
CSC 431

# SyllaPlan

## System Architecture Specification (SAS)

Team 07


| | |
|---|---|
| **Aashay Badgamia** | **Project Manager** |
| **Andrew Shatsky** | Developer |
| **Julian Cantillo** | Prototyper |

# Version History

| Version | Date | Author(s) | Change Comments |
|---|---|---|---|
| **1.0** | 04/06/23 | Aashay Badgamia<br>Andrew Shatsky<br>Julian Cantillo | First Draft |
| | | | |
| | | | |
| | | | |

# Table of Contents

# Table of Figures

# 1. System Analysis

## 1.1 System Overview

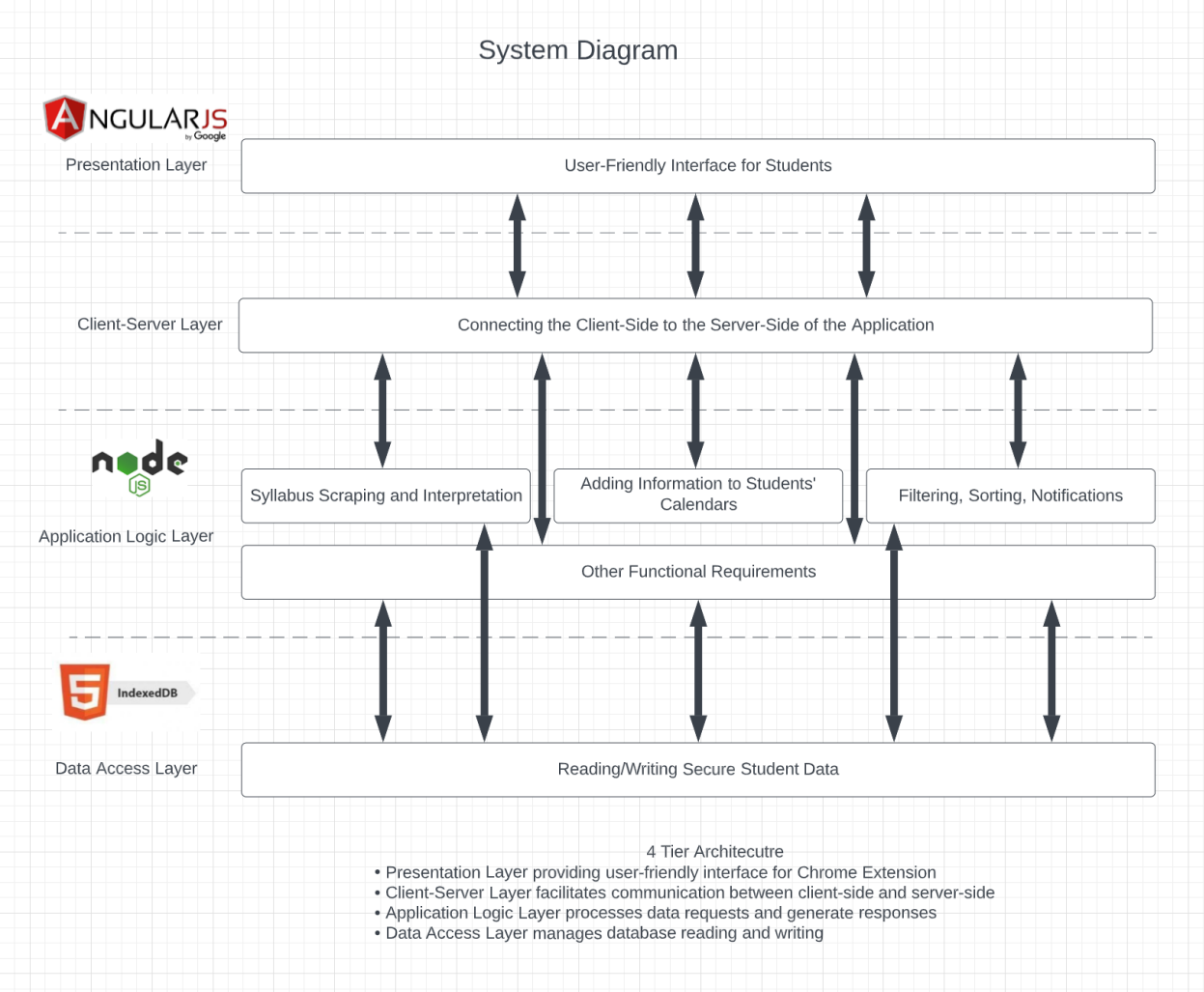
Our system, SyllaPlan, is a Chrome Extension designed to help students plan and organize their semesters more efficiently. The system is built using a web development framework, Angular, and is compatible with Chrome Extension regulations.
The architecture of our system is based on the client-server model. The Chrome Extension runs on the client-side and interacts with the server-side through APIs. The server-side is responsible for processing data requests and storing data in a secure database, such as IndexedDB.
To ensure a seamless user experience, we have implemented features like automatic data extraction from syllabi documents and a user-friendly interface displaying all the student's classes.

We will use a 4-tiered architecture for Syllaplan:

1. Presentation Layer: This layer includes the user interface of the SyllaPlan Chrome Extension, which is built using a web development framework, Angular
2. Client-Server Layer: This layer is responsible for communication between the client-side and server-side. The client-side is the SyllaPlan Chrome Extension, while the server-side is responsible for processing data requests and storing data in the database. The client and server communicate through APIs.
3. Application Logic Layer: This layer includes the business logic of the SyllaPlan system, which is responsible for processing data requests from the client-side and generating responses. It includes the core syllabi scraping and interpretation feature, which runs NodeJS scripts. The application logic layer is implemented on the server-side and communicates with the client-side through APIs.
4. Data Access Layer: This layer includes the database management system (DBMS) used to store and retrieve data. In this case, we are using database compatible with Chrome Extensions, IndexedDB, to store user data in a secure manner.

Overall, this 4-tiered architecture provides a scalable, efficient, and secure system for SyllaPlan, with each layer responsible for a specific function. The presentation layer provides a user-friendly interface for the Chrome Extension, while the application logic layer processes data requests and generates responses. The data access layer manages the database, and the client-server layer facilitates communication between the client-side and server-side.

# 1.2 System Diagram




# 1.3 Actor Identification

**User (human):** The primary actor of SyllaPlan is the student, who interacts with the system through the Chrome Extension. The student uses the system to plan and organize their semester, and to view their class schedules and other relevant information.

**SyllaPlan Server (external system):** The server-side component of the SyllaPlan system, which is responsible for processing data requests from the Chrome Extension and interacting with the database to store and retrieve data.

**Database (external system):** The database is a critical actor in the system, responsible for storing all the data used by SyllaPlan. The database interacts with the API server to provide the data needed by the client-side, and is responsible for ensuring data integrity and security.

**System Clock (time):** The system clock is an important actor in the system, providing a time reference that is used by various components of SyllaPlan. For example, the system clock is used to schedule notifications and reminders for upcoming assignments or exams, helping students to stay on top of their workload

# 1.4 Design Rationale

## 1.4.1 Architectural Style

Overall, this 4-tiered architecture provides a scalable, efficient, and secure system for SyllaPlan, with each layer responsible for a specific function. The presentation layer provides a user-friendly interface for the Chrome Extension, while the application logic layer processes data requests and generates responses. The data access layer manages the database, and the client-server layer facilitates communication between the client-side and server-side. We plan to have the presentation layer on the client-side, while having the application logic and database access layer on the server-side. One advantage of this design is that we can update our application logic and database back-end scripts without having the user download additional versions. However, this will call for higher server capability leading to higher costs.

## 1.4.2 Design Pattern(s)

**Singleton Pattern**: This pattern ensures that there is only one instance of a class in the system, making it easier to manage shared resources such as database connections or configuration settings. In SyllaPlan, a singleton pattern could be used to manage database connections or other shared resources.

**Decorator Pattern**: This pattern allows behavior to be added to an object dynamically, making it easier to extend the functionality of the system without modifying the existing code. In SyllaPlan, a decorator pattern could be used to add additional security measures, such as rate limiting or IP blocking, to the API.

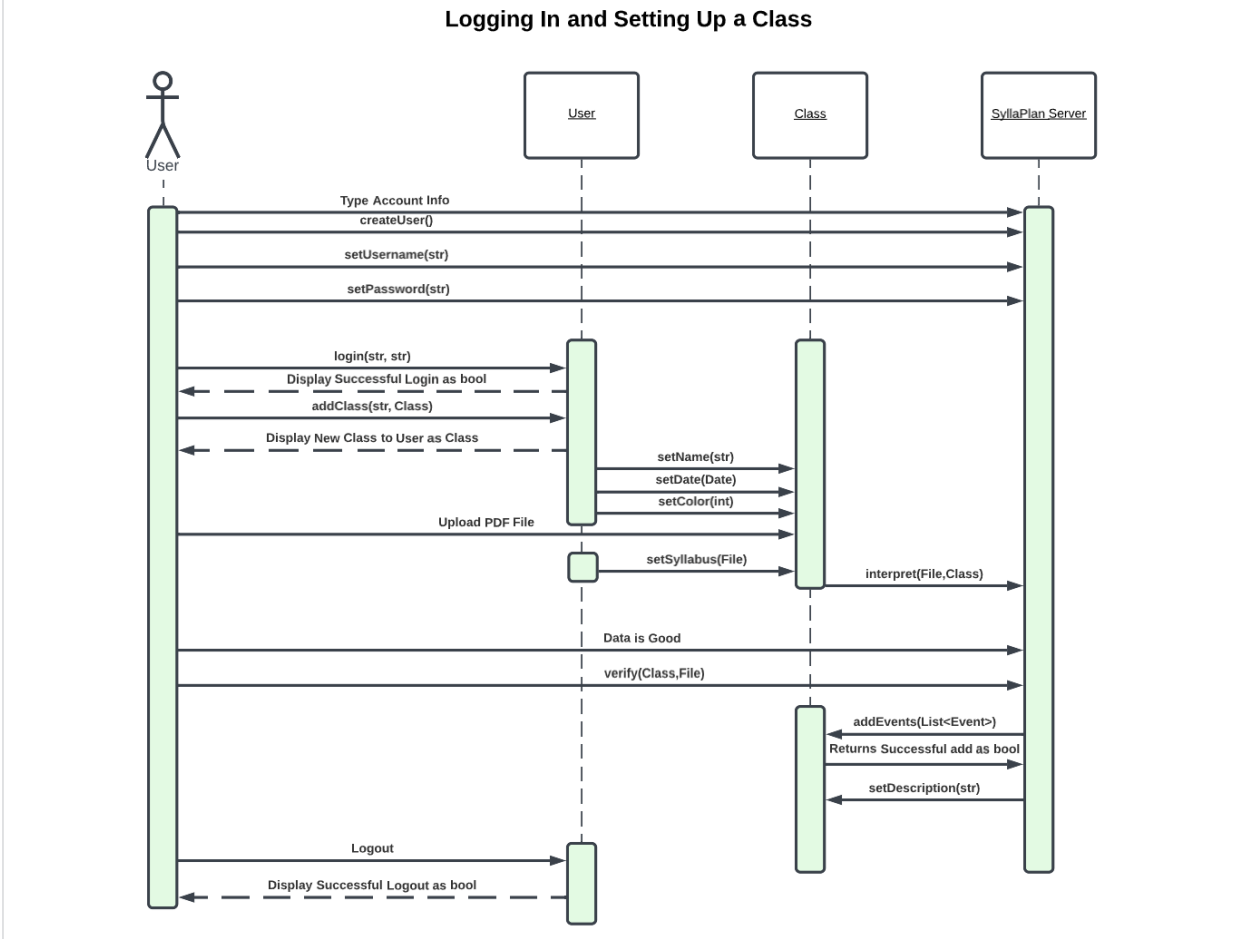
## 1.4.3 Framework

Angular is a popular open-source framework maintained by Google for building complex single-page web applications using TypeScript. It provides various features such as data binding, dependency injection, and component-based architecture to simplify code organization, management, and scalability. SyllaPlan has selected Angular for its project due to its comprehensive feature set, strong community support, and compatibility with TypeScript. By using Angular, SyllaPlan aims to create a scalable and complex application with improved code organization and early error detection.

Chrome Extension Framework, which includes a set of APIs and tools provided by Google Chrome to build extensions. These APIs include content scripts, background scripts, popups, storage, and messaging. These will be used alongside the Angular Framework to facilitate integration with Google Chrome.;

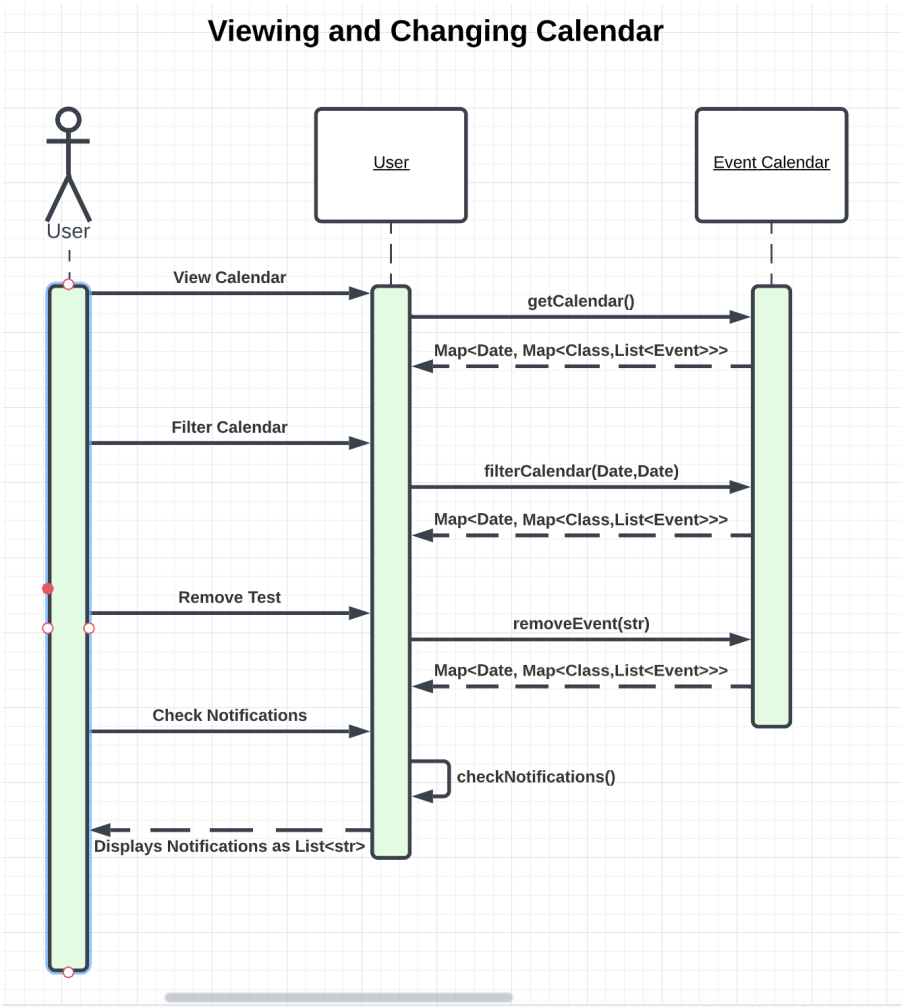
# 2. Functional Design

## 2.1 Logging In and Setting Up a Class




- User creates an account, setting username and password
- SyllaPlan Server adds the User object to the list of all users
- User logs in, receives a success message
- User sees new class added
- Color, name, and date are set
- User uploads syllabus PDF for class
- Syllabus is scraped and interpreted through SyllaPlan Server
- User verifies and adds any data to the interpretation
- All events are added with all information and class description
- User logs out and is displayed a successful logout message

# 2.2 Viewing and Changing Calendar




- User is logged in and wants to view calendar of assignments/events
- EventCalendar is shown to user
- User wants to filter calendar from date x to date y
- EventCalendar uses filterCalendar(Date, Date) and returns new filtered calendar
- User wants to remove an event from the Calendar
- EventCalendar uses removeEvent(str) to remove event and returns updated calendar without event
- Extension automatically checks for notifications and returns a list of events to user if any notifications found

# 3. Structural Design