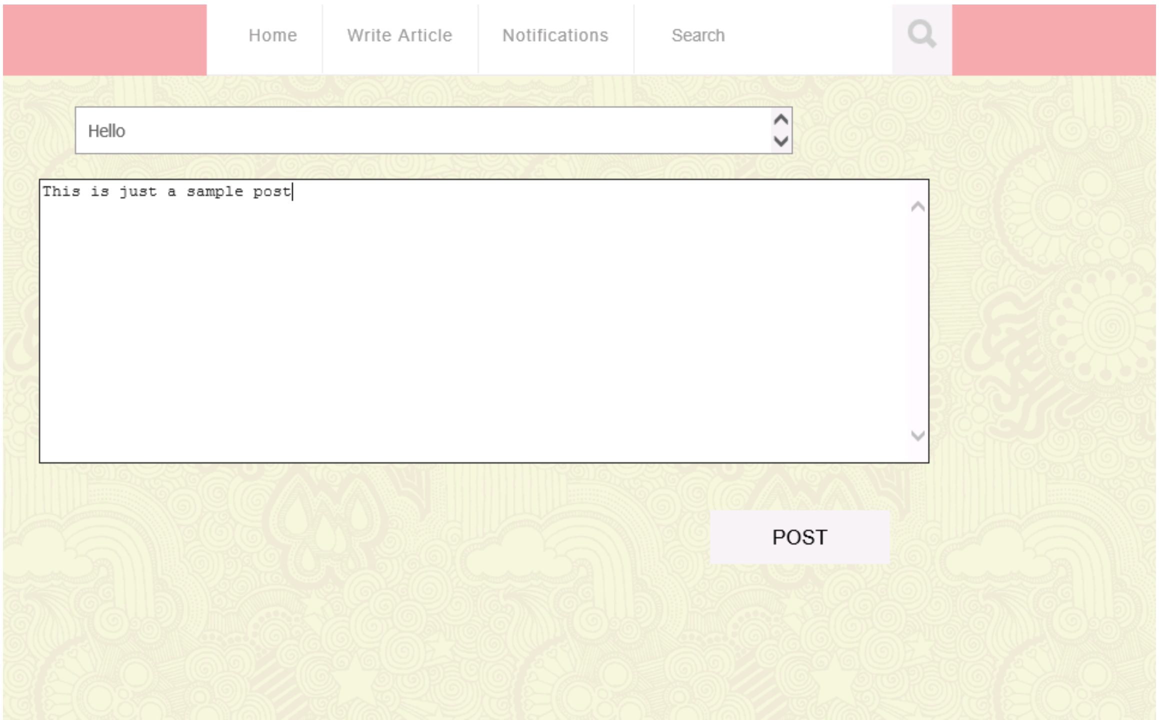
Ankush Dhar

About Me
I like only reading stuff. Dummy description to test variations.

Last Seen 11 - Apr - 2014

Tag 5 : providing
Tag 6 : features
Tag 7 : current
Tag 8 : programming language
Tag 9 : level
Tag 10 : efficiency
Tag 11 : Java Java
Tag 12 : GNU Compiler for Java
Tag 13 : Java Community
Tag 14 : Java
Tag 15 : James Gosling at Sun Microsystems
Tag 16 : developers
Tag 17 : Sun
Tag 18 : compiled

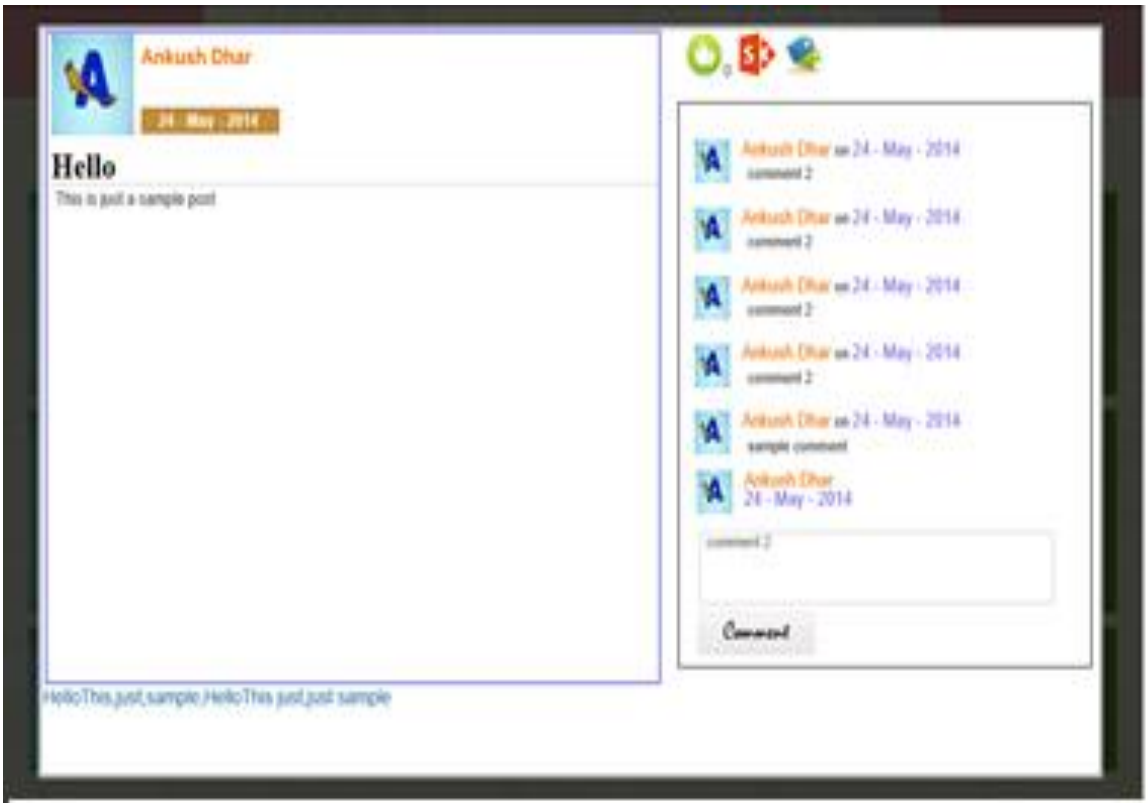
Figure 8.4 UserProfile.aspx



Hello

This is just a sample post

Figure 8.5: WritePost.aspx



Ankush Dhar

24 - May - 2014

Hello

This is just a sample post

Ankush Dhar on 24 - May - 2014
comment 2

Ankush Dhar on 24 - May - 2014
comment 2

Ankush Dhar on 24 - May - 2014
comment 2

Ankush Dhar on 24 - May - 2014
comment 2

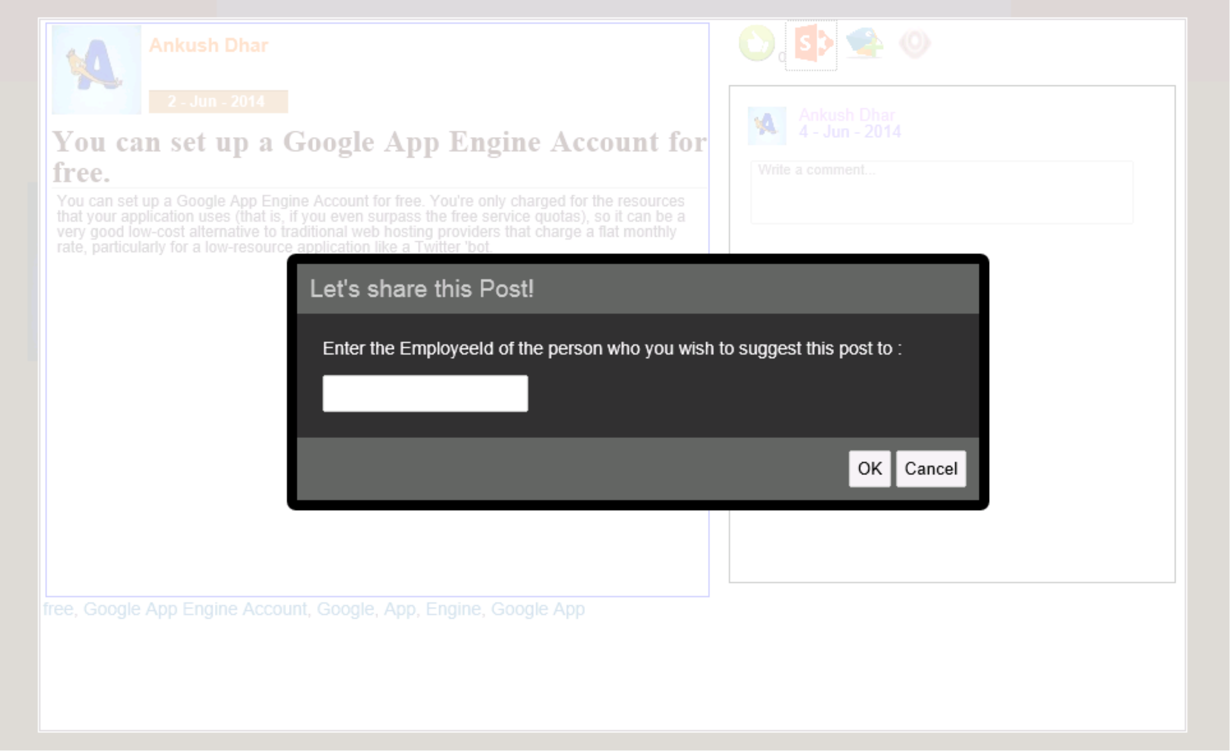
Ankush Dhar on 24 - May - 2014
sample comment

Ankush Dhar
24 - May - 2014

comment 2

HelloThis just sample HelloThis just just sample

Figure 8.6: ViewPost.aspx



Ankush Dhar

2 - Jun - 2014

You can set up a Google App Engine Account for free.

You can set up a Google App Engine Account for free. You're only charged for the resources that your application uses (that is, if you even surpass the free service quotas), so it can be a very good low-cost alternative to traditional web hosting providers that charge a flat monthly rate, particularly for a low-resource application like a Twitter 'bot.

Ankush Dhar
4 - Jun - 2014

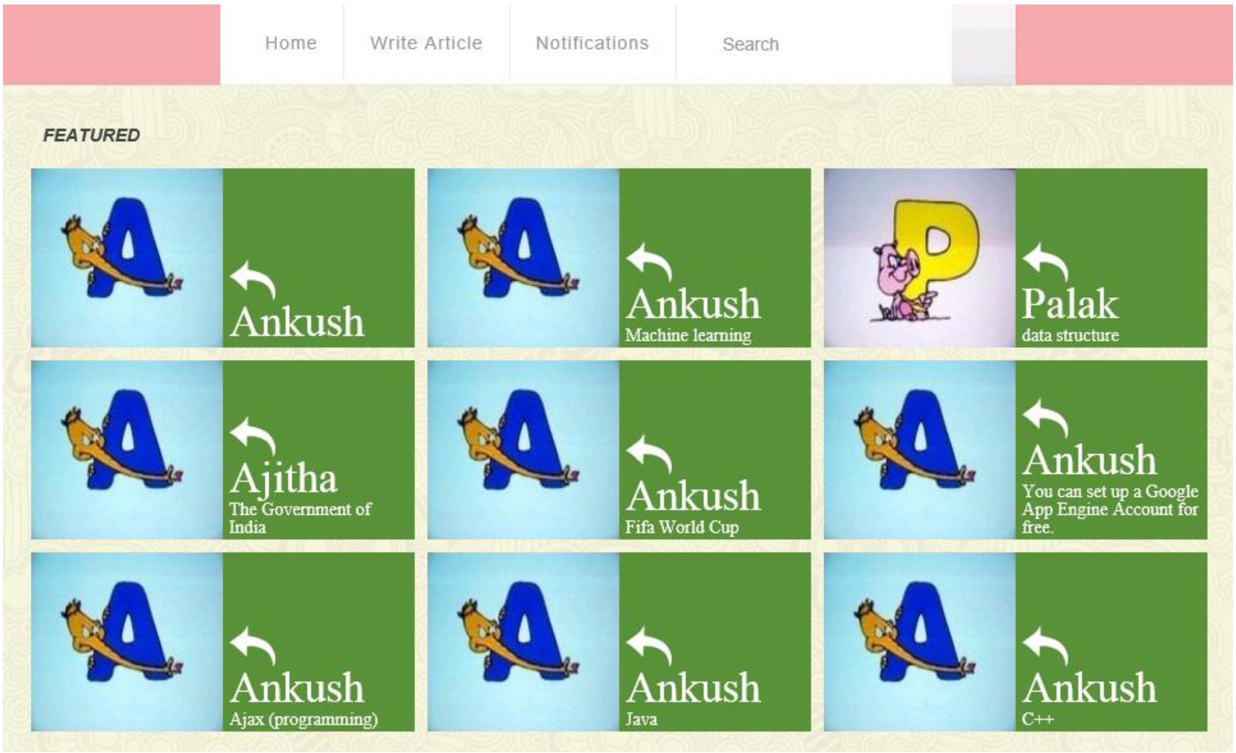
Write a comment...

Let's share this Post!

Enter the EmployeeId of the person who you wish to suggest this post to :

OK
Cancel
free, Google App Engine Account, Google, App, Engine, Google App

Figure 8.7 ViewPost.aspx (Suggest Prompt)



Ankush

Ankush
Machine learning

Palak
data structure

Ajitha
The Government of India

Ankush
Fifa World Cup

Ankush
You can set up a Google App Engine Account for free.

Ankush
Ajax (programming)

Ankush
Java

Ankush
C++

Figure 8.8: FeaturedPost.aspx

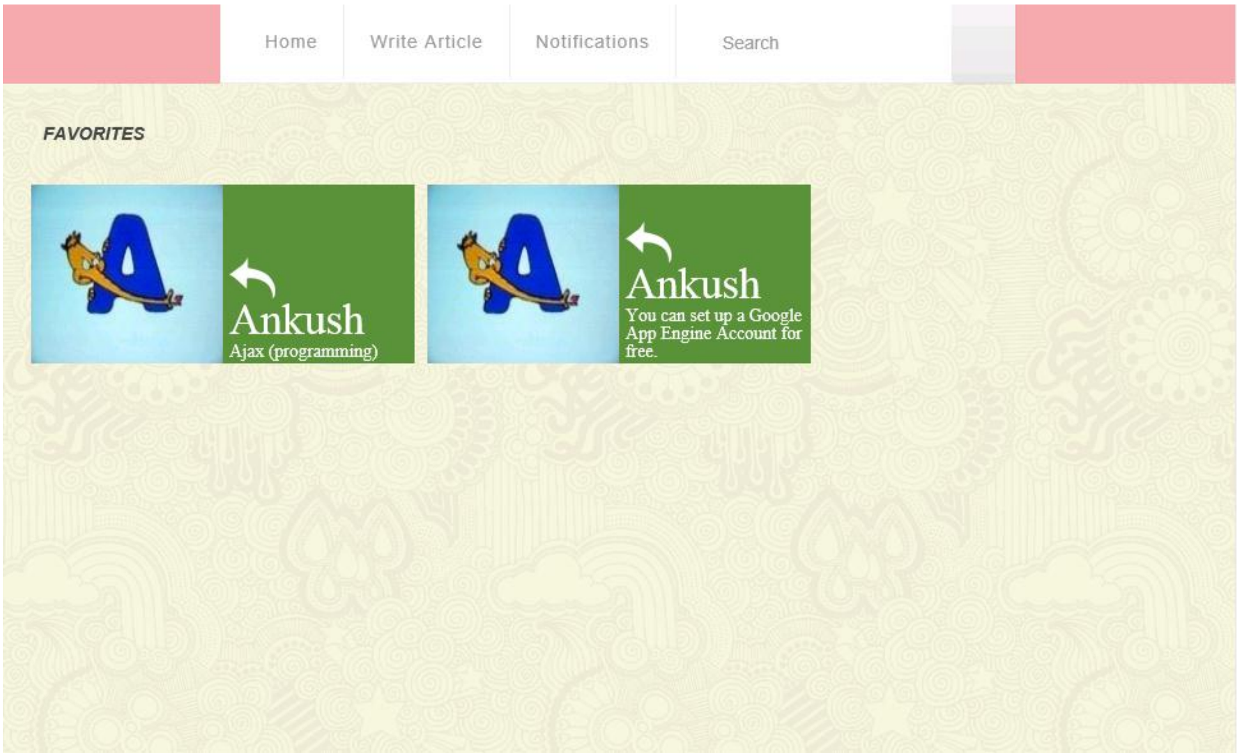


Figure 8.9: Favorites.aspx

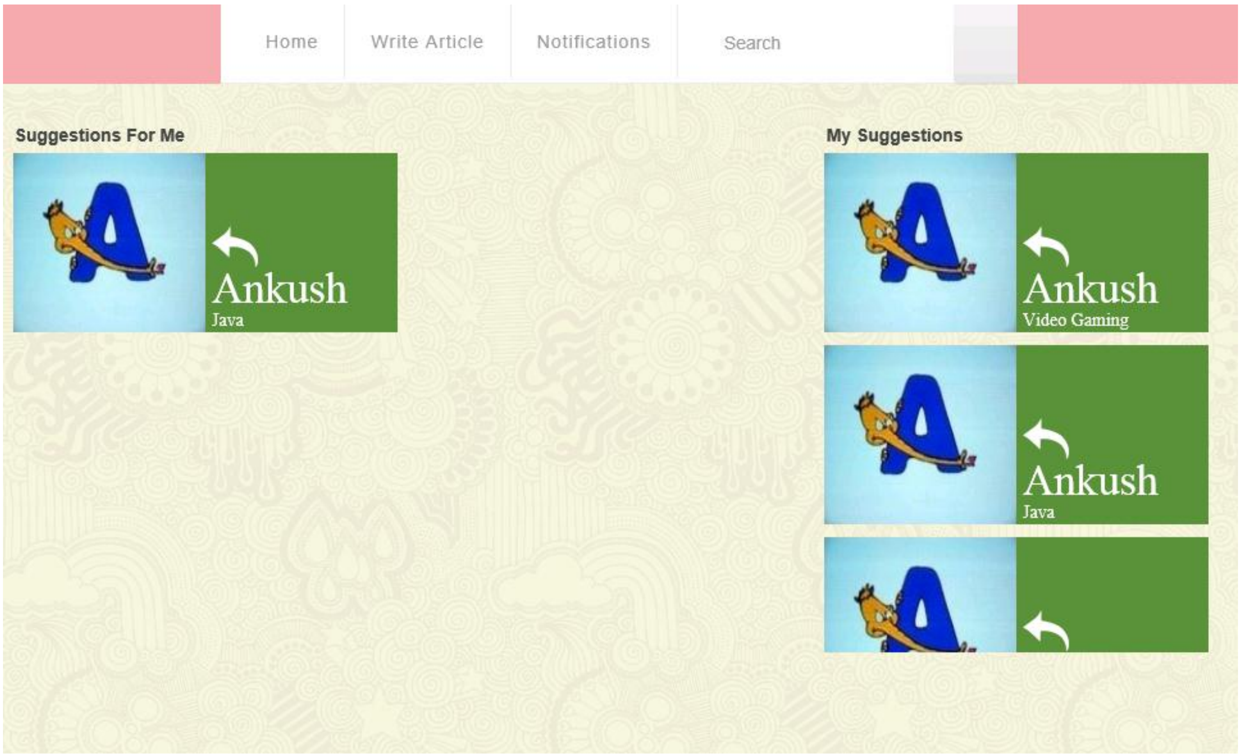


Figure 8.10: Suggestions.asp

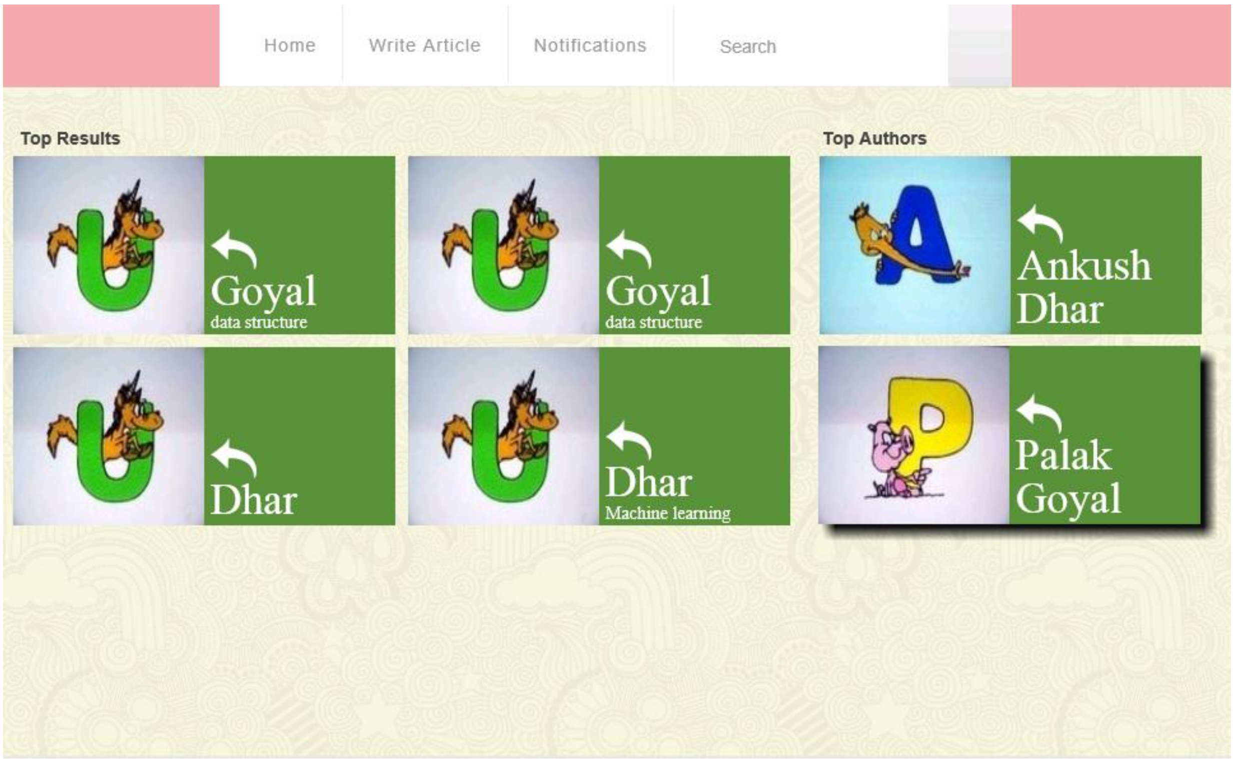
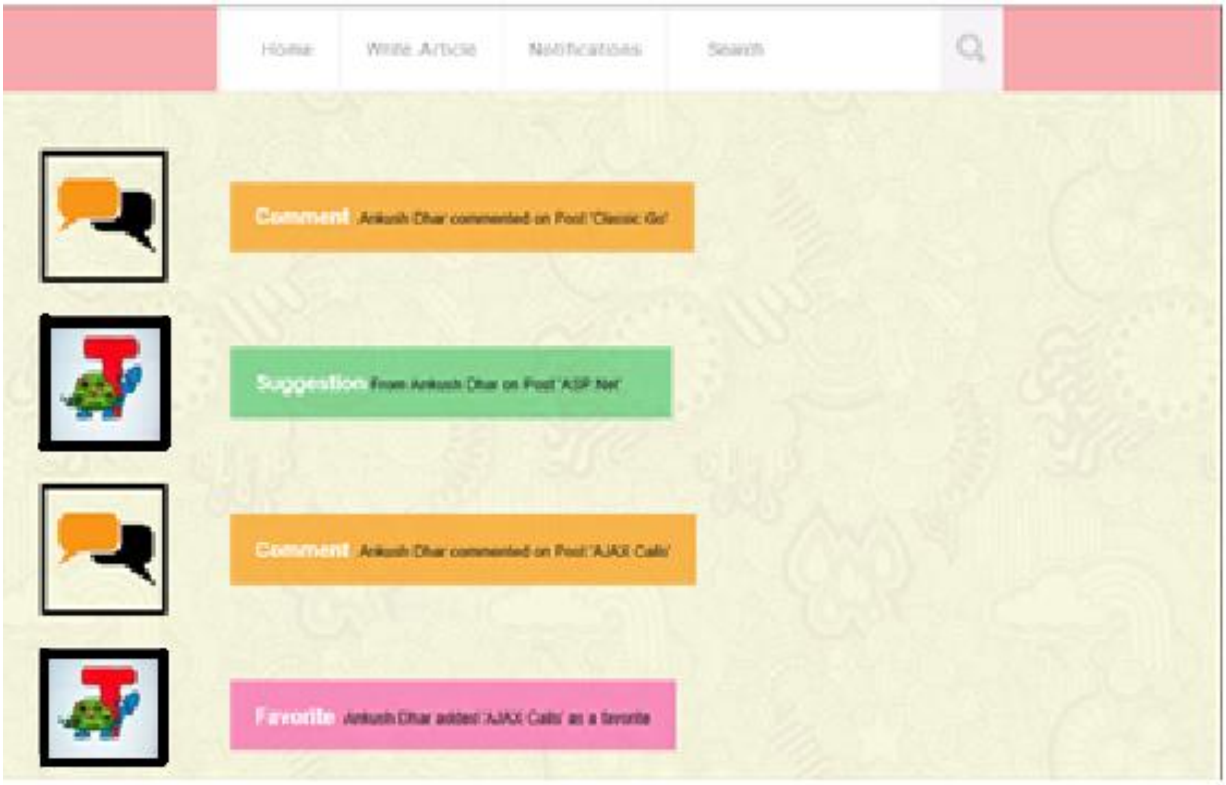


Figure 8.11: SearchResults.aspx



Comment :Ankush Dhar commented on Post 'Classic Go'

Suggestion From Ankush Dhar on Post 'ASP.Net'

Comment :Ankush Dhar commented on Post 'AJAX Calls'

Favorite Ankush Dhar added 'AJAX Calls' as a favorite

Figure 8.12: Notification.asp

9. Conclusion

The project is successfully implemented considering all functional requirements; the system is checked with various constraints. Although the system is working fine under the given platform provided by the company, but it has to go to the real environment where it has to be successfully implemented. Overall conclusion is that after implementing local policies, rules and security the system is working fine.

Smart Blogger still lacks lot of features that high end blog users demand. For example password protected post, gallery, Blog by email etc.

10. Future Scope and Future Enhancements of the project

- Ability to include media in a blog post.
- Ability to tag picture.
- Ability to receive notifications on phone when user is offline.

11. Bibliography

Books:

[1] Head First HTML and CSS

[2] Learning jQuery, Fourth Edition

[3] ASP.Net Professional – By: *Scott Mitchell*

[4] C Sharp 3.0 – The Complete Reference – By: *Herbert Schildt*

Web Sites:

http://www.w3schools.com/jQuery

http://en.wikipedia.org/wiki/Blogger_%28service%29

http://blogsofnote.blogspot.in/

http://dev.digi-corp.com/category/aspnet/

https://dannyacuna.wordpress.com/tag/n-tier-application-design/

12. Glossary

- **1C – One Cognizant Platform**

  One Cognizant Platform has been envisioned to provide next generation of information workplace to Cognizant associates, by addressing needs around user context, seamlessness, process centricity, multi-modality, personalization, rich interaction and social computing based on key features of Web2.0.

- **Blog**

  A personal website or web page, on which an individual records opinions, links to other sites, etc. on a regular basis.

- **Post**

  An entry written and published to a blog.