CSCI 3308 Group Project
Milestone III - Database

Group name: StudyOutlet

Zoë Koppenhofer, Alexander Ray,

Woosung Jang, Ryan Whitmer, and Pengqi Yin(Bill)

March 21, 2017

1   Database Description

There are three things that need to be stored in this database:

1. Users and their information. We want users to be able to log in to the application, mostly to be able to access their set of incorrect questions. This set of incorrect questions discussed later allows us to play around with giving users more practice with questions they find difficult, and saves this information for later on down the line.

    For the first iteration of this app we are not storing actual "tests" (tests will simply be randomly picked questions with the same topic). We are also not planning on saving great deals of information on the history of the user (subjects, topics, time spent on a topic, user comments on the relevance of a test, etc) as we believe that is outside the scope of the initial functionality of the app. Any user preferences or history data that we feel a need to save in the initial run of the app can be stored using the `NSUserDefaults` class of the Swift `Foundation` framework.

    If it is deemed necessary to centralize this information in the future, we would likely store it in a combination of the "Users" table and new "Subjects" and "Topics" tables.

2. Questions and their associated answers, subjects, and topics. This will be stored in a "Questions" table.

3. Incorrect questions. This table is necessary to ensure we know which questions a user has gotten wrong; storing incorrect questions is necessary if we plan to do anything special when a user misses a question.
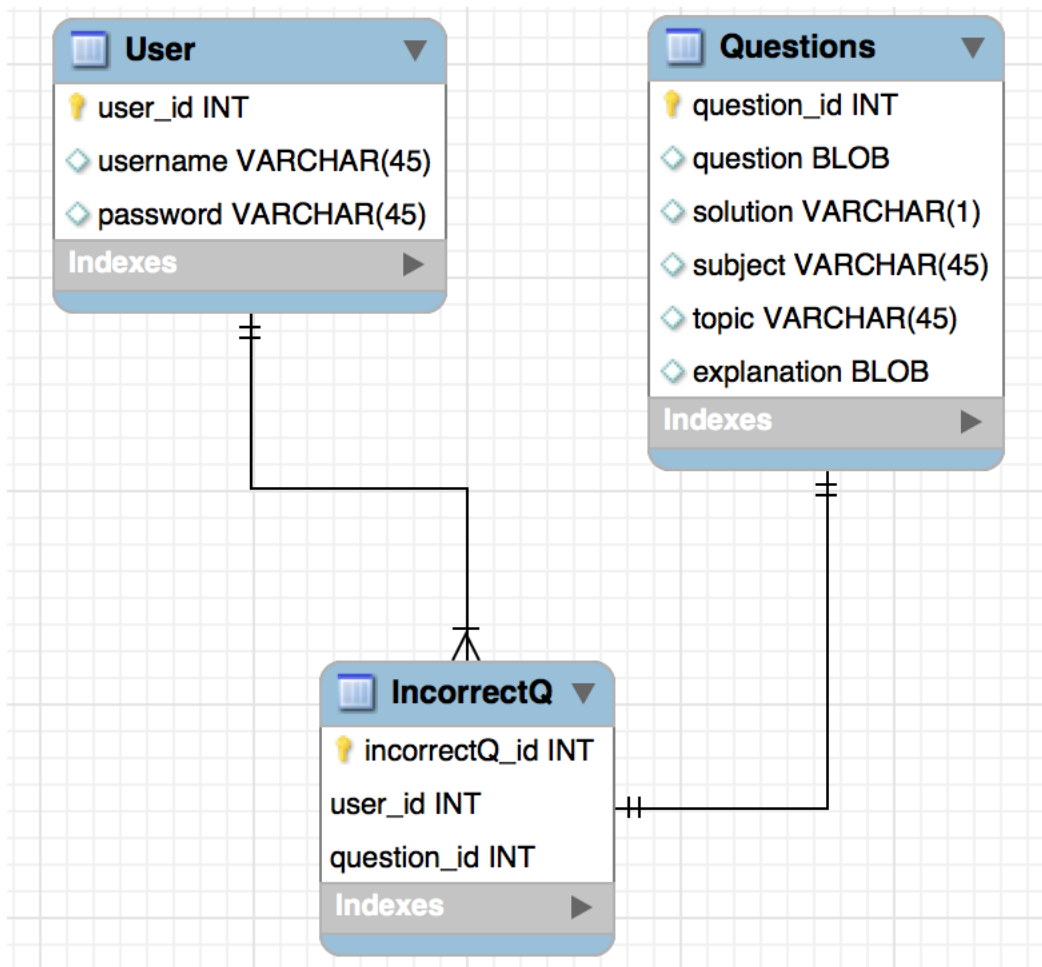
2 Data Model

Figure 1: ER Diagram describing relationships between tables in the database.

3   Database Description

```sql
create table if not exists `Users` (
 `user_id` int(1) not null auto_increment,
 `username` varchar(45) not null,
 `password` varchar(45) not null,
 primary key (`user_id`)
);

create table if not exists `Questions` (
 `question_id` int(1) not null auto_increment,
 `question` mediumblob not null,
 `answer` varchar(1) not null,
 `soulution` mediumblob not null,
 `subject` varchar(45) not null,
 `topic` varchar(45) not null,
 primary key(`question_id`)
);

create table if not exists `IncorrectQ`(
 `incorrectQ_id` int(1) not null,
 `question_id` int(1) not null,
 `user_id` int(1) not null,
 primary key(`incorrectQ_id`, `question_id` , `user_id`),
 foreign key(`user_id`) references User(`user_id`),
 foreign key(`question_id`) references Questions(`question_id`)
);

INSERT INTO `Questions` (`question`, `answer`, `solution`, `subject`, `topic`)
    VALUES (LOAD_FILE(Q1.png),'A', LOAD_FILE(Q1sol.png), 'Physics', 'Mechanics');
INSERT INTO `Questions` (`question`, `answer`, `solution`, `subject`, `topic`)
    VALUES (LOAD_FILE(Q2.png), 'D', LOAD_FILE(Q2sol.png), 'Physics', 'E&M');
INSERT INTO `Questions` (`question`, `answer`, `solution`, `subject`, `topic`)
    VALUES (LOAD_FILE(Q3.png), 'B', LOAD_FILE(Q3sol.png), 'Physics', 'Mechanics');
INSERT INTO `Questions` (`question`, `answer`, `solution`, `subject`, `topic`);
    VALUES (LOAD_FILE(Q4.png), 'A', LOAD_FILE(Q4sol.png), 'Physics', 'E&M')
INSERT INTO `Questions` (`question`, `answer`, `solution`, `subject`, `topic`)
    VALUES (LOAD_FILE(Q5.png), 'C', LOAD_FILE(Q5sol.png), 'Physics', 'Mechanics');

INSERT INTO `Users` (`username`, `password`)
    VALUES ('AlexRay', 'pass');
INSERT INTO `Users` (`username`, `password`)
    VALUES ('OneMoreUser', '123456');
```

Figure 2: Example code to create tables and add sample questions and users into the database.