# ABSTRACT

In a Question Answering system, the user submits a question and waits for the answer as the response. If the system is capable of predicting the user's future interest as the next question, its performance will improve greatly. This project predicts users' future questions based on the current interaction records of the user with the system. Their current interactions with the system show what they are interested in. These interaction records are maintained in the form of Questions log from which the user sessions are extracted. Based on the user sessions, the system predicts the next question for which the user may become interested in near future. A sample Questions Log is selected for the purpose of performing experiments. The model of Association rule mining is applied to predict the future question of the user.

The online community consists of several discussion groups, help communities and forums. The Question Answering system is a useful tool in such communities. The proposed system is a Question Answering system with an extended functionality of predicting the users' next question. By prediction, we eliminate the work required for posting each question and improve the efficiency and user friendliness of the system. The prediction is done using the data mining technique of Association rule mining. In particular, we make use of the Apriori algorithm to generate association rules among the questions posted by the user. We analyze the question log of the users to obtain their history with the system and derive their area of interest. The predictions are obtained using the question log as an input to the algorithm

# ACKNOWLEDGEMENT

If words are considered as the token of acknowledgement, then the words play the heralding role of expressing our gratitude.

We owe our gratitude to The Principal, **Dr. B. Sadashive Gowda** for his whole hearted support.

We wish to express our deepest gratitude to **Dr. Ravi Kumar V,** Head of Department, Information Science and Engineering, VVCE, for his profound alacrity in our progress and his valuable suggestions.

We hereby like to thank **Prof. Vidyashree K P**, Assistant Professor, Information Science and Engineering for her constant guidance and encouragement

We also thank the project coordinator **Prof. Vinutha D C**, Associate Professor, Information Science and Engineering for her support and encouragement.

At the end, we are anxious to offer our sincere thanks to our family members and friends for their valuable suggestions, encouragement and unwavering support.

**AMULYA R**

**ASHRITH C**

**H V TEJAS TAUNAD**

**KAVYA SHREE**

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# INTRODUCTION

Since the inception of the search engines, much of the research has been done on extracting information from Search logs. This procedure is referred as Log Analysis. The records of interactions between a search engine and users interested for information is maintained in the Log. Since, during interaction a user writes a set of queries that specify information about his/her topics of interest. These interactions in a specified interval of time are maintained in the Log. The major objective of the technique of log analysis is the information extraction. The process of Data Mining can be applied to the user interactions maintained in the log for knowledge extraction about the user interests and the information in which he may become interested in near future. The major objective is to keep track of the tasks performed by the users in the current search sessions, and to predict the tasks they will do in the sessions that may occur in future.

Association rule mining is concerned with checking the databases consisting of transactions, relations, and other information repositories for a set of transactions and identification of patterns occurring frequently, associations, correlations, or causal structures among the item sets on the basis of the existing item sets in those transactions. These are used to express how items (or objects) are related to each other, and how they tend to group together. The model of Association rules mining has many interesting research directions. Question Answering System offers a more intuitive approach to information processing. For a set of documents and a question fired by user in the natural language, a question answering system aims to find the at least a portion of the text or a precise answer to the question.

When a user poses a question to the QA, the system responds with the answer to the question. A search session of a user involves a set of questions that have been asked by the user. Along with responding the user with answer to his question, the system also predicts the next question in which the user is interested in. By doing so, the efficiency of the system increases greatly. The response time, the user friendliness increases and the extra time required by the user to enter each question can be avoided. This is the extra feature being added to the proposed system which differs from the existing system which is a normal QA system without prediction.

## 1.1 Input & Output

- **Input** - Question Log based on the user
- **Output -  -** system predicts users future questions

## 1.2 Aim

In a Question Answering system, If the system is capable of predicting the user's future interest as the next question, its performance will improve greatly.

## 1.3 Objective

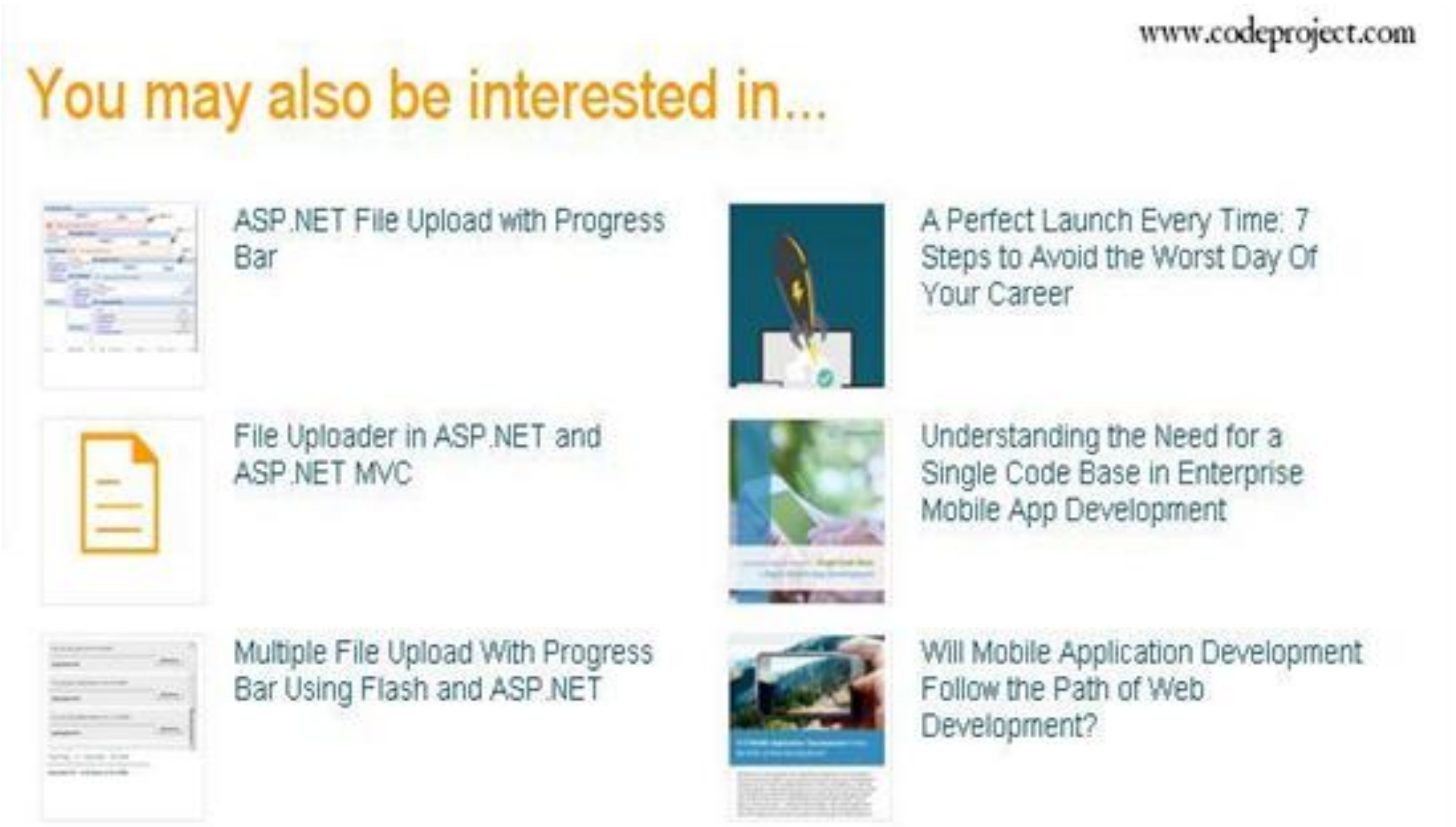
- Proposed system is a online based application.
- Proposed system is a new QA system which is meant for education sector.
- Proposed system generates the answers for the questions posted on the users.
- Proposed system major objective is to predict the users' next questions and to provide the services according to the users' area of interests.
- Proposed system predicts user's future questions based on the current interaction records of the user with the system.
- Proposed system makes use of data mining technique called as “Association Rules” to discover patterns between the users questions.

## 1.4 Existing System

In a Question Answering system, the user submits a question and waits for the answer as the response. Example for question answering system www.aspsnippets.com, www.codeproject.com,www.c-sharpcorner.com,www.msdn.microsoft.com, www.stackoverflow.com, www.codeguru.com etc.

In the above mentioned QA systems, the user inputs the query/question, the system retrieves the most relevant answers from the database/server. The QA system displays n number of answers for the inputted question, along with the answers system displays the related blogs and this suggestion is not based on the user interaction with the system, but it is based on the question type inputted by the user. The following are some of the snaps of related blogs collected from different existing QA systems.

*For the Question - "Uploading file using Asp.Net C#", the related blogs*



www.codeproject.com

You may also be interested in...

ASP.NET File Upload with Progress Bar

A Perfect Launch Every Time: 7 Steps to Avoid the Worst Day Of Your Career

File Uploader in ASP.NET and ASP.NET MVC

Understanding the Need for a Single Code Base in Enterprise Mobile App Development

Multiple File Upload With Progress Bar Using Flash and ASP.NET

Will Mobile Application Development Follow the Path of Web Development?

**Fig 1.1** Example showing existing systems' output

## 1.5 Limitations of Existing System

- Suggestions are related to inputted query, not based on users' interest.
- Lack of user satisfaction
- Lack of knowledge discovery

- Time consumption for searching questions
- Less reliable
- Less Efficient

## 1.6 Proposed System

Proposed system is a new QA system where user submits the question and waits for the answer as the response. Proposed system is a general QA system where members submit the questions and other members sends the response. Proposed is a QA system which predicts the users' future questions based on the current interaction records of the user with the system, current interaction shows the users' area of interest. For prediction system makes use of data mining technique called as “Association Rules”.

# Chapter 2

# SYSTEM REQUIREMENTS AND ANALYSIS

## 2.1 Hardware Requirements:

Hardware required to develop the software is as listed below

| | |
|---|---|
| • **Processor** | Pentium IV onwards |
| • **Processor Speed** | 1.2GHz |
| • **RAM** | 2GB + |
| • **Hard disk space** | 40GB + |

**Table 2.1** Hardware requirements

## 2.2 Software Requirements:

Software required in development is as listed below

| | |
|---|---|
| • **Operating System** | Windows XP version & Higher |
| • **Design Tool** | Visual Studio 2010 |
| • **Front End** | ASP.NET 4.0 |
| • **Language** | C# |
| • **Architectural design** | N – Tier |

**Table 2.2** Software requirements

## 2.3 Specific Requirements

### 2.3.1 External Interface Requirements

**User Requirements**

- Application contains an interface where the member can get registered to the application by specifying their details.
- Application contains interface where member and admin can get login by specifying their credentials.
- Application has interface where the admin can manage the users data.
- Application has an interface where the members can get their predicted questions.
- Interface where the members can view the questions generated by the system.
- Interface where the admin validate the answers.
- Interface where admin can view the profile by site members.

### 2.3.2 Hardware Interfaces

***Server Side***

The web application will be hosted on the Internet Information server and connecting to the MS SQL Database server.

***Client Side***

The system is a web based application; users require a web browser such as Internet Explorer, Mozilla Firefox, etc. The computer must have an Internet connection in order to be able to access the system.

### 2.3.3 Software Interfaces

***Server Side***

The required software is .NET web application. An IIS Web server will accept all requests from the client and forward the specific requests of application to server. A development database will be hosted locally (using MS SQL).

***Client Side***

An OS is capable of running a modern web browser which supports ASP.NET.

**Communications Interfaces**

The HTTP protocol will be used to facilitate communications between the client and server.

## 2.4 Functional Requirements

- Administrator of the QA system uploads all the question types into the server,so this will help the users of the QA system to filter the QA based on the type which reduces the time.
- Administrator of the QA system uploads possible keywords based on the question type,so part is used to classify the query posted by the users of the application.
- Administrator has the option to manage the types means adding the new category,editing,updating and deleting the existing question categories.
- Administrator can view the registered user profiles and questions and answers posted by the users of the application.
- Members of the QA system can post the queries,for which the system generates the answers.
- Members of the QA system can browse the QA system.
- Members can post the questions based the question type,for the posted question system generates all possible answers stored in the server.
- Users of the QA system should get registered to access the application to access the services.

## 2.5 Other Non-functional requirements

Non-functional requirements are constraints that must be adhered to during development. They limit what resources can be used and set bounds on aspects of the software's quality.

One of the most important things about non-functional requirements is to make them verifiable. The verification is normally done by measuring various aspects of the system and seeing if the measurements confirm to the requirements.

Non-functional requirements are divided into several groups:

The first group of categories reflects the five qualities attributes

1. Usability

2. Efficiency

3. Reliability

4. Maintainability

5. Reusability

These requirements constrain the design to meet specified levels of quality.

The second group of non-functional requirements categories constrains the environment and technology of the system.

## 2.6 Data Mining

Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to search large databases in order to find novel and useful patterns that might otherwise remain unknown.

In other words, Data mining is a process of analyzing the data from different perspectives and summarizing it into useful information.

### Data Mining Techniques

Several core techniques that are used in data mining describe the type of mining and data recovery operation. The techniques are,

1 Association Rule

2 Classification

3 Clustering

4 Sequential Patterns and Predictions

### Association Rule

Association (or relation) is probably the better known and most familiar and straightforward data mining technique. Here, we make a simple correlation between two or more items, often of the same type to identify patterns.

For example, Market-basket analysis, where we track people's buying habits, we might identify that a customer always buys cream when they buy strawberries, and therefore suggest that the next time that they buy strawberries they might also want to buy cream.

### Classification

Classification is used to build up an idea of the type of customer, item, or object by describing multiple attributes to identify a particular class.

For example, you can easily classify cars into different types (sedan, 4x4, convertible) by identifying different attributes (number of seats, car shape, driven wheels). Given a new car, we might apply it into a particular class by comparing the attributes with our known definition. We can apply the same principles to customers.

For example , by classifying them by age and social group.

### Clustering

Clustering is done using one or more attributes by identifying a cluster of correlating results. Clustering is useful to identify different information because it correlates with other examples so we can see where the similarities and ranges agree.

Clustering can work both ways. We can assume that there is a cluster at a certain point and then use our identification criteria to see if we are correct.

### Sequential Patterns and Prediction

**Sequential patterns** are a useful method for identifying trends, or regular occurrences of similar events. For example, with customer data we can identify that customers buy a particular collection of products together at different times of the year. In a shopping basket application, we can use this information to automatically suggest that certain items be added to a basket based on their frequency and past purchasing history.

**Prediction** is a wide topic and runs from predicting the failure of components or machinery, to identifying fraud and even the prediction of company profits. Used in combination with the other data mining techniques, prediction involves analyzing trends, classification, pattern matching, and relation. By analyzing past events or instances, you can make a prediction about an event.

Using the credit card authorization, for example, you might combine decision tree analysis of individual past transactions with classification and historical pattern matches to identify whether a transaction is fraudulent. Making a match between the purchase of flights to the US and transactions in the US, it is likely that the transaction is valid.

### Data Mining Phases

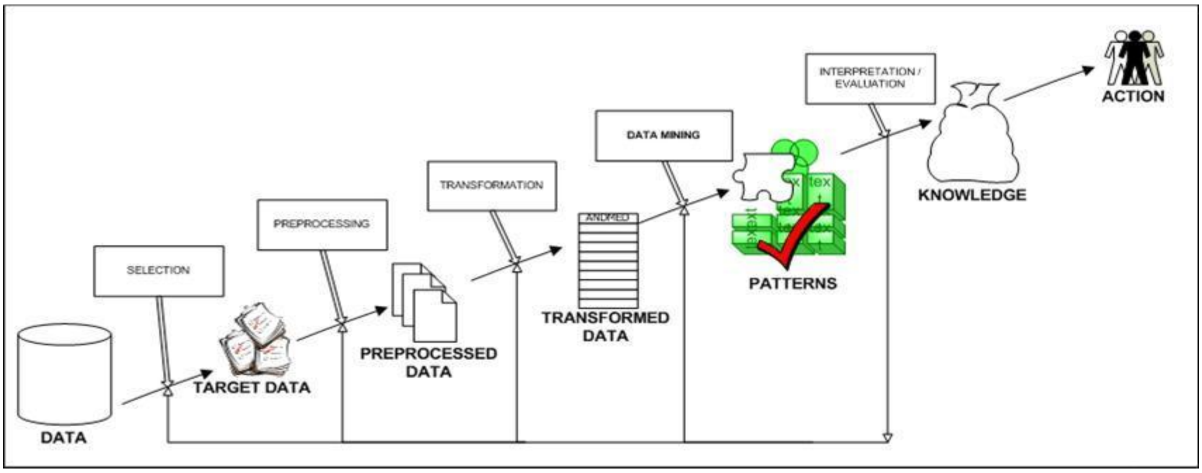


**Fig 2.1** Data mining process consists of a sequence of the following steps

- **Data cleaning** – to remove noise and irrelevant data.

  Data quality is a main issue in quality information management. Data quality problems occur anywhere in information systems. These problems are solved by data cleaning. Data cleaning is a process used to determine inaccurate, incomplete or unreasonable data and then improve the quality through correcting of detected errors and omissions. Generally data cleaning reduces errors and improves the data quality. Correcting errors in data and eliminating bad records can be a time consuming and tedious process but it cannot be ignored. Data mining is a key technique for data cleaning. Data mining is a technique for discovery interesting information in data. Data quality mining is a recent approach applying data mining techniques to identify and recover data quality problems in large databases. Data mining automatically extract hidden and intrinsic information from the collections of data. Data mining has various techniques that are suitable for data cleaning. In this paper we discuss three major data mining methods, namely functional dependency mining, association rule mining and Bagging SVMs for data cleaning. We discuss strengths and weakness of these data mining methods for data cleaning.

- **Data integration** – where multiple data sources are combined

  Data integration involves combining data from several disparate sources, which are stored using various technologies and provide a unified view of the data. Data integration becomes increasingly important in cases of merging systems of two companies or consolidating applications within one company to provide a unified view of the company's data assets. The later initiative is often called a data warehouse.

  Probably the most well known implementation of data integration is building an enterprise's data warehouse. The benefit of a data warehouse enables a business to perform analyses based on the data in the data warehouse. This would not be possible to do on the data available only in the source system. The reason is that the source systems may not contain corresponding data, even though the data are identically named, they may refer to different entities.

- **Data selection** – for retrieve from the database only the relevant data for the analyze

- **Data transformation** – where data are transformed or consolidated into forms appropriate for mining
- **Data mining** – the phase where the algorithms are applied in order to extract data patterns
- **Pattern evaluation** – to find the interesting patterns which representing new knowledge.
- **Knowledge presentation** – where the visualization techniques are used to present the mined knowledge to the user

In order to ensure that the extracted information generated by the data mining algorithms is useful, additional activities are required, like incorporating appropriate prior knowledge and proper interpretation of the data mining results.

## 2.7 User and Characteristics Requirements:

### Project Planning

**Actors - Application mainly consists of 3 actors;**

### Administrator

Administrator is a one who maintains the entire QA system. Administrator has the full accessibility of the application (QA system).

### Members

Member is a registered user. Member is a one who receives the services from the application. The key service given by this QA system is "prediction of next question" based on the previous questions posted by the users.

### Guest

Guest is a new comer to the application, Guest has only the limited accessibility.

## General Constraints

The system need to design based on the database using MS SQL.

- ***Security:*** The files in which the information regarding securities and back up should be secured against malicious deformations.
- ***Reliable:*** system which can hold large database and can perform various operation on it.
- ***Fault Tolerance:*** Data should not become corrupted in case of system crash or power failure.

## Assumptions and Dependencies

- Members should have knowledge of accessing web applications.
- Members should be educated

## 2.8 Modules

## Administrator has the following modules

### Login Module

In this module, administrator of the application gets login to the application by inputting the credentials such as login id and password which is set in the server.

### Manage Types [Question Types]

In this module, administrator of the application uploads the question types into the server. This type option is given so that members can post the questions based on the selected type. Managing of the types means adding the new category, editing, updating and deleting the existing question categories.

### Set Special words

Here administrator of the application sets the special words based on question type. This module is used to identify the question type posted by the users.

### Manage Members

In this module, administrator can view the registered user profiles.

### Browse QA System

Here administrator can view the questions and answers posted by the users of the application. Here admin can browse the questions and answers based on the question type.

### QA Verification Module

In this module administrator of the application verifies the answers posted by the members.

## Member has the following modules

### Login Module

Here user gets login to the application by providing the login id and password set by the administrator.

### QA Module

In this module student can post the question. For the posted question system generates all possible answers stored in the server and the specified question is dumped into the server as a new question.

### Answer Module

In this module, member can send the answers for the questions posted by other users

## Visitor has the following modules

### Registration module

Here, the guest can register himself to the QA system.

# Chapter 3

# TOOLS AND TECHNOLOGY USED

## Preferred Technology: .NET [Visual Studio 2010]

## 3.1 Reason for choosing .NET:

**Limitations of C:**

- C developers are forced to contend with manual memory management.
- Ugly pointer arithmetic.
- C is structured programming language.
- Programmers require complete knowledge of best programming technique.

**Limitations of C++:**

- C++ can be thought as an Object Oriented layer on top of C.
- It involves manual memory management.
- Ugly pointer arithmetic.
- Ugly syntactical constructs.

**Limitations of JAVA/J2EE:**

- Java programmers must use java front to back during development cycle.
- It is not appropriate for many graphical or numerical intensive applications.

.NET provides solution to all the above mentioned problems.

## 3.2 Overview of .NET:

.NET Framework:

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives: To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.

- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components:

- The common language runtime and
- The .NET Framework class library

The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

## 3.3 Introduction to ASP.NET

ASP.NET is a unified Web development platform that provides the services necessary for you to build enterprise-class Web applications. While ASP.NET is largely syntax compatible with Active Server Pages (ASP), it provides a new programming model and infrastructure that allow you to create a powerful new class of applications. ASP.NET is part of the .NET Framework and allows you to take full advantage of the features of the common language runtime, such as type safety, inheritance, language interoperability, and versioning.

ASP.NET is supported on Windows 2000 (Professional, Server, and Advanced Server), Windows XP Professional, and the Windows Server 2003 family for both client and server applications.

In addition, to develop ASP.NET server applications, the following software is also required:

- Windows 2000 Server or Advanced Server with Service Pack 2, Windows XP Professional or 64-Bit Edition, or one of the Windows Server 2003 family products.
- MDAC 2.7 for Data.
- Internet Information Services.

## 3.4 Introduction to C#:

C# (pronounced as ‘C Sharp’) is a new computer-programming language developed by Microsoft Corporation, USA. C# is a fully object-oriented language like Java and is the first Component-oriented language. It has been designed to support the key features of .NET Framework, the new development platform of Microsoft for building component-based software solutions. It is a simple, efficient, productive and type-safe language derived from the popular C and C++ languages. Although it belongs to the family of

C/C++, it is a purely object-oriented, modern language suitable for developing Web-based applications.

C# is designed for building robust, reliable and durable components to handle real-world applications. Major highlights of C# are:

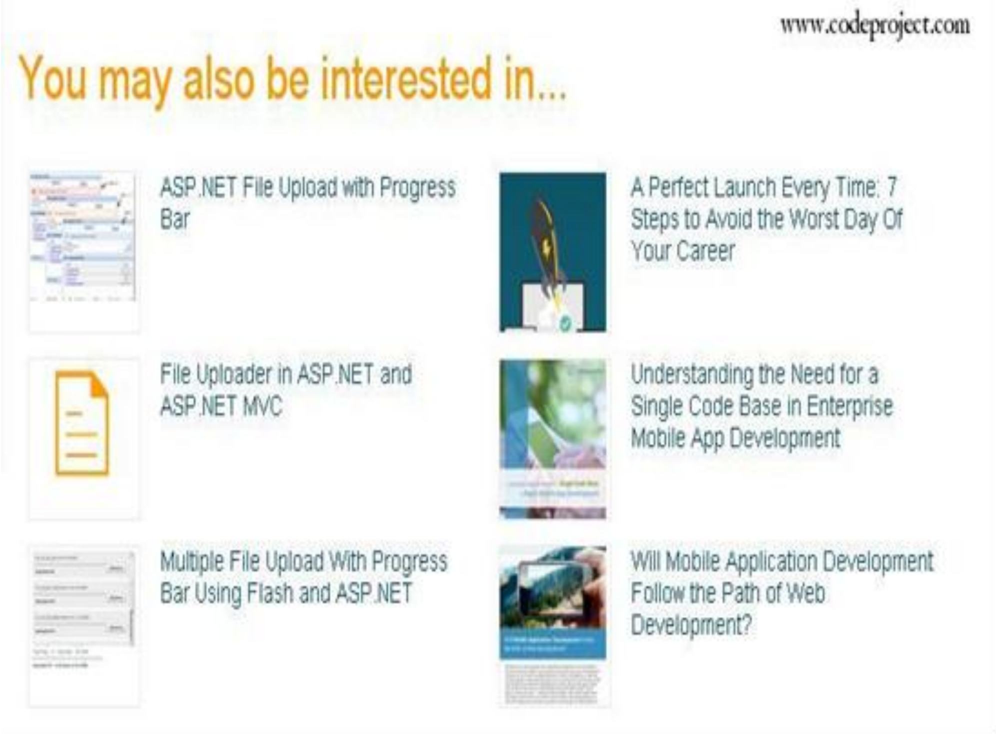
- It is a brand new language derived from the C/C++ family.
- It simplifies and modernizes C++.
- It is the only component-oriented language available today.
- It is the only language designed for the .NET Framework.
- It is a concise, lean and modern language.
- It combines the best features of many commonly used languages: the productivity of Visual Basic, the power of C++ and the elegance of Java.
- It is intrinsically object-oriented and web-enabled.
- It has a lean and consistent syntax.
- It embodies today's concern for simplicity, productivity and robustness.
- It will become the language of choice for .NET programming.
- Major parts of .NET Framework are actually coded in C#.

# Chapter 4

# LITERATURE SURVEY

Several research topics similar to ours have been used as references and a great help as knowledge sources. The primary goal is to obtain predictions of the users' next questions based on his interests. Limam, Coquil, Kosch and Brunie [1] worked on the enhancing semantic relations between the search query logs of the user using real life data from search engines. Fonseca et al. [2] research is very similar to ours. They worked on obtaining association rules between queries which are previously posted by the users. Zhang and Nasraoui [3] used the combined method of sequential search behaviour and content based similarity search to obtain query recommendations for a search engine. They use the method of graph analysis where the queries are represented as nodes in a graph and relation between the nodes is determined by the distance between each node. Cheng et al. [4] works on predicting the search pattern of the user based on the diverse browsing patterns of the user on the internet. They present three stages in their analogy – query extraction, building a model for prediction and an optimization algorithm for the model obtained in the previous stage. These diverse browsing patterns are similar to the diverse categories of questions posted by the users in our research. In Wang et al. [5] discusses the possibility of browsing the internet beyond hyperlinks. They provide a system of browsing the question logs of a user in a commercial search engine to obtain the browsing patterns and build a multi resolution topic map which can be zoomed in or zoomed out to vary the sensitivity of information. K.H Lin et al. [6] use a very similar approach to ours in predicting the users' next query. They use 3 different methods WTAL, SRPF and ACTF do encounter this issue. They determine that WTAL is the best among the methods. The paper also shows that historical data of a long time may not be suitable for prediction. Dupret and Mendoza [8] propose a system of recommending better queries for the user. They provide an improvised query as a recommendation for the user instead of providing related queries based on the user's question log. They also predict the date even when there are no matching keywords to be searched. They achieve this by ranking the data in the query log of the user. In [8], Madaan, Sharma and Dixit have used the Apriori algorithm to obtain association rules among the keywords present in the query log of the users for prediction of the next query to the user. This research is very much similar to ours in terms of working and algorithm usage. Much of the previous knowledge required has been obtained from their research. After analyzing all the mentioned resources, we have come up with our project which aims on achieving a better result and overcoming all the disadvantages presented in the other researches. We have taken a sample question log of a user for 100 days and analyzed it offline to obtain our experimental results. Our aim is to enhance the QA system to have efficient prediction and a better interaction with the users to make it user friendly.

*For the Question - "Uploading file using Asp.Net C#", the related blogs.*



www.codeproject.com

You may also be interested in...

ASP.NET File Upload with Progress Bar

A Perfect Launch Every Time: 7 Steps to Avoid the Worst Day Of Your Career

File Uploader in ASP.NET and ASP.NET MVC

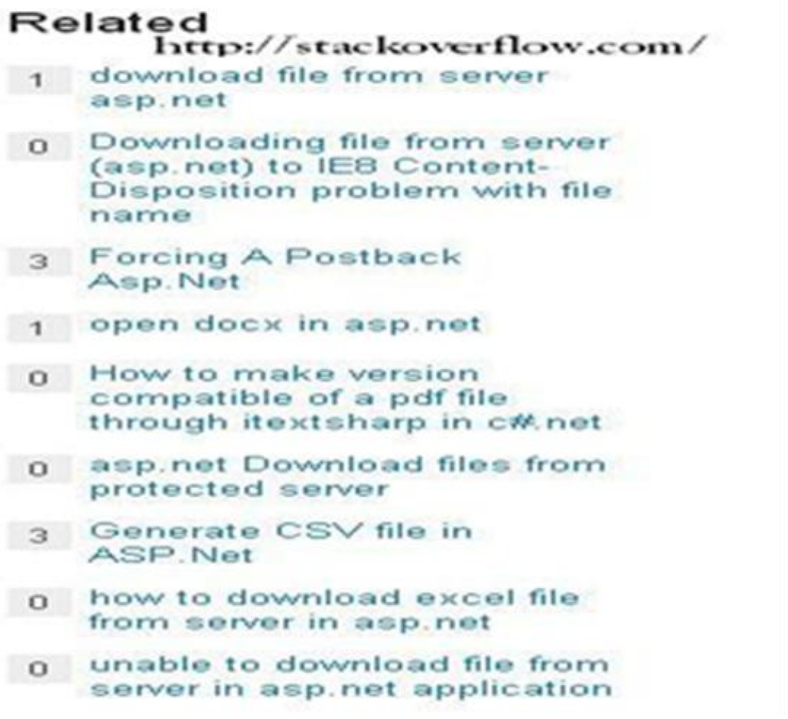
Understanding the Need for a Single Code Base in Enterprise Mobile App Development

Multiple File Upload With Progress Bar Using Flash and ASP.NET

Will Mobile Application Development Follow the Path of Web Development?

**Fig 4.1** Snap(1)of related blogs collected from different existing QA systems

*For the Question - "downloading file using asp.net and c#", the related blogs.*



Related

http://stackoverflow.com/

1 download file from server asp.net

0 Downloading file from server (asp.net) to IE8 Content-Disposition problem with file name

3 Forcing A Postback Asp.Net

1 open docx in asp.net

0 How to make version compatible of a pdf file through itextsharp in c#.net

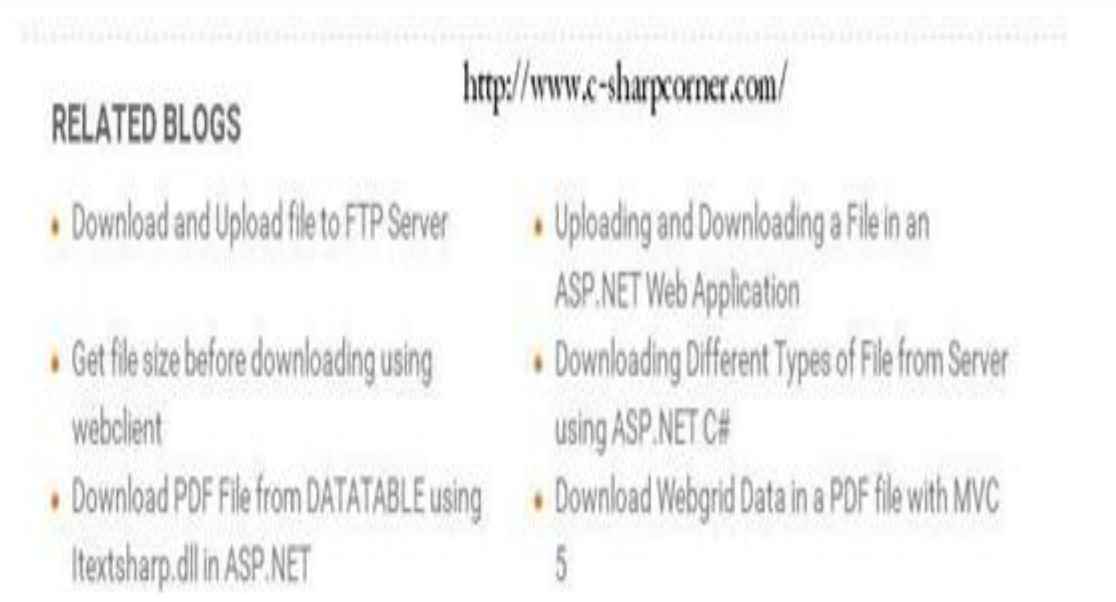
0 asp.net Download files from protected server

3 Generate CSV file in ASP.Net

0 how to download excel file from server in asp.net

0 unable to download file from server in asp.net application

**Fig 4.2** Snap (2)of related blogs collected from different existing QA systems



http://www.c-sharpcorner.com/

RELATED BLOGS

- Download and Upload file to FTP Server
- Get file size before downloading using webclient
- Download PDF File from DATATABLE using Itextsharp.dll in ASP.NET
- Uploading and Downloading a File in an ASP.NET Web Application
- Downloading Different Types of File from Server using ASP.NET C#
- Download Webgrid Data in a PDF file with MVC 5

**Fig 4.3** Snap( 3) of related blogs collected from different existing QA systems

# Chapter 5

# SYSTEM DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases particularly testing and maintenance.

The design activity often results in three separate outputs –

- Architecture design.
- High level design.
- Detailed design.

**Architecture Design:**

Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired result. The focus is on identifying components or subsystems and how they connect. In other words, the focus is on what major components are needed.

**High Level Design:**

In high level design identifies the modules that should be built for developing the system and the specifications of these modules. At the end of system design all major data structures, file format, output formats, etc., are also fixed. The focus is on identifying the modules. In other words, the attention is on what modules are needed.

**Detailed Design:**

In the detailed design the internal logic of each of the modules is specified. The focus is on designing the logic for each of the modules. In other words how modules can be implemented in software is the issue.

A design methodology is a systematic approach to creating a design by application of a set of techniques and guidelines. Most methodologies focus on high level design.

## 5.1 Architectural Design:

In this project **three** tier architecture is used.

**Introduction:**

As a developer, the .NET framework and Visual Studio present many choices for choosing the right architecture, from placing the data access code directly in the UI through datasets and data source controls, to creating a data access layer that talks to the database, all the way to creating an n-tier architecture approach that consists of multiple layers, and use data-transfer objects to pass data back and forth.

**Layer:**

A layer is a reusable portion of code that performs a specific function. In the .NET environment, a layer is usually setup as a project that represents this specific function. This specific layer is in charge of working with other layers to perform some specific goal. In an application where the presentation layer needs to extract information from a backend database, the presentation would utilize a series of layers to retrieve the data, rather than having the database calls embedded directly within itself. Now we will look briefly at the latter situation first.

## Two-Tier Architecture

When the .NET 2.0 framework became available to the world, there were some neat features that allowed the developer to connect the framework’s GUI controls directly to the database. This approach is very handy when rapidly developing applications. However, it’s not always favorable to embed all of the business logic and data access code directly in the website, for several reasons:

- Putting all of the code in the web site (business logic and data access) can make the application harder to maintain and understand.
- Reusing database queries in the presentation layer often isn’t done, because of the typical data source control setup in the ASP.NET framework.

- Relying on the data source controls can make debugging more difficult, often due to vague error messages.

So in looking for an alternative, we can separate the data access code and business logic into separate “layers”.

## Three tier Architecture:

Three tier architecture consists of three layers. They are:

a) **The Data Layer:**

The key component to most applications is the data. The data has to be served to the presentation layer somehow. The data layer is a separate component (often setup as a separate single or group of projects in a .NET solution), whose sole purpose is to serve up the data from the database and return it to the caller. Through this approach, data can be logically reused, meaning that a portion of an application reusing the same query can make a call to one data layer method, instead of embedding the query multiple times. This is generally more maintainable.

b) **Business Layer:**

Though a web site could talk to the data access layer directly, it usually goes through another layer called the business layer. The business layer is vital in that it validates the input conditions before calling a method from the data layer. This ensures the data input is correct before proceeding, and can often ensure that the outputs are correct as well. This validation of input is called business rules, meaning the rules that the business layer uses to make “judgments” about the data.

One of the best reasons for reusing logic is that applications that start off small usually grow in functionality. The business layer helps move logic to a central layer for “maximum reusability.”

c) **Presentation Layer:**

The ASP.NET web site or windows forms application (the UI for the project) is called the presentation layer. The presentation layer is the most important layer simply because it’s the one that everyone sees and uses. Even with a well structured business and data layer, if the presentation layer is designed poorly, this gives the users a poor view of the system.

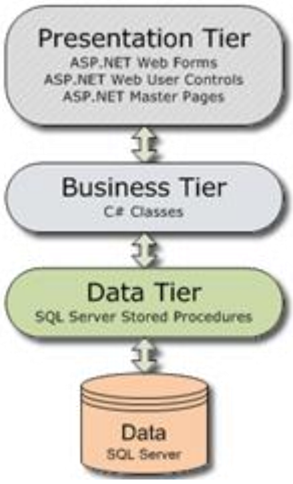


**Fig 5.1** Three tier Architecture.

The **presentation tier** contains the UI (User Interface) elements of the site, and includes all the logic that managers the interaction between the visitor and the client’s business. (ASP.NET Web Forms, Web User Controls, ASP.NET Master Pages)

The **business tier** receives requests from the presentation tier and returns a result to the presentation tier depending on the business logic it contains. (C# Classes)

The **data tier** is responsible for storing the application’s data and sending it to the business tier when requested. (SQL Server Stored Procedures).

## 5.2 High level design:

## Context data flow diagram:

It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the 'Level 0 DFD') the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

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

Some of the benefits of a Context Diagram are:

- Shows the scope and boundaries of a system at a glance including the other systems that interface with it
- No technical knowledge is assumed or required to understand the diagram
- Easy to draw and amend due to its limited notation
- Easy to expand by adding different levels of DFDs
- Can benefit a wide audience including stakeholders, business analyst, data analysts, developers

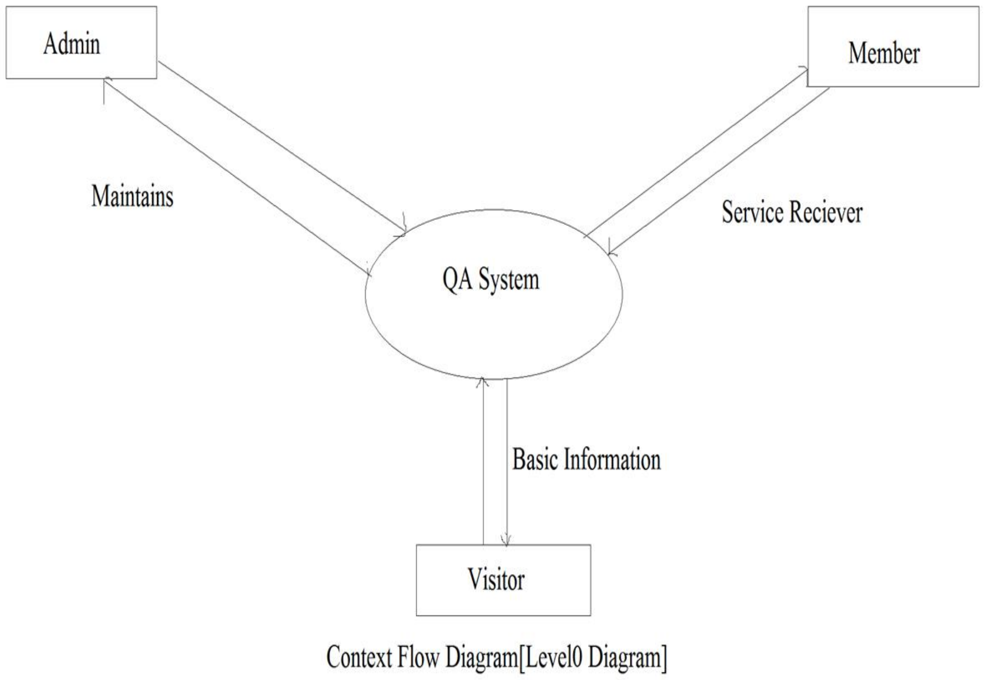


**Fig 5.2** Context Flow Diagram ( level 0 diagram)

## Data Flow Diagram:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

A DFD provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD).

## Symbols used in DFD's:

### Processes:

A process transforms data values. The lowest processes are our functions without side effects.

### Data Flows:

A data flow connects the output of an object or process to the input of another object or process. It represents the intermediate data values within the computation. It is draws as an arrow between the procedure and the consumer of the data value. The arrow is labeled with the description of the data, usually its name or type.

### Actors:

An actor is an active object that drives the data flow graph by producing or consuming values. Actors are attached to the inputs and the outputs of a dataflow graph. In sense, the actors lie on the boundary of the flow graph but terminate the flow of data as sources and sinks of data, and so are sometimes called terminators.

**Data Store:**

A data store is a passive object within a data flow diagram that stores data for later access. Unlike an actor, a data store does not generate any operations on its own but merely responds to requests to store and access data.

Data flow diagrams (DFDs) are the method of choice over technical descriptions for three principal reasons.

1. DFDs are easier to understand by technical and nontechnical audiences

2. DFDs can provide a high level system overview, complete with boundaries and connections to other systems

3. DFDs can provide a detailed representation of system components

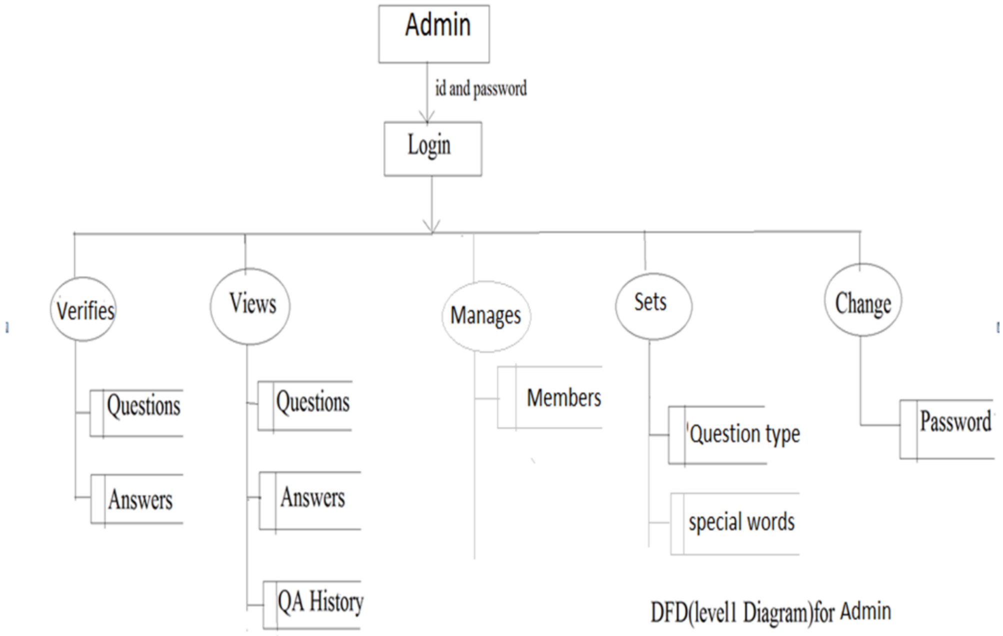


**Fig 5.3** DFD (level 1 diagram) for Admin

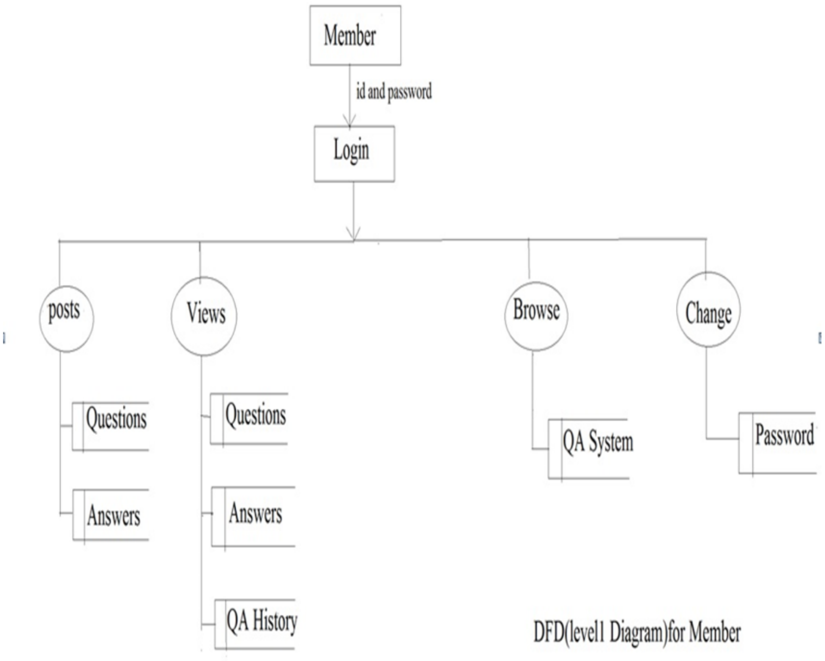


**Fig 5.4** DFD( level 1 diagram ) for Member

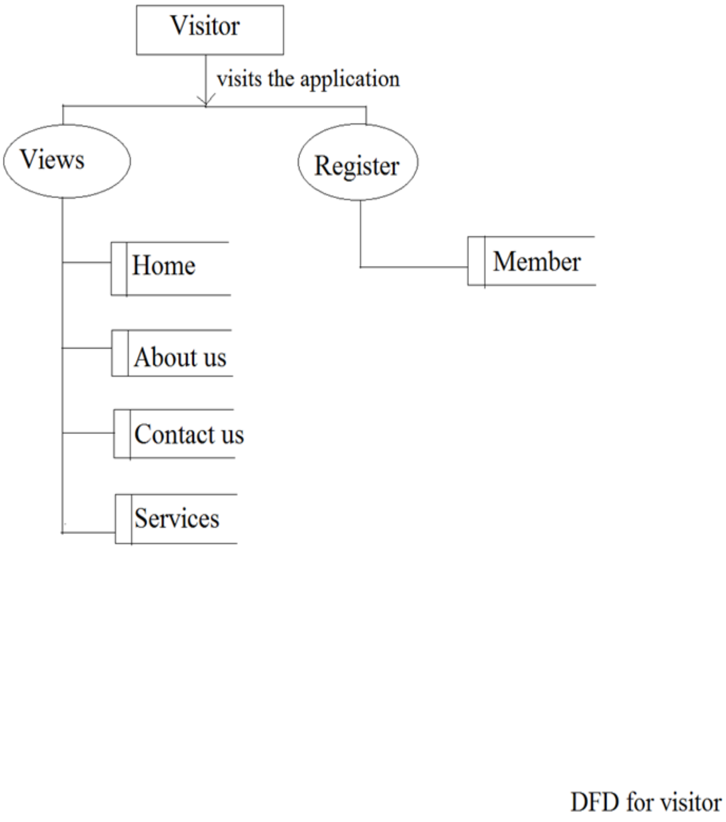


**Fig 5.5** DFD of Visitor

## 5.3 Detailed Design:

### USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.An important part of the Unified Modelling Language (UML) is the facilities for drawing use case diagrams. Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors and use cases.

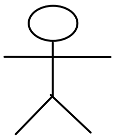
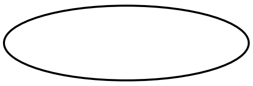
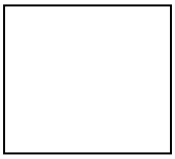
| **SYMBOL** | **PURPOSE** |
|---|---|
|  | Actor |
| | Association |
|  | Business use case |
|  | Subject |

**Table 5.1** Use Case Diagram Symbols

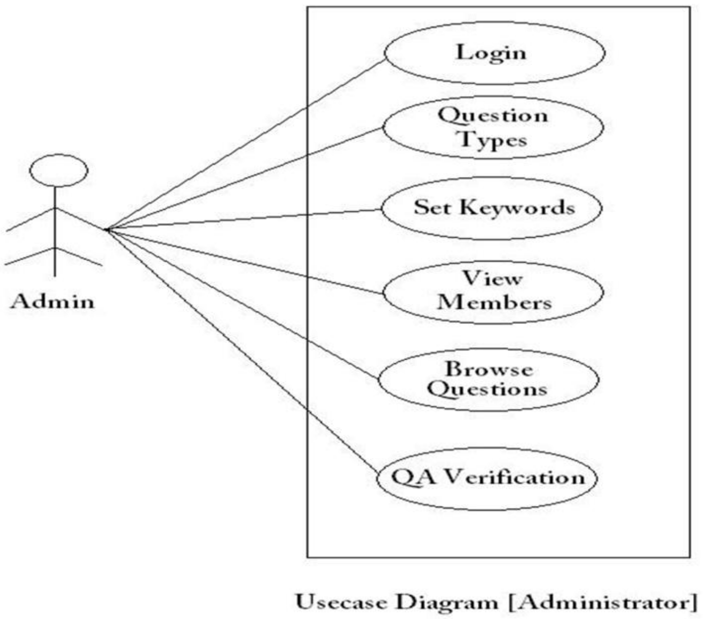


**Fig 5.6** Usecase Diagram (Admin)

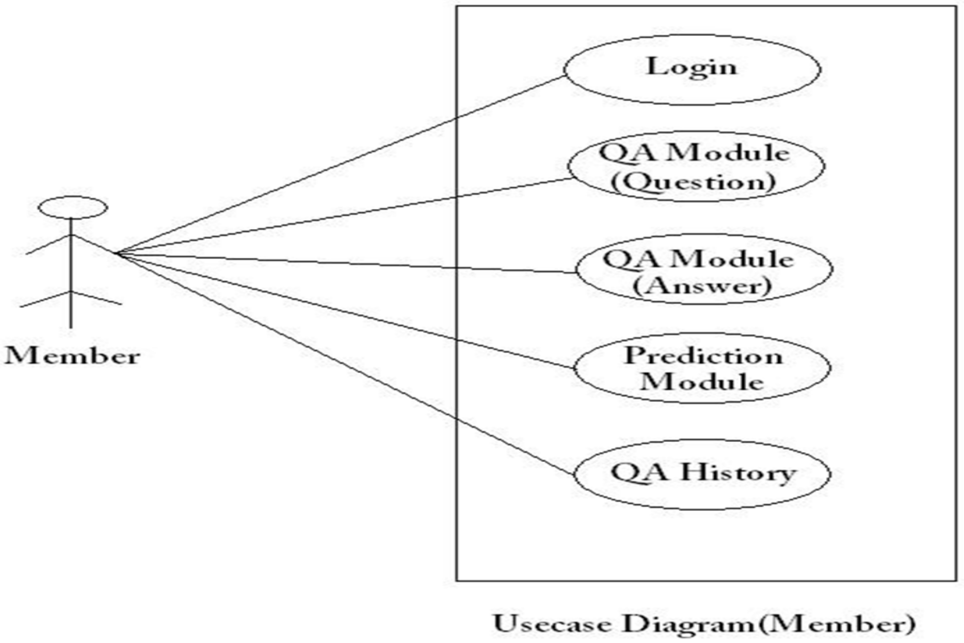


**Fig 5.7** Usecase Diagram (Member)

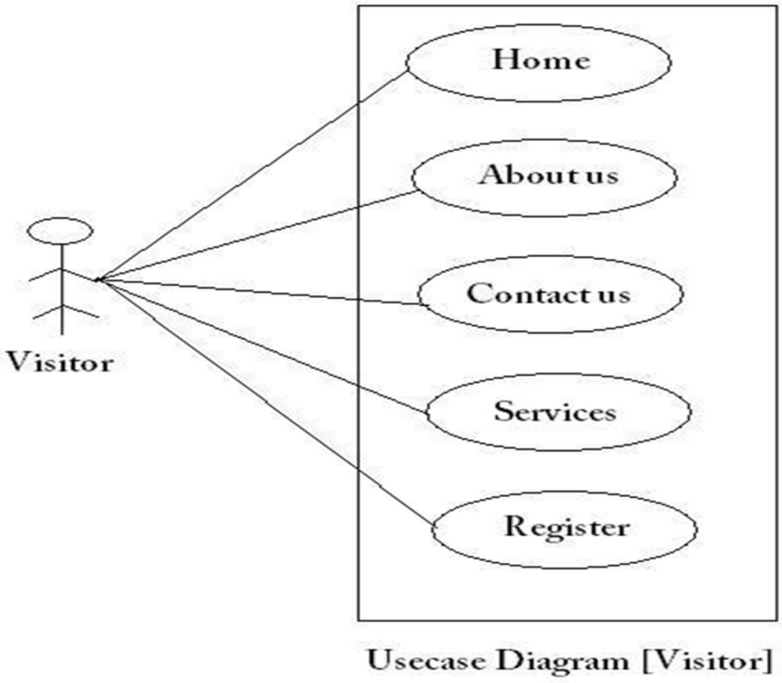
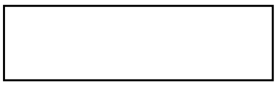


**Fig 5.8** Usecase Diagram (Visitor)

## E-R DIAGRAM:

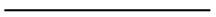
E-R diagram also termed as entity relationship diagram shows relationship between various tables in the data base. It includes symbols like entity(strong),weak entity, attribute, primary key, relation ,link, multi valued attributes, derived attributes, total participation. The symbols used in this E-R diagram are as follows:

**ENTITY:**



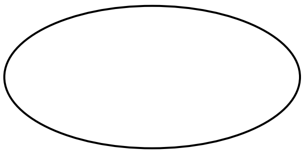
The basic object that the E-R model represents is an entity, which is a thing in the real world with an independent existence. An entity may be an object with an physical existence or it may be an object with a conceptual existence.

**LINKS:**



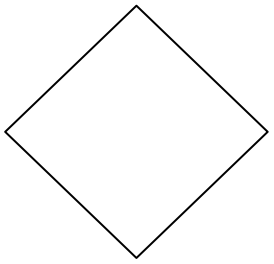
It is connection between an each entities and relationship

**ATTRIBUTES:**



Each entity has attributes –the particular properties that describe it. The attribute values that describe each entity become a major part of the data stored in the database.

**RELATIONSHIP:**



It describes the relation between two entities.

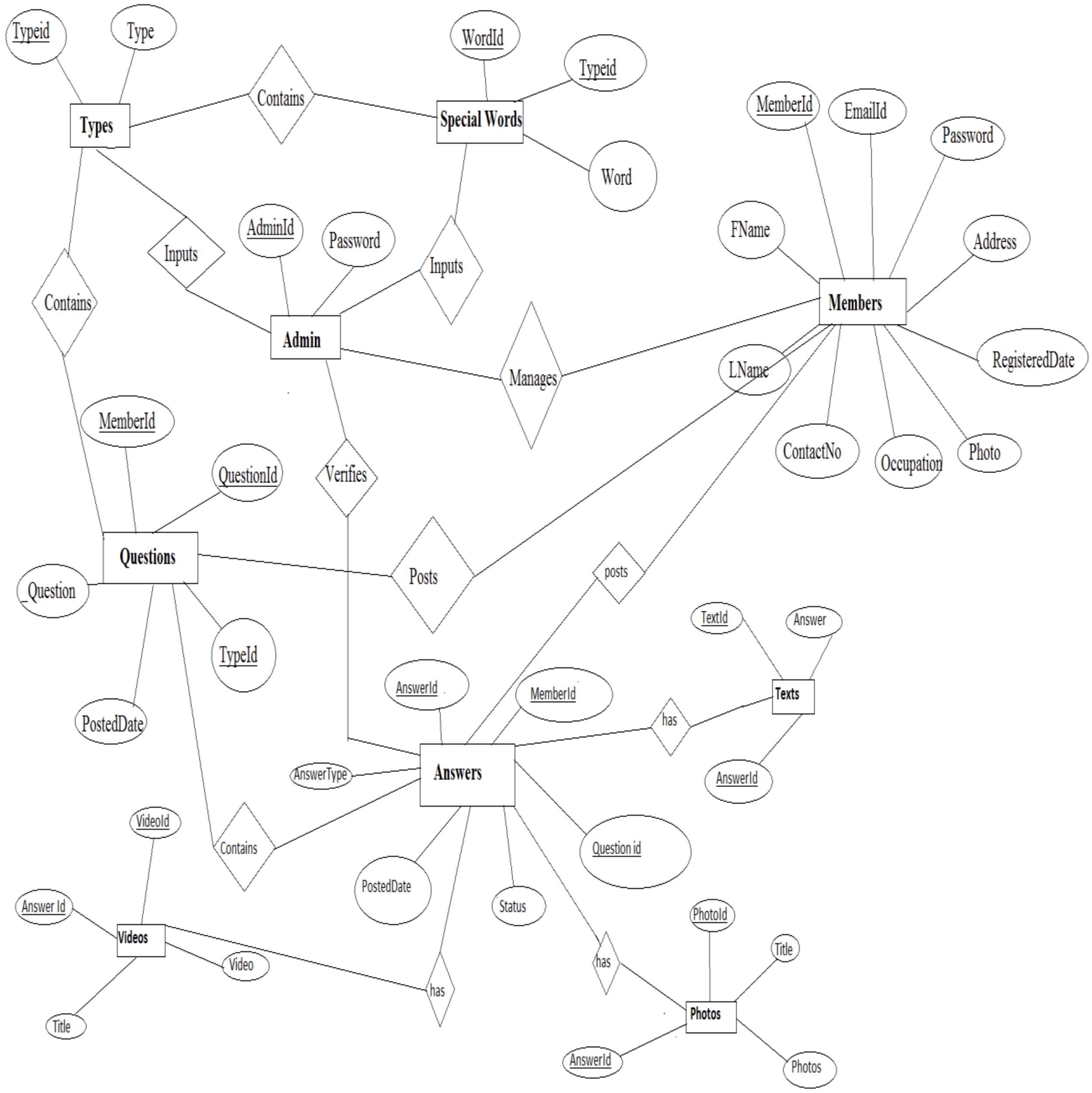


**Fig 5.9** ER diagram of QA System

## 5.4 Storage Structure:

**Database Name - QADatabase**

**Table 1 – tblAdmin**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| AdminId | String (Varchar[20]) | Not null | Primary Key |
| Password | String (Varchar[20]) | | |

**Table 2 – tblTypes**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| TypeId | Int, Auto Generate | Not null | Primary Key |
| Type | String (Varchar[20]) | | |

**Table 3 – tblWords**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| WordId | Int, Auto Generate | Not null | Primary Key |
| TypeId | Int | | Foreign Key |
| Word | String (Varchar[20]) | | |

**Table 4 – tblMembers**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| MemberId | Int, Auto Generate | Not null | Primary Key |
| EmailId | String (Varchar[20]) | | |
| Password | String (Varchar[20]) | | |
| Fname | String (Varchar[20]) | | |
| Lname | String (Varchar[20]) | | |
| Address | String (Varchar[50]) | | |
| ContactNo | String (Varchar[10]) | | |
| Occupation | String (Varchar[20]) | | |
| Photo | String (Varchar[500]) | | |
| RegisteredDate | DateTime | | |

**Table 5 – tblQuestions**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| QuestionId | Int, Auto Generate | Not null | Primary Key |
| TypeId | Int | Not null | Foreign Key |

| | | | |
|---|---|---|---|
| Question | String (Varchar[500]) | | |
| MemberId | Int | Not Null | Foreign Key |
| PostedDate | DateTime | | |

**Table 6 – tblAnswers**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| AnswerId | Int, Auto Generate | Not null | Primary Key |
| QuestionId | Int | Not null | Foreign Key |
| MemberId | Int | Not null | Foreign Key |
| AnswerType | String (Varchar[50]) | | |
| PostedDate | DateTime | | |
| Status | String (Varchar[50]) | | |

**Table 7 – tblTexts**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| TextId | Int, Auto Generate | Not null | Primary Key |
| AnswerId | Int | Not null | Foreign Key |
| Answer | String (Varchar[max]) | | |

**Table 8 – tblPhotos**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| PhotoId | Int, Auto Generate | Not null | Primary Key |
| AnswerId | Int | Not null | Foreign Key |
| Title | String (Varchar[500]) | | |
| Photo | String (Varchar[max]) | | |

**Table 9 – tblVideos**

| FIELD | TYPE | NULL | KEY |
|---|---|---|---|
| VideoId | Int, Auto Generate | Not null | Primary Key |
| AnswerId | Int | Not null | Foreign Key |
| Title | String (Varchar[500]) | | |
| Video | String (Varchar[max]) | | |

## Relations established in database tables

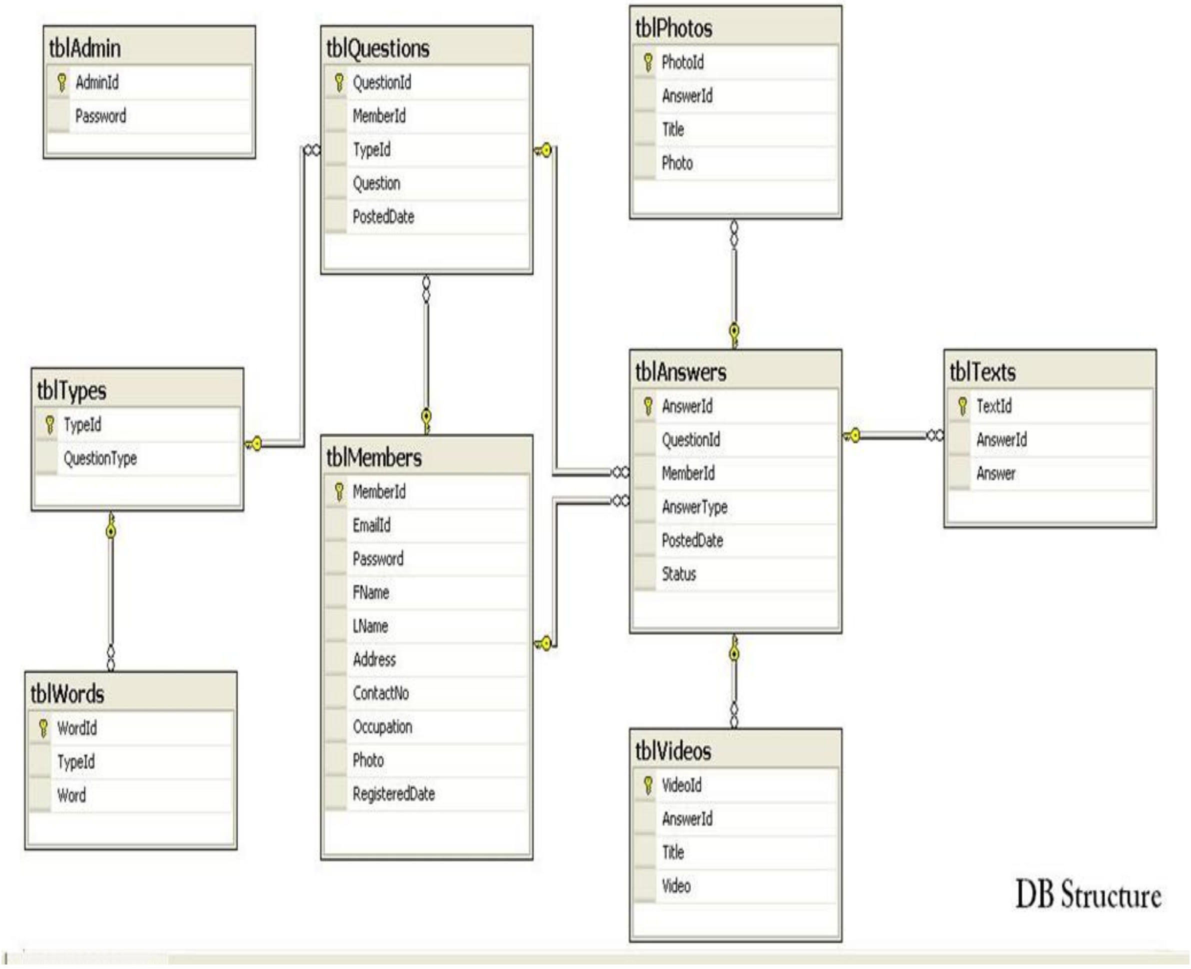


**Fig 5.10** Relations established between the tables in QA System

# Chapter 6

# SYSTEM IMPLEMENTATION

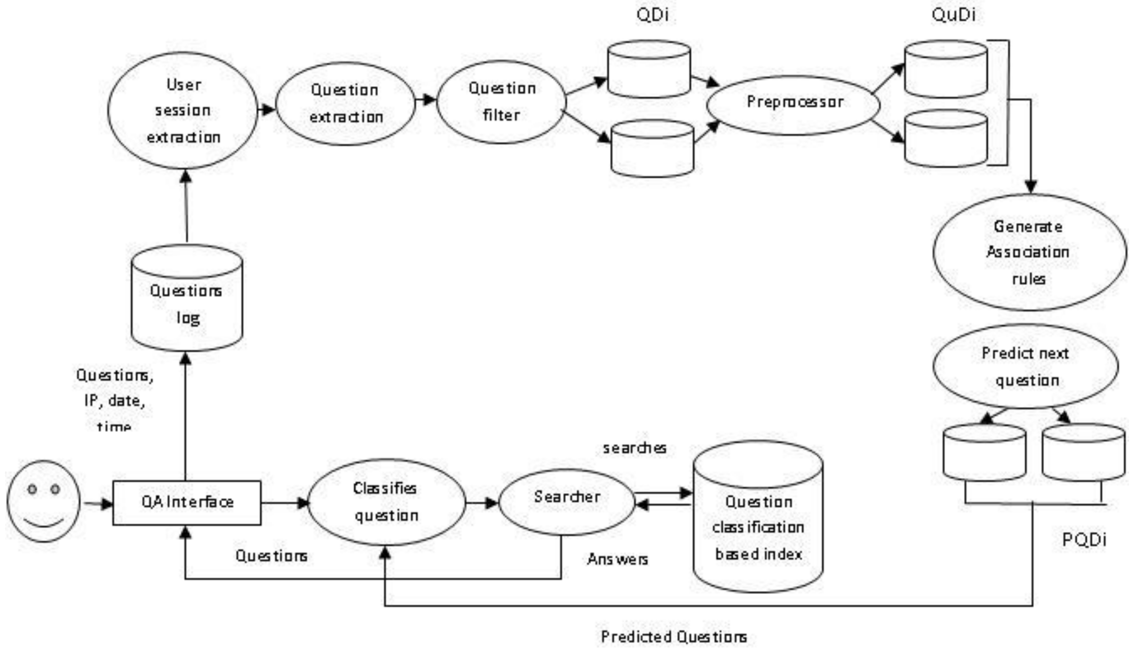


**Fig 6.1** Diagram showing the working of QA System

## 6.1 Working of QA system

- The user enters his question on the interface of the QA system.
- This question is classified by the Question classifier module.
- Search is done by the searcher module for the answer(s) to the questions.
- The interactions between the user and the system are maintained in a file termed as Question Log

**User's Next Question Prediction**

- **User Session Extraction**

Let us consider a set of user sessions S= {S1, S2,... Sn} where n is the number of user sessions that have been extracted and a set of QIDs Q={QID1, QID2,.... QIDm} where m is the number of Questions in Si each identified by a QID.

- **Question Extraction**

This module works for the extraction of the Questions from the user sessions that have been extracted. For this purpose, this module uses the Questions log that has been maintained for the users' interactions.

- **Question Filter Module**

This module takes as its input the questions extracted by the Question extraction module and then separates the Questions on the basis of their Question type. A separate database QDi is maintained for the questions belonging to each Question type QTi. Each QDi contains QTi and a set of questions.

- **Question Pre-processor**

The Question pre-processor separates the Question type from the Questions and converts the remaining part of the Question into a query. A query is a set of terms contained in the Question.

- **Association Rules Generation**

There are two major steps involved in the Apriori algorithm:

Step1: Finding the itemsets having minimum support (frequent itemsets, also called large itemsets). Let the transaction data is given as follows:

- query1: Student, Teach, School
- query 2: Student, School
- query 3: Teach, School, City, Game
- query 4: Baseball, Basketball, teach, school
- query 5: Basketball, Baseball, Player, Spectator

Assume the two thresholds be minsup=30% (0.3) and minconf= 80% (0.8).

Some example frequent itemsets:

- Itemset1: {Teach, school} has sup= 0.6.
- Itemset 2:{Baseball, Basketball} has sup=0.4.

  Both of these itemsets satisfy the criterion sup>=minsup, so these are selected as frequent itemsets and are considered for the next step to generate association rules.

- Step 2: Generation of association rules using frequent itemsets.

  Using the frequent itemsets generated in the above step for generating association rules as follows:

- Rule 1:teach->school, sup=2/5 (0.6), conf=3/3 (1)
- Rule 2: school-> teach sup= 2/5 (0.6), conf=3/4 (0.7)
- Rule 3:baseball->basketball, sup= 2/5 (0.4), conf=2/2 (1)
- Rule 4:basketball->baseball, sup= 2/5 (0.4), conf=2/2 (1)

  So, associations rules that satisfy the criterion of conf>=minconf are Rule1, 3 and 4.

- **Next Question Predictor**

  This module takes as its input the association rules that have been generated by Association Rule Generation module and predict the next Question by the association rules generated. A predicted questions database is maintained say PQDi for each QuDi. Each PQDi contains QTi and a set of predicted questions.

- For e.g., if the user is looking for “what is testing”, then the module predicts that he or she may be interested in “what is testing in software”.
- Similarly, if the user is looking for “how to play basketball” then he or she may be interested in “how to play basketball and baseball”

**SUPPORT:**

The support of the item set. It is the relationship between the total numbers of transaction containing the item with the total number of transaction in the data set.

$$\text{Support (A->B)} = \frac{\text{The total number of transaction containing A and B}}{\text{The total number of transaction in the data set}}$$

**CONFIDENCE:**

Confidence is defined as a relationship between the total number of transaction containing the frequently bought items with the total number of transaction Containing only A.

$$\text{Confidence (A->B)} = \frac{\text{The total number of transaction containing A and B}}{\text{The total number of transaction containing only A}}$$

| | | |
|---|---|---|
| • **DM** | **:** | Data Mining |
| • **OS** | **:** | Operating System |
| • **SRS** | **:** | Software Requirement Specifications |

## 6.2 Association Rules

## Apriori Algorithm

*STEP 1:* Scan the opinion data set and determine the support(s) of each item.

*STEP 2:* Generate L1 (Frequent one item set).

*STEP 3:* Use Lk-1, join Lk-1 to generate the set of candidate k - item set.

*STEP 4:* Scan the candidate k item set and generate the support of each candidate k – item set.

*STEP 5:* Add to frequent item set, until C=Null Set.

*STEP 6:* For each item in the frequent item set generate all non empty subsets.

*STEP 7:* For each non empty subset determine the confidence. If confidence is greater than or equal to this specified confidence .Then add to Strong Association Rule.

**Sample Example**

| TID | Symptoms-diseases-drugs |
|---|---|
| 1 | A,C,D |
| 2 | A,C,E |
| 3 | A,B,C,E |
| 4 | B,E |

**Minimum Support = 50%**

**Minimum Confidence = 80%**

Item set □ A, B, C, D, and E

C1

| Items | Support |
|---|---|
| A | 75% |
| B | 50% |
| C | 75% |
| D | 25% |
| E | 75% |

**L1**

| Items | Support |
|---|---|
| A | 75% |
| B | 50% |
| C | 75% |
| E | 75% |

**C2**

| Items | Support |
|---|---|
| AB | 25% |
| AC | 75% |
| AE | 50% |
| BC | 25% |
| BE | 50% |
| CE | 50% |

**L2**

| Items | Support |
|---|---|
| AC | 75% |
| AE | 50% |
| BE | 50% |
| CE | 50% |

**C3**

| Items | Support |
|---|---|
| ACE | 50% |
| ABC | 25% |
| ABE | 25% |
| BCE | 25% |

**L3**

| Items | Support |
|---|---|
| ACE | 50% |

**FREQUENT ITEM SET (L)**

| Items | Support |
|---|---|
| A | 75% |
| B | 50% |
| C | 75% |
| E | 75% |

| | |
|---|---|
| AC | 75.00% |
| AE | 50% |
| BE | 50% |
| CE | 50% |
| ACE | 50% |

**GENERATE CONFIDENCE:**

| RULE X RULE Y | CONFIDENCE |
|---|---|
| {A} □ {C} | 100.00% |
| {C} □ {A} | 100.00% |
| {A} □ {E} | 66% |
| {E} □ {A} | 66% |
| {B} □ {E} | 100% |
| {E} □ {B} | 66% |
| {C} □ {E} | 66% |
| {E} □ {C} | 66% |
| {A} □ {CE} | 66% |
| {C} □ {AE} | 66% |
| {E} □ {AC} | 66% |
| {CE} □ {A} | 100% |
| {AE} □ {C} | 100% |
| {AC} □ {E} | 66.00% |

**STRONG ASSOCIATION RULE:**

- {B} □ {E}
- {CE} □ {A}
- {AE} □ {C}
- {A} □ {C}
- {C} □ {A}

**Association Analysis**

• This is used to discover patterns that describe strongly associated features in the data.

• The goal is to extract the most interesting patterns in an efficient manner.

• Useful applications include

→ finding groups of genes that have related functionality or

→ identifying web pages that are accessed together

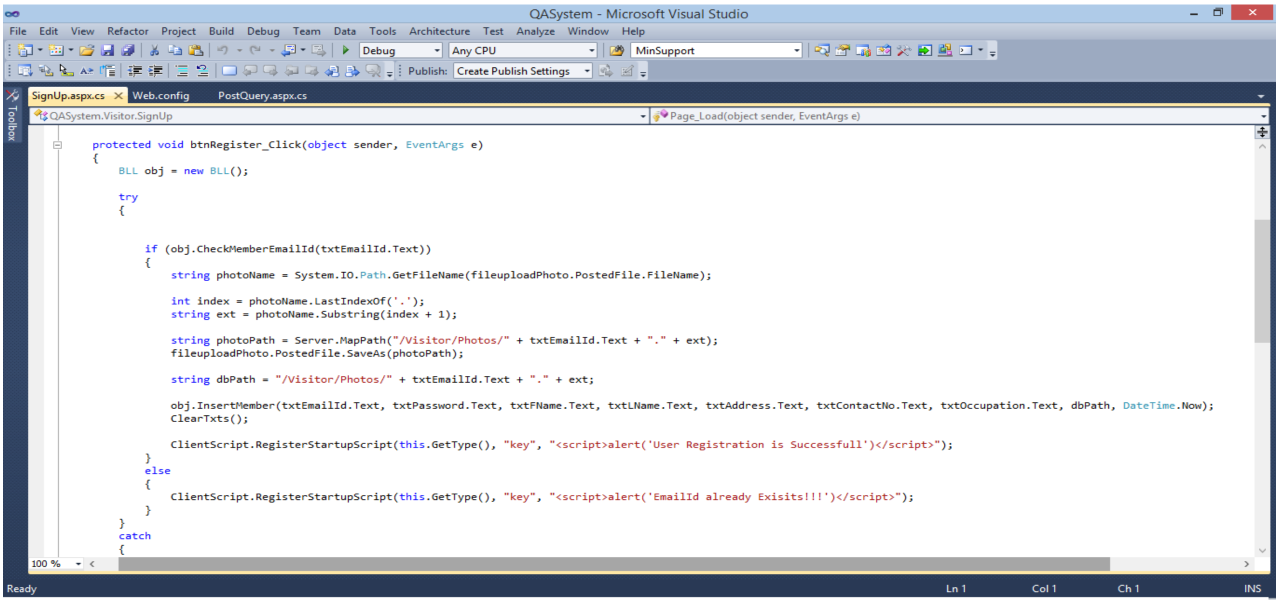
• Ex: market based analysis

We may discover the rule that {diapers} -> {Milk}, which suggests that customers who buy diapers also tend to buy milk.

# Chapter 7

# CODING

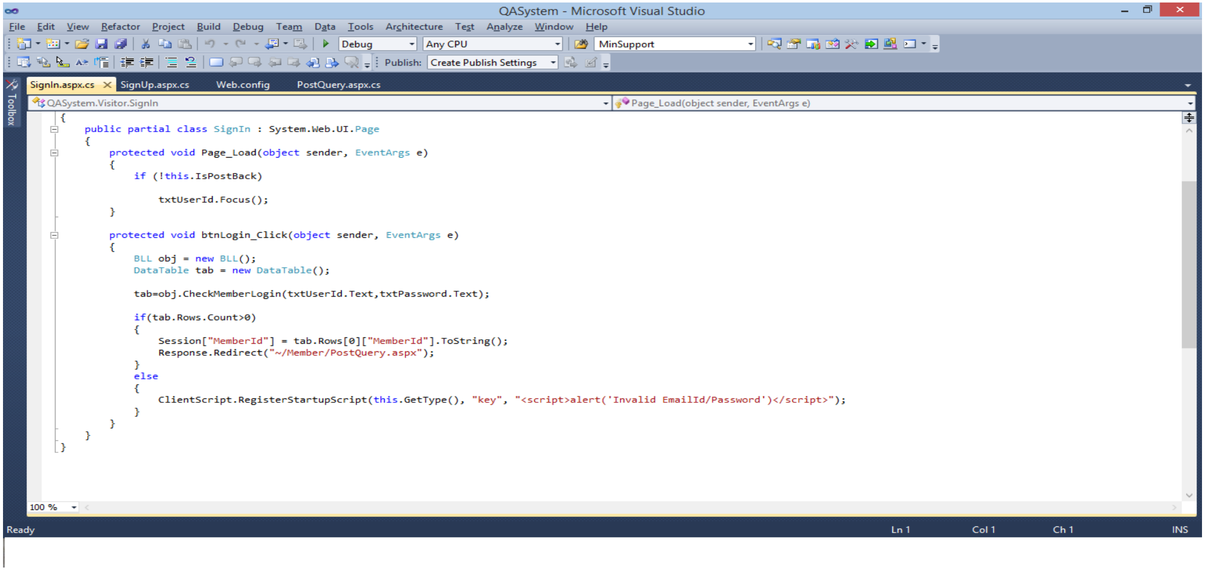
## 7.1 Registration



QASystem - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Any CPU
MinSupport
Publish: Create Publish Settings
SignUp.aspx.cs
Web.config
PostQuery.aspx.cs
QASystem.Visitor.SignUp
Page_Load(object sender, EventArgs e)
100 %
Ready
Ln 1
Col 1
Ch 1
INS

```
protected void btnRegister_Click(object sender, EventArgs e)
{
    BLL obj = new BLL();

    try
    {


        if (obj.CheckMemberEmailId(txtEmailId.Text))
        {
            string photoName = System.IO.Path.GetFileName(fileuploadPhoto.PostedFile.FileName);

            int index = photoName.LastIndexOf('.');
            string ext = photoName.Substring(index + 1);

            string photoPath = Server.MapPath("/Visitor/Photos/" + txtEmailId.Text + "." + ext);
            fileuploadPhoto.PostedFile.SaveAs(photoPath);

            string dbPath = "/Visitor/Photos/" + txtEmailId.Text + "." + ext;

            obj.InsertMember(txtEmailId.Text, txtPassword.Text, txtFName.Text, txtLName.Text, txtAddress.Text, txtContactNo.Text, txtOccupation.Text, dbPath, DateTime.Now);
            ClearTxts();

            ClientScript.RegisterStartupScript(this.GetType(), "key", "<script>alert('User Registration is Successfull')</script>");
        }
        else
        {
            ClientScript.RegisterStartupScript(this.GetType(), "key", "<script>alert('EmailId already Exisits!!!')</script>");
        }
    }
    catch
    {
```

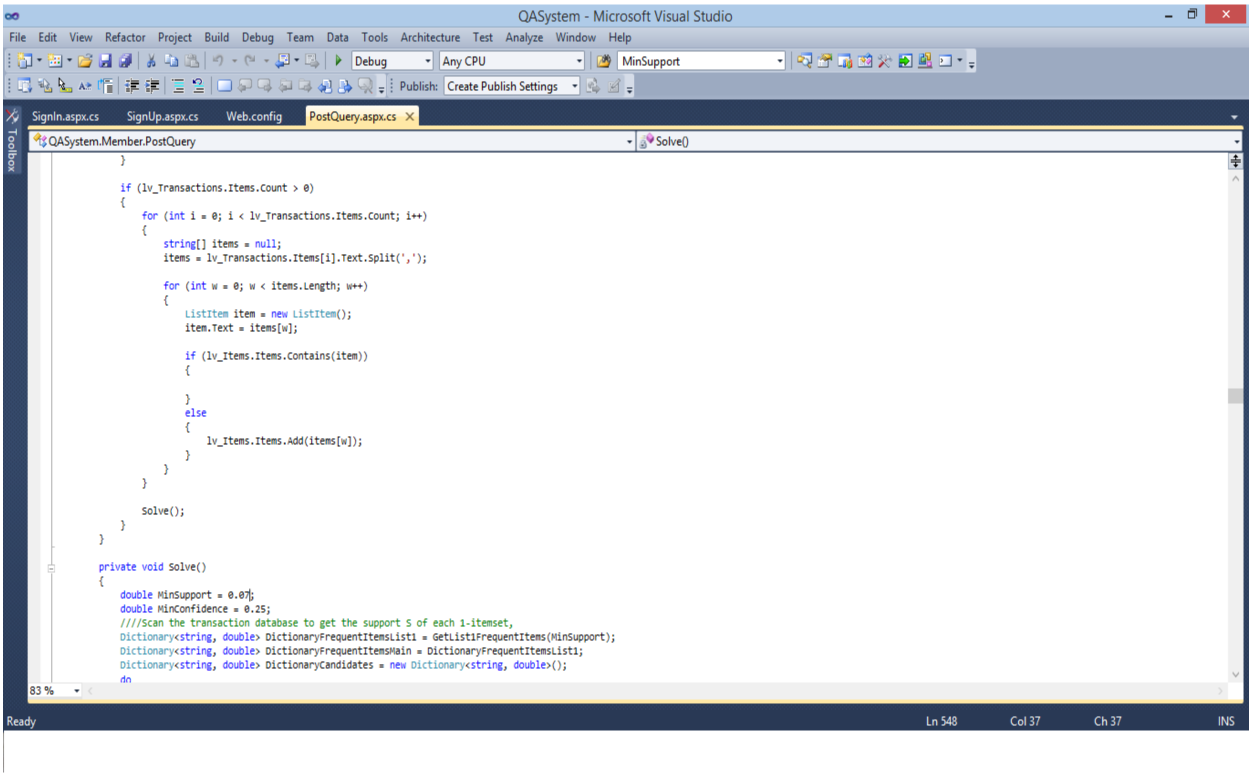
## 7.2 Member Login



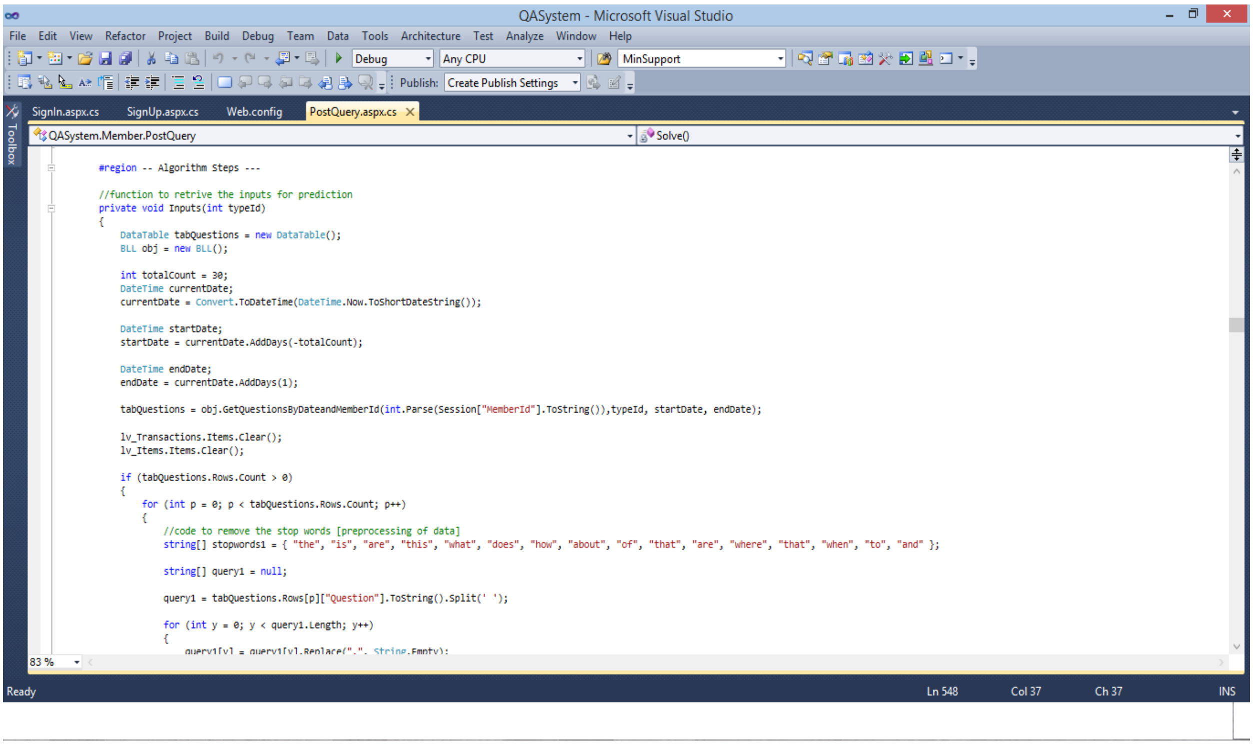
QASystem - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Any CPU
MinSupport
Publish: Create Publish Settings
SignIn.aspx.cs
SignUp.aspx.cs
Web.config
PostQuery.aspx.cs
QASystem.Visitor.SignIn
Page_Load(object sender, EventArgs e)
100 %
Ready
Ln 1
Col 1
Ch 1
INS

```
{
    public partial class SignIn : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!this.IsPostBack)

                txtUserId.Focus();
        }

        protected void btnLogin_Click(object sender, EventArgs e)
        {
            BLL obj = new BLL();
            DataTable tab = new DataTable();

            tab=obj.CheckMemberLogin(txtUserId.Text,txtPassword.Text);

            if(tab.Rows.Count>0)
            {
                Session["MemberId"] = tab.Rows[0]["MemberId"].ToString();
                Response.Redirect("~/Member/PostQuery.aspx");
            }
            else
            {
                ClientScript.RegisterStartupScript(this.GetType(), "key", "<script>alert('Invalid EmailId/Password')</script>");
            }
        }
    }
}
```

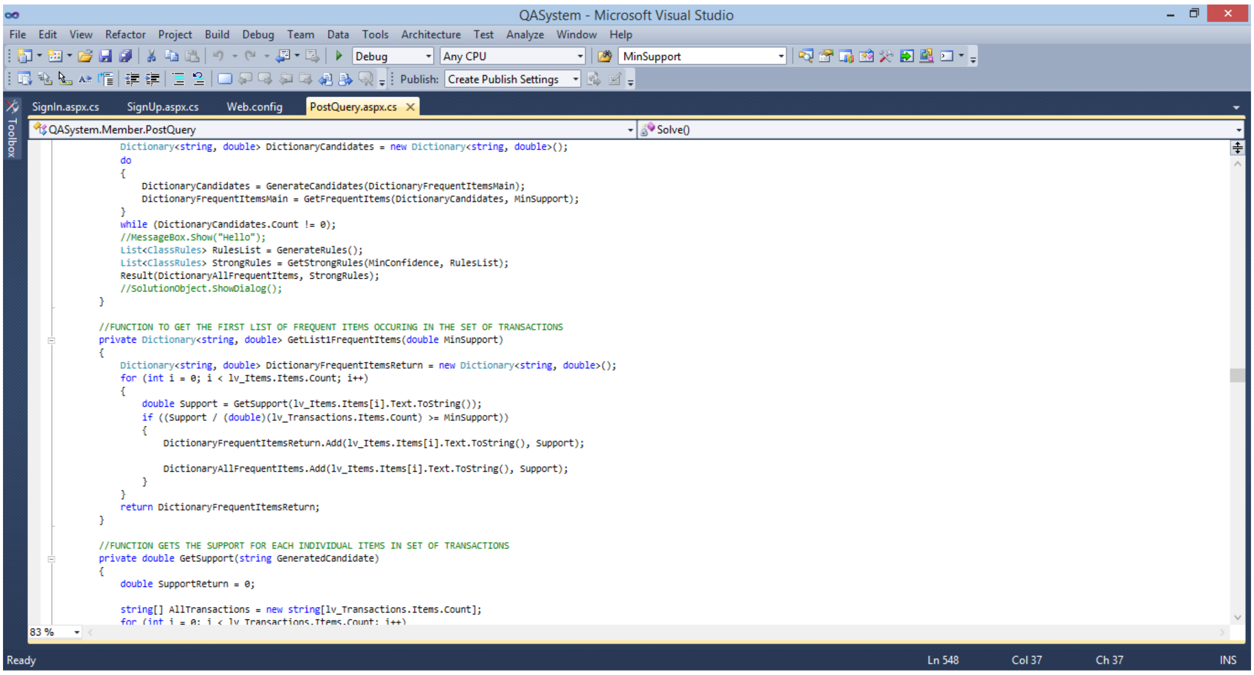
## 7.3 Algorithm


QASystem - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Any CPU
MinSupport
Publish: Create Publish Settings
Toolbox
SignIn.aspx.cs
SignUp.aspx.cs
Web.config
PostQuery.aspx.cs
QASystem.Member.PostQuery
Solve()
83 %
Ready
Ln 548
Col 37
Ch 37
INS

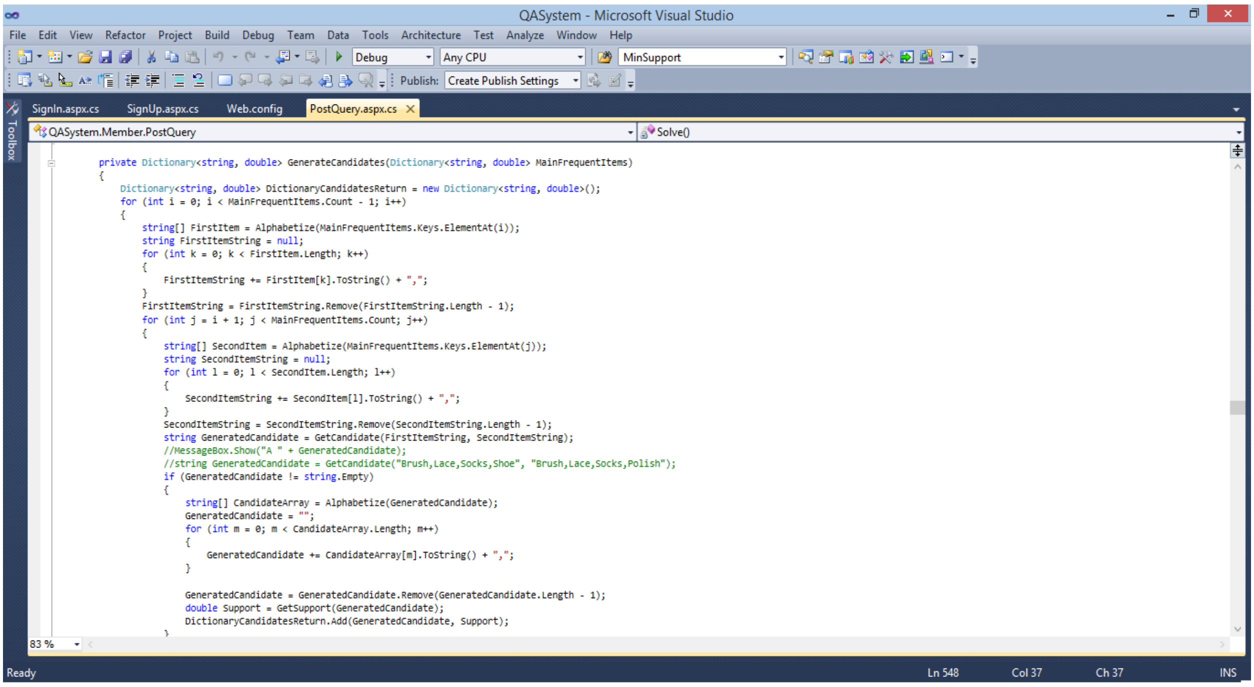
```
            }

            if (lv_Transactions.Items.Count > 0)
            {
                for (int i = 0; i < lv_Transactions.Items.Count; i++)
                {
                    string[] items = null;
                    items = lv_Transactions.Items[i].Text.Split(',');

                    for (int w = 0; w < items.Length; w++)
                    {
                        ListItem item = new ListItem();
                        item.Text = items[w];

                        if (lv_Items.Items.Contains(item))
                        {

                        }
                        else
                        {
                            lv_Items.Items.Add(items[w]);
                        }
                    }
                }

                Solve();
            }
        }

        private void Solve()
        {
            double MinSupport = 0.07;
            double MinConfidence = 0.25;
            ////Scan the transaction database to get the support S of each 1-itemset,
            Dictionary<string, double> DictionaryFrequentItemsList1 = GetList1FrequentItems(MinSupport);
            Dictionary<string, double> DictionaryFrequentItemsMain = DictionaryFrequentItemsList1;
            Dictionary<string, double> DictionaryCandidates = new Dictionary<string, double>();
            do
```


QASystem - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Any CPU
MinSupport
Publish: Create Publish Settings
Toolbox
SignIn.aspx.cs
SignUp.aspx.cs
Web.config
PostQuery.aspx.cs
QASystem.Member.PostQuery
Solve()
83 %
Ready
Ln 548
Col 37
Ch 37
INS

```
        #region -- Algorithm Steps ---

        //function to retrive the inputs for prediction
        private void Inputs(int typeId)
        {
            DataTable tabQuestions = new DataTable();
            BLL obj = new BLL();

            int totalCount = 30;
            DateTime currentDate;
            currentDate = Convert.ToDateTime(DateTime.Now.ToShortDateString());

            DateTime startDate;
            startDate = currentDate.AddDays(-totalCount);

            DateTime endDate;
            endDate = currentDate.AddDays(1);

            tabQuestions = obj.GetQuestionsByDateandMemberId(int.Parse(Session["MemberId"].ToString()),typeId, startDate, endDate);

            lv_Transactions.Items.Clear();
            lv_Items.Items.Clear();

            if (tabQuestions.Rows.Count > 0)
            {
                for (int p = 0; p < tabQuestions.Rows.Count; p++)
                {
                    //code to remove the stop words [preprocessing of data]
                    string[] stopwords1 = { "the", "is", "are", "this", "what", "does", "how", "about", "of", "that", "are", "where", "that", "when", "to", "and" };

                    string[] query1 = null;

                    query1 = tabQuestions.Rows[p]["Question"].ToString().Split(' ');

                    for (int y = 0; y < query1.Length; y++)
                    {
                        query1[y] = query1[y].Replace(".", String.Empty);
```


QASystem - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Any CPU
MinSupport
Publish: Create Publish Settings
SignIn.aspx.cs
SignUp.aspx.cs
Web.config
PostQuery.aspx.cs
Toolbox
QASystem.Member.PostQuery
Solve()
83 %
Ready
Ln 548
Col 37
Ch 37
INS

```
            Dictionary<string, double> DictionaryCandidates = new Dictionary<string, double>();
            do
            {
                DictionaryCandidates = GenerateCandidates(DictionaryFrequentItemsMain);
                DictionaryFrequentItemsMain = GetFrequentItems(DictionaryCandidates, MinSupport);
            }
            while (DictionaryCandidates.Count != 0);
            //MessageBox.Show("Hello");
            List<ClassRules> RulesList = GenerateRules();
            List<ClassRules> StrongRules = GetStrongRules(MinConfidence, RulesList);
            Result(DictionaryAllFrequentItems, StrongRules);
            //SolutionObject.ShowDialog();
        }

        //FUNCTION TO GET THE FIRST LIST OF FREQUENT ITEMS OCCURING IN THE SET OF TRANSACTIONS
        private Dictionary<string, double> GetList1FrequentItems(double MinSupport)
        {
            Dictionary<string, double> DictionaryFrequentItemsReturn = new Dictionary<string, double>();
            for (int i = 0; i < lv_Items.Items.Count; i++)
            {
                double Support = GetSupport(lv_Items.Items[i].Text.ToString());
                if ((Support / (double)(lv_Transactions.Items.Count) >= MinSupport))
                {
                    DictionaryFrequentItemsReturn.Add(lv_Items.Items[i].Text.ToString(), Support);

                    DictionaryAllFrequentItems.Add(lv_Items.Items[i].Text.ToString(), Support);
                }
            }
            return DictionaryFrequentItemsReturn;
        }

        //FUNCTION GETS THE SUPPORT FOR EACH INDIVIDUAL ITEMS IN SET OF TRANSACTIONS
        private double GetSupport(string GeneratedCandidate)
        {
            double SupportReturn = 0;

            string[] AllTransactions = new string[lv_Transactions.Items.Count];
            for (int i = 0; i < lv_Transactions.Items.Count; i++)
```


QASystem - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Any CPU
MinSupport
Publish: Create Publish Settings
SignIn.aspx.cs
SignUp.aspx.cs
Web.config
PostQuery.aspx.cs
Toolbox
QASystem.Member.PostQuery
Solve()
83 %
Ready
Ln 548
Col 37
Ch 37
INS

```
        private Dictionary<string, double> GenerateCandidates(Dictionary<string, double> MainFrequentItems)
        {
            Dictionary<string, double> DictionaryCandidatesReturn = new Dictionary<string, double>();
            for (int i = 0; i < MainFrequentItems.Count - 1; i++)
            {
                string[] FirstItem = Alphabetize(MainFrequentItems.Keys.ElementAt(i));
                string FirstItemString = null;
                for (int k = 0; k < FirstItem.Length; k++)
                {
                    FirstItemString += FirstItem[k].ToString() + ",";
                }
                FirstItemString = FirstItemString.Remove(FirstItemString.Length - 1);
                for (int j = i + 1; j < MainFrequentItems.Count; j++)
                {
                    string[] SecondItem = Alphabetize(MainFrequentItems.Keys.ElementAt(j));
                    string SecondItemString = null;
                    for (int l = 0; l < SecondItem.Length; l++)
                    {
                        SecondItemString += SecondItem[l].ToString() + ",";
                    }
                    SecondItemString = SecondItemString.Remove(SecondItemString.Length - 1);
                    string GeneratedCandidate = GetCandidate(FirstItemString, SecondItemString);
                    //MessageBox.Show("A " + GeneratedCandidate);
                    //string GeneratedCandidate = GetCandidate("Brush,Lace,Socks,Shoe", "Brush,Lace,Socks,Polish");
                    if (GeneratedCandidate != string.Empty)
                    {
                        string[] CandidateArray = Alphabetize(GeneratedCandidate);
                        GeneratedCandidate = "";
                        for (int m = 0; m < CandidateArray.Length; m++)
                        {
                            GeneratedCandidate += CandidateArray[m].ToString() + ",";
                        }

                        GeneratedCandidate = GeneratedCandidate.Remove(GeneratedCandidate.Length - 1);
                        double Support = GetSupport(GeneratedCandidate);
                        DictionaryCandidatesReturn.Add(GeneratedCandidate, Support);
                    }
```

# CONCLUSION

A next question prediction system that predicts the users' next requests based on their current interactions with the system has been presented and experimented.The approach of Association rules for the prediction of the user's next question to the QA system has been discussed and applied. The proposed system works for the prediction of the next question in which the user may become interested in his next interaction with the QA system. The system uses the concept of Association rules for finding out that the user's next question to the QA system. The technique of Association rule discovery is found as one of the most important techniques in the field of data mining. The objective of the technique is to find patterns that exist in the database. The proposed system has shown remarkable performance for the experimental corpus taken and has shown promising results.

# REFERENCES

[1] L. Limam, D. Coquil, H. Kosch, and L. Brunie, "Extracting user interests from search query logs: A clustering approach," DEXA '10 Proceedings of the 2010 Workshops on Database and Expert Systems Applications, 2010.

[2] http://www.cs.helsinki.fi/u/ronkaine/tilome/luentomateriaali/TiLoMe-140305.pdf

[3] B. M. Fonseca, P. B. Golgher, E. S. de Moura,, and N. Ziviani, "Using Association Rules to Discover Search Engines Related Queries," in Proceedings of the First Conference on Latin American Web Congress, pp. 66–71, 2003.

[4] Z. Zhang, and O. Nasraoui, "Mining search engine query logs for query recommendation," in Proceedings of the $15^{th}$ international conference on World Wide Web, pp. 1039–1040, 2006.

[5] Z. Cheng, B. Gao, and T. Liu, "Actively predicting diverse search intent from user browsing behaviors," in Proceedings of the 19th international conference on World wide web, pp. 221–230, 2010.

[6] X. Wang, B. Tan, A. Shakery, and C. Zhai, "Beyond hyperlinks: organizing information footprints in search logs to support effective browsing," in Proceeding of the 18th ACM conference on Information and knowledge management, pp.1237–1246, 2009.

[7] D. Gupta, A. Puniya, and K.K. Bhatia, "Prediction of the Query of the Search Engine using Backpropogation Algorithm," IJCSE, 2011.

[8] K.H Lin, "Predicting Next Search Actions with Search Engine Query Logs," Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE/WIC/ACM International Conference on (Volume: 1),2011.

[9] G. Dupret , and M. Mendoza, "Recommending Better Queries Based on Click-Through Data," LNCS, Springer, 2005.

# APPENDIX A

## Snapshots

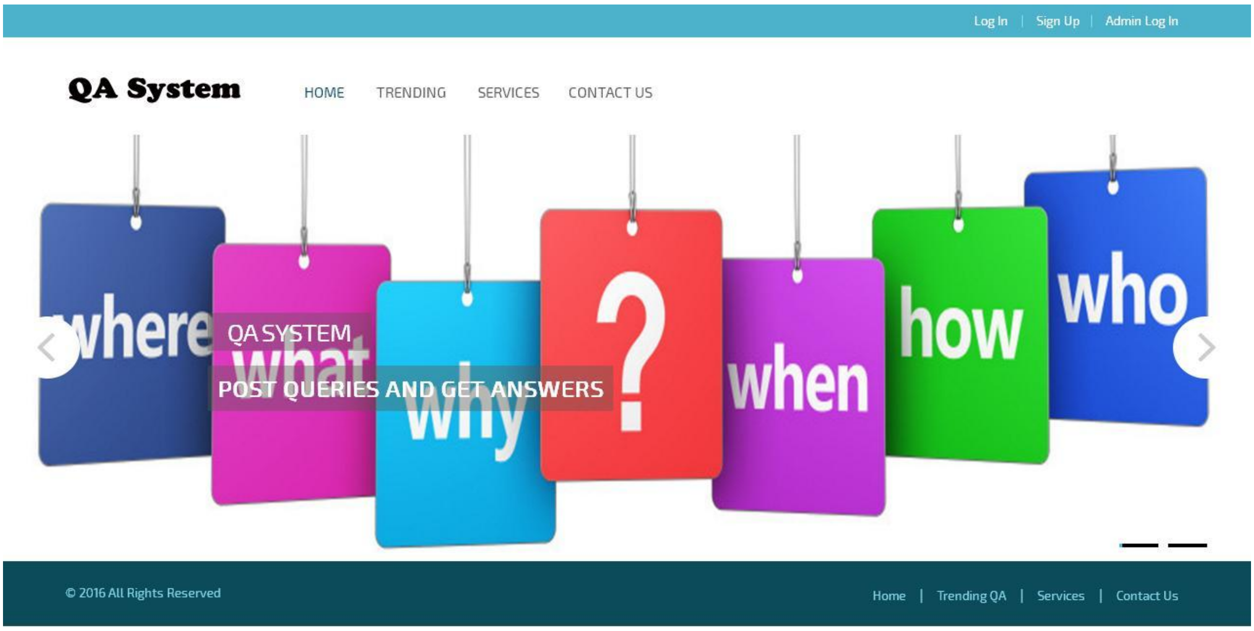
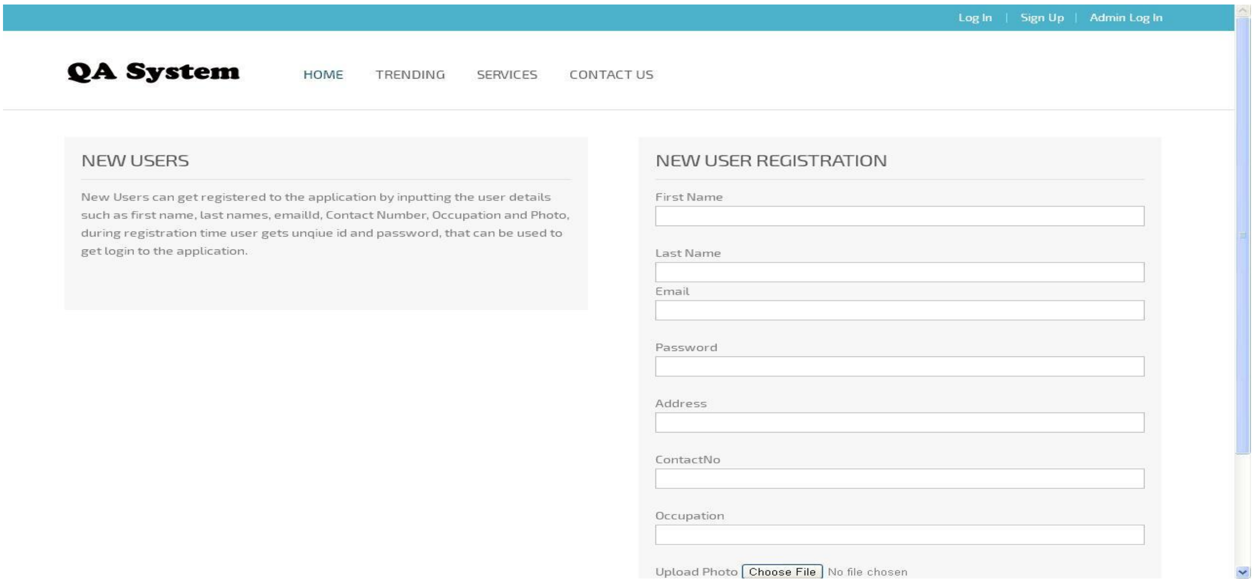

Log In | Sign Up | Admin Log In
QA System
HOME
TRENDING
SERVICES
CONTACT US
where
what
why
?
when
how
who
QA SYSTEM
POST QUERIES AND GET ANSWERS
© 2016 All Rights Reserved
Home | Trending QA | Services | Contact Us

**Fig A.1** Master page or Home page of QA System



NEW USERS

New Users can get registered to the application by inputting the user details such as first name, last names, emailId, Contact Number, Occupation and Photo, during registration time user gets unqiue id and password, that can be used to get login to the application.

NEW USER REGISTRATION

First Name

Last Name

Email

Password

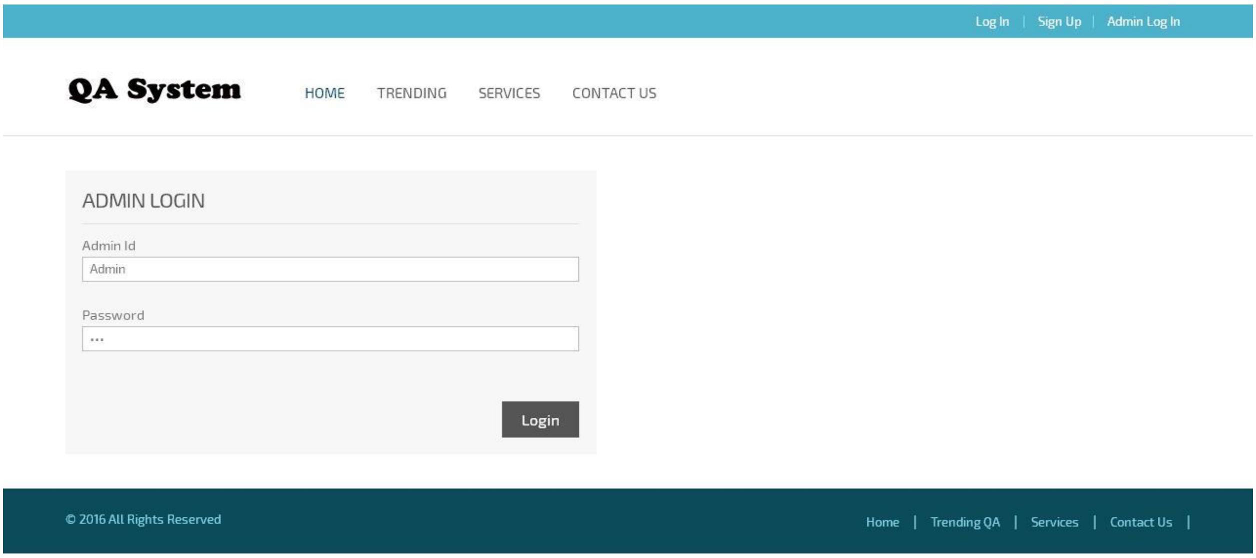
Address

ContactNo

Occupation

Upload Photo Choose File No file chosen

**Fig A.2** Registration page for new user


Log In | Sign Up | Admin Log In
QA System
HOME
TRENDING
SERVICES
CONTACT US
Login
© 2016 All Rights Reserved
Home | Trending QA | Services | Contact Us |

ADMIN LOGIN

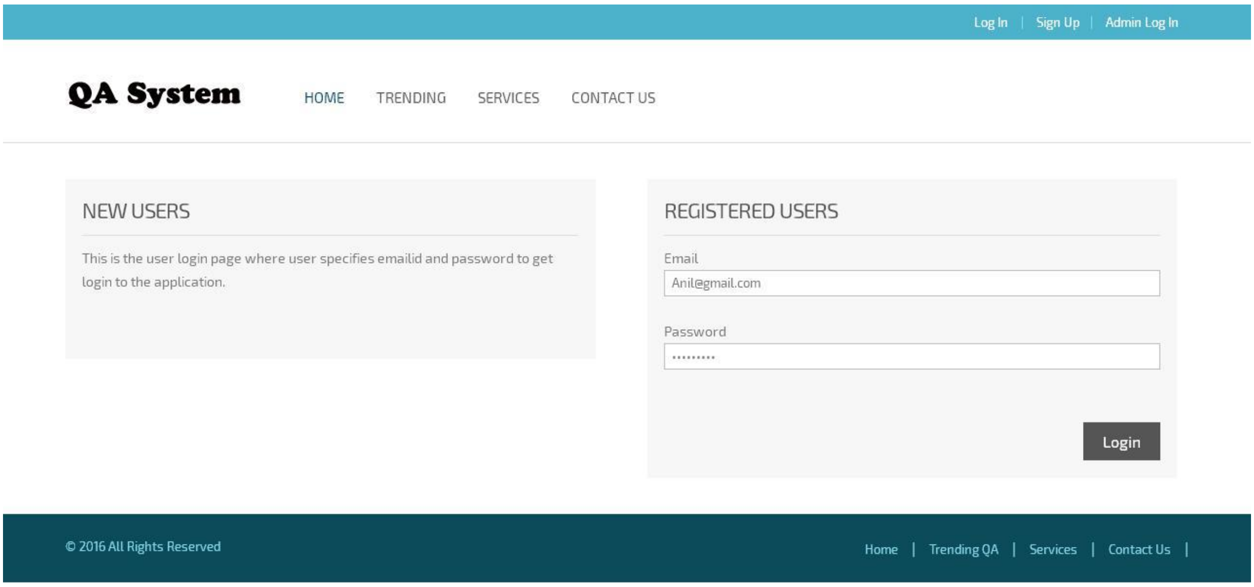
Admin Id

Admin

Password

•••

**Fig A.3** Admin login page


Log In | Sign Up | Admin Log In
QA System
HOME
TRENDING
SERVICES
CONTACT US
Login
© 2016 All Rights Reserved
Home | Trending QA | Services | Contact Us |

NEW USERS

This is the user login page where user specifies emailid and password to get login to the application.

REGISTERED USERS

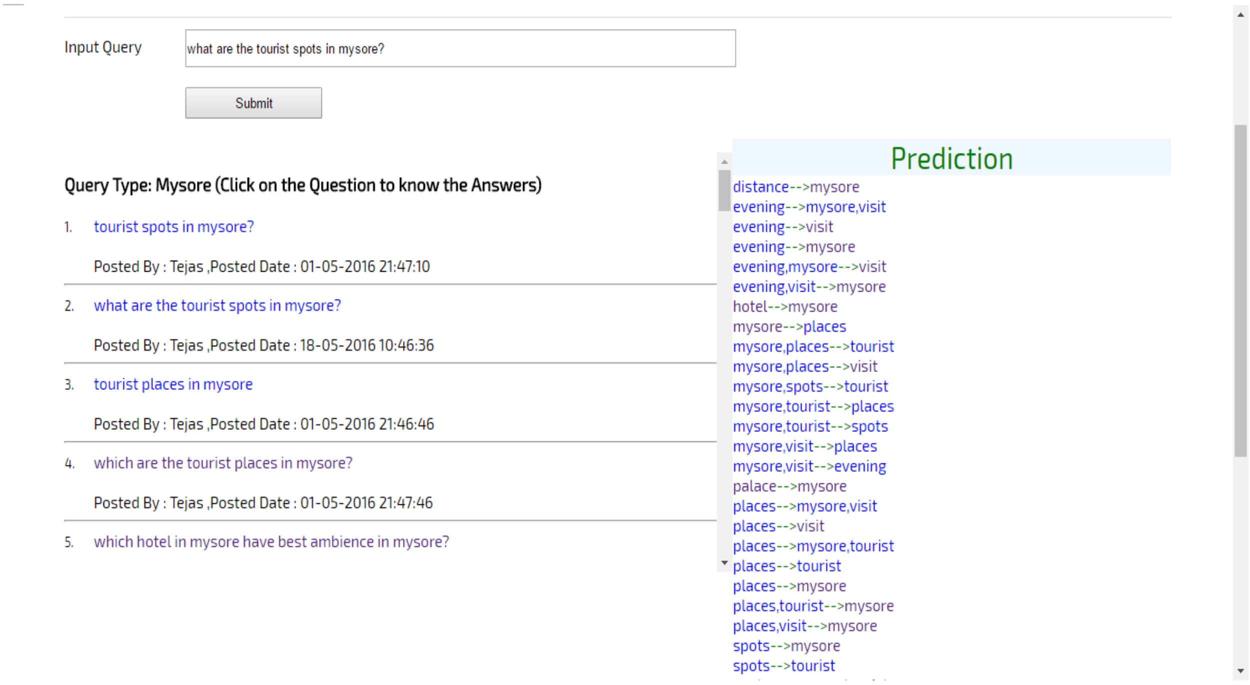
Email

Anil@gmail.com

Password

•••••••••

**Fig A.4** Member login page



Input Query: what are the tourist spots in mysore?

Submit

Query Type: Mysore (Click on the Question to know the Answers)

1. tourist spots in mysore?

   Posted By : Tejas ,Posted Date : 01-05-2016 21:47:10

2. what are the tourist spots in mysore?

   Posted By : Tejas ,Posted Date : 18-05-2016 10:46:36

3. tourist places in mysore

   Posted By : Tejas ,Posted Date : 01-05-2016 21:46:46

4. which are the tourist places in mysore?

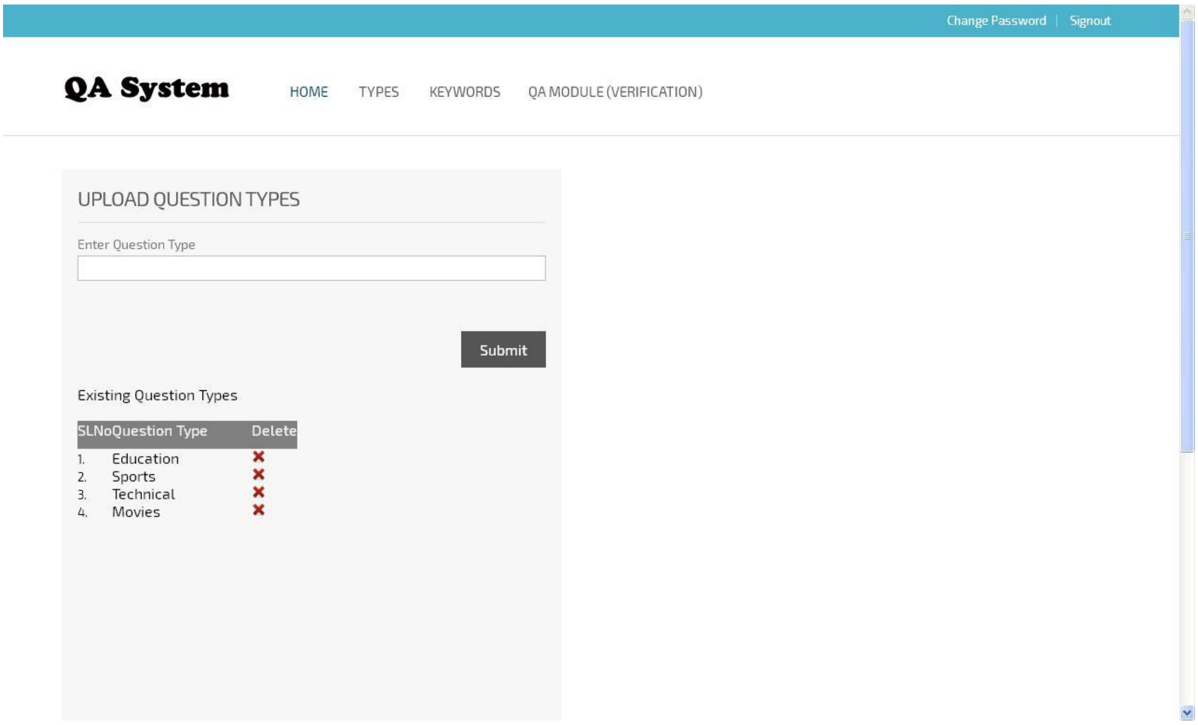
   Posted By : Tejas ,Posted Date : 01-05-2016 21:47:46

5. which hotel in mysore have best ambience in mysore?

Prediction

distance-->mysore
evening-->mysore,visit
evening-->visit
evening-->mysore
evening,mysore-->visit
evening,visit-->mysore
hotel-->mysore
mysore-->places
mysore,places-->tourist
mysore,places-->visit
mysore,spots-->tourist
mysore,tourist-->places
mysore,tourist-->spots
mysore,visit-->places
mysore,visit-->evening
palace-->mysore
places-->mysore,visit
places-->visit
places-->mysore,tourist
places-->tourist
places-->mysore
places,tourist-->mysore
places,visit-->mysore
spots-->mysore
spots-->tourist

**Fig A.5** QA Module with predictions



Change Password | Signout

UPLOAD QUESTION TYPES

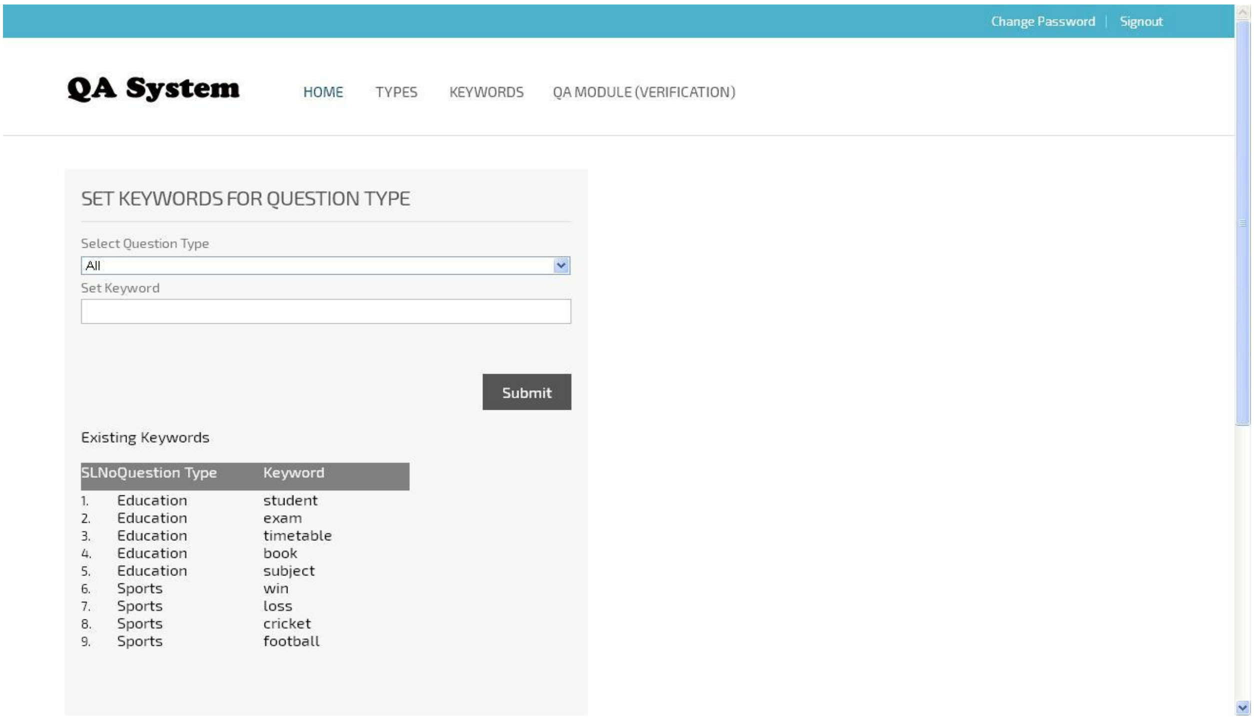
Enter Question Type

Submit

Existing Question Types

| SLNo | Question Type | Delete |
|---|---|---|
| 1. | Education | ✗ |
| 2. | Sports | ✗ |
| 3. | Technical | ✗ |
| 4. | Movies | ✗ |

**Fig A.6** Page that shows admin setting question types



SET KEYWORDS FOR QUESTION TYPE

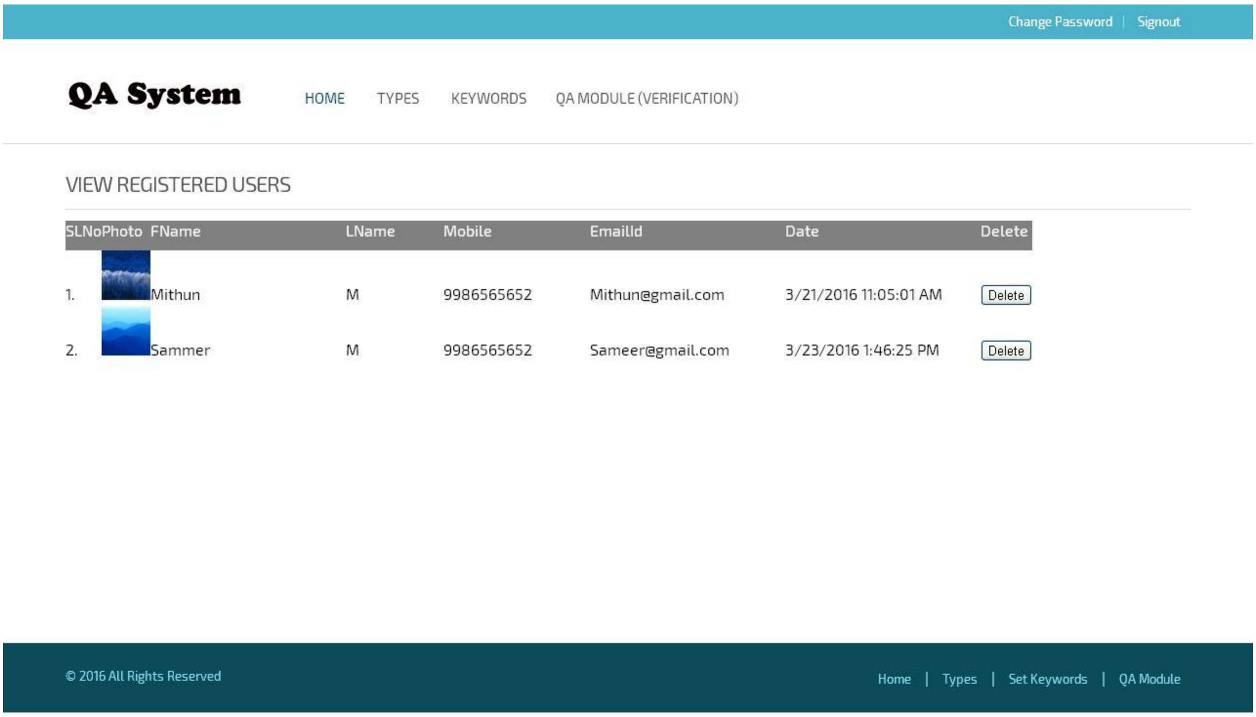
Select Question Type: All

Set Keyword

Existing Keywords

| SLNo | Question Type | Keyword |
|---|---|---|
| 1. | Education | student |
| 2. | Education | exam |
| 3. | Education | timetable |
| 4. | Education | book |
| 5. | Education | subject |
| 6. | Sports | win |
| 7. | Sports | loss |
| 8. | Sports | cricket |
| 9. | Sports | football |

**Fig A.7** Page that shows admin setting keywords


Change Password | Signout
QA System
HOME
TYPES
KEYWORDS
QA MODULE (VERIFICATION)
© 2016 All Rights Reserved
Home | Types | Set Keywords | QA Module

VIEW REGISTERED USERS

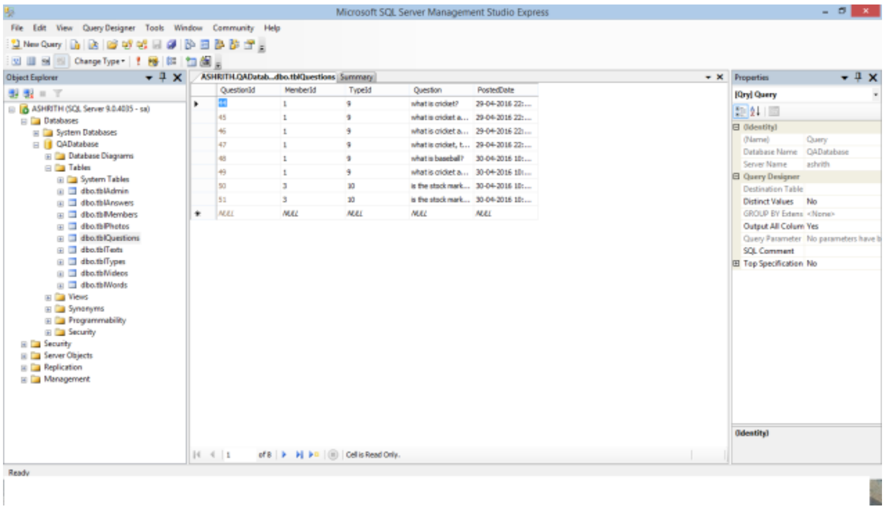
| SLNo | Photo | FName | LName | Mobile | EmailId | Date | Delete |
|---|---|---|---|---|---|---|---|
| 1. | | Mithun | M | 9986565652 | Mithun@gmail.com | 3/21/2016 11:05:01 AM | Delete |
| 2. | | Sammer | M | 9986565652 | Sameer@gmail.com | 3/23/2016 1:46:25 PM | Delete |

**Fig A.8** Page that shows details about the admins

# APPENDIX B

## Experimental Results

For the purpose of analyzing the working of the system, we have taken some sample experimental measures. C# and .NET have been used on a 64 bit windows system to create the system. Two members are created who post the questions. The question log of the users is shown in fig 5.

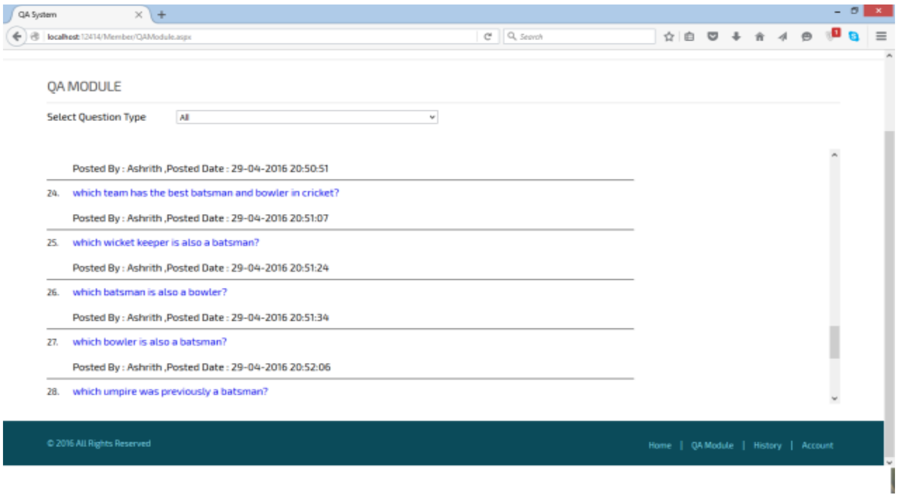


| QuestionId | MemberId | TypeId | Question | PostedDate |
|---|---|---|---|---|
| 44 | 1 | 9 | what is cricket? | 29-04-2016 22:... |
| 45 | 1 | 9 | what is cricket a... | 29-04-2016 22:... |
| 46 | 1 | 9 | what is cricket a... | 29-04-2016 22:... |
| 47 | 1 | 9 | what is cricket, t... | 29-04-2016 22:... |
| 48 | 1 | 9 | what is baseball? | 30-04-2016 10:... |
| 49 | 1 | 9 | what is cricket a... | 30-04-2016 10:... |
| 50 | 3 | 10 | is the stock mark... | 30-04-2016 10:... |
| 51 | 3 | 10 | is the stock mark... | 30-04-2016 10:... |
| NULL | NULL | NULL | NULL | NULL |

Cell is Read Only.

**Fig B.1** Questions posted by the users

Each question belongs to either the sports or the finance category. This question answer module can be viewed by any of the members. There is also a filter to view the questions related only to a required category. Fig 6 shows the QA module of the system



QA System
localhost:12414/Member/QAModule.aspx
Search
© 2016 All Rights Reserved
Home | QA Module | History | Account

QA MODULE

Select Question Type: All

Posted By : Ashrith ,Posted Date : 29-04-2016 20:50:51

24. which team has the best batsman and bowler in cricket?

Posted By : Ashrith ,Posted Date : 29-04-2016 20:51:07

25. which wicket keeper is also a batsman?

Posted By : Ashrith ,Posted Date : 29-04-2016 20:51:24

26. which batsman is also a bowler?

Posted By : Ashrith ,Posted Date : 29-04-2016 20:51:34

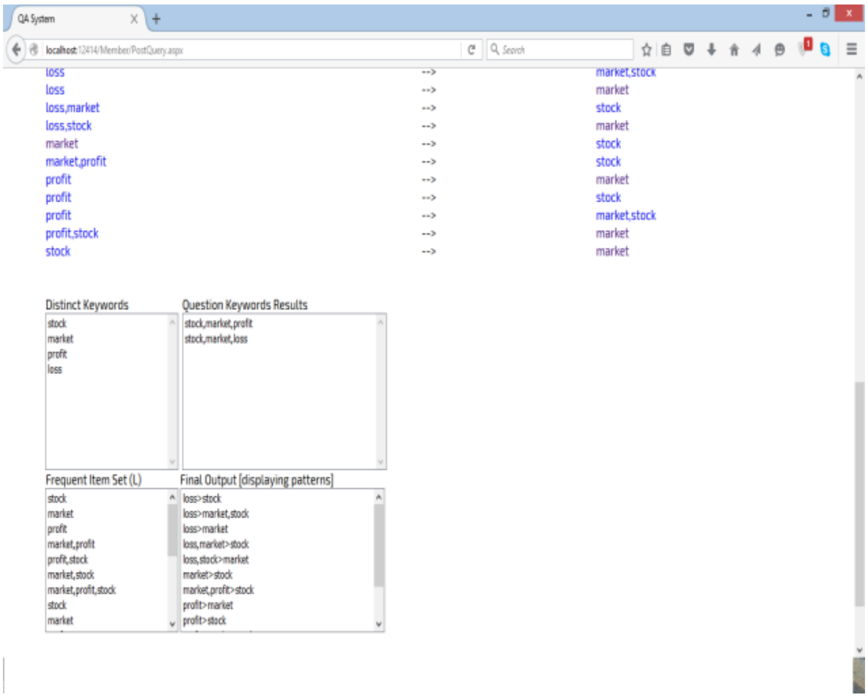
27. which bowler is also a batsman?

Posted By : Ashrith ,Posted Date : 29-04-2016 20:52:06

28. which umpire was previously a batsman?

**Fig. B.2** QA module of the system

For the purpose of prediction, we have taken the minimum support as 50% and the minimum confidence as 80%. As soon as a query is submitted, the keywords are identified, frequent itemset is generated and association rules are obtained from this frequent itemset. In our case, we have entered the query as “Which stock purchase would incur loss”. The previous question logs contain other queries such as “Is the stock market in loss?” or “Is the stock market in profit?”. The system analyzes these logs and brings about the associations required. The association rules generated for the query input by the user using the previous question logs is shown in fig 7.



```
loss              -->   market,stock
loss              -->   market
loss,market       -->   stock
loss,stock        -->   market
market            -->   stock
market,profit     -->   stock
profit            -->   market
profit            -->   stock
profit            -->   market,stock
profit,stock      -->   market
stock             -->   market
```

Distinct Keywords

```
stock
market
profit
loss
```

Question Keywords Results

```
stock,market,profit
stock,market,loss
```

Frequent Item Set (L)

```
stock
market
profit
market,profit
profit,stock
market,stock
market,profit,stock
stock
market
```

Final Output [displaying patterns]

```
loss>stock
loss>market,stock
loss>market
loss,market>stock
loss,stock>market
market>stock
market,profit>stock
profit>market
profit>stock
```

**Fig. B.3** The association rules generated after posting query

The experimental data show that the associations generated and the predicted questions are based on the user’s previous interactions with the system. If the user has no previous interactions, the interactions of other members with the system are considered. The accuracy of the predictions increases with a larger training set for the system. It can be seen that the proposed system is very interactive, efficient and smart.