CSE AUTO-RESPONSE BOT

Group 11

Anh Nguyen - Emily Wang - Trieu Hung Tran - Yuxuan Luan

System Design Document

Version: (1) **Date: 09/25/2017**

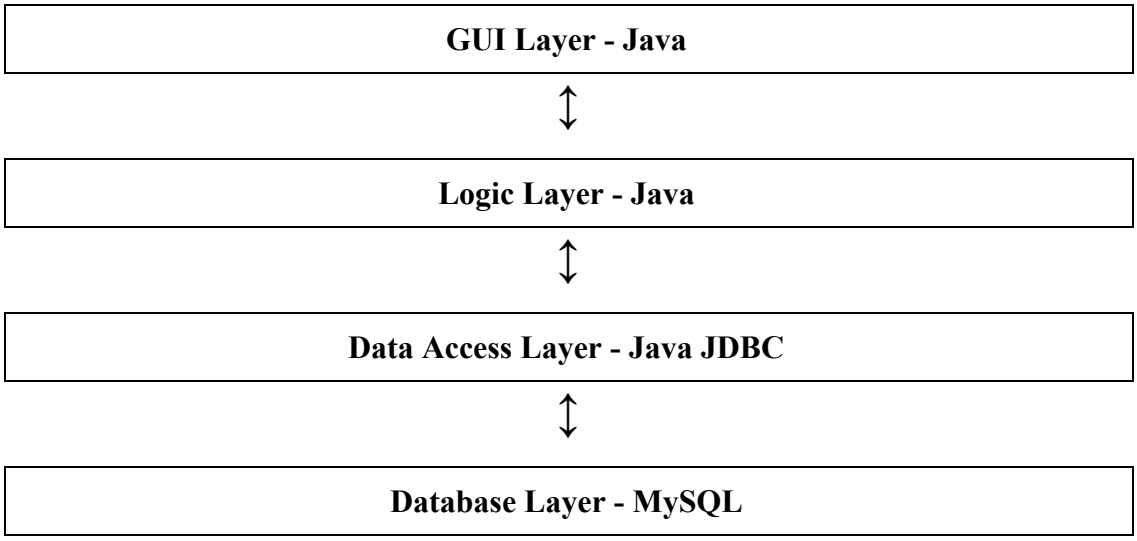
Table of Contents

1. Introduction

This system design document describes the architecture and system design of the CSE Auto-Response Bot. It includes the modular program structure and explains the relationship between the modules to achieve the complete functionality of the system. It also provides information about how the major data and system entities are stored, processed and organized. The audience of this document is software engineers and system architects who will be implementing and maintaining the described application.

2. Architecture

2.1 Introduction



The high level architectural design of the system will be a layered model. The lowest level will be MySQL database to store all permanent information. All of data in MySQL can be read and written using Java Database Connectivity. From there the logic layer, written in Java, will run the searching algorithm to transfer the questions into specific keywords which user inputs in the top layer - GUI.

2.2 Modules

2.2.1 Database Layer

The database layer is responsible for holding all the persistent data for the system and defining the relationships between this data. The system will be using an MySQL database that can easily be incorporated into the project via MySQL Workbench. The data relationships are described in more detail below in section 3.1.

2.2.2 Data Access Layer

This module is responsible for pulling and storing data from and to the database layer. All access to the database must go through this layer. This layer is also responsible for converting database into Java objects that can be manipulated by the Logic and GUI Layers. The primary internal functions of this module involve querying tables in the database to access required information and storing this information into Java objects. It will also have functions that will permanently store any changes made to these objects by other layers into the database. This layer will be implemented with the help of JDBC which is part of the Java frameworks.

2.2.3 Logic Layer

In Logic Layer, there will be two parts of the system. First, there will be all essential getter and setter for every Java Objects. Second part is the searching algorithm which is the most important part of the whole system. For any types of input from users, Logic Layer will translate and take out the most necessary keywords. Using these keywords, the system will retrieve to the GUI Layer the most appropriate response. However, if the system cannot find a specific question in the database, the system will automatically create a report and add to the database through Data Access Layer.

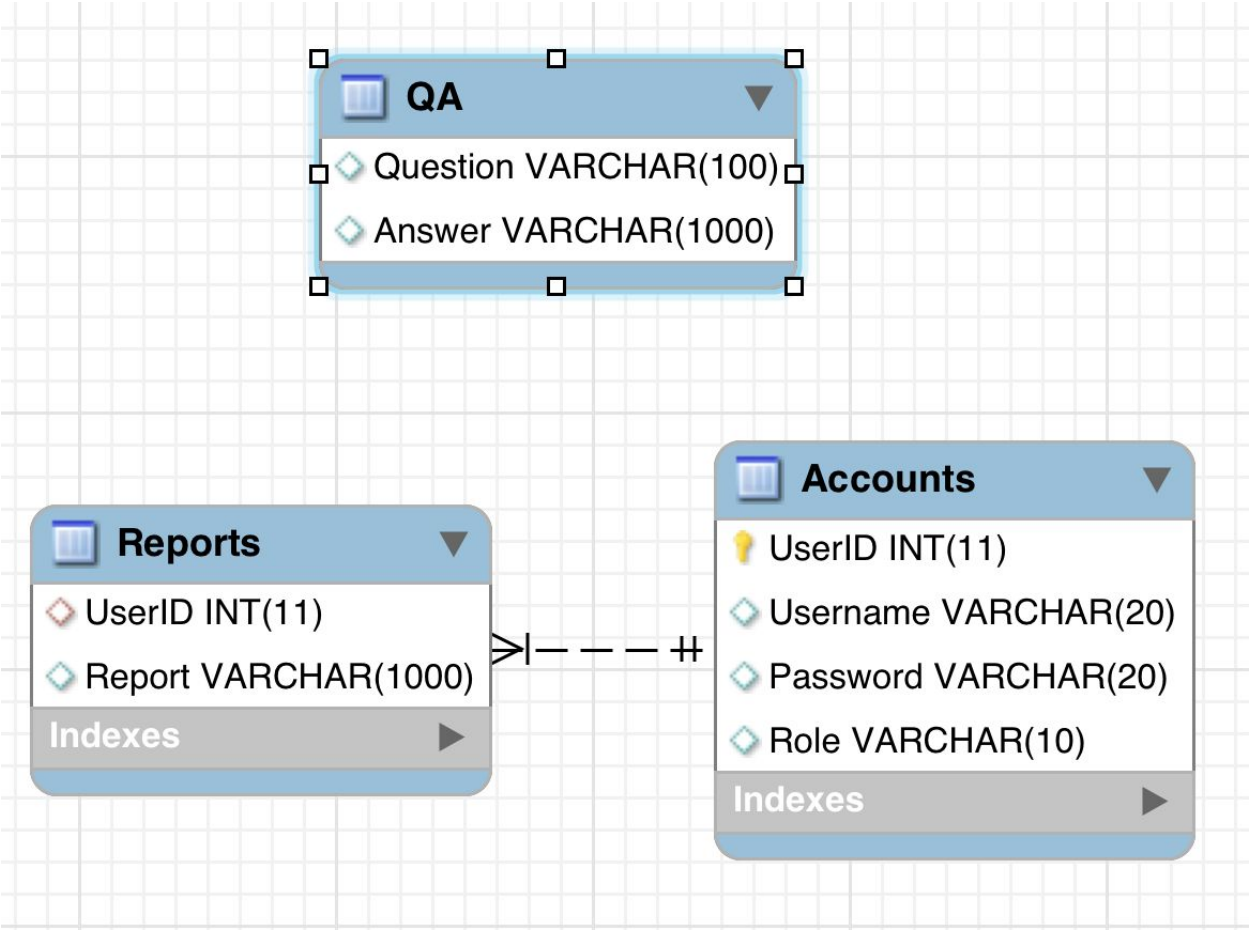
2.2.4 GUI Layer

GUI Layer is the primary layer that the user will interact with and is responsible for generating the user interface. The GUI Layer structure is described in more detail below in section 3.3. Because of creating by Java GUI, the layout of system will look really simple and beginner-friendly. In future releases, Login Screen will be added. However, all the layers are still connected the same way.

3. Class Diagram

3.1 Data Table Classes

3.1.1 Schema



3.1.2 Schema Information

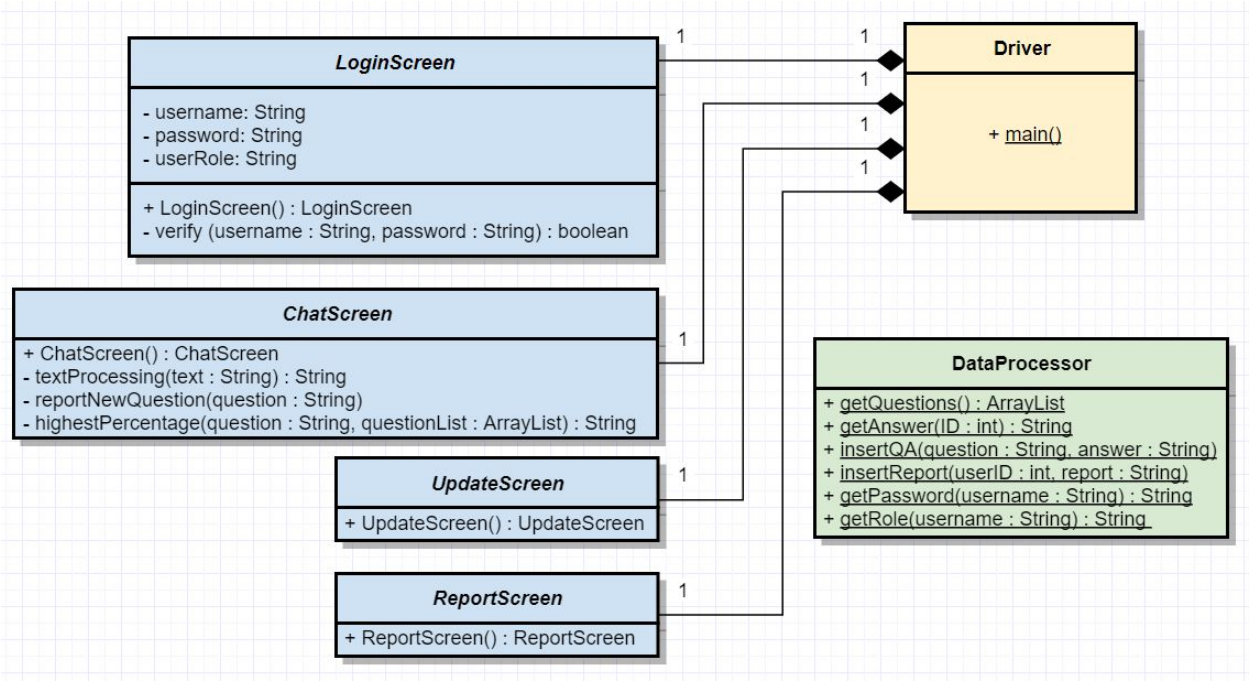
QA: this table contains all CSE questions along with their answers.

Reports: this table holds all reports from all users.

Accounts: this table stores all CSE accounts. Each account comprises of one username, one password, and the specific role of that person (Student/Professor/Staff).

3.2 Class Information

3.2.1 Schema



3.2.2 Schema Information

The data that implemented into Java could be organized into the following classes:

1. **Driver:** contains the main method of the whole program.
    a. **login**: a LoginScreen instance that handles all the login process.
    b. **chat**: a ChatScreen instance that handles all the chatting process.
    c. **update:** an UpdateScreen instance that handles all the updating process.
    d. **report**: a ReportScreen instance that handles all the reporting process.
2. **LoginScreen:** provides a login GUI that handles all the login process.
    a. **username:** string type, contains the username
    b. **password:** string type, contains the password
    c. **userRole:** Student/Staff/Professor
    d. **verify (username, password)**: verifies the correctness of the entered username and password, if the method return true, user role can be retrieved from database automatically.
3. **UpdateScreen:** provides an update GUI that handles all the updating process.
4. **ChatScreen:** provides a chatting GUI that handles all the chatting process.
    a. **textProcessing(string):** retrieve the String type question from the box in the ChatScreen and split the question into an array of words while get rid of all irrelevant keywords.

b. **highestPercentage(question, questionList):** compare the input question with all the questions in the database, retrieve and return the best-matched answer.
c. **reportNewQuestion(question):** automatically create and insert a report about a new question into the report table.

5. **ReportScreen:** provides a report GUI that handles all of the reporting processes.
6. **DataProcessor:** a helper class provides all methods that use to interact with the database.
    a. **getQuestions():** retrieve the list of all questions from the database.
    b. **getAnswer(ID):** retrieve the answer with a specific ID from the database.
    c. **insertQA(question, answer):** insert a question along with its answer to the database.
    d. **insertReport(userID, report):** insert a report from a specific user into the database.
    e. **getPassword(username):** retrieve the password of a specific user from the database.
    f. **getRole(username):** retrieve the role of a specific user from the database.

3.3 GUI Layer

The GUI Layer of the system consists of four basic pages: Login Screen, Chat Screen, Update Screen, and Report Screen.

When a user first opens the application, the Login Screen presents where he/she can enter his/her username and password. The entered information is then verified against the database. If the login information is invalid, an login error will be displayed, otherwise, the appropriate chat screen will be presented to the users depending on their roles. In addition, only Professors and Staffs can update the database.

In the Chat Screen, users can only enter questions that are related to CSE department. If the entered question is found in the database, its corresponding answer will be presented automatically in the textbox, otherwise, the system will inform users about the problem and then automatically send an update report into the report table .

When the role of the user is Staff or Professor, the user can access to the Update option, and the Update Screen will appear where user can type in a new question along with its answer. After clicking the submit button, both problem and answer will be inserted into the database.

All users are allow to submit reports by accessing into the Report Screen. In the textbox, users can describe what kinds of errors that were occurred during the process, or any

potential improvement of the system. All reports will be inserted into the database after users click the submit button.