

Requirements Specification Document

Async

5.0 Requirements Specification

5.1 Introduction

5.2 CSCI Component Breakdown

5.3 Functional Requirements by CSC

5.3.1 Functional Requirement 1

.
. .
.

5.3.n Functional Requirement n

5.4 Performance Requirements by CSC

5.4.1 Performance Requirement 1

.
. .
.

5.4.n Performance Requirement n

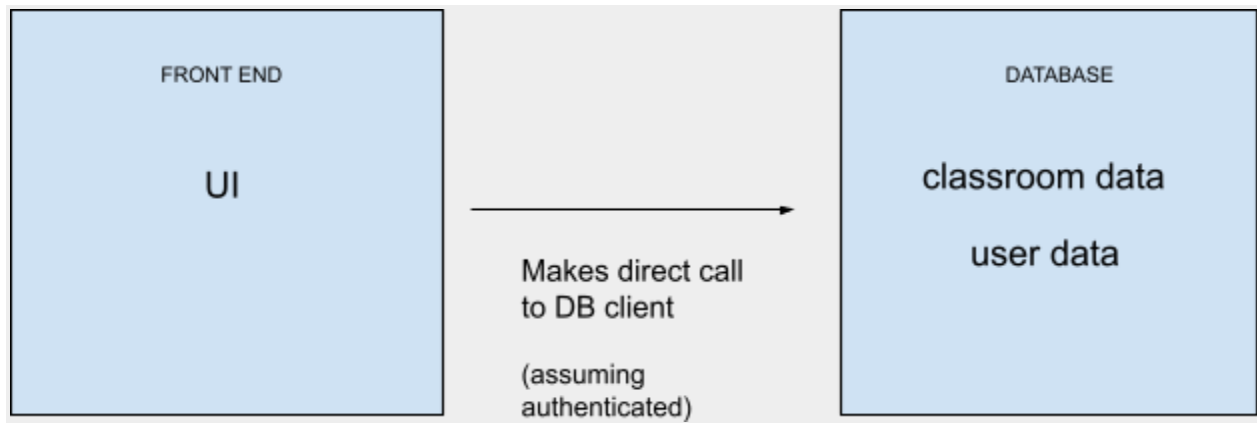
5.5 Project Environment Requirements

5.5.1 Development Environment Requirements

5.5.2 Execution Environment Requirements

5.1 Introduction

The design of this system will focus on creating a polished and feature-rich application in the education space. Brightspace, one of the primary learning management systems, has features that provide the bare minimum, but leaves much to be desired in regards to user experience. This system will focus on providing improved discussion board and class interaction features for both students and teachers. The following diagram presents the basic system-structure of this application:



5.2 CSCI Component Breakdown

CSCI Async is composed of the following CSCs:

5.2.1 Front-End

- 5.2.1.1 Client Main Screen
- 5.2.1.2 Client Discussion Board
- 5.2.1.3 Client Notes Upload
- 5.2.1.4 Client Chat Interface
- 5.2.1.5 Client-Side Routing Library
- 5.2.1.6 Client-State Management Library (optional)

5.2.2 Back-End

- 5.2.2.1 Server-State Management Library
- 5.2.2.2 Database Integration Library
- 5.2.2.3 Query Schema Module
- 5.2.2.4 Endpoint Structure Module

5.3 *Functional Requirement*

- 5.3.1 The web application shall first display a revamped login main page for users.
 - 5.3.1.1 Users who attempt to access a specific page route will be automatically redirected to the login page.
 - 5.3.1.2 Users will be able to easily sign in and out with an email-password and Google account.
- 5.3.2 The web application shall authenticate users and ensure that their information is secure.
- 5.3.3 The web application shall then display a new home-page dashboard of relevant information, including a calendar preview and links to specific classes.
- 5.3.4 The web application shall have a calendar system that automatically displays for assignments, tests, and other key dates.
- 5.3.5 The web application shall utilize back-end authentication for users.
- 5.3.6 The web application shall include a sidebar for key app features.
- 5.3.7 The web application shall allow users to upload notes for class.
- 5.3.8 The web application shall allow users to set up and respond to discussion board posts.

5.4 *Performance Requirements*

- 5.4.1 Re-Renders Should Be Minimized As Much As Possible
 - Client State management will be a very important consideration as it relates directly to the performance of the React front-end.
- 5.4.2 Improved User Authentication
 - User authentication information will be accessed from a single source of truth to minimize errors and allow for easier debugging. Additionally, authentication status is used to prevent non-authenticated users from accessing specific application pages.
- 5.4.3 Responsive Rendering
 - The application should load smoothly and be responsive to user clicks and events. Client-Side Routing and Server-Side Rendering are potential tools to reach and optimize this goal.
- 5.4.4 Class Homepage Design Must Be Intuitive
 - Users must be able to navigate their class pages easily and intuitively.
- 5.4.5 Follow Data Fetching Best-Practices
 - Data that is fetched from the database or other external services should be handled procedurally with error handling and fallbacks.
 - Loading States must be taken into consideration.
- 5.4.6 Design a simple but effective Database structure

- A well-organized database will improve performance across the application and will lead to less errors during fetching.

5.5 *Environment Requirements*

5.5.1 Client Laptop or Desktop Computer:

5.5.2 Client OS: Windows, macOS, Linux

5.5.3 Client Minimum Processor: Intel Core i5 or equivalent

5.5.4 Client Minimum Storage: 256 GB

5.5.5 Client Minimum RAM: 8 GB

5.5.6 Developer Software Requirements: VSCode, NPM, Node, Fastify, Supabase, React, Vitest, and various libraries.