

Drake Hovsepian

Christina Porter

Software Engineering Assignment 3

CS400-01

25 Oct. 2021

### Architectural Design of Valor's Schedule Planner

As discussed in Software Engineering Assignment 2, Valor's Schedule Planner will have 12 or more different features including a database that contains information about classes, schedules, users, professors, and majors, the ability to generate schedules, add in personal scheduling, compare different schedules, add users as friends, share schedules using a link or through friends, and view professor ratings. Each of these features is a critical component for providing a unique interface for scheduling at Bellarmine University and should eliminate the need of having to manually create schedules that are viewable to the user only. To implement these features, we will plan on using MySQL for the database, Node.js for the backend, and React for the frontend while using multiple services that allow for signing in, storage, security, and more.

Each component plays a critical feature in our application. Our first, most visible, component is the frontend webpage application, which we plan to build with React. React is a declarative and component-based JavaScript library that is made for rendering UI in a web application. One useful function of React is the ability to create and use templates so we don't have to duplicate code. React supports HTTP communication with Node.js using React Native, which is vital to sending information over the web and communicating with our database. Our next component is the webserver, which will handle traffic generated by each user accessing the

webpage. This webserver will handle HTTP requests sent to it by the React webpage. We plan to use Node.js to implement this webserver. This language supports responsive and robust communication that can handle thousands of simultaneous requests. Node.js is a powerful networking tool with a wide selection of libraries.

The database that will store the information that will be accessed by Node.js will be made with MySQL. MySQL is a relational database management system that communicates with other applications to provide powerful data-driven functionality. This will be essential to constructing schedules out of the many possible classes, sections, and semester opportunities. Our application will provide functionality that allows allow developers and administrators to adjust the database, when necessary, with more complex functionality than normal users so high-privilege users can access every aspect of the application.

The main component of the architecture will be Azure Cloud Services, which can provide a server, an Azure SQL Database, and the ability to host our Node.js architecture in one platform. Not only does Azure provide a platform that will host everything, Azure also provides its own architecture that provides a layer of security that wouldn't be available with regular Node.js that includes the Azure Security Center, Application Gateway, Azure Active Directory, Azure DDoS Protection, Key Vault, and Azure Information Protection. Using this, we can allow users to sign in to the application using their Bellarmine Single Sign On. The diagram representing our application framework is in Figure 1.

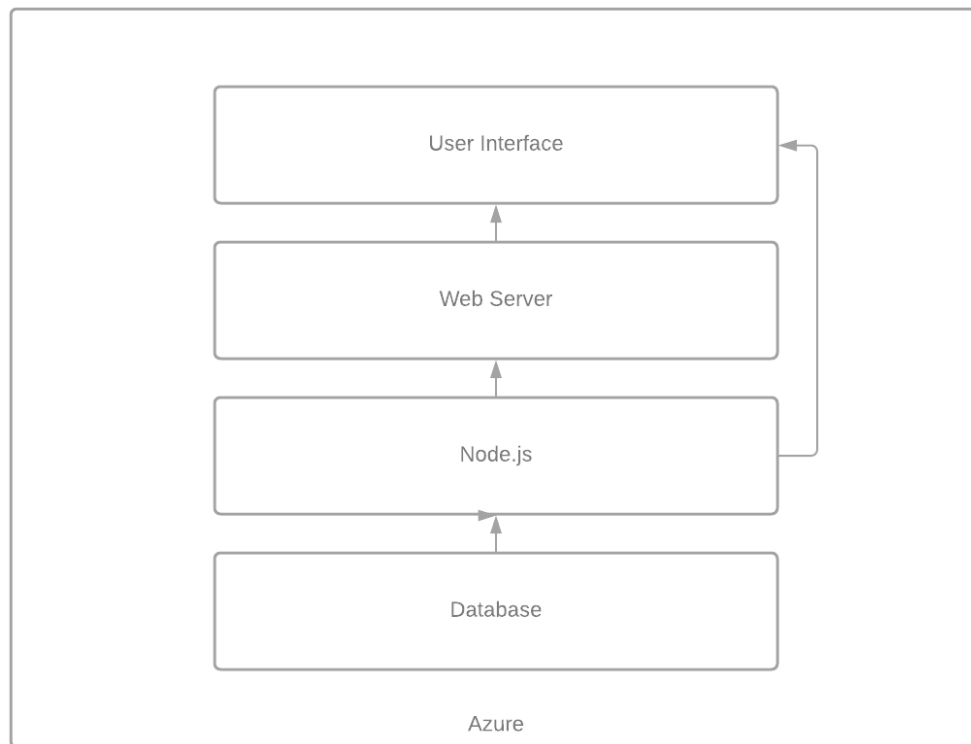
In order for Node.js to communicate with our Azure MySQL database, we will need to install tedious, which is a library used within Node.js that allows the user to connect to a MySQL Server database. From here, we can access our database, which contains tables for Classes, Schedules, Users, Professors, and linking tables. Our database diagram is contained in Figure 2.

From here, we can use the Express framework to host our web server, which works very similarly to the http module in Node.js but extends functionality via a framework to allow for asset hosting, templating, cookie parsing, data handling, and more. We are still exploring how we can connect the React page to the Express server, but a couple options include using a proxy to handle requests from the React page to the Express server or to include the React files in the same folder as the Express documents. Table 1 includes additional frameworks that we can use to implement features.

Feature	Framework
Friend Request	Socket.IO
Schedule Sharing	Redis

*Table 1 – Features and Frameworks*

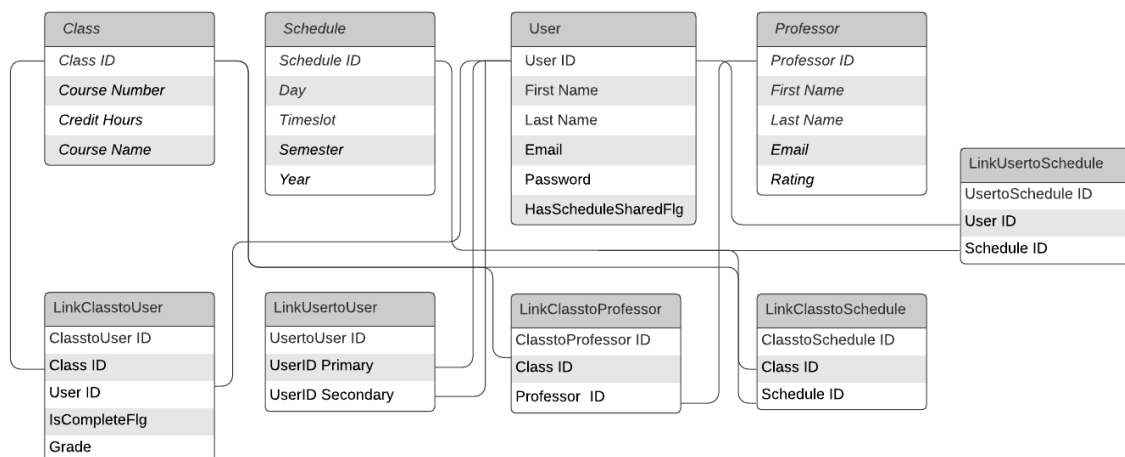
Microsoft’s Visual Studio (or VS) Code software will allow us to implement each of these components and write the software that allows each of them to communicate with each other. This editor supports a wide variety of languages and tools, giving our team flexibility in case we need to add or change planned components in our architecture. Many of the same importable tools and libraries in VS Code that extend Node.js functionality also support React. VS Code natively supports debugging for Node.js, and has built-in Git tools that will allow us to easily push and pull code from dedicated repositories in increments. As a Microsoft product, VS Code allows Azure services to be run directly from the editor, which enhances our testing and update deployment abilities.



*Figure 1 - Architecture Diagram*

### Valor's Schedule Planner SQL Database Diagram

Christina Porter and Drake Hovsepian | October 24, 2021



*Figure 2 - Valor's Schedule Planner SQL Diagram*