**1. INTRODUCTION**

Our project is proposed to replace human by a machine which can think and can share information just like the humans does. With the fast growth of internet and web technologies humans can connect to each other and can exchange or share information. The sharing of information takes place either through textual methods, through speech or audio, or by having a video conference. Consider the textual method which is commonly called as chatting, where two humans share information. In industries it is not possible to keep a large range of employees for sharing information with the clients. So why not this human can be replaced by a machine which can think and can share information just like the humans does. Such a system called an Intelligent Response System can have a big impact in the future of information sharing. These chat bots can be embedded in web pages or can act as a standalone program. While doing so the visitors of the website can have information related to the website topic without the help of humans, whether it is a tourism website or any social website.

This project is proposed to replace human by a machine which can think and can share information. These chat bots can be embedded in web pages or can act as a standalone program. While doing so the visitors of the website can have information related to these website topic without the help of humans, whether it is a tourism website or any social website. It is one of the few developments of data mining technology over the decade. The Virtual Agent should have the capacity to chat like a human. Till now many intelligent response engine have been evolved, mainly ELIZA, verbots, A.L.I.C.E and ELLA. They all have are developed with the same thought, i.e. to chat like humans.

**2. PROBLEM DEFENITION**

With the fast growth of internet and web technologies humans can connect to each other and can exchange or share information The Antwort Konig is chatterbot, which should have the capacity to chat like a human. Till now many intelligent response engine have been evolved, mainly ELIZA, verbots, A.L.I.C.E and ELLA. They all have are developed with the same thought, i.e. to chat like humans. But the problem is that most of them either have learning problem or won’t have common sense to carry out a conversation. Another problem with these chat bots is they can be easily caught during a chat, since they might reply your queries with answers that are irrelevant to it but will be in the database. So a user can easily interpret whether it is a human or a chat bot. Most of them don’t have sessions or User-Password pairs. These should be implemented, so that, even after the completion of one session, next time, whenever the user enters his Username and Password, the chat bot will get all the relevant data and previous talks, related to the user, from the database itself.

**3. OBJECTIVES**

• Reduce the Human effort in various organizations

• The ease and convenience of access of the information by the clients can be ensured

• Efficient and significant time management is implemented

• Unauthorized persons can be restricted

We can implement this software in various fields such as

• Online tutoring sites

• Online live helps

• Call centre executives

• Customer support executives

• Customer relationship management

• Travel agencies

• Tourist guides

**4. SOFTWARE REQUIREMENT SPECIFICATION**

**4.1 PURPOSE OF THIS DOCUMENT**

This SRS describes the function and the performances allocated to our project. SRS provides an overview of the product including functional and non-functional requirements, abbreviations used, product and function etc.

**4.2 SCOPE OF THE DEVELOPMENT PROJECT**

The system is designed to focus on chatbot which is authorized and user friendly. Here we focused on learning process based on the user’s feedback instead of dictionary mode type, it can be implements through tokenizing the question at each sessions and the tokens are stored in the database that tokens are used for next time.

**4.3 FUNCTIONAL REQUIREMENTS**

Like any system the proposed system also require a user and an admin. The USER is the one who can login to the website. He can search for the question he has and can get the answer. He can get the answer later to his mail id if the bot can’t answer at the particular time. The events to be done by a client is given below:

1. Fill up the Registration Form if the client is entering the system for first time sign in.

2. Otherwise enter in to the Login page and sign in by using the user’s user id, password.

3. If the sign in is successful client can enter into the chatbot page .there the request of client can be typed as text and then response will get displayed.

4. Log out.

The ADMIN is the person who can accept the questions that are not answered by the chatbot and give response back to the customer. He is the person who teaches the question and answer for the chatbot which later is answered by the chatbot. It gives the opportunity to get answer like the lively man. These are done by the help of technology called Intelligent response system. It provides more easy and convenient way of search of knowledge. The events to be done by the admin are given below:

1. The administrator enter directly into the login page since they have own user id and password.

2. A Server page will be displayed, then status is turn on.

3. When the Robo fails to answer, that question is passed to the administrator and he send reply to that question.

4. The administrator can teach the Robo, teaching option is available in the server page.

5. There is a profile icon, by clicking in it the server can view the client’s profile.

6. When the client login for the second time the history of the past conversations can be available.

7. The status will go to off state when the administrator

8. Logout

**5. SYSTEM STUDY**

**5.1 EXISTING SYSTEM**

This project analyses and compares the data gathered from two previously conducted artificial linguistic Internet chat entity chatterbot studies that were focused on response accuracy and user satisfaction measures for six chatterbots. These chatter bots were further loaded with varying degrees of conversational, telecommunications, and terrorism knowledge. From our prior experiments using 347 participants, we obtained 33 446 human or chatterbot interactions. It was found that asking the Alice chatter bots questions resulted in higher Response satisfaction levels, as compared to other interrogative-style inputs because of their acceptability to vague, binary, chatterbot responses. We also found a relationship between the length of a query and the users’ perceived satisfaction of the chatbot response, where shorter queries led to more satisfying responses.

DISADVANTAGES

• Stand alone application

• Not ease and convenience of access

• Unauthorized access

• Quick response is not ensure

**5.2 PROPOSED SYSTEM**

The system is designed to focus on a chatbot which is authorized and user friendly .Here we focused on learning process based on the users feedback instead of dictionary mode type, it can be implements through tokenizing the question at each sessions and the tokens are stored in the database that tokens are used for next time. Also we are using registration methods only authorized user can access the chat bot, through this we can able increase privacy for our chatbot. The clients are entered to the chatbot through their username and password, so that previous sessions are available The chatbot is packed with a online assistants connected to server, he can able to ON/OFF his status, while the status is ON the untaught question to the chatbot is passed to him, that administrator can able to give replay to the client. This helps the firms to handle the customers more efficiently. It elevates a feeling in the user that he is interacting with a human rather than a mere machine. Workers employed for the purpose of usual customer care service can be reduced to a large extent with the advent of chatterBOt. It handles all sorts of users in an equivalent basis irrespective of their diversifying nature. An all-time available customer service is ensured. The learning capability of chatterBOt can be exploited by the administrator in saving a lot of time which is otherwise spent for the training of employees**.**

The proposed system has six modules:

 Graphical User Interface module

 Parser module

 Concept map module

 Data Mining module

 Knowledge base module.

**5.3 FEASIBILITY STUDY**

A feasibility study precedes the decision to begin a project. It is used to determine if the project should get the go-ahead. The proposed system is tested thoroughly in many ways for its feasibility by conducting the feasibility studies.

**5.3.1 Economical feasibility**

The resources required for the system is comparatively less. The resources that may be needed for the further developments are also assessed. In terms of the labour and the manpower, the present strength is sufficient and no extra personal to be required. So the present status of the company is sufficient enough for the project to be undertaken**.**

**5.3.2 Operational feasibility**

It has been found that the proposed system is much better than the existing system where a person need to be trained to handle the user requests. There is no barrier for implementing the proposed system, which offers greater level of convenience to users, and it is operationally feasible.

**5.3.3 Technical *feasibility***

The hardware and the softwares requirements of this propose system is available everywhere. Since the software is platform independent, it can be implemented anywhere and thus making it widespread. Thus the system is technically feasible. Service can be provided to anyone who logins in the correct manner.

 **Fast:** since the system is fast all actions can be performed with less time which is more convenient and effective for both the company and the customers.

 **Reduce errors:** since all actions are performed by the system side there is no chance getting error while updating the data.

 **Time consuming is less:** It less time consuming system hence it is very convenient.

 **User friendly:** It is user friendly. Hence can be used very effectively by the user.

 **Powerful:-** It is a power full mechanism since it is error free, less time consumption, and can stand independently.

 **Flexible: -**Flexible to use. It is modifiable.

 **Planned approach towards working: -** The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

 **Accuracy: -** The levels of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.

**6. SYSTEM DESIGN**

**6.1 MODULE DESCRIPTION**

The modules that are in the project are:

1. Graphical User Interface module

2. Parser module

3. Concept map module

4. Data retrieval module

5. Knowledge base module

6. Learning module

**6.1.1 GUI module**

The GUI module will be designed in Java programming language. The GUI is the only module that the user will interact with, when the program executes. Once the user enters in a username, he/she can begin the conversation. The graphical user interface is the module which allows the user to interact with the system. No other modules can connect for client.

Input: Once the user types a message the message is sent from the GUI as a character string to the Parser module.

Output: The GUI receives sentence structure back from the Data Minig module and displays it to the screen. Also the output is sent to the voice synthesizer that will produce voice output.

**6.1.2 Parser Module**

The parser module will be designed in Java programming language. The parser module separates the user sentence into separate parts of speech. The Parts of speech are stored within a tree structure with links generated between related words. In parser module we use an algorithm known as matrix algorithm. In matrix algorithm we use the maximum number of combination. From the maximum number of combinations the parser module can choose the correct combination of word that satisfies the users query.

In parser module we have three types combination set such as fact, direct question, wishes, and indirect questions. The parser then will check among the packages and determine the type of questions.

Let’s look into an example;

A user logged into the system. Then he makes his queries. By the help of the module called the graphical user interface the module get connected to the parser module. Hence the parser module gets the query of the user say “who is Abdul kalam?” this question by the user match for maximum number of combinations in this module.

Let the combinations be “who Abdul kalam”,”Abdul kalam who?” Since there is only one combination we get the answer easily. In parser module the parser does the combination by removing the auxiliary verbs and looks the combination among words. From the maximum number of combinations the matrix algorithm will chose the answer satisfiable for the query of the user.

Input: The parser receives a character string from the GUI.

Output: The parser outputs the data of interconnected words, in tree form, for integration within the concept map.

**6.1.3 Concept map module**

The concept map module will be designed in Java programming language. The concept map module stores all the data that is extracted from the user’s input by the parser. The concept map develops the parser tree structure into a network topology. The concept map stores the links between the parsed words and also stores all nouns, verbs, and adjectives within super-lists for searching.

The parser tree is just like a tree where all the data of the user such as parsed words, nouns, verbs, etc. anything that is entered by the user. These are from the module called parser module. Then this parser tree is converted to a network topology in order to make the data’s available to the next module say data mining module just like the internet does.

As the word implies the details in this module are concepts which are used for the further evaluation of the system output. If the input is only stored in this module, then correct knowledge can be applied to each module for the better and efficient output of the system.

Hence the robot can speak efficiently making the client to feel that he is a man rather than a robot.

Input: The input is the tree structure from the parser.

Output: The network is sent to the DM module and analysed so a conceptual output can be produced.

**6.1.4 Data retrieval module**

This module will be designed in Java programming language. The Data retrievalmodule is the main subsystem that interprets all of the data that has been generated so far in the program execution. The module interprets the words stored in the concept map to determine what the proper output should be. If the Data Mining module cannot determine a meaning from the data then a default message will be displayed in the GUI.

By this module, the user gets all the information’s what he wants. Data which is stored will be not as much as usable for the user. The particular information available is extracted for that particular user. Each user will expect different answers that is logically understandable by the robo and retrieve the results to the user.

Input: The network structure developed in the concept map is the input to the Data Mining module.

Output: Once the Data Mining module derives a sentence from the data it outputs that sentence to a character string. The string is sent to the GUI, which displays it, for the user to read.

**6.1.5 Knowledge base module**

A knowledge base will be designed in Java programming Language. It is a special kind of database for knowledge management. It provides the means for the computerized collection, organization, and retrieval of knowledge.

With the fast growth of internet and web technologies humans can connect to each other and can exchange/share information. The sharing of information takes place either through textual methods, through speech or audio, or by having to a video conference. Consider the textual method which is commonly called as chatting, where two humans share information. In industries it is not possible to keep a large range of employees for sharing information with the clients. So why not this human can be replaced by a machine which can think and can share information just like the humans does. Such a system called Intelligent Response System can have a big impact in the future of information sharing. These chat bots can be embedded in web pages or can act as a standalone program. While doing so the visitors of the website can have information related to the website of a topic without the help of humans, whether it is a tourism website or any social website. The Antwort Konig should have the capacity to chat like a human. Till now many intelligent response engine have been evolved, mainly ELIZA, verbots, A.L.I.C.E and ELLA. They all have are developed with the same thought, i.e. to chat like humans. But the problem is that most of them either have learning problem or won’t have common sense to carry out a conversation. Another problem with these chat bots is they can be easily caught during a chat, since they might reply your queries with answers that are irrelevant to it but will be in the database. So a user can easily interpret whether it is a human or a chat bot. Most of them don’t have sessions or User-Password pairs. These should be implemented, so that, even after the completion of one session, next time, whenever the user enters his Username and Password, the chat bot will get all the relevant data and previous talks, related to the user, from the database itself.

We can implement this software in various fields such as

•Online tutoring sites

•Online live helps

•Call centre executives

•Customer support executives

•Customer relationship management

•Travel agencies

•Tourist guides

Input: The Knowledge base module receives input from the Data Mining module which includes a collection of information about a particular topic and is used for organization and retrieval of knowledge.

Output: It generates information as per the request from the DM module.

**6.1.6 Learning module**

The main function of this module is to enhance the Learning capabilities of the system and also to expand knowledge base. This module is the core part of the Antwort Konig, without learning capabilities an Intelligent Response system will be a simple program which will produce output which will have no relation to the previous conversations.

Let’s look into an example;

A user logged into the system. Then he makes his queries. By the help of the module called graphical user interface the module get connected to the parser module. Hence the parser module gets the query of the user say “who’s Abdul kalam?” this question by the user match for maximum number of combinations in this module.

Let the combinations be “who Abdul kalam”,”Abdul kalam who?” Since there is only one combination we get the answer easily. In parser module the parser does the combination by removing the auxiliary verbs and looks the combination among words. From the maximum number of combinations the matrix algorithm will chose the answer satisfiable for the query of the user. Then it is moved to the concept map module. From the module treelike structure obtained is moved along network for making it accessible for the data mining module.

From the data mining module it’s moved to knowledge if data is not available in the previous module. Then comes the module learning, which studies about each combinations. It checks the needed combination and pass over the network to the user through previous module.

Input: It collects conversations from the DM module and will use these conversations.

Output: It will pass the information which is learned to the Knowledge base module, so that the learned knowledge can be organized and retrieved in future.

**6.2 DATA FLOW DIAGRAM**

**Context level DFD**

User

User

Request

Response

**Level 1 DFD**

Response

User

User

Admin

Username password

URL

Username password

Response

**Level 2 DFD for Process 2.0**

Questions

Unanswered

User table

Admin

Admin

Unanswered questions

Request

Unanswered questions

Request

User details

User details

Admin details

Username password

Request

Admin details

Message

Request

**Level 2 DFD for Process 3.0**

Users

Question

User

User

Question

Answer

Username password

Question

Request

Username password

Answer

User details

User details

User details

# 7. USE CASE DIAGRAM

Use case is a set of scenarios that describing an interaction between a user and a system. The two main component of use case diagram are use cases and actors. They are helpful in exposing requirements and planning the projects. During the initial stage of project, most use cases should be defined. But as the project continues more might become visible, the use case diagram for showing the project is that it measures the size of the software. It reduces users need for training and operating the system.

In use case diagram, we all know, the main two things are the actors and the actions. Here the actors is the user. Apart from the actors there are actions to be performed by these actors. These actions are shown in oval.

Use case diagrams are relatively easy UML diagram. They are helpful in exposing requirements and planning of the project. During the initial stage of a project most use cases should be defined, but as the project continues more might become visible. A use case is an external view of the system that represents some action the user might perform in order to complete a task. A use case diagram displays the relationship among actors and use cases.

**8. DATABASE DESIGN**

**USERS**

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Constraints |
| Sid | Int | Primary key |
| Firstname | Varchar | Not Null |
| Lastname | Varchar | Not Null |
| Type | Varchar | Not Null |
| Phone | Varchar | Not Null |
| Email | Varchar | Not Null |
| Pwd | Varchar | Not Null |
| Uname | Varchar | Not Null |

**ANSWERD QUESTION ANSWER**

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Constraints |
| questAnsID | Int | Not Null |
| question | varchar | Not Null |
| answer | varchar | Not Null |

**UNANSWERED QUESTIONS**

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Constraints |
| qstID | Int | Not Null |
| Question | Varchar | Not Null |
| Username | Varchar | Not Null |

**WHAT**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whatID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| could | Varchar | null |

**WHO**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whoID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| could | Varchar | null |

**WHEN**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whenID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| could | Varchar | null |

**WHICH**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whatID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| Could | Varchar | null |

**WHERE**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whatID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| Could | Varchar | null |

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whomeID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| Could | Varchar | null |

**WHOME**

**WHOSE**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whoseID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| could | Varchar | null |

**WHY**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| whatID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| could | Varchar | null |

**HOW**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| howID | Int | Not null |
| Full combination | Varchar | Not null |
| Subject | Varchar | Not null |
| Is | Varchar | null |
| Are | Varchar | null |
| Was | Varchar | null |
| Were | Varchar | null |
| Has | Varchar | null |
| Have | Varchar | null |
| Had | Varchar | null |
| Do | Varchar | null |
| Does | Varchar | null |
| Did | Varchar | null |
| Will | Varchar | null |
| Would | Varchar | null |
| Shall | Varchar | null |
| could | Varchar | null |

**DIRECTQUESTIONS**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| Id | int | Not null |
| Combination | Varchar | Not null |
| Term | Varchar | Null |

**YESNO**

|  |  |  |
| --- | --- | --- |
| Field name | Type | Constraints |
| Id | Int | Not null |
| Combination | Varchar | Not null |
| Term | Varchar | null |

**9. INPUT/OUTPUT DESIGN**

Physical design produces working system by defining specification and the necessary programs are written based on this, which performs necessary calculations through the existing database produce report and maintains updated database at all times.

Physical design consists of:

• Input Design

• Output Design

**9.1** **INPUT DESIGN**

Input design is the process of converting user-originated inputs to a computer based format. The goal of designing input data is to make data entry as easy and free from errors as possible.

Well-designed inputs serve five purposes

1 Control work flow

2 Reduce redundancies

3 Record data

4 Increase clerical accuracy

5 Allow easier checking of data

Inputs are important because in many instances, they are the contact a user has with a system. When the data keyed into the system, the operator must receive the data in a form that are readily understood. The key operator should key the data in order in which it occurs on the form and the computer should reform it when required.

Various inputs used in this system are mainly the primary information of the website. The primary information includes Mail id, Username, Password, Site URL, Title, Description, Page start value, Visitor start value. Website administrator provides the primary information. Mail id is our primary key. For searching a record, we use mainly Mail id. Although he entered a single field, the search will be successful and he will get the corresponding record.

**9.2** **OUTPUT DESIGN**

Outputs from the computer system are required primarily to communicate the results of processing to the users. Output design is an ongoing activity during the study phase. The objective of output design is to define the contents and format of all documents and reports in an attractive and useful format. Other reasons for output generations are: To provide proper communication of data to the user.

1 To re-input to the computer for being connected with other data and further processing.

2 To provide permanent storage

Output generally refers to the results and information that are generated by the system. It can be in the form of operational documents a report. Since some of the end users will not actually operate the information system or input through workstations. Our proposed system provides various outputs. If the input entered is successful the corresponding record will be displayed as output. When the user clicks on the button, then corresponding action will be carried out. The corresponding action includes submission, removal and searching of the record.

**10. SYSTEM IMPLEMENTATION**

**10.1 HARDWARE REQUIREMENT**

Processor : 1.2 GHz

Hard Disk : 10 GB or Above

Monitor : 15’’

Keyboard : USB/Wireless

Mouse : Any

**10.2** **SOFTWARE REQUIREMENT**

Browser : Mozilla Firefox/Internet Explorer

Web Server : Apache Tomcat

Tools : Netbeans

Front End : J2EE, AJAX

Back End : MySQL 5.0 Server

Operating System : Ubuntu

**10.3 SELECTION OF SOFTWARE**

**10.3.1 JAVA LANGUAGE**

Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun's Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code which can run on any Java virtual machine (JVM) regardless of computer architecture. As of May 2007, in compliance with the specifications of the Java Community Process, Sun made available most of their Java technologies as free software under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Class path.

Java's design, industry backing and portability have made Java one of the fastest-growing and most widely used programming languages in the modern computing industry. The Java language was created by James Gosling in June 1991 for use in a set top box project. The language was initially called Oak, after an oak tree that stood outside Gosling's office and also went by the name Green and ended up later being renamed to Java, from a list of random words. It promised "Write Once, Run anywhere" (WORA), providing no-cost run times on popular platforms. It was fairly secure and its security was configurable, allowing network and file access to be restricted. Major web browsers soon incorporated the ability to run secure Java applets within web pages. With the advent of Java 2, new versions had multiple configurations built for different types of platforms. For example, J2EE was for enterprise applications and the greatly stripped down version J2ME was for mobile applications. J2SE was the designation for the Standard Edition. In 2006, for marketing purposes, new J2 versions were renamed Java EE, Java ME, and Java SE, respectively.

Java technology is both a programming language and a platform. The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the Java compiler (javac). A .class file does not contain code that is native to your processor; it instead contains byte codes-- the machine language of the Java Virtual Machine. The Java launcher tool (java) then runs your application with an instance of the Java Virtual Machine. Because the Java Virtual Machine is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris TM Operating System (Solaris OS), Linux, or MacOS. Some virtual machines, such as the Java HotSpot Virtual Machine, perform additional steps at runtime to give your application a performance boost. This includes various tasks such as finding performance bottlenecks and recompiling (to native code) frequently-used sections of your code.

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms. The Java platform has two components:

 The Java Virtual Machine

 The Java Application Programming Interface (API).

Class

A class defines the shape and behaviour of an object and it is a template for multiple objects with similar features. The instances of the class can be created by the new operator is called objects or instances. By creating classes that uses classes provided by java, highly maintainable program can be constructed.

Packages

Java provides a large number of classes grouped into different packages according to functionality. Java System package includes

Java.Lang: Language support classes. They include classes for primitive types, strings, math functions, threads and exception.

 Java.util: utility classes such as vectors, hash tables, date, random numbers etc.

 Java.io: input/output support classes.

 Java.awt:set of classes for implementing graphical user interface.

 Java.net: classes for networking.

 Javax.swing: providing a graphical user interface (GUI) .

Java isa high level programming language that is cross-platform, object oriented, and can be run in a browser, on a server, or as a stand-alone application. It was created by Sun Microsystems and first released in 1995.

Features of JAVA:

* Java programs can run in a browser**.**  With the help of the Java Plug-In, most users can run complicated programs from their home, without having to buy any software.  The programs can also be accessed in the middle of a lecture provided the lecturer has a computer with web access.
* Java is cross-platform**:** As long as the operating system in question has a current java runtime environment (jre), that OS can run the Java program.  Java programs can be run in Windows, Mac, Linux, Unix, and Solaris.
* Java is Object Oriented**.**  Object oriented languages allow the programmer to build a program using blocks of code which are called objects.  Objects encapsulate functions and data.  By "encapsulate" I mean that a programmer does not need to know the exact mechanics of the object in order to program with them, but instead gets all required data from built in behaviours of that object.  It is then easy for even the most novice programmer to create a complicated program simply by using built in behaviours

**10.3.2 MYSQL SERVER**

MySQL is the world's second most widely used open-source [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). It is named after co-founder [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, My. The [SQL](http://en.wikipedia.org/wiki/SQL) phrase stands for [Structured Query Language](http://en.wikipedia.org/wiki/Structured_Query_Language). The MySQL development project has made its [source code](http://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of[proprietary](http://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-profit](http://en.wikipedia.org/wiki/Business) firm, the [Swedish](http://en.wikipedia.org/wiki/Sweden) company [MySQL AB](http://en.wikipedia.org/wiki/MySQL_AB), now owned by [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation).

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_(software_bundle)) open source web application software stack (and other ['AMP'](http://en.wikipedia.org/wiki/List_of_AMP_packages) stacks). LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux),[Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](http://en.wikipedia.org/wiki/Perl)/[PHP](http://en.wikipedia.org/wiki/PHP)/[Python](http://en.wikipedia.org/wiki/Python_(programming_language))." [Free-software](http://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL.

MySQL is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), and ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](http://en.wikipedia.org/wiki/Command_line) tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, [MySQL Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) is actively developed by Oracle, and is freely available for use.

Features of MySQL

* A broad subset of [ANSI SQL 99](http://en.wikipedia.org/wiki/SQL:1999), as well as extensions
* Cross-platform support
* [Stored procedures](http://en.wikipedia.org/wiki/Stored_procedure), using a procedural language that closely adheres to [SQL/PSM](http://en.wikipedia.org/wiki/SQL/PSM)[[36]](http://en.wikipedia.org/wiki/MySQL#cite_note-HarrisonFeuerstein2008-37)
* [Triggers](http://en.wikipedia.org/wiki/Database_trigger) and [Cursors](http://en.wikipedia.org/wiki/Cursor_(databases))
* Updatable [views](http://en.wikipedia.org/wiki/View_(SQL))
* [Information schema](http://en.wikipedia.org/wiki/Information_schema)
* Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)
* [X/Open XA](http://en.wikipedia.org/wiki/X/Open_XA) [distributed transaction processing](http://en.wikipedia.org/wiki/Distributed_transaction_processing) (DTP) support; [two phase commit](http://en.wikipedia.org/wiki/Two-phase-commit_protocol) as part of this, using Oracle's [InnoDB](http://en.wikipedia.org/wiki/InnoDB) engine
* Independent [storage engines](http://en.wikipedia.org/wiki/Storage_engine) ([MyISAM](http://en.wikipedia.org/wiki/MyISAM) for read speed, InnoDB for transactions and [referential integrity](http://en.wikipedia.org/wiki/Referential_integrity), [MySQL Archive](http://en.wikipedia.org/wiki/MySQL_Archive) for storing historical data in little space)
* Transactions with the InnoDB and [NDB Cluster](http://en.wikipedia.org/wiki/NDB_Cluster) storage engines; [savepoints](http://en.wikipedia.org/wiki/Savepoint) with InnoDB
* [SSL](http://en.wikipedia.org/wiki/Secure_Sockets_Layer) support
* Query [caching](http://en.wikipedia.org/wiki/Cache_(computing))
* Sub-[SELECTs](http://en.wikipedia.org/wiki/Select_(SQL)) (i.e. nested SELECTs)
* [Replication](http://en.wikipedia.org/wiki/Database_replication) support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master.[[37]](http://en.wikipedia.org/wiki/MySQL#cite_note-38) [Multi-master replication](http://en.wikipedia.org/wiki/Multi-master_replication) is provided in [MySQL Cluster](http://en.wikipedia.org/wiki/MySQL_Cluster)  and multi-master support can be added to unclustered configurations using Galera Cluster.
* Full-text [indexing](http://en.wikipedia.org/wiki/Index_(database)) and searching.

**10.3.3 NETBEANS**

NetBeans refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Groovy, C, C++and others. The NetBeans IDE 7.0 no longer supports Ruby and Ruby on Rails. The NetBeans IDE is written in Java and can run anywhere a JVM (Java Virtual Machine) is installed, including Windows, Mac OS, Linux, and Solaris. A JDK (Java Development Kit) is required for Java development functionality, but is not required for development in other programming languages. In our project we used JDK 1.7.The NetBeans platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans platform (including the NetBeans IDE) can be extended by third party developers. The NetBeans Platform is a reusable framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

NetBeans manages the following platform features and components:

* User settings
* Windows (placement, appearance, etc.)
* NetBeans Visual Library
* Storage
* Integrated development tools
* Framework wizard

NetBeans uses components, also known as modules, to enable software development. NetBeans dynamically installs modules and allows users to download updated features and digitally-authenticated-upgrades.

NetBeans IDE modules include NetBeans Profiler, a Graphical User Interface (GUI) design tool, and-NetBeans framework reusability simplifies Java Swing desktop application development, which provide platform extension capabilities to third-party developers.

**11. SOURCE CODE**

**Code for preparing Answer**

package vsrd.chatbot.datamining;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Random;

import java.util.Set;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.mail.Session;

import javax.servlet.http.HttpSession;

//import javax.swing.JOptionPane;

import vsrd.chatbot.datamining.online.OnlineAnswer;

import vsrd.chatbot.dbase.DatabaseCalls;

import vsrd.chatbot.grammar.ClarityImprovements;

import vsrd.chatbot.grammar.SentanceManupulations;

import vsrd.chatbot.grammar.SentanceTypes;

public class PrepareAnswers {

private String answer;

private List<String> askedQuestions=new ArrayList<String>();

public String getAnswer(String question, String userName, HttpSession session) throws SQLException {

if(session.getAttribute("askedQuestions")!=null) {

askedQuestions=(List<String>) session.getAttribute("askedQuestions");

}

try{

int ii = new db.DBConnect().update("update sound set flag='1' where id='0'");

}catch(Exception e){

System.out.println(e);

}

String originalQuestion = question;

String[] dontknow = {"thtz all", "It doesnt matter for you", "thtz all about it", "Sorry ,,I dont know", "I wont say", "hmm", "hmm", "hhh ", "god knows", "you dont have to worry abt that", "No more question about that", "I dont say more", "donno", "i dont have full details about that", "You got have to think about that", "No buddy..you dont need to know that", "hmm", "hmm hah", "Ah..I dont rem the exact answer", "ahh", "ahh", "better to try urself", "hmmm", "pmm"};

String[] instantReplay = {"I cant say.", "hmmm i wont say..", "hmm..Why do you ask so ?", "Wait, I got a phone call.", "Oh my keyboard hangs again", "You dont need to know that", "ahh", "ok", "Dont xpt answer for evry questions....", "Tht answer is beyond my ability", " I knew that", "Itz not a matter for you", "really ?", "I cant express that ?", "I dont understand", "I didnt get you", "I dont know", "are you crazy", "I am feeling too headache", "words can speek but letters not", "I am not sure about that", "may be", "??????"};

String[] good = {"itz k..", "Sorry.. itz my duty to help you!!", "oh..fine", "yah...", "No mention..", "k..", "k gud.", "thank you!!", "am sooo happy", "am happy to help you", "any more doubts!!"};

String[] un = {"hm", "yup...", "yaeh", "k..!!", "aahhh.."};

String[] repeatedQuestions = {"why you are asking that question again", "I have already said about that", "I think you are not listening what i am saying..I have told about that", "are you blaming me.. i had told you early about this", "oh i think you are testing me..", "Excuse me.. are you there.. i think you are playing, not listening what i'm saying", "i had answered for this..", "are you playing with me.."};

String[] replay = {"sorry..i can't get you", "I didn't get you", "I don't get you, ask me clearly"};

String[] noanswer = {"dont worry about that I will mail you as soon as possible", "I will get bact to you by mail", "I'm not able to provide you that information right now, I will get back to you by mail",

"sorry, I will inform you later by mail"};

Wishes wishes = new Wishes();

ClarityImprovements ci = new ClarityImprovements();

SentanceManupulations sm = new SentanceManupulations();

OnlineAnswer onlineAnswer = new OnlineAnswer();

question = ci.getCorrectedSentance(question);

SentanceTypes sentType = new SentanceTypes();

sentType.nlp.setAnalyzedOutput(question);

MineAnswers mineAns = new MineAnswers();

if (wishes.isItaWish(question)) {

answer = wishes.getReplayWish(question);

} // else if (question.split(" ").length == 100) {

// answer = singleWord(question);

// }

else if (askedQuestions.contains(question)) {

int nextInt = new Random().nextInt(repeatedQuestions.length);

answer = repeatedQuestions[nextInt];

} else if (sentType.isYesOrNo()) {

answer = mineAns.getYesOrNoAnswer(question);

if (answer == null) {

// session.setAttribute(answer, un);

answer = onlineAnswer.getAnswer(originalQuestion);

if (answer.isEmpty()) {

int nextInt = new Random().nextInt(noanswer.length);

answer = noanswer[nextInt];

storeUnansweredQuestion(question, userName);

}

}

} else if (sentType.isDirectQuestion()) {

answer = mineAns.getDirectAnswer(question);

if (answer == null) {

answer = onlineAnswer.getAnswer(originalQuestion);

if (answer.isEmpty()) {

int nextInt = new Random().nextInt(noanswer.length);

answer = noanswer[nextInt];

storeUnansweredQuestion(question, userName);

}

}

} else {

if (question.equalsIgnoreCase("yes")) {

answer = "ok";

} else {

String convertedToQuestion = sm.convertToQuestion(question);

String directAnswer = mineAns.getDirectAnswer(convertedToQuestion);

if (directAnswer == null) {

answer = onlineAnswer.getAnswer(originalQuestion);

if (answer.isEmpty()) {

int nextInt = new Random().nextInt(replay.length);

answer = replay[nextInt];

storeUnansweredQuestion(question, userName);

}

} else {

answer = directAnswer;

}

}

}

askedQuestions.add(question);

session.setAttribute("askedQuestions", askedQuestions);

return answer;

}

private String singleWord(String word) {

String[] dontknow = {"thtz all", "It doesnt matter for you", "thtz all about it", "Sorry ,,I dont know", "I wont say", "hmm", "hmm", "hhh ", "god knows", "you dont have to worry abt that", "No more question about that", "I dont say more", "donno", "i dont have full details about that", "You got have to think about that", "No buddy..you dont need to know that", "hmm", "hmm hah", "Ah..I dont rem the exact answer", "ahh", "ahh", "better to try urself", "hmmm", "pmm"};

String[] good = {"itz k..", "Sorry.. itz my duty to help you!!", "oh..fine", "yah...", "No mention..", "k..", "k gud.", "thank you!!", "am sooo happy", "am happy to help you", "any more doubts!!"};

String[] un = {"hm", "yup...", "yaeh", "k..!!", "aahhh.."};

if ((word.equals("why")) || (word.equals("what")) || (word.equals("who"))) {

int nextInt = new Random().nextInt(dontknow.length);

return dontknow[nextInt];

} else if ((word.equals("good")) || (word.equals("gud")) || (word.equals("fine"))) {

int nextInt = new Random().nextInt(good.length);

return good[nextInt];

} else if (word.startsWith("hm")) {

int nextInt = new Random().nextInt(un.length);

return un[nextInt];

} else {

return "what???";

}

}

private void storeUnansweredQuestion(String question, String userName) {

DatabaseCalls dbc = new DatabaseCalls();

String query1 = "select \* from UnansweredQuestions where question='" + question + "'";

String query2 = "insert into UnansweredQuestions (question, userName) values('" + question + "', '" + userName + "')";

dbc.openConnection();

ResultSet result = dbc.getResult(query1);

try {

if (!result.next()) {

dbc.getUpdated(query2);

}

} catch (SQLException ex) {

Logger.getLogger(PrepareAnswers.class.getName()).log(Level.SEVERE, null, ex);

}

dbc.closeConnection();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

PrepareAnswers preAns = new PrepareAnswers();

// String answer1 = preAns.getAnswer("are you fine", "admin");

// System.out.println(answer1);

// TODO code application logic here

}

}

**Code for clarity improvements**

package vsrd.chatbot.grammar;

import java.util.HashSet;

import java.util.Set;

public class ClarityImprovements {

private String correctedSentance;

public ClarityImprovements() {

}

public ClarityImprovements(String correctedSentance) {

this.correctedSentance = correctedSentance;

}

public String getCorrectedSentance(String inputSentance) {

inputSentance = convertToSingleSpace(inputSentance);

Set where = new HashSet();

where.add("where");

where.add("wher");

where.add("whr");

where.add("whre");

Set what = new HashSet();

what.add("what");

what.add("wat");

what.add("wt");

what.add("wht");

Set when = new HashSet();

when.add("when");

when.add("whn");

when.add("wen");

Set who = new HashSet();

who.add("who");

who.add("whu");

Set how = new HashSet();

how.add("how");

how.add("hw");

Set which = new HashSet();

which.add("which");

which.add("whch");

which.add("wch");

which.add("wich");

Set is = new HashSet();

is.add("is");

is.add("s");

is.add("z");

Set are = new HashSet();

are.add("are");

are.add("ar");

are.add("r");

Set was = new HashSet();

was.add("was");

was.add("ws");

was.add("waz");

Set were = new HashSet();

were.add("were");

were.add("wer");

were.add("wr");

Set your = new HashSet();

your.add("your");

your.add("ur");

your.add("yur");

your.add("yor");

your.add("yo");

your.add("yr");

Set whats = new HashSet();

whats.add("what's");

whats.add("whatz");

whats.add("whats");

whats.add("wat's");

whats.add("wats");

whats.add("wots");

whats.add("wot's");

whats.add("watz");

whats.add("wotz");

Set wheres = new HashSet();

wheres.add("wheres");

wheres.add("where's");

wheres.add("whers");

wheres.add("whrs");

wheres.add("whres");

wheres.add("wherez");

wheres.add("wherz");

wheres.add("whrz");

wheres.add("whrez");

wheres.add("where's");

wheres.add("wher's");

wheres.add("whr's");

wheres.add("whre's");

wheres.add("where'z");

wheres.add("wher'z");

wheres.add("whr'z");

wheres.add("whre'z");

Set whens = new HashSet();

whens.add("whens");

whens.add("whenz");

whens.add("when's");

whens.add("when'z");

whens.add("whnz");

whens.add("wenz");

whens.add("wens");

whens.add("wen's");

whens.add("wen'z");

Set hows = new HashSet();

hows.add("how's");

hows.add("hows");

hows.add("howz");

hows.add("how'z");

hows.add("hws");

hows.add("hwz");

Set whichs = new HashSet();

whichs.add("whichs");

whichs.add("whichz");

whichs.add("whchz");

whichs.add("whchs");

whichs.add("wichz");

whichs.add("wichs");

whichs.add("wchz");

whichs.add("wchz");

Set whos = new HashSet();

whos.add("whoz");

whos.add("who's");

whos.add("whos");

whos.add("whos");

whos.add("whz");

whos.add("whs");

Set whose = new HashSet();

whose.add("whoz");

whose.add("whos");

whose.add("whz");

whose.add("whs");

Set you = new HashSet();

you.add("u");

you.add("yu");

for (int i = 0; i < ary.length; i++) {

if (where.contains(ary[i].toLowerCase())) {

ary[i] = "where";

} else if (what.contains(ary[i].toLowerCase())) {

ary[i] = "what";

} else if (when.contains(ary[i].toLowerCase())) {

ary[i] = "when";

} else if (who.contains(ary[i].toLowerCase())) {

ary[i] = "who";

} else if (how.contains(ary[i].toLowerCase())) {

ary[i] = "how";

} else if (which.contains(ary[i].toLowerCase())) {

ary[i] = "which";

} else if (is.contains(ary[i].toLowerCase())) {

ary[i] = "is";

} else if (was.contains(ary[i].toLowerCase())) {

ary[i] = "was";

} else if (are.contains(ary[i].toLowerCase())) {

ary[i] = "are";

} else if (were.contains(ary[i].toLowerCase())) {

ary[i] = "were";

} else if (your.contains(ary[i].toLowerCase())) {

ary[i] = "your";

} else if (whats.contains(ary[i].toLowerCase())) {

ary[i] = "what is";

} else if (wheres.contains(ary[i].toLowerCase())) {

ary[i] = "where is";

} else if (whens.contains(ary[i].toLowerCase())) {

ary[i] = "when is";

} else if (hows.contains(ary[i].toLowerCase())) {

ary[i] = "how is";

} else if (whichs.contains(ary[i].toLowerCase())) {

ary[i] = "which is";

} else if (whos.contains(ary[i].toLowerCase())) {

ary[i] = "who is";

}else if (you.contains(ary[i].toLowerCase())) {

ary[i] = "you";

}

}

StringBuffer sb = new StringBuffer();

for (int i = 0; i < ary.length; i++) {

sb.append(ary[i]);

if (i < ary.length - 1) {

sb.append(" ");

}

}

correctedSentance=removeSymbols(sb.toString());

correctedSentance=convertToSingleSpace(correctedSentance);

return correctedSentance;

}

public String convertToSingleSpace(String sentance) {

boolean contains = sentance.contains(" ");

while (contains) {

sentance = sentance.replaceAll(" ", " ");

contains = sentance.contains(" ");

}

return sentance;

}

public boolean isNumber(char letter) {

boolean status=false;

int zero=(int)'0';

int nine=(int) '9';

int letterAsci=(int)letter;

if (letterAsci >= zero && letterAsci <= nine ) {

status=true;

} else {

status=false;

}

return status;

}

public boolean isLetter(char letter) {

boolean status=false;

int A=(int)'A';

int Z=(int) 'Z';

int a=(int)'a';

int z=(int)'z';

int letterAsci=(int)letter;

if ((letterAsci >= A && letterAsci <= Z) || (letterAsci >= a && letterAsci <= z) ) {

status=true;

} else {

status=false;

}

return status;

}

public boolean isSpace(char letter) {

boolean status=false;

if(letter==' ') {

status=true;

} else {

status=false;

}

return status;

}

public String removeSymbols(String input) {

for(int i=0; i<input.length(); i++) {

char letter=input.charAt(i);

if(!isNumber(letter) && !isLetter(letter) && !isSpace(letter)) {

input=input.replace(letter, ' ');

}

}

return input;

}

public static void main(String[] args) {

ClarityImprovements ci = new ClarityImprovements();

// String correctedSentance1 = ci.getCorrectedSentance("what's up");

System.out.println(ci.removeSymbols("H?/ai?/ ?dfd/s? dfd?"));

}}

**Code for sentence manipulation**

package vsrd.chatbot.grammar;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.Iterator;

import java.util.List;

import java.util.Set;

import vsrd.chatbot.conceptmap.Combinations;

import vsrd.chatbot.datamining.MineAnswers;

public class SentanceManupulations extends Combinations {

public String removeAuxiliaryVerb(String sentance) {

String[] words = sentance.split(" ");

String sent = "";

for (int i = 0; i < words.length; i++) {

if (words[i].equalsIgnoreCase("is")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("are")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("am")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("do")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("does")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("will")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("shall")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("can")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("could")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("would")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("has")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("have")) {

words[i] = "";

} else if (words[i].equalsIgnoreCase("should")) {

words[i] = "";

}

}

boolean first = true;

for (int i = 0; i < words.length; i++) {

if (!words[i].isEmpty()) {

if (first) {

sent = words[i];

first = false;

} else {

sent += " " + words[i];

}

}

}

return sent;

}

public String removeQuestionWordFromDQ(String sentance) {

String[] words = sentance.split(" ");

String sent = "";

boolean first = true;

for (int i = 1; i < words.length; i++) {

if (!words[i].isEmpty()) {

if (first) {

sent = words[i];

first = false;

} else {

sent += " " + words[i];

}

}

}

return sent;

}

public String firstLetterUpperCase(String word) {

String first = "" + word.charAt(0);

first = first.toUpperCase();

String converted = first + word.substring(1);

return converted;

}

public String objectToSubject(String objectPart) {

String[] words = objectPart.split(" ");

for (int i = 0; i < words.length; i++) {

if (words[i].equalsIgnoreCase("your")) {

words[i] = "my";

} else if (words[i].equalsIgnoreCase("my")) {

words[i] = "your";

} else if (words[i].equalsIgnoreCase("you")) {

words[i] = "I";

} else if (words[i].equalsIgnoreCase("me")) {

words[i] = "you";

} else if (words[i].equalsIgnoreCase("mine")) {

words[i] = "yours";

} else if (words[i].equalsIgnoreCase("yours")) {

words[i] = "mine";

}

}

String subject = "";

boolean first = true;

for (int i = 0; i < words.length; i++) {

if (first) {

subject = words[i];

first = false;

} else {

subject += " " + words[i];

}

}

return subject;

}

public String isToiss(String aux) {

aux = aux.equalsIgnoreCase("is") ? "iss" : aux;

return aux;

}

public String issTois(String aux) {

aux = aux.equalsIgnoreCase("iss") ? "is" : aux;

return aux;

}

public String whereTowher(String qWord) {

qWord = qWord.equalsIgnoreCase("where") ? "Wher" : qWord;

return qWord;

}

public String wherTowhere(String qWord) {

qWord = qWord.equalsIgnoreCase("wher") ? "Where" : qWord;

return qWord;

}

public String cleanSentance(String sentance) {

sentance = sentance.replace("'", "");

String[] words = sentance.split(" ");

String outPut = "";

boolean first = true;

for (int i = 0; i < words.length; i++) {

if (!(words[i].equalsIgnoreCase("for"))

&& !(words[i].equalsIgnoreCase("on"))

&& !(words[i].equalsIgnoreCase("the"))

&& !(words[i].equalsIgnoreCase("up"))

&& !(words[i].equalsIgnoreCase("down"))

// && !(words[i].equalsIgnoreCase("was"))

&& !(words[i].equalsIgnoreCase("but"))

&& !(words[i].equalsIgnoreCase("then"))

&& !(words[i].equalsIgnoreCase("of"))

&& !(words[i].equalsIgnoreCase("in"))

&& !(words[i].equalsIgnoreCase("by"))

&& !(words[i].equalsIgnoreCase("a"))

&& !(words[i].equalsIgnoreCase("an"))) {

if (first) {

outPut = words[i];

first = false;

} else {

outPut += " " + words[i];

}

}

}

return outPut;

}

public String lettersSeperatedBySpace(String word) {

word = word.toLowerCase();

String outPut = "";

boolean first = true;

for (int i = 0; i < word.length(); i++) {

if (first) {

outPut += word.charAt(i);

first = false;

} else {

outPut += " " + word.charAt(i);

}

}

return outPut;

}

public Set<String> anotherWord(String combination, String fromSentance) {

Set<String> outPut = new HashSet<String>();

String[] words = fromSentance.split(" ");

for (int i = 0; i < words.length; i++) {

if (!combination.contains(words[i])) {

outPut.add(words[i]);

}

}

return outPut;

}

public String convertToQuestion(String sentance) {

String outPut = "";

StringBuffer question = new StringBuffer();

String[] words = sentance.toLowerCase().split(" ");

Set<String> qWords = new HashSet<String>();

qWords.add("how");

qWords.add("who");

qWords.add("when");

qWords.add("where");

qWords.add("why");

qWords.add("what");

qWords.add("which");

qWords.add("whom");

qWords.add("whose");

Set aux = new HashSet();

aux.add("is");

aux.add("are");

aux.add("am");

aux.add("do");

aux.add("does");

aux.add("shall");

aux.add("will");

aux.add("would");

aux.add("has");

aux.add("have");

aux.add("should");

aux.add("can");

aux.add("could");

// aux.add("many");

aux.add("was");

aux.add("were");

aux.add("had");

aux.add("did");

boolean qw = false;

for (int i = 0; i < words.length; i++) {

if (qWords.contains(words[i]) && i < words.length - 1) {

qw = true;

}

}

boolean ax = false;

for (int i = 0; i < words.length; i++) {

if (aux.contains(words[i]) && i < words.length - 1) {

ax = true;

}

}

if (qw && ax) {

return sentance;

} else if (!qw && !ax) {

question.append("what");

question.append(" is");

for (int i = 0; i < words.length; i++) {

question.append(" ");

question.append(words[i]);

}

} else if (!qWords.contains(words[0])) {

if (ax) {

question.append("what");

for (int i = 0; i < words.length; i++) {

question.append(" ");

question.append(words[i]);

}

}

} else if (qWords.contains(words[0])) {

question.append(words[0]);

question.append(" ");

if (words[1].equalsIgnoreCase("many")) {

question.append(words[1]);

question.append(" ");

question.append(words[2]);

question.append(" ");

question.append("are");

for (int i = 3; i < words.length; i++) {

question.append(" ");

question.append(words[i]);

}

} else {

question.append("is");

for (int i = 1; i < words.length; i++) {

question.append(" ");

question.append(words[i]);

}

}

}

return question.toString();

}

public static void main(String[] args) {

SentanceManupulations sm = new SentanceManupulations();

System.out.println(sm.convertToQuestion("many staffs working in your company"));

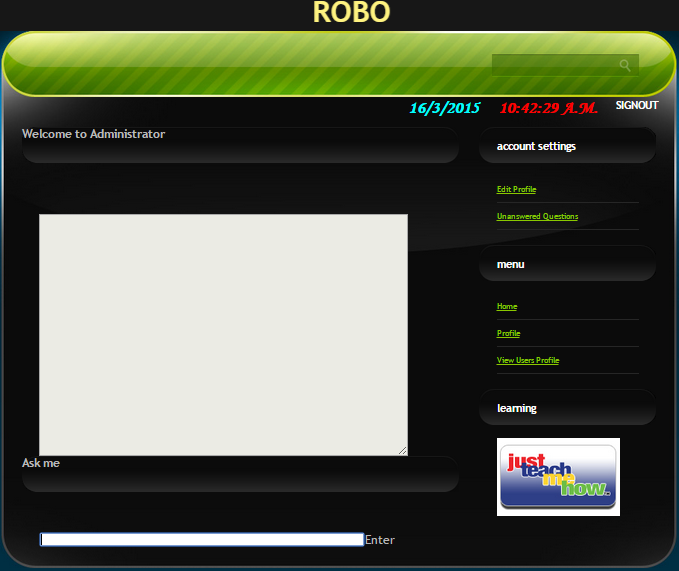
}

}

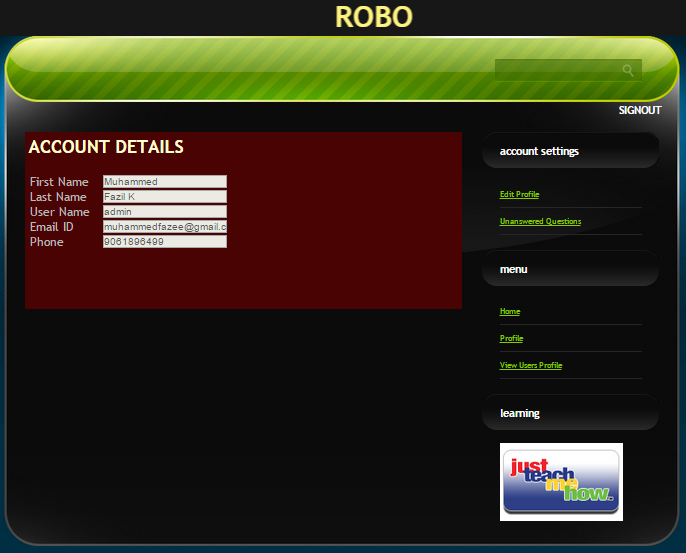
**12. RESULT AND FINDINGS**

**12.1 FORM DESIGN**

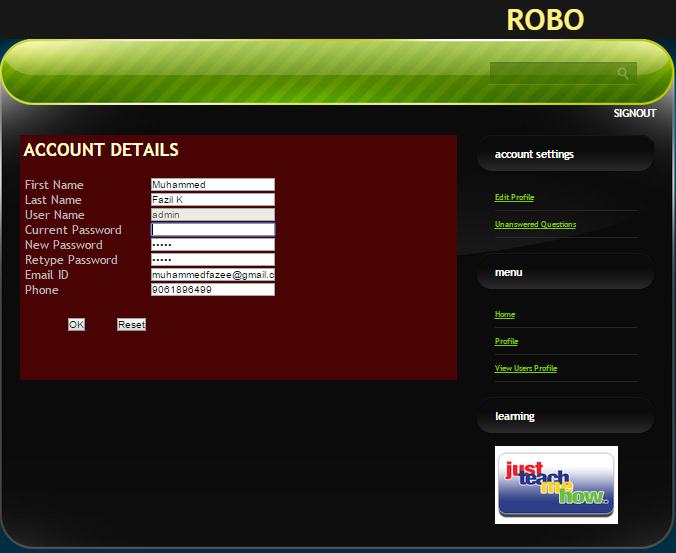
**Admin home**

****

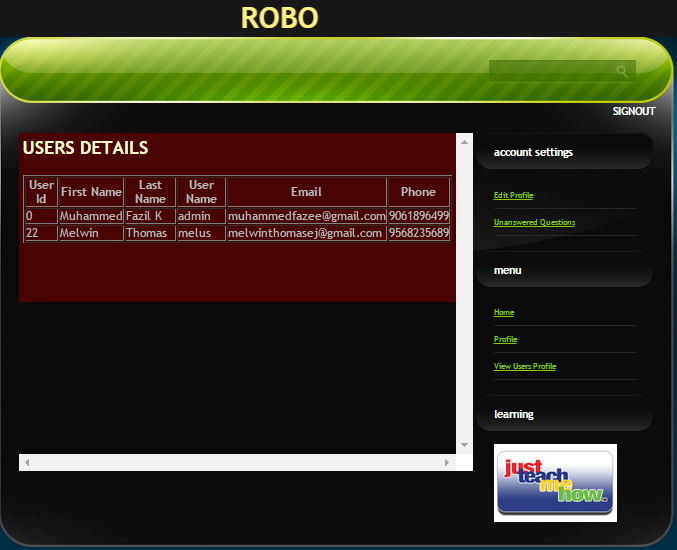
**Admin Profile**

****

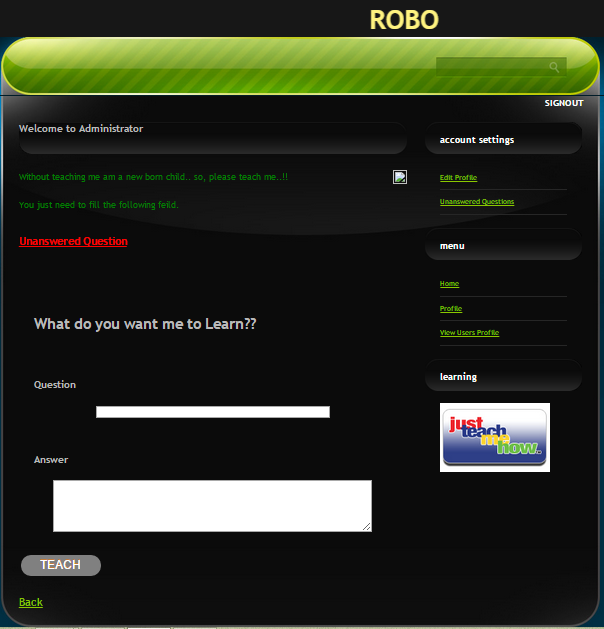
**Admin Edit Profile**

****

**Admin view user**

****

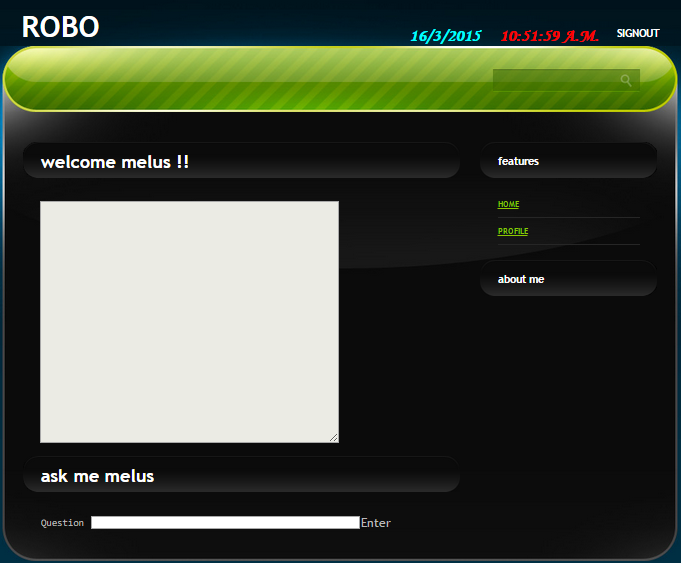
**Admin Teach**

****

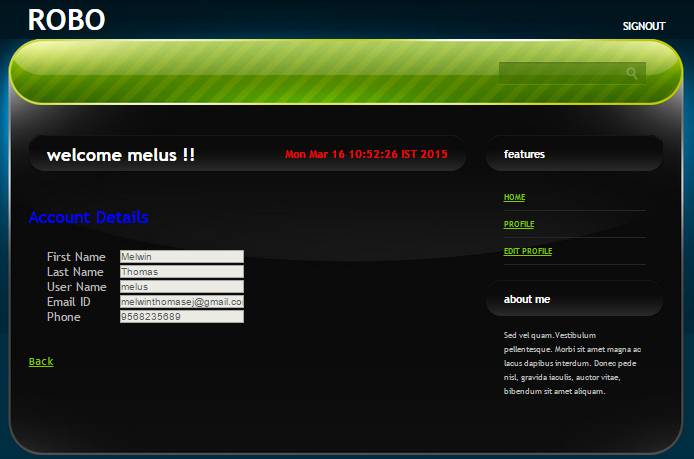
**Admin unanswered**

****

**User Home**

****

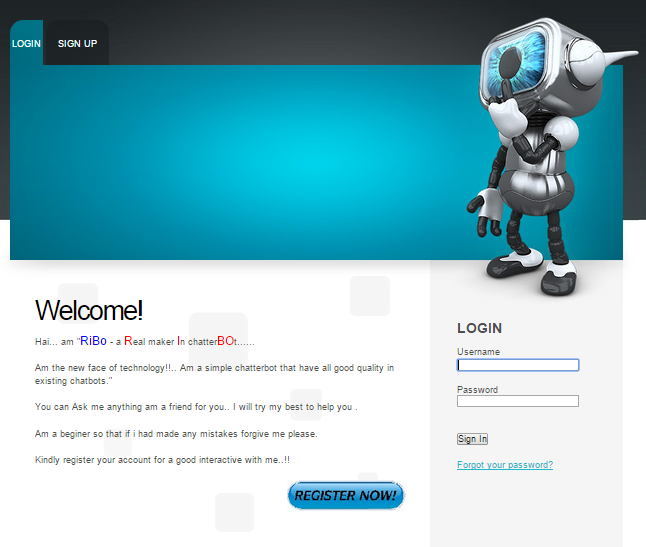
**User Profile**

****

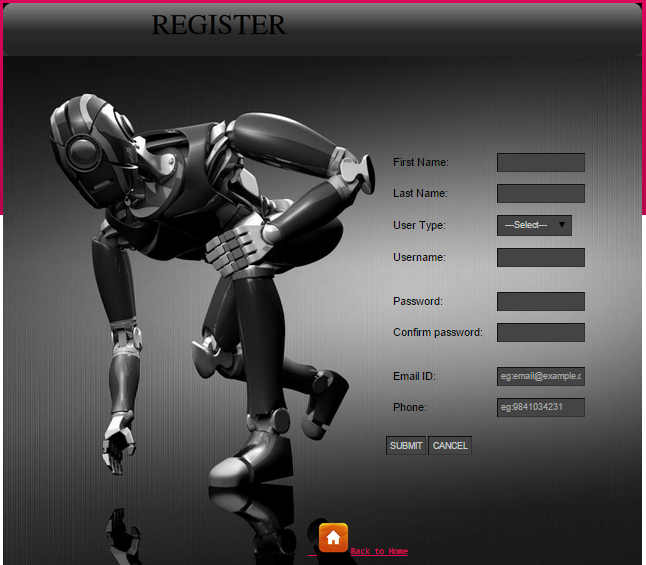
**User Edit Profile**

****

**Index**

****

**Signup**

****

**12.2 FUTURE SCOPE**

In future a chat bot which give response through the voice can be being in practice and also which can be response by identifying the different pronunciations of client. In future a chatbot which give response through the voice can be being in practice and also which can be response by identifying the different pronunciations of client. The resources required for the system is comparatively less. The resources that may be needed for the further developments are also assessed. In terms of the labour and the manpower, the present strength is sufficient and no extra personal to be required. So the present status of the company is sufficient enough for the project to be undertaken**.** It has been found that the proposed system is much better than the existing system where a person need to be trained to handle the user requests. There is no barrier for implementing the proposed system, which offers greater level of convenience to users, and it is operationally feasible The hardware and software requirements of the propose system is available everywhere. Since the software is platform independent, it can be implemented anywhere and thus making it widespread. Thus the system is technically feasible. Service can be provided to anyone who logins in the correct manner.

**13. CONCLUSION**

Chatbot is a software application of Data Mining that helps the firms to handle the customers more efficiently. It elevates a feeling in the user that he is interacting with a human rather than a mere machine. Workers employed for the purpose of usual customer care service can be reduced to a large extent with the advent of chatterbot. It handles all sorts of users in an equivalent basis irrespective of their diversifying nature. An all-time available customer service is ensured. The learning capability of chatterbot can be exploited by the administrator in saving a lot of time which is otherwise spent for the training of employees.

So far the administrative part of our project has been implemented with the provisions for adding subjects (as in English grammar) and departments. The administrator is also given the power to handle the learning module of chatbot. The clients can carry on a conversation about the college with the chatbot after submitting the user name. The format of answers for single subject questions has been made available. This project runs in a maximum user friendly manner.

**14. REFERENCES**

1. O. Ibrahim, H. ElGendy, and A. M. ElShafee “Towards Speed Detection Camera System for a RADAR Alternative,” in Proc. 11th International Conf. on ITS Communications, Sait-Petersburg, Russia, August 2011.
2. Y. Dedeo˘glu, “Moving Object Detection, Tracking and Classification for Smart Video Surveillance,” department of computer engineering and the institute of engineering and science of bilkent university, August, 2004.
3. J. C. Russ, “Image Processing Handbook,” the Fifth Edition, 2007.
4. R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, “a System for Video Surveillance and Monitoring”, The Robotics Institute, Carnegie Mellon University, Pittsburgh PAThe SarnoffCorporation, Princeton, NJ.
5. http://ieeexplore.ieee.org
6. http://www.researchgate.net