

Design Pattern Documentation

Our group was tasked with creating the SMART system, a system to manage classrooms for extracurricular English and computer classes in African communities. This gave us a unique challenge of designing a system that is robust enough to keep track of many different types of users, while keeping it flexible enough to remain usable for those who might not be as tech literate.

We decided to use an Express Node javascript program to create the system. Express allows us to easily link to a MySQL database, create different web pages with session variables for the different types of users, and have a system that can be accessed from many different machines while it only needs one on the network on which to run. The Express files are split into creation and maintenance files such as database.js and app.js, and the two different components of each webpage, the .js and .ejs files. For example, when the program is launched, it will construct itself and connect with the database using app.js and database.js, and then the index page is built with a combination of the login.ejs and login.js files working together.

These webpages make up the vast majority of the program, with the other half being how they interact with the MySQL database. Almost all SQL is written in the database.js file, using stored procedures so that the information is efficient and easily accessed by different pages. The first time the program is booted up, it will create the database if it does not exist. The Entity Relationship Diagram also in the Docs folder of the project explains the exact tables created and their relationships to each other.

The core of navigation throughout the system is related to the user type. When a user is created, it is given a specific role. The roles are Owner, Admin, Instructor, Sponsor, and Social Worker. The system uses encryption to have a secure login for each user, and the pages that they can access are given to them with a session token based on this login. This ensures that the system is able to flexibly meet the requirements of whichever kind of user logs in, and is able to keep important protected functions away from users that don't have proper authentication.

Since all users are kept separate and have authentication through their session tokens, the functionality for each user is available right when they log in. Users without a login are able to submit applications to become a student. Instructors are able to create courses and assignments, grade students, take attendance, and pass a student in their class with a certificate. Admins are able to create semesters and new course names, and accept applications from potential students. Social Workers are able to enroll students in classes. Sponsors are able to sponsor students and check on their current grades. All of these functions work with the stored procedures to dynamically update the database, ensuring integrity of information.