White Volley Girls

CSUN Dashboard

Software Design Document

Name (s):
Christian J.
Nima S
Michael B.
David H.

Section:
Workstation:

Date: (24/October/2022)

Software Design Document

TABLE OF CONTENTS

Software Design Document

1.0 INTRODUCTION

1.1 Purpose

Identify the purpose of this SDD and its intended audience. (e.g. “This software design document describes the architecture and system design of XX. ….”).

This software design document describes the architecture and system design of the CSUN Dashboard

1.2 Scope

Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product.

The defragmentation of information for the consumers is the ultimate end goal of this project will allow them to effectively plan and coordinate their financial and short-term/long-term projections and movement.

1.3 Overview

Provide an overview of this document and its organization.

1.4 Reference Material

This section is optional.

List any documents, if any, which were used as sources of information for the test plan.

1.5 Definitions and Acronyms

This section is optional.

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

2.0 SYSTEM OVERVIEW

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.
There exists information about CSUN in terms of Catalogs, Majors, and Professors that are stored in various schemas across a database in MySQL and JSON documents that are handled and served by Python to a ReactJS built frontend as requested by the user.

3.0 SYSTEM ARCHITECTURE

3.1 Architectural Design

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don't go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.
Sample text:
Each Computer Software Component (CSC) in the *XYSW* CSCI Architectural Design section is described in more detail in the paragraphs under Section 4.1. The top level architecture is as shown in Figure 2.

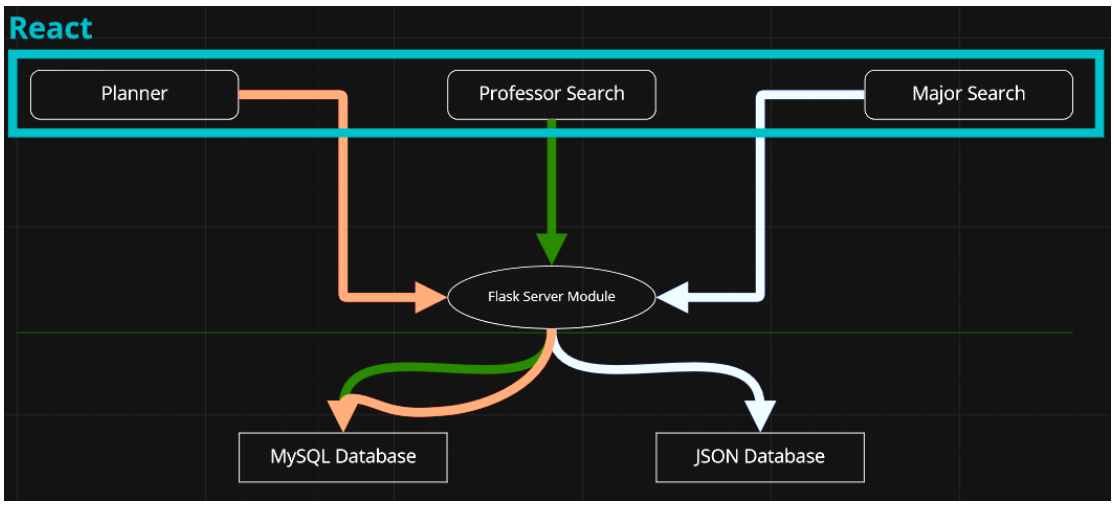


Figure 3.1: Modular System Design

The colors indicate what database each tool is pulling from.
The middle man between the client and server is the Flask Server Module that facilitates communication between the two. The planner and 2 search tools send GET requests to the server module which then accesses the appropriate databases for the information requested.

3.2 Interface Design

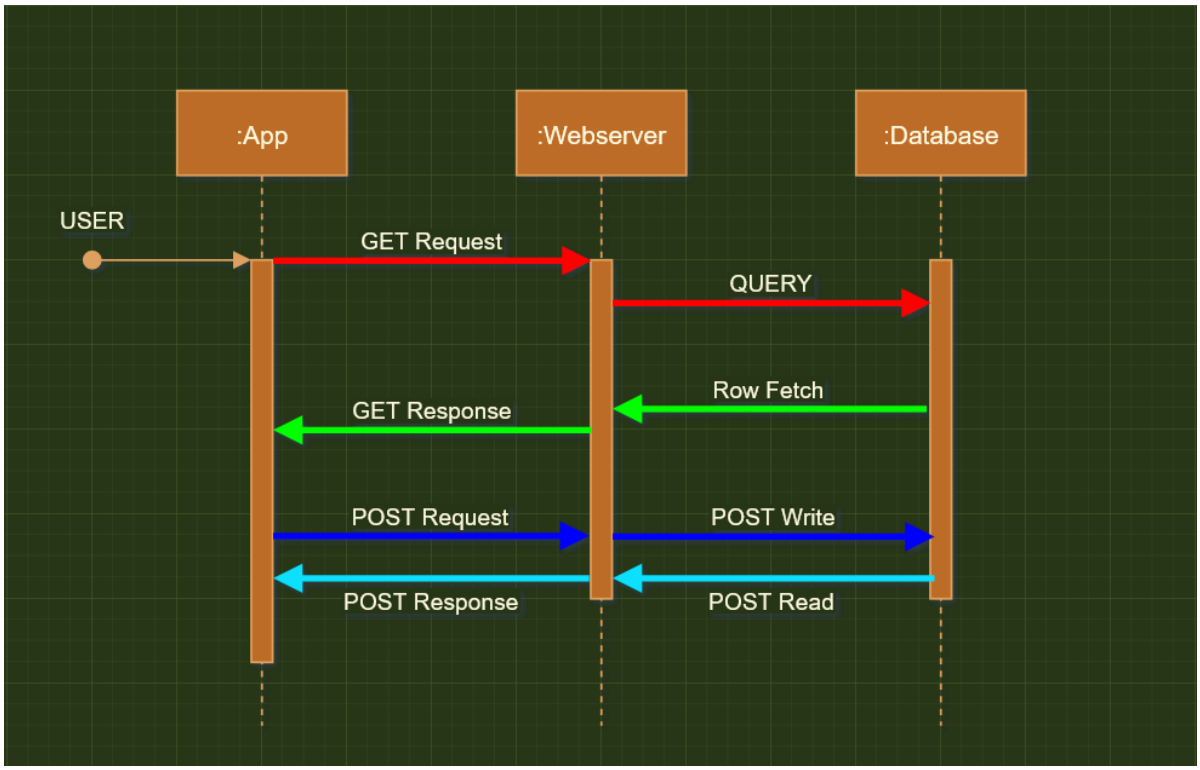
Provide textual or image/picture of how the CSCIs interface/relate to each other

3.3 Decomposition Description

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object oriented (OO) description. For a functional description, put a top level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object

diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

Reference the SDD_WebGLGame.pdf in Files (Canvas)



Having a middle man between the databases and the client was important for the simplification of client-side operations regarding deserialization of data received from the databases. This reduces the resource usage both client side and server side as the server module is configured such a way to not send unneeded data.

3.4 Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

4.0 COMPONENT DESIGN/DETAILED DESIGN

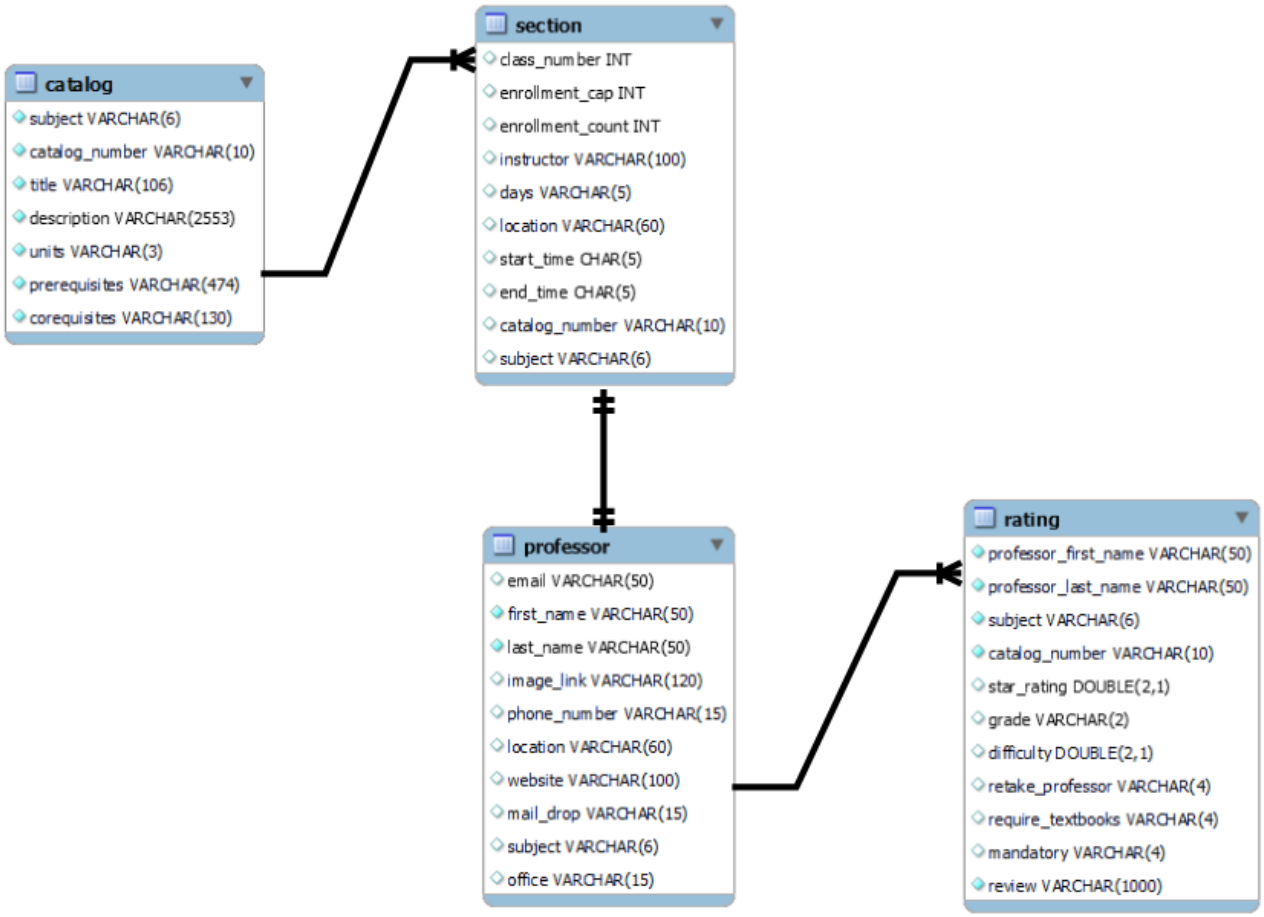
In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary. Also describe interface characteristics of

software units here as well as software units that are databases.

Sample text:
The purpose of each software component, its development status, and planned utilization of computer hardware resources is discussed in this section.
The *XYSW* consists of a single software CSCI. This CSCI handles the Guidance, Navigation, and Control (GNC) for the vehicle through all flight phases, as well as the mode control tasks. The software executes *whatever it executes* and flies *whatever it flies*.
This CSCI consists of two major CSCs: GNC and Mode Control. The GNC CSC contains three Computer Software Units (CSUs) (Navigation, Guidance Mode, and Autopilot). The mode control CSC contains CSUs which handle the executive, communications to external devices, task scheduler, Input and Output (I/O), and Built-In-Test (BIT). These CSUs are divided to provide a Modular Open Systems Approach (MOSA) to the software architecture.

4.1 Class Diagrams

This is where you put your UML/Class diagrams for your application, put your FULL class diagram here:



4.1.1 Sequence Diagram of the applicable Base Class (4.1, 4.2, 4.3, etc)

4.1.2 Database Schemas

- MySQL was used to store all subject catalogs and professor information
    - For every subject in the catalog schema, views were created on a per subject basis
- **Schemas**
    - **Professor Schema (Name: DATATYPE)**
        - **Email: VARCHAR(50)**
        - **First_name: VARCHAR(50)**
        - **Last_name: VARCHAR(50)**
        - **Image_link: VARCHAR(120)**
        - **Phone_number: VARCHAR(15)**
        - **Location: VARCHAR(60)**
        - **Website: VARCHAR(100)**
        - **Mail_drop: VARCHAR(15)**
        - **Subject: VARCHAR(6)**
        - **Office: VARCHAR(15)**
    - **Rating Schema (Name: DATATYPE)**
        - **Professor_First_Name: VARCHAR(50)**
        - **Professor_Last_Name: VARCHAR(50)**
        - **Subject: VARCHAR(6)**
        - **Catalog_number: VARCHAR(10)**
        - **Star_rating: DOUBLE(2,1)**
        - **Grade: VARCHAR(2)**
        - **Difficulty: DOUBLE(2,1)**
        - **Retake_Professor: VARCHAR(4)**
        - **Require_Textbooks: VARCHAR(4)**
        - **Mandatory: VARCHAR(4)**
        - **Review: VARCHAR(1000)**
    - **Catalog Schema (Name: DATATYPE)**
        - **Subject: VARCHAR(6)**
        - **Catalog_Number: VARCHAR(10)**
        - **Title: VARCHAR(106)**
        - **Description: VARCHAR(2553)**
        - **Units: VARCHAR(3)**
        - **Prerequisites: VARCHAR(474)**
        - **Corequisites: VARCHAR(130)**
    - **Section Schema (Name: DATATYPE)**
        - **Class_Number: INT**
        - **Enrollment_Cap: INT**
        - **Enrollment_Count: INT**
        - **Instructor: VARCHAR(100)**
        - **Days: VARCHAR(5)**
        - **Location: VARCHAR(60)**
        - **Start_time: CHAR(5)**
        - **End_time: CHAR(5)**
        - **Catalog_number: VARCHAR(10)**
        - **Subject: VARCHAR(6)**

- JSON Documents were used to store the Major descriptions. They are just strings of explanation about the major stored in an array that is meant to be fetched on load and organized according to what the major is.

5.0 User Interface

5.1 Overview of User Interface

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

Each client side module can be imagined as a Domain-Specific search engine.

- In the planner module, the parameters for every search are as follows
    - Semester (Already present dropdown)
    - Subject (Already present dropdown)
    - Which then displays a list of all the {Subject} courses scheduled in the {Semester}
- In the Professor Search, the parameters for every search are as follows
    - Subject (Already present dropdown)
    - Professor (Dropdown that loads on {Subject} choice)

5.2 Screen Images

Display screenshots showing the interface from the user's perspective. These can be hand drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

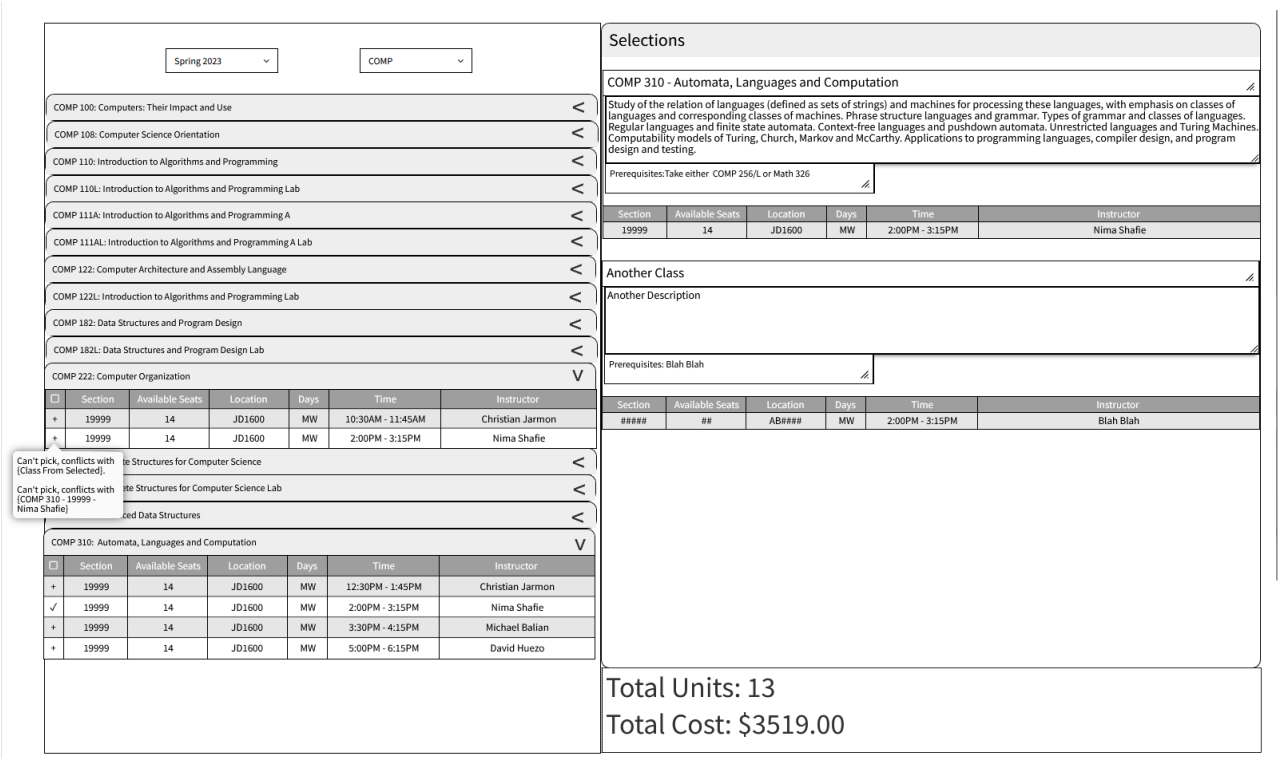


Figure 6.1: Planner Page Wireframe

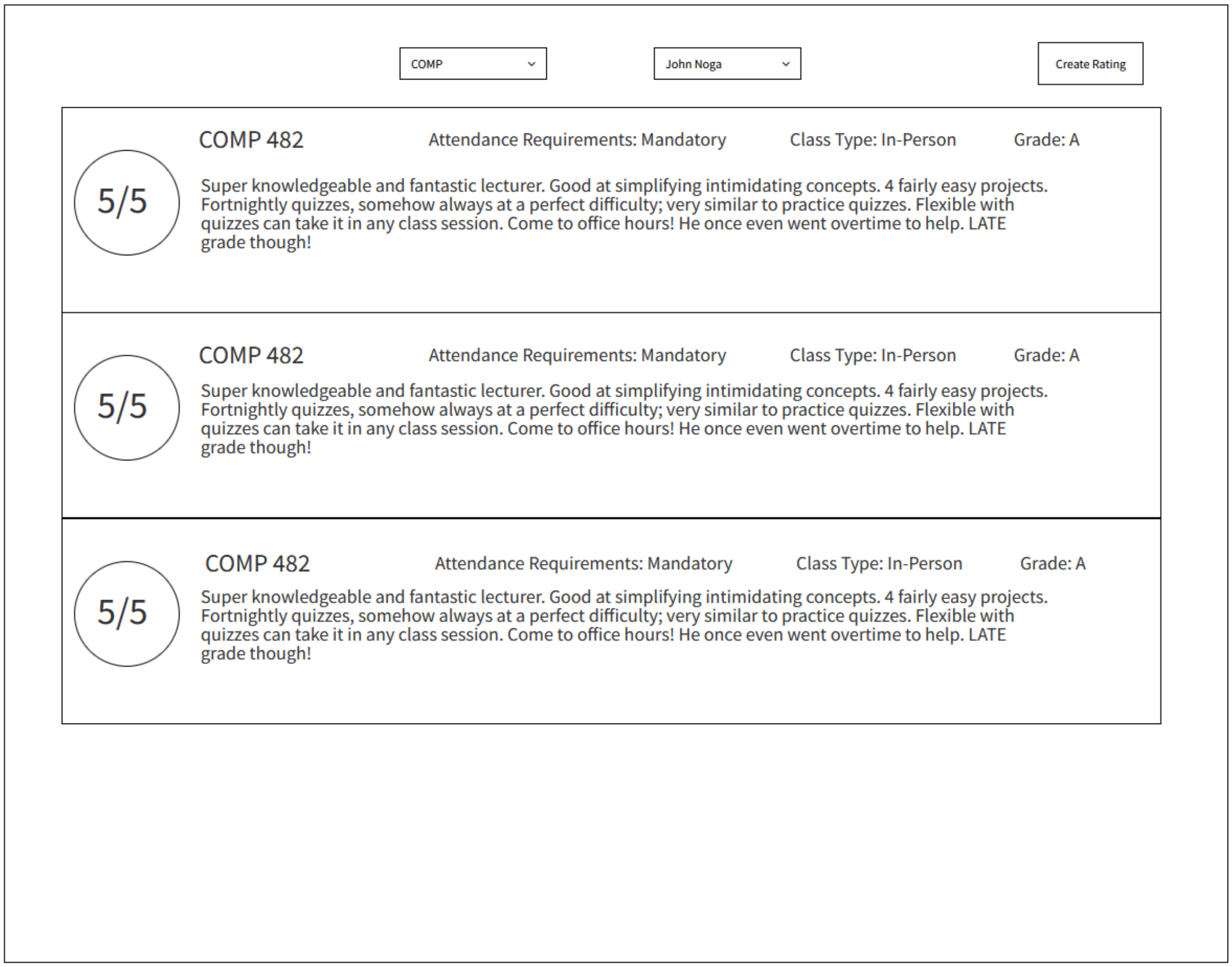


Figure 6.2: Professor Search and Ratings wireframe

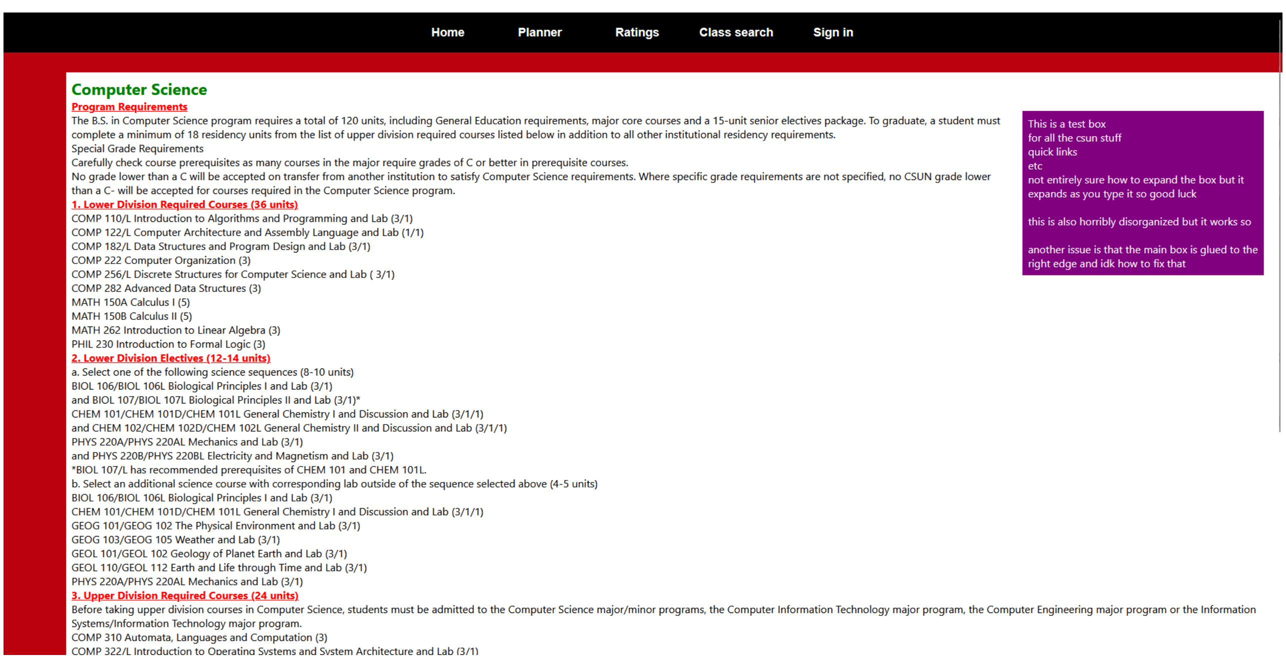


Figure 6.3: Computer Science Major page Rough Draft

.

SRS Req. ID	Satisfied (Yes/No)	Satisfaction Component/Proof
FUNC_SRS_(1.0)	**Yes**	Client Side layer in Figure 3.1
FUNC_SRS_(2.0)	**Yes**	Figure 6.1
FUNC_SRS_(2.1)	**Yes**	Figure 6.1
FUNC_SRS_(2.2)	**Yes**	Figure 6.1
FUNC_SRS_(2.3)	**Yes**	Figure 6.1
FUNC_SRS_(2.4)	**Yes**	Figure 6.1
FUNC_SRS_(3.0)	**Yes**	Figure 6.2, the dropdown is not shown, but should be
FUNC_SRS_(3.1)	**Yes**	Figure 6.2
FUNC_SRS_(3.2)	**No**	*Sample in progress
FUNC_SRS_(4.0)	**No**	Figure 6.3

APPENDICES

This section is optional.

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.