Student Polling App

Reece Panek

Dr. Bo Hatfield

9 December 2019

Project Objective:

The objective of my project is to create an interactive web application for students and instructors to allow for a more interactive and engaged classroom as well as providing data on how well the students are understanding the concept taught while also providing a resource for students to look back on to study.

My Objective:

My objective is to apply the knowledge I have learned over the past four years in developing this project and enhance my skills. I will gain a greater understanding of what I have learned including scripting techniques, database design and GUI design. I will also be able to show this to possible future employers during my interview for reference.

- Experience completing a project from start to finish
- Gain a better understanding of connecting a back-end with a front-end
- Enhance my abilities with scripting techniques
- Practice creating a database from scratch that is easy to manage.
- Gain experience in creating an easy to use and understandable user interface for multiple types of users.

Problem Specification:

The student polling app is made to keep students engaged and interactive in the classroom. The web application will allow students to sign into classes and answer questions posted by the instructor as class goes along.

The instructor can take a screenshot of their screen and upload it in order to display its contents to the students. Then the instructor can choose the type of question it is such as multiple choice and true and false and the students will receive the image on their end as well as the option to answer the question. Afterwards the answers will be displayed anonymously as a bar and the instructor can choose which of the answers are correct and the question will be saved for reference. The answers are originally outputted anonymously so they instructor can share all of the answers with the class, but the instructor will also be able to see who exactly gave which response when looking for more details.

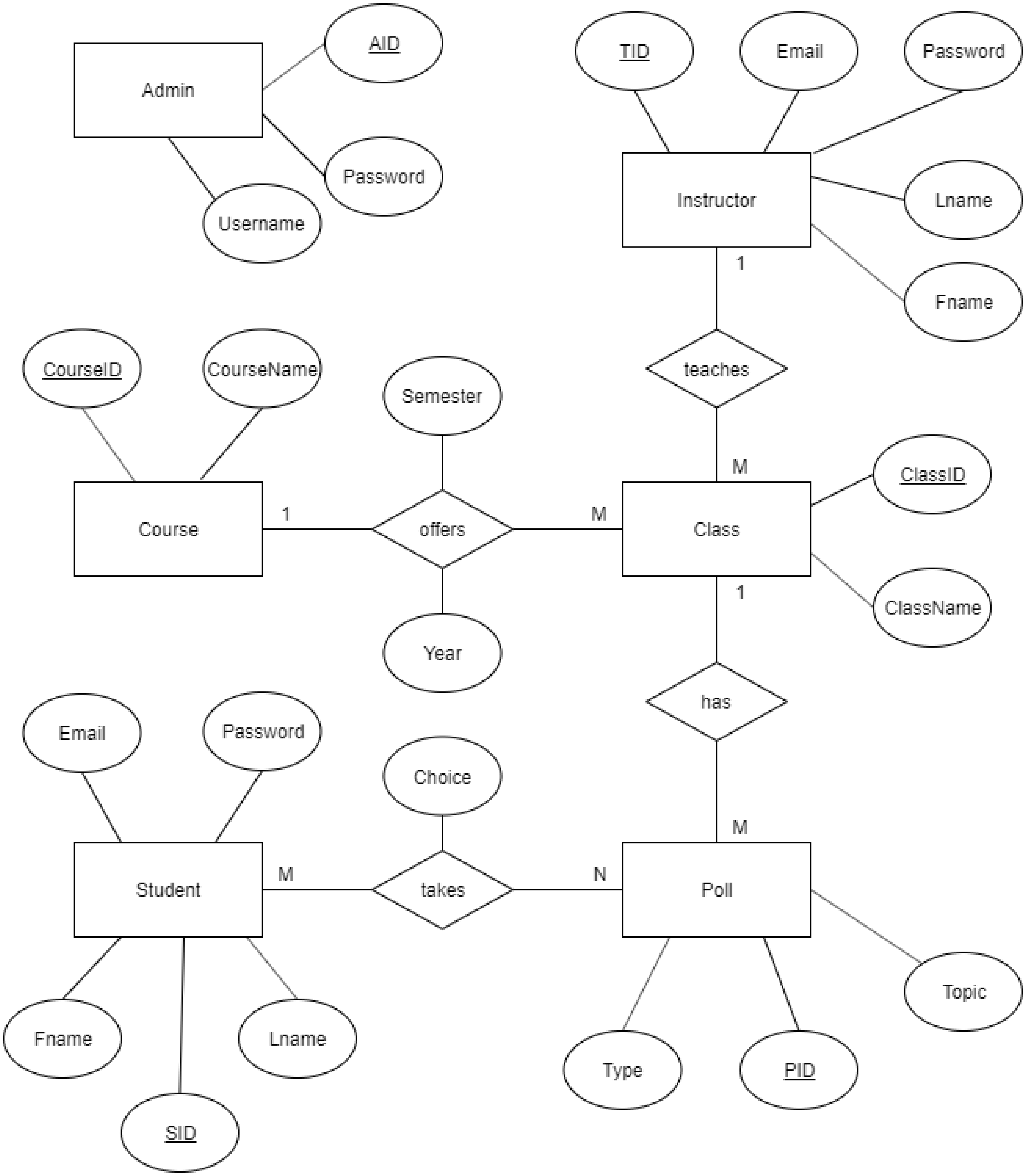
Since every question is automatically saved, both the instructors and students can use the web application at home in order to see how they responded to every question and what the correct answer is in order to study.

The web application will use each students SSU login in order to keep track of them and sign into class in order to keep track of grades and attendance. Each course will be separated into classes that each have a unique code the instructor can give to the students so they can join the class. Every lecture the instructor starts a class a new session will be created that can be sorted by the topic to keep things more organized. Since everything is saved and sorted, the instructors can look back to previous classes from earlier semesters to compare averages so they can better understand what areas students tend to struggle on more and revamp the way they teach that topic.

The web application will have three separate types of users with different permissions or privileges to use the app. The Student user will have the least amount of permissions only being allowed to view the classes they are in and their own answers. The Instructor user will be able to create and delete classes, view all the classes they teach, be able to upload and create questions, and view every student’s answers in their classes. Lastly, the Admin user will be able to view all of the information in every class, create and delete courses, and will have all of the permissions of the previous account types as well.

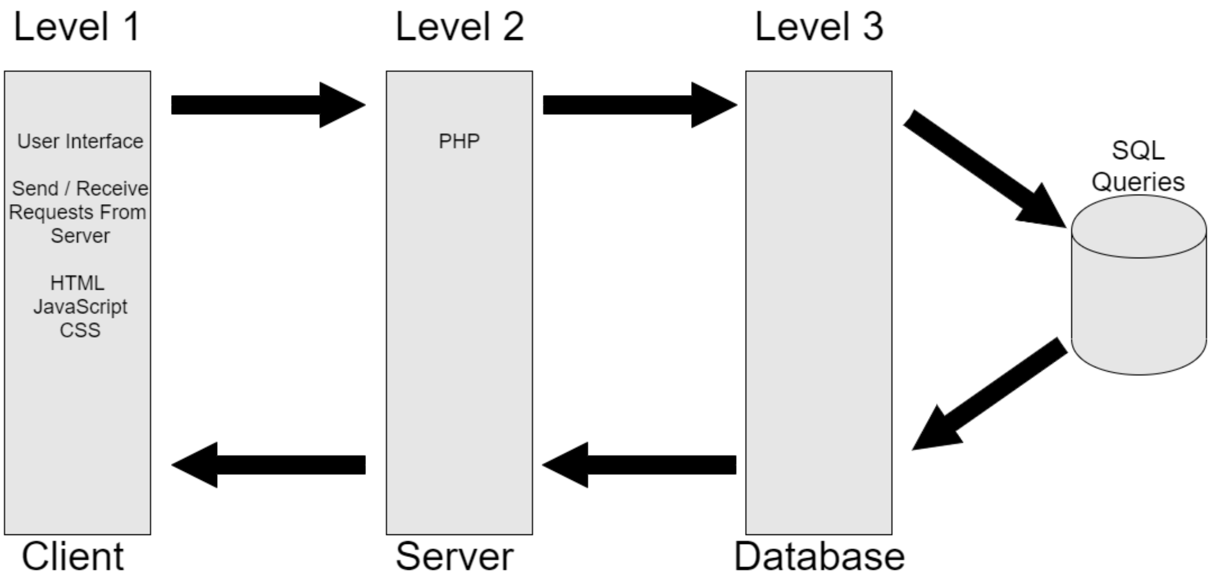
Solution Designs:

ERD:



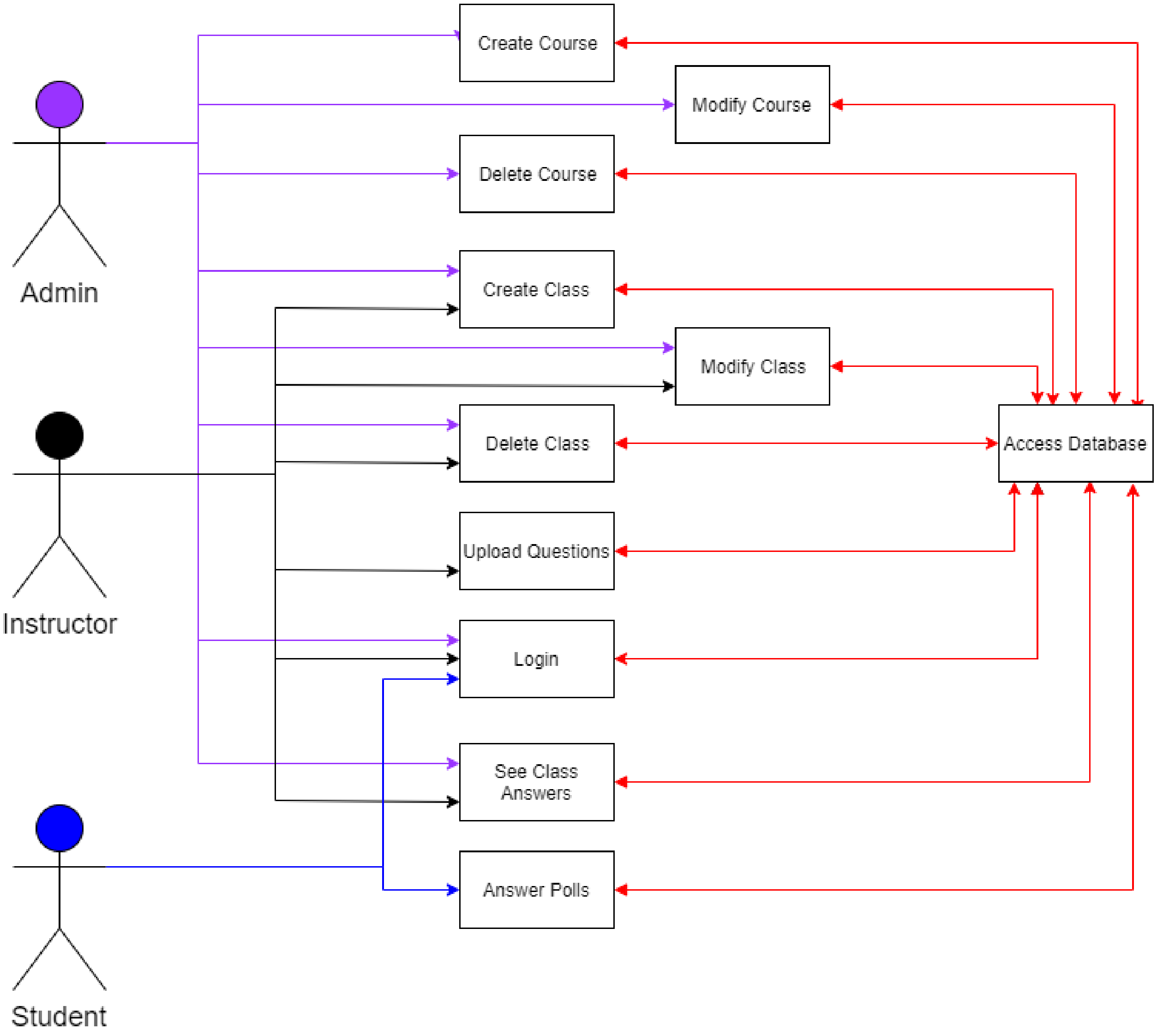
The ERD shows the information of each of the tables in the database. Each of the rectangles are the tables and the bubbles that connect to them are columns that hold the information in the table with the underlined bubbles being the Primary Key. Since the tables need to connect with each other the diamonds are the linking tables so they can connect the information between them while also showing how they correlate for each other. So, for instance the Instructor table connects to the Class table through Instructor_teaches_Class. These connect 1 to many since 1 instructor can teach many classes but each class is only taught by one instructor.

System Architecture:



The system architecture shows how my three-layer architecture is being used. The first level is the client level. This level is what the user sees and does, it has the information that the browser needs in order to display the site to the user and sends and receives requests from the user to the server. The second level is the server that it is stored on, this is where the PHP code is handled. Lastly, level 3 is the database where all of the information is stored, the database can send SQL queries to get information and send it back to where it is needed. All of these levels can send and receive information with each other and are needed in order for the web application to be functional.

Usage Diagram:



The usage diagram shows what each account type can do in a more simplified manner. Each account is colored coded so it is easier to read and its arrow points directly to what they can specifically do. For instance, while an admin create courses and classes, they are not able to upload questions or answer polls. The diagram also shows how every action the user does can send and receive information from the database, but they are never specifically accessing the database.

Tool List

- Server – weblab.salemstate.edu
    - LAMP STACK
        - APACHE
        - MySQL
        - PHP
    - PHPMYADMIN
- OS
    - Linux
    - Windows 10
- Languages
    - CSS
    - HTML
    - PHP
    - Javascript
- IDE
    - Visual Studio
- Mobirise4 (an offline website builder)
- ECharts (drawing tool)
- Version Control
    - GitHub

Benchmarks

Back-End Database Implementation

- Instructor
- Student
- Admin

Instructor Functionality, Implementation / Testing

Student Functionality, Implementation / Testing

Admin Functionality, Implementation / Testing

Integrated Testing

Time Schedule

Back-End Database Implementation – 3 weeks

- Instructor (1)
- Student (1)
- Admin (1)

Instructor Functionality, Implementation / Testing – 3 weeks

Student Functionality, Implementation / Testing – 3 weeks

Instructor and Student must be completed before moving further.

Admin Functionality, Implementation / Testing – 2 weeks

Integrated Testing – 2 weeks

Report / Presentation Preparation – 1 week

Grading Scheme

CSC 521 Presentation – 10%

Back-end – 25%

Instructor – 20%

Student - 15%

Admin - 10%

Report – 10%

Integrated Testing – 10%

Deliverables

- Original Presentation
- Original Proposal
- Amendments
- System Architecture Diagrams
- Project Version Control URL
- User Manual
- Sample Output
- Project Journal
- Postmortem
- Final Presentation
- Final Report
- Zip of All Files