Journal of Computer Engineering & Information Technology

Research Article

Design and Development of an Ajax Enabled Knowledge Sharing System

Jennifer Mosheshe*

Abstract

The paper seeks to describe the design and development process of an Ajax enabled knowledge sharing system using ASP. Net, which provides detailed up-to-date Questions and Answers information on different popular topics. Using Object Oriented Web Modelling approach we were able to model the system requirement specification to generate Use case, Class object and navigational diagrams which are used as a blue print for the system development. In particular more attention was given to the ability of new visitors and registered members to Ask Questions by posting their questions on visited destinations to alert other members who have the answers to their required questions in the Questions and Answers Forum. The result shows that with the use of AJAX, the developed system allowed various students, professionals and the general public to receive information from other individuals online as required. Furthermore, the system design, Ajax and ASP.Net framework were implemented for web content management in order to develop the system. Using ASP.Net several API (Application Programming Interface) were integrated inform of module extensions to develop the user interface functionalities and collect real-time information from other websites inform of RSS feeds and weather report. Also the system utilized hierarchy base structural model to develop the system interface layout allowing us to breakdown the system interface into sections consisting of categories and articles.

Keywords

AJAX enabled knowledge sharing System; ASP.Net Framework; Object oriented web modelling; Application programming interfaces; Hierarchy modelling; Entity relationship diagram; Use case

Introduction

The rise of web 2.0 has increased which makes it possible to develop web applications that increase the rate of interactivity between users and dynamic web applications. This paper describes the use of Ajax technology and server side scripting language to develop a dynamic web based knowledge sharing web application. Ajax provides the framework for building web applications that doesn't require automatic post back there by preventing the need to load an entire page anytime a user makes an http request to the web server. To successfully implement this framework of the web application the ASP.Net programming language will be used in designing and developing the Ajax enabled sharing system [1].

Aims and Objectives

The goal requires designing and prototype implementation of

a technological solution utilizing AJAX technologies for building multi-tier Web applications. It is to specify, design and describe the implementation of a Web site for use with a search engine and a backend database. The prototyping is for development of a simple Web application according to the above specification. It can be developed on a different level of sophistication, depending on the personal preference of the students and the success of incorporating of various technologies. It utilizes synchronous server-side processing of search information using ASP.Net and asynchronous client-side hinting of the search terms (using JavaScript and AJAX) and the demonstration is to show the developed solution in operation [2].

Approach

- To design the architecture of a multi-tier system for a Web application which incorporates a Web presentation tier, AJAX communication tier, data processing tier with search engine, domain tier with terminological data and a storage tier with a backend database.

- To design a database and a database broker for storing information about search keywords and their description

- To build a simple Web site consisting of a Web page front-end, a server-side search engine and a database backend allowing to search for keywords in the database together with a Web page for showing the results of the search

- To extend the Web application with an AJAX engine which allows hinting for the search options based on the first letter of the keyword [3].

Report structure

The report has four major sections, which are further broken down into sub-sections. Section one emphasizes more on introductory part of Ajax which involve the designing and prototype of a technological solution using Ajax (who, when, how, what are the expectances), aims, and objectives of the intend system. Furthermore this converses about the report structure, how the new system will be analyzed, designed and implemented, it went further to specify what each section holds. Section two discuss about the software architecture design. In planning for the system development we review different methodologies which may be suitable for developing the intended system, the plan of work that will facilitate an easy implementation and evaluation of the assignment. OO (Object Oriented) methodology was preferred for the new system for a specific reason which was revealed further in the next sections. Section three reflects on the database design on how the new system is designed and developed. Also we present various system analytical diagrams from contextual diagram for capturing the process flow to the system navigational diagram. Finally section four presents the system development, evaluation and testing process using Ajax technologies [4].

Software architecture design

Prior to system implementation this section presents a detail plan which defines the approach to the software architecture design and testing for the system. This is used as a guide to the system implementation [5].

*Corresponding author: Jennifer Mosheshe, Cardiff Metropolitan University, Greater London, United Kingdom, Tel: +44 2073570077; Fax: +442074031163; E-mail: jennymosh@yahoo.com

Layers: I am using Ajax tool kit, the application is separated into 4 basic layers. The first layer is the data access layer it handles connections to Microsoft server 2008. The second layer of the system deals with xml web services, it handles xml web services, html response. The third layer is the server side scripting layer; this layer handles anything associated with business logic within the application. The last level is the presentation layer; this is the layer where users interact with the application [6].

Tiers: I am using the multi-tier application architecture which provides a model for developers to create a flexible and reusable application by breaking up an application into tiers, developers only have to modify or add a specific layer, rather than have to rewrite the entire application over, if they decide to change technologies or scale up. In the term "N-tier," "N" implies any number -- like 2-tier, or 4-tier; basically, any number of distinct tiers used in your architecture. Application architectures are part of Layer 7 of the OSI model [7].

Packages: A package is a grouping and organizing element in which other elements reside, which must be uniquely named. In the UML, packages are used in a manner similar to the way directories and folders in an operating system group and organize files [6].

Subsystems: A system is an organized collection of elements that may be recursively decomposed into smaller subsystems and eventually into non decomposable primitive elements [7]. Subsystems have various types namely:

- A user interface subsystem responsible for providing a user interface through which users may interact with the systems.

- A business processing subsystem responsible for implementing business functionality.

- A data subsystem responsible for implementing data storage functionality.

Partitions: Partitioning is the separation of software processes to prevent unintended interaction. I am using a physical partitioning as my software will be able to run on different hardware [8].

Object oriented modelling approach

In other to develop a system that meets the system specifications and acceptance criteria. We have decided to implement Object Oriented Web Modelling (OOWM) approach in our Content Management System development. This web development approach was selected, based on its support for relationships and abstractions between web objects which aids fast web application development. OOWM allow us to view the entire system as a collection of objects with defined methods and function calls. Additionally, Object-oriented model involves designing object, classes and the relationships between these classes [3]. To fully implement this approach I have decided to give more time to system design, this will allow us to identify each object within the system and there relational modularity.

Using OOWM, our approach to the Content Management System design and development are stated below:

- Development of a project schedule: We have decided to develop a project plan to schedule each stage of the system design and development using Microsoft Project.

- Identification of project quality criteria and risk analysis: To prevent risk such as file corruption, virus or accidental deletion of program files we have decide to identify various

risk that serve as a threat to the project develop a contingency plan to prevent them.

- Review of existing Ajax enabled websites: This will give us an overview of how information is presented to user's general layout of content management systems. Some of the websites to be reviewed is Google. Since they have been in the system for long we hope to gain detail information from this site.

- Identification of system Entities: Entities are the objects which the system needs to store and analyse their information. Egg Users, page sections, blogs, etc. Our target is to identify all the system entities in other for us to identify their types of relationships.

- Development of an Entity Relational Modeling: After identifying all existing entities within the system. We will develop an entity relational diagram to model their individual relationship. E.g. Many: Many or 1: Many

- Development of Class diagrams to represent each entity: At this stage we will develop a class diagram which will allow us to encapsulate and inherit from other objects within the system. The class diagram will show class names, class attribute methods and there function calls. For instance a Blog Entity Class will have an Update, Create, and Delete method.

- System Data analysis and development of contextual flow diagrams: To capture information flow within the system object, we have decided to develop a contextual flow diagram to analyze our system data.

- System interface Design and Development of System Navigational Diagram: Having Identified our system entities and the data flow within the system: I will move forward to design our site map and major navigations within the system. This will give us a general overview of the work-flow based on the system requirement specifications.

- System Development Using ASP.Net and Ajax: At this stage I will start developing each section, category and web pages of the content management system.

- Regular review of system development process and any issue: I have decided to regularly check system development progress and to solve any issue as they arise. This will help to ensure that the project is delivered on time.

Several applications and operating systems have been successfully developed using object oriented approach. Since my development tool is ASP.Net, which have several components inform of web page template, menu etc. I believe using OOM approach will give me the ability to integrate this component into each other without much difficulty [8].

System development specification

- View others' entries including blogs, audio, images, videos, etc.

- Add a RSS feed for Question and Answers

- Create a front-end interface for the system of your choice, which includes a list (not dynamic services) of browse-able popular destinations

- Create communication tier for the Question and Answer Forum using Ajax technologies

- Create advanced 'search for destinations' facility

- Create a user registration form & user login facility

- Allow users to view, add and update their text entries in Question and Answer Forum

- Allow users to upload their audios, videos, images, etc. (describing their experiences about the destination) [9].

Database design

The aim at this stage of the system design is to establish an analytical framework for system development. I have presented a wider conceptual view of the system architecture which will serve as a blueprint for the system. This precisely defines the push and pull of information in and out of the system domain [10].

Below is the system database definition language for the database design:

create Database TechQuestionsDB

---USE [TechQuestionsDB]

create table TechQuestionUsers

(

 UserID int identity(1,1) not null primary key,

 UserEmail varchar(100) not null,

 UserName Varchar(50) not null,

 UsersMobile varchar(20),

 UserPassword Varchar(20) not null

)

Create table Topics

(

TopicID int identity(1,1) primary key,

MainTopics varchar(200)

)

create table Questions

(

questionID int identity(1,1) not null primary key,

TopicID int,

UserID int, Question varchar(max)

)

create table Answers

(

AnswerID int identity(1,1) primary key,

questionID int ,

Answer varchar(max),

UserID int

)

create table UserSearchKeywords

(

KeywordID int identity(1,1) primary key,

SearchKeyword varchar(100)) [10].

System contexual design

In the quest to design the system architecture, we develop a contextual diagram of the system workflow to model the flow of information. The figure presents a contextual model of the entire system. The contextual diagram shows the data flow and process used to transform data into information as it moves through the system. Additionally, the contextual diagram shows the major entities and modules [11].

From the contextual diagram shown above, the systems consist of four major entities. The first entity is the new visitors who are recognized as anonymous users, our analyses summaries that they perform transactions such as searching for information on popular destination and entering search criteria. The second entity which is the Registered Users is given more access right to the system. First they are authenticated through the login module which verifies their password and username. After successful authentication they are given access right to the system to share their experience. The fourth entity is the registration module which accepts new user details as input to create a new user record in the system. The last entity is the Admin users who perform CRUD operation in the system. Finally the context diagram reveals the type of interactions and the stage boundaries that occur within each entity in the system and how their operations are modelled. The system entities are further modelled to create individual class diagrams [12].

High level system class diagram

Additionally to further develop the system, we present a class diagram which defines the methods and operations linking each entity within the system. This creates a clear picture of the type of relationship that exists between each entity which acts as an object within the system. Figures showing system class diagram [13] (Figures 1-5).

The above figure models each entity operation with a class diagram. As shown in the class diagram, rectangular box are used to represent each entity class within the system, the rectangles are divided into three sections. The first section shows the class name the second presents the properties of the entity which are the input parameters of each class. The last section shows the methods and functions performed by each entity. Also based on our design the Registered Users Class performs four basic operations, the Create Blogs, Comment on Blog, sharing of experience and updating created blogs. Furthermore based on critical analysis of access control to anonymous users which are visitors to the website the system will record the date of visit as an input parameter, the basic operations allowed for anonymous users are searching for destination details and commenting on blogs. Each class exchange messages inform of data, the Questions class pass articles, pictures and video into the blog class to create a new blog. While the Answers class accept search criteria's to display search result on various destinations [14].
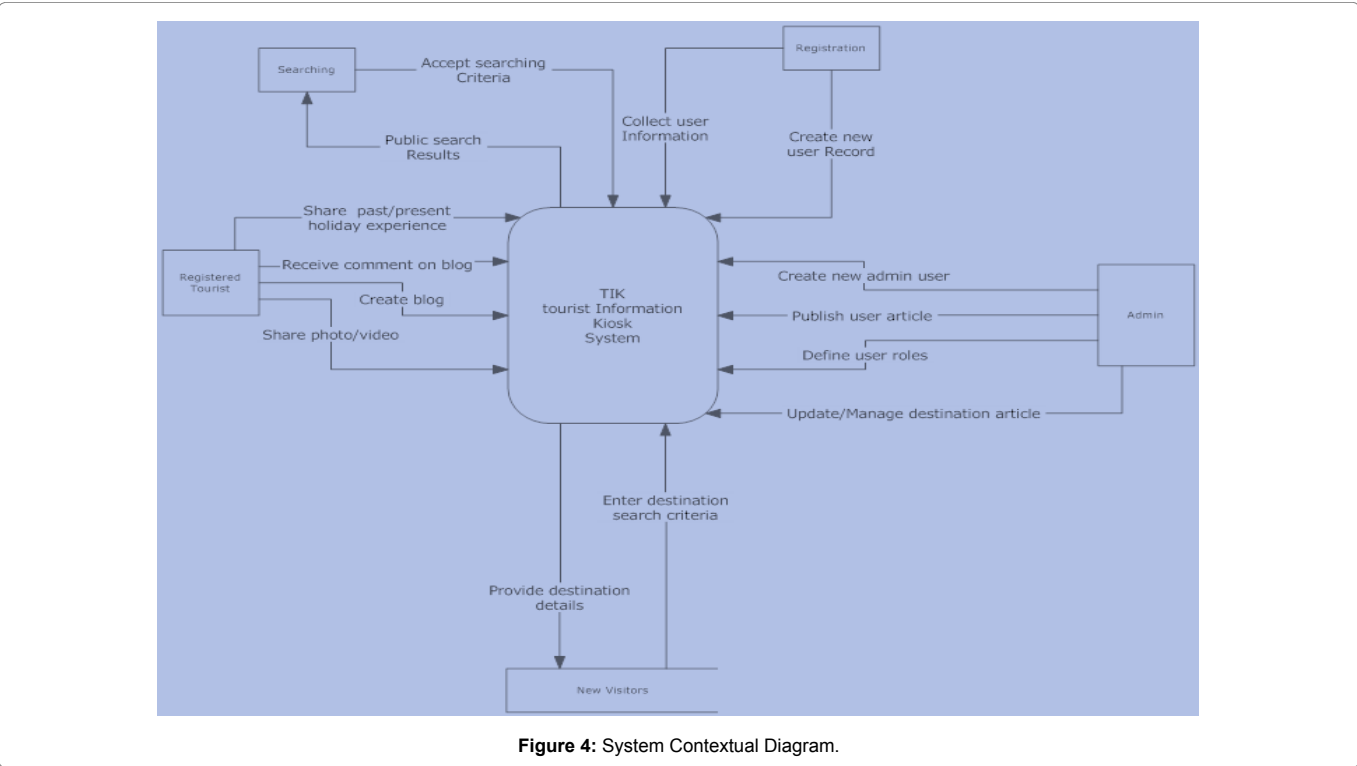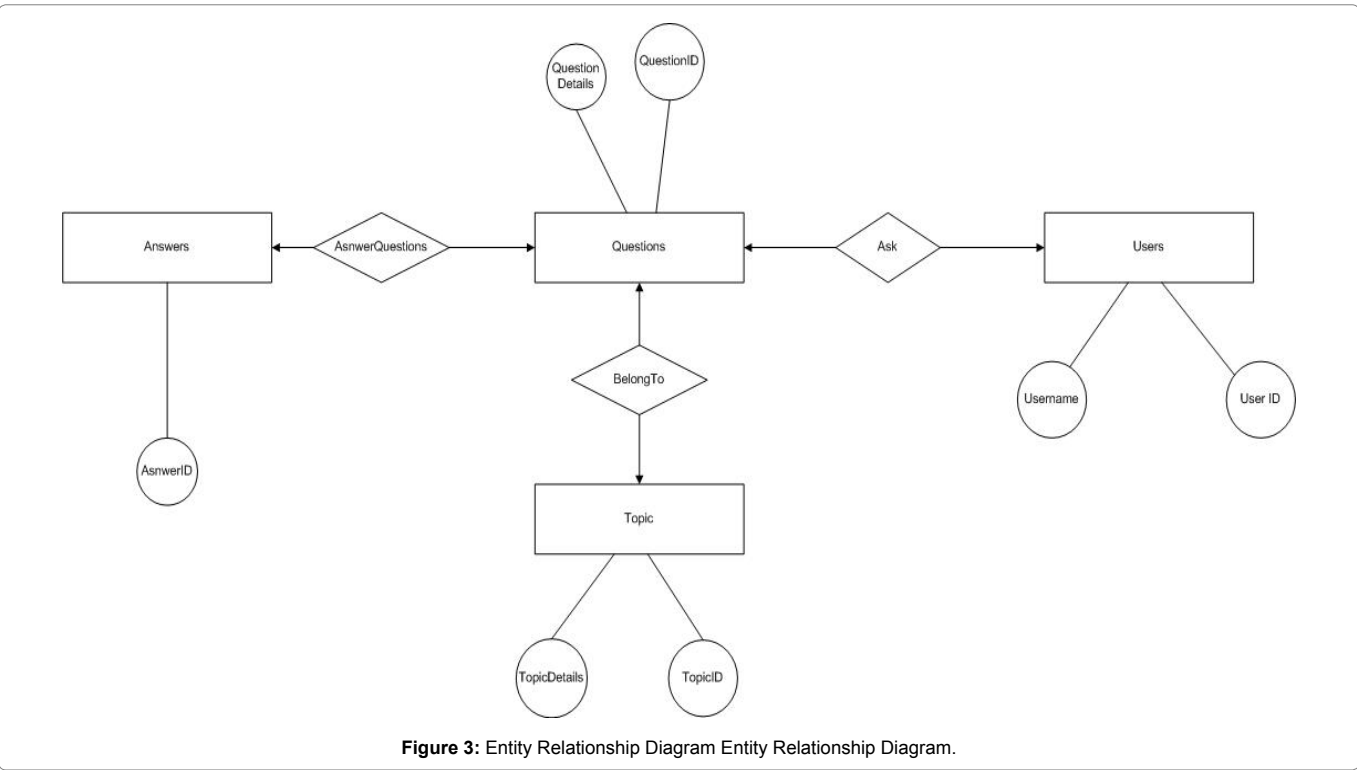
High level system use case

In our pursuit to model real world scenarios which reflect the

Figure 1: System Architectural Diagram

Figure 2: Database Diagram.
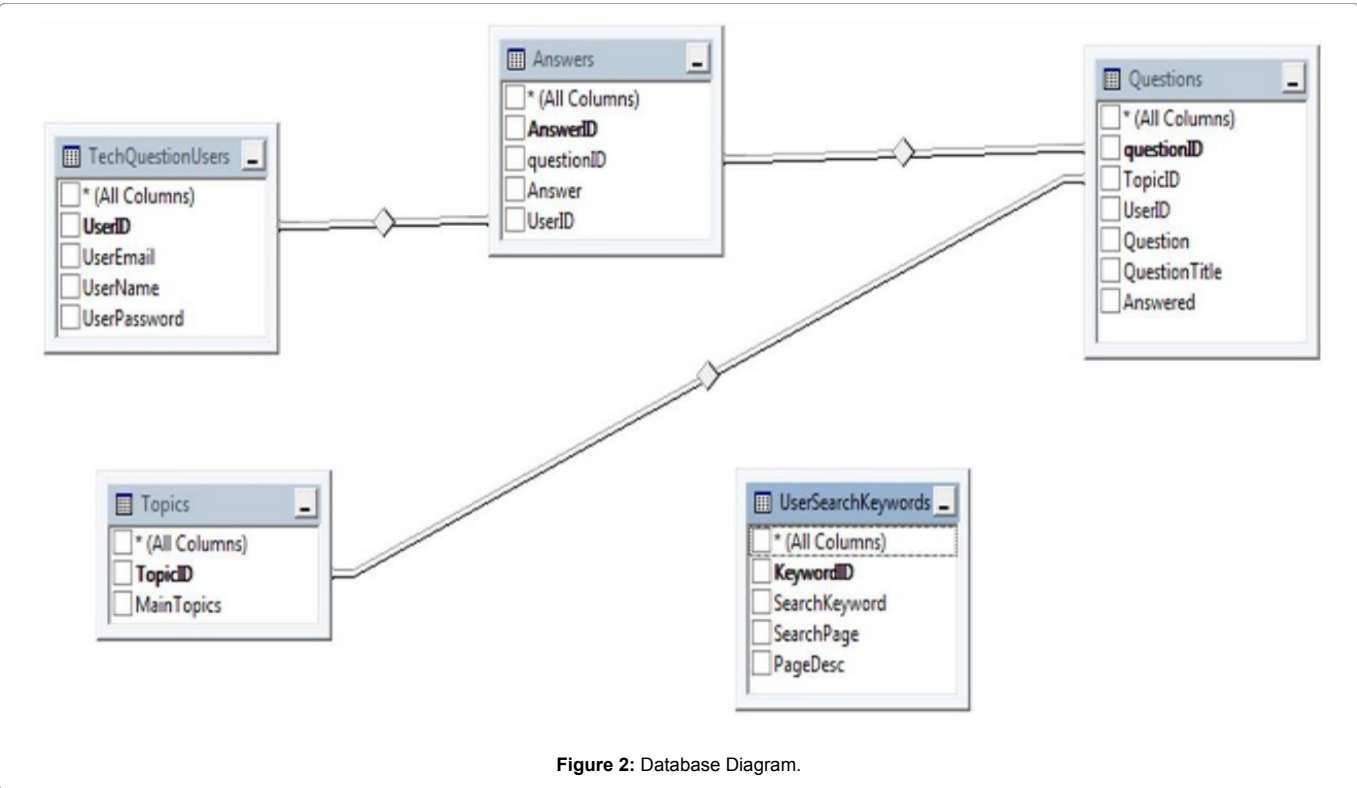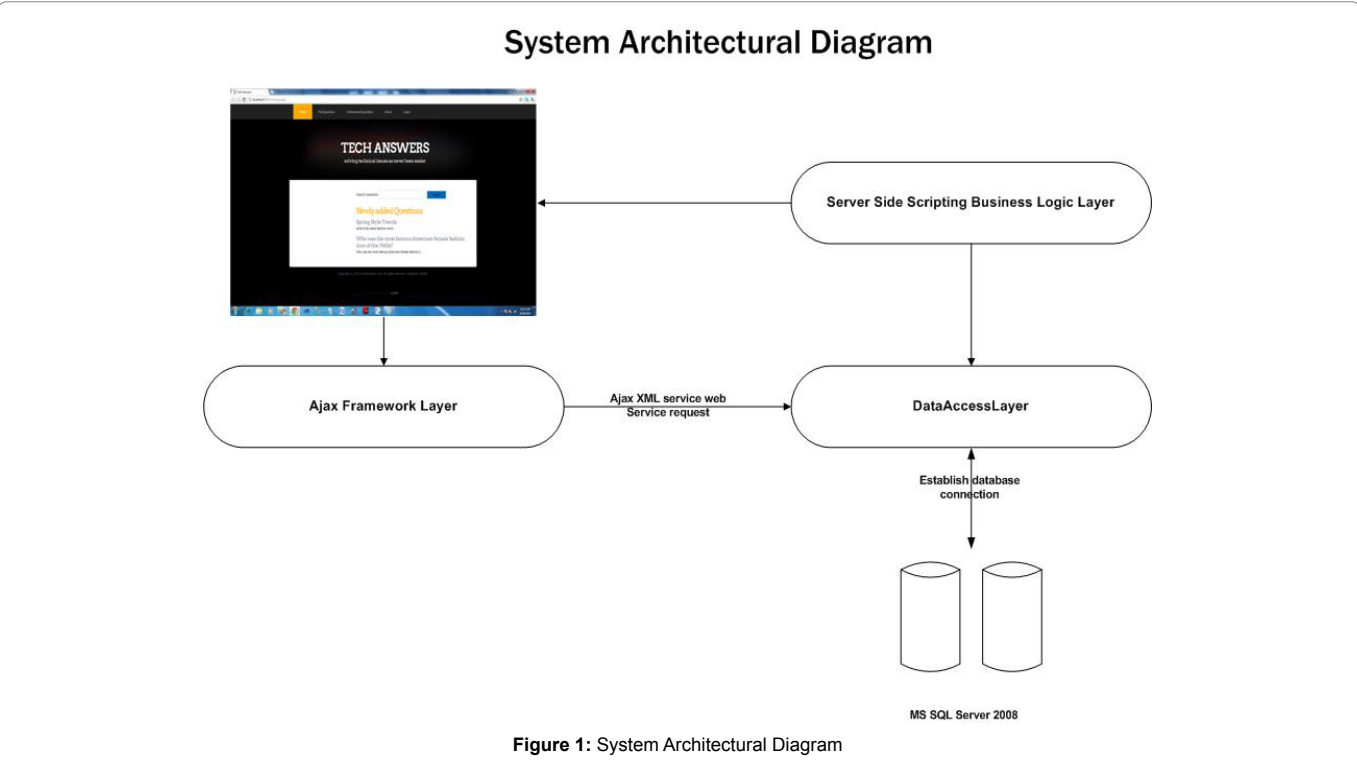
system requirement specifications, within the system actors (users). A use case diagram which clearly defines case and issues that can occur within the system is developed. Figure illustrates a typical use case scenario of the system (Figure 6).

Firstly case 1 registered user's login into the system by submitting their password and user names, new visitors are asked to create an account by filling a user registration form which allow them to create a user profile.

Figure 3: Entity Relationship Diagram Entity Relationship Diagram.

Figure 4: System Contextual Diagram.

Case 2 the system administrator creates an advance user search interface. Users search for information on a particular system and retrieve detail information.

Case 3 sharing of user information, Admin users collect and publish detail information on a popular destination, users review information's and post a blog on the web. Admin users read and publish appropriate blogs, other members comment on the web. Admin users read and publish appropriate blogs, other members comment on the publish blog in relation to a particular destination. Finally Admin Users continue to update information published on a particular destination [15].
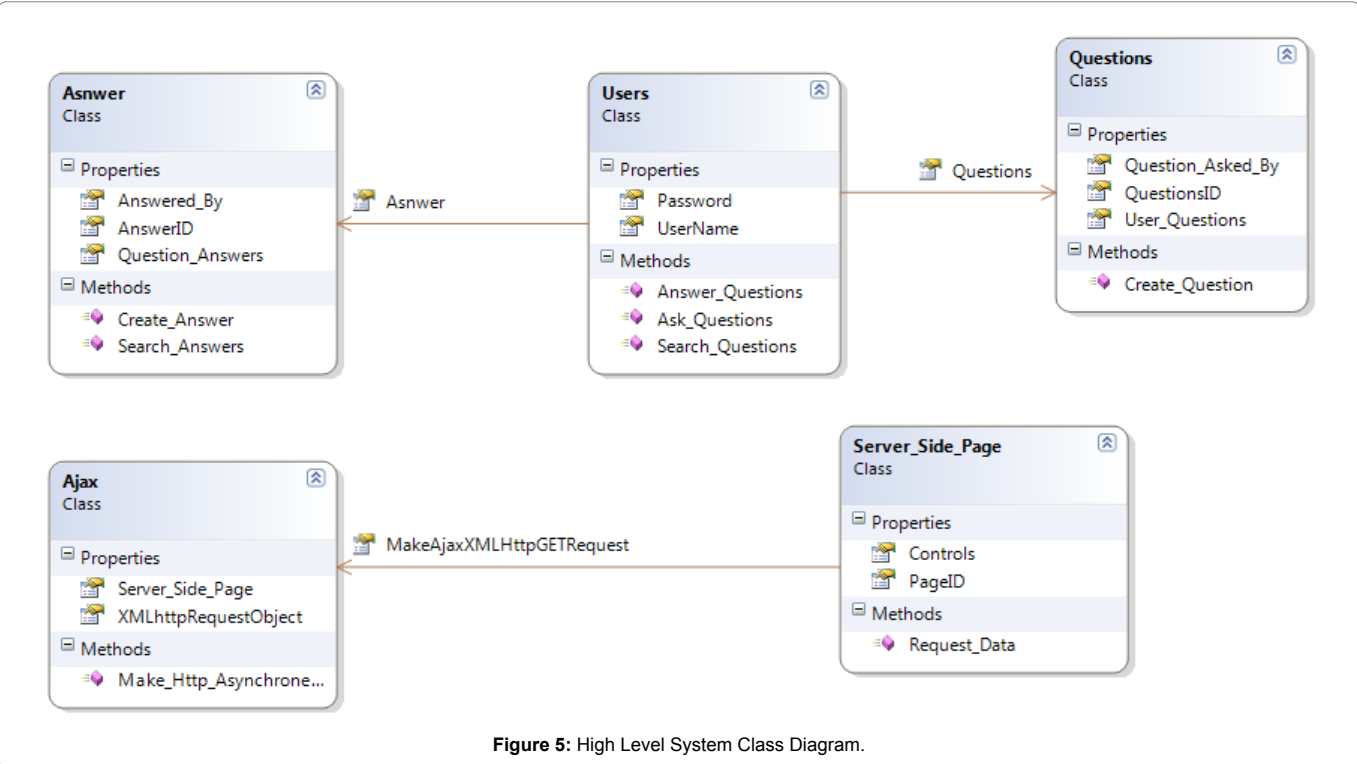
Figure 5: High Level System Class Diagram.
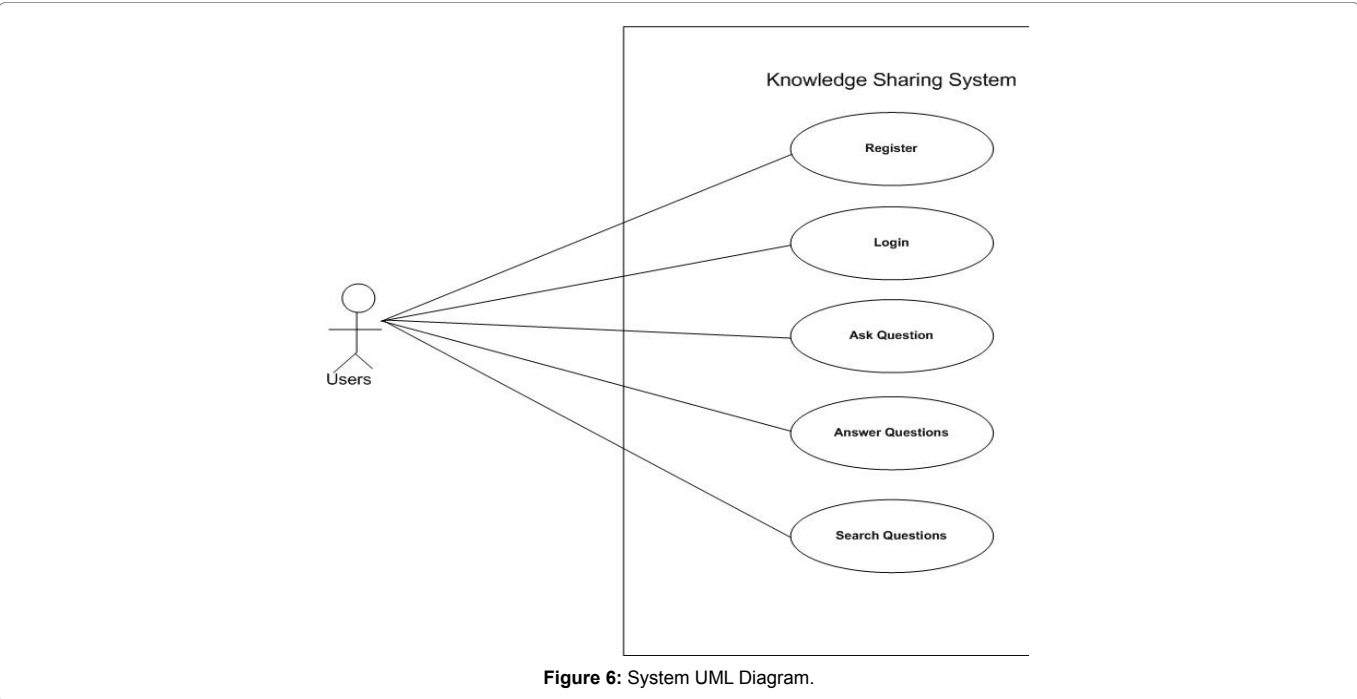
Figure 6: System UML Diagram.

System interface design

With detail understanding of the overall system architecture and workflow we proceed to develop web interface of the system. Our interface design is driven by the system requirements and target end-users, with this in mind a navigational diagram which shows how end-users will access the developed system is presented below. From the navigational diagram presented in figure the system will be accessed by three users each with their own navigational map (Figure 7). The anonymous users also identified has new visitors are giving limited navigational access. These users only have access to the general home page and the destination search page as shown in the diagram below. The registered users will be allowed to login into the user home after valid authentication. The admin users will be redirected to the administrator page after successful login were they will be able to perform update and publishing of blogs on the system [16].

The link menus displayed on each page are based on the user access right into the system. However the admin user page has complete access to all system pages. Finally the interface design will be implemented using Ajax technologies with layout that meet the system design.

At this stage we move to the system implementation using Ajax as the system development tool. The next section shows how the system design was implemented in building the system.

System implementation using Ajax

Based on the project specification, this section illustrates how ASP.Net was used to develop the web interface layer and data access into the Ajax enabled knowledge sharing system.

Content hierarchy model

To successfully build the System using Ajax, the contents of the system are partitioned into hierarchy. Base on the Ajax database infrastructure, we split the entire data into level based hierarchies from section to categories which is further broken down into articles [17] (Figures 8 and 9).

Each popular destination in the system is partitioned into categories. An article is attached to each category this allow the system to display different articles base on the destination selected in the menu. The aim of using this model is to model the hierarchy of relationships that exist between each entity that form the entire system. Based on our implementation we further divide each entity as a child to a parent and parent to parent relationships. This approach is further illustrated in the modelled hierarchical relationship diagram shown in the figure above. Figure 10 illustrate the level of dependences and child to parent relationships in the system by breaking the system down into structured hierarchy. Also each location consists of its

own comments and blogs which allow new visitor to express their comments and questions.

Using this model each object within the system environment is linked to each other allowing users to gain detail information on each destination and other top attraction and reviews. With detail development of each page and sections we move further to define the system user access control [18].
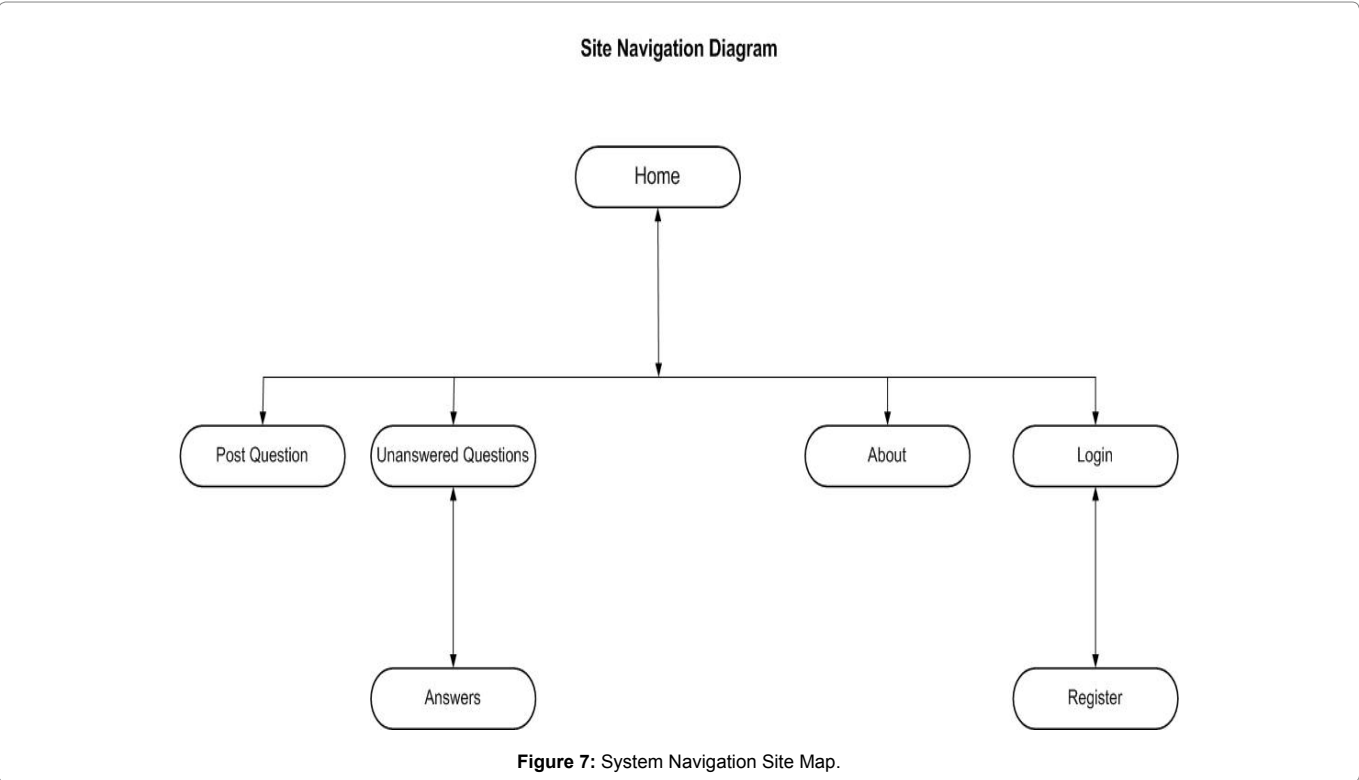
User group and roles

Subsequent to the system requirement specification we developed user groups and roles to control access right of each user based on their group and allocated roles. In total the system consist of three basic groups as shown below.

- Anonymous Users (New Visitors)
- Registered Users
- Administrator

Anonymous Users: They are recognized as new tourist visitors which are yet to register their details into the system. They are given the right to search using the advance searching module in website. Secondly they are allowed to comment on published articles and visited popular destinations.

Registered Users: Also recognized as authors, they are given the privilege to create article or blog and upload videos on a particular destination or hotels they have visited. This gives them the chance to share their experience with new visitors.

Administrator: The admin group is responsible for the system maintenance. They will review posted articles and publish appropriate once. This group is also known as the publisher since they are given the role to publish articles.

Figure 7: System Navigation Site Map.

```
  |
□ create Database TechQuestionsDB
□ create table TechQuestionUsers
  (
   UserID  int identity(1,1) not null primary key,
   UserEmail varchar(100) not null,
   UserName Varchar(50) not null,
   UsersMobile varchar(20),
   UserPassword Varchar(20) not null
  )

□ Create table Topics
  (
  TopicID int identity(1,1) primary key,
  MainTopics varchar(200)
  )

□ create table Questions
  (
  questionID int identity(1,1) not null primary key,
  TopicID int,
  UserID  int,
  Question varchar(max)
  )

□ create table Answers
  (
  AnswerID int identity(1,1) primary key,
  questionID int ,
  Answer varchar(max),
  UserID  int
  )

□ create table UserSearchKeywords
  (
   KeywordID int identity(1,1) primary key,
   SearchKeyword varchar(100)
  )
```

Figure 8: SQL DLL.

Figure 9: System Hierarchical Model.

System interface development

In developing the system interface a web template which fit into ASP.Net frameworks was added into the system. Using this template the following pages and sections were developed. Figures 11 and 12 shows a detail diagram of the system developed webpage.

The system home page shown above provides a general introduction to the system and allows existing users to login from the user login section (Figure 11). Also by clicking the Register button the system loads the user registration form presented in Figure which is used to create a user account within the system (Figure 12).

Figure 13 presents the system advance search and popular destination web page. Users can either select a popular destination from the destination section of the home page or enter a destination search query using the advance search function. It is used to Post Questions to other users within the system.

Figure 14 presents the Questions which are asked in the question and Answer Forum and are open to other users in the system to be answered. Using the search link older Questions and Answers can be recovered.
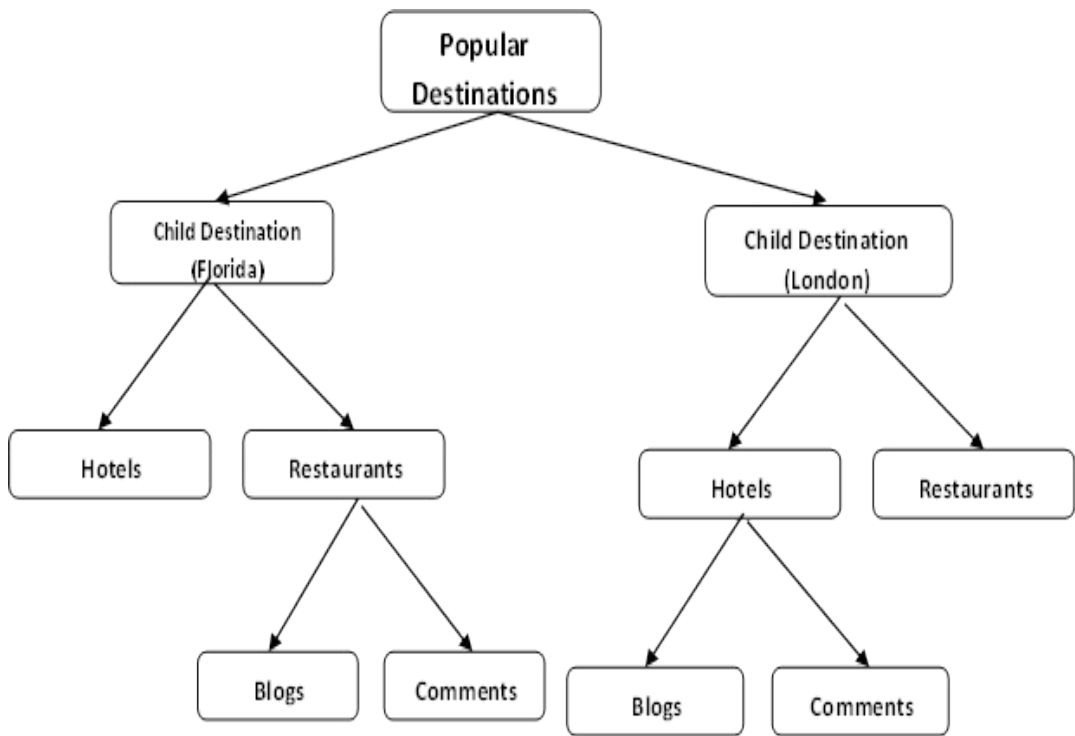
Figure 10: Partial Blueprint of System Implementation Using Hierarchical Model.
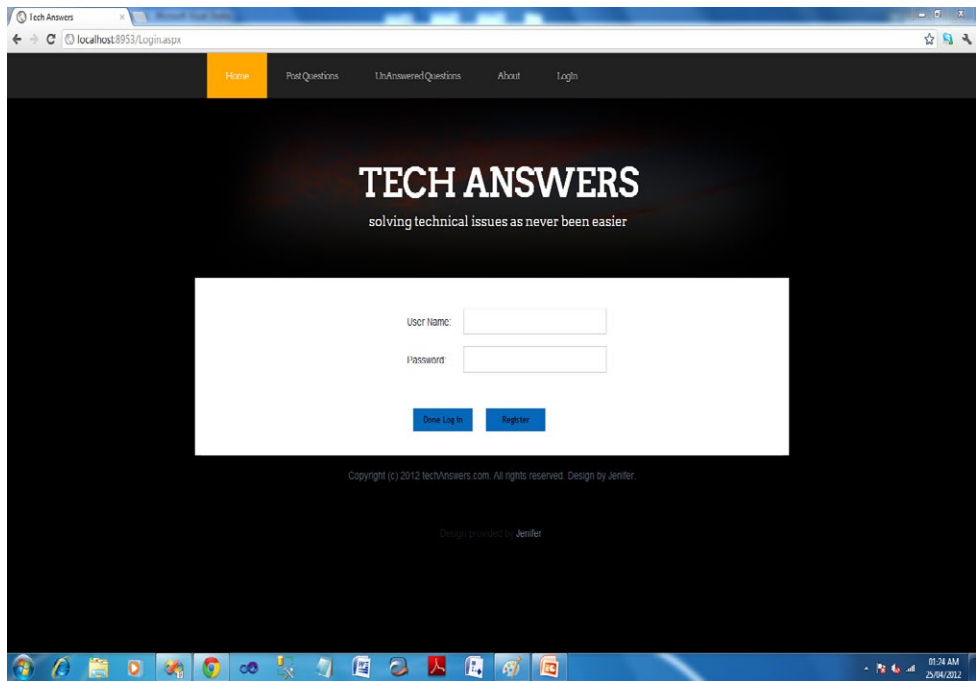

Figure 11: System Home Page and Login Page.

Conclusion

This paper described the design and development process of an AJAX enabled knowledge sharing system, indicating each steps encountered during the system development, as well as realizing the strengths and limitations of the system. System design and development experience and knowledge gained were also realized in this paper. Hence, each challenge encountered during the design and development process was solved utilizing suitable solutions through the aid of hierarchy modelling [3]. This project has created a profitable

Figure 12: User Registration Form.

Figure 13: System Advanced Search Page.

knowledge and eloquent experience to the world at large, especially for professionals who require a more robust approach during the software development life cycle and the designing of databases, in order to achieve their respective goals and objectives effectively as planned. Also, the AJAX enabled knowledge sharing system is considered to improve various organizational service quality and performance of organizational proceedings, by enabling user friendly and dynamic platforms that are secured [7]. Thus, creating improving organizational productivity, reliability and profits. Although the

developed system implements the major system requirements further research work is needed to allow various objects within the system across various geographical region to collaborate in other to share and deliver detailed information to end users.

References

1. Klamma R, Spaniol M, Cao Y, Jarke M (2006) Pattern Based Cross Media Social Network Analysis for Technology Enhanced Learning in Europe. Proceedings of the 1st European Conference on Technology Enhanced Learning (EC-TEL 2006), Hersonissou, Greece.
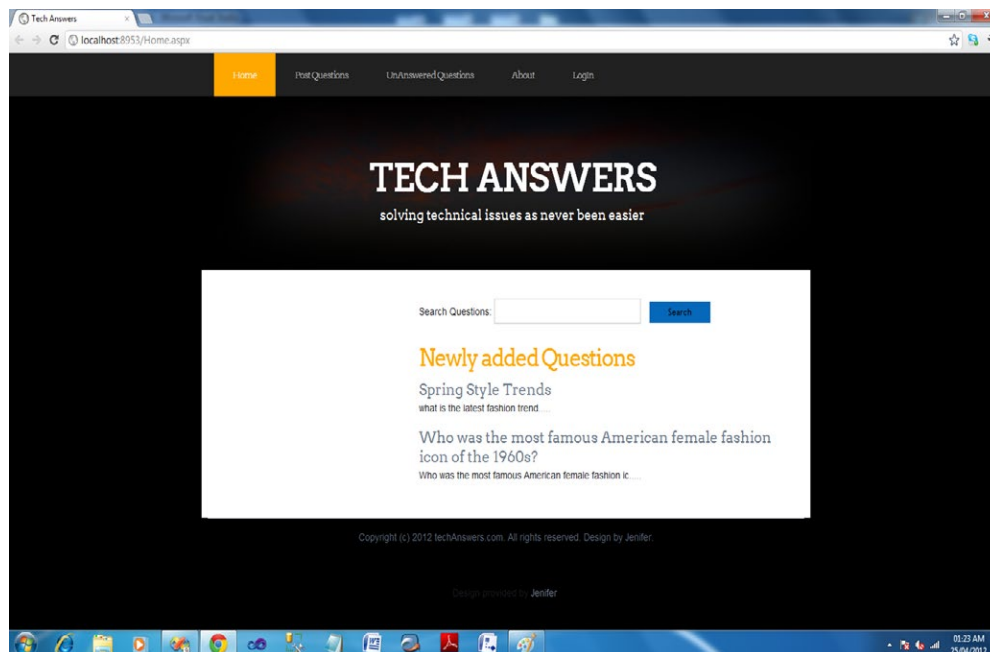
Figure 14: Newly added Questions to the Web Page.

2. Weisfield M (2003) The Object-Oriented Thought Process, an Introduction to Object-Oriented Concept for Programmers. (2nd edtn), Addison-Wesley, Boston, Massachusetts, United States.

3. Sommerville I (2000) Object-oriented-design.

4. Adar E, Adamic LA (2005) Tracking Information Epidemics in Blogspace. Proceedings of the International Conference on Web Intelligence (WI 2005), 19-22 Compiegne, France.

5. Coad P, Yourdon E (1991) Object Oriented Design. Prentice Hall, Englewood Cliffs, NJ.

6. Drucker PF (1999) Knowledge Work Productivity: The biggest Challenge. California Management Review 1: 79- 94.

7. Beckett D (2004) Modernising Semantic Web Markup" in XML Europe 2004, Amsterdam, The Netherlands.

8. Aumueller D, Auer S (2005) Towards a Semantic Wiki Experience - Desktop Integration and Interactivity in WikSAR. Semantic Desktop Workshop.

9. Frank U, Klein S (1992) Three integrated tools for designing and prototyping object-oriented enterprise models. GMD-Research Report, no. 689, Sankt Augustin, Germany.

10. O'Reilly T (2006) What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software.

11. Paulson LD (2005) Building rich web application with Ajax, Computer, 38 IEEE, 14-17.

12. Brown JS, Duguid P (2000) The Social Life of Information, Harvard Business Press, Cambridge, MA.

13. Bausch P, Haughey M, Hourihan M (2002) We Blog: Publishing Online with Weblogs, Wiley.

14. Rizova P (2006) Are you networked for Successful Innovation? MIT Sloan Management Review 47: 49-55.

15. Nonaka I, Takeuchi H (1995) The Knowledge-creating Company. Oxford University Press, Oxford.

16. Rumbaugh (1991) Object-oriented Modelling and Design. Prentice Hall, Englewood Cliffs/NJ 1991.

17. Gottesdiener E (1998) Methodologies: Process & Products Patterns.

18. Rosenbloom A (2004) Communications of the ACM - The Blogosphere. CACM Communications of the ACM, ACM New York, NY, USA.

Author Affiliation   Top

Cardiff Metropolitan University, Greater London, Cardiff Metropolitan University, United Kingdom