BeachCS

Design Specifications Document

Authors:	Parth Chhasatiya, Shujoy Islam, Aaron Ramirez, Rios Rios
Date:	May 8, 2021
Version:	3.0

Version 1.0	Date: October 20, 2020
Version 2.0	Date: March 12, 2021
Version 3.0	Date: May 8, 2021
	Summary of changes: 1. Updated Amendment 4: Sitemaps 2. Added Amendment 5: Technology Used 3. Added Amendment 6: Behavior Specifications - View Roadmap 4. Added Amendment 7: Architecture Specifications - Class Diagram 5. Added Amendment 8: Behavior Specifications - Add Course

Table of Contents

Table of Contents 2

Executive Abstract 3

Abstract 3

Terminology 4

References 5

Technology Used 5

Behavior Specifications 5

Feature #1 View Course 5

Feature #2 View Road Map 7

Feature #3 View Student Affairs 8

Feature #4 Add Course 9

Architecture Specifications 11

Class Diagram 11

Site Maps 12

Appendix 13

Amendment 1: Technology Used 13

Amendment 2: Behavior Specifications - View Courses 13

Amendment 3: Behavior Specifications - View Road Map 13

Amendment 4: Site Maps 13

Amendment 5: Technology Used 13

Amendment 6: Behavior Specifications - View Road Map 14

Amendment 7: Architecture Specifications - Class Diagram 14

Amendment 8: Behavior Specifications - Add Course 15

Executive Abstract

Abstract

This document lays out the software architecture and design decisions for the implementation of CSULB CECS WIKI. The target audience of this document is the database and website developers and designers, and the quality assurance team of CSULB CECS WIKI. Creating a dynamic web application that attempts to collect all information relevant to being a student studying Computer Science at California State University Long Beach requires an in-depth software development and design process. The designers will be the foundation of this program and will lay out detailed schematics for the creation of this website. Post-creation, the developers will work hand-in-hand with administration (CSULB faculty) to provide students (people who are studying CS at CSULB) a beneficial guide to becoming a successful graduate at CSULB. Furthermore, there will only be one major release of the software by May 2021. Prior to this release, the development team will work tirelessly throughout the development process to find and fix bugs and ensure the web application functions as expected via unit testing. Post-release, the developers will work closely with administration to dynamically update the web application as CSULB changes requirements related to their CS department. In-line with the above idea, the following principles will guide the design of CSULB CECS WIKI.

Components of the project are integral parts of the program and changes applied to them should have no effect on associated components. Developers must be able to make changes to parts of the program without affecting the work being done by other developers in order to maintain a stable project during the development process. Because of this, all major functions/features must be kept independently within their own respective private classes and databases. The application takes in course search requests in real time and consists of the system constantly requesting information from the database so it requires optimized back-end performance. Consequently, our server run-time should be consistent and latency-free. These principles will provide our users with high-quality service and experience.

Terminology

Technical terms

- User Interface(UI)
    - The menu that allows the user to search,select, and navigate through the website . UI is responsible for displaying class menu, searching, and changing settings in order to provide the user with a smooth experience
- Client
    - A computer that connects to and uses the resources of a remote computer.
- Server
    - System that provides resources, data, services or programs to other clients.
- JavaScript
    - Programming language that is commonly used in the web development client side.
- Node.js (Node)
    - Open source development platform for executing JavaScript code server-side.
- Express.js
    - Back end web application framework for Node.js
- React.js
    - Another open source development platform for executing front end JavaScript code.
- Hypertext Markup Language
        - Programming language that will be used to create web pages.

Non-Technical terms

- Student
    - The entity that voluntarily accepts the conditions and restraints that come within the website.
- Programmer
    - The developer that makes the website function as expected and responsible for a smoother UI
- Administration
    - The university faculty members that have permission to add and remove courses from the website

Abbreviations

- CSULB
    - California State University Long Beach
- CS
    - Computer Science
- CECS

    - College of Engineering and Computer Science
- HTML
    - Hypertext Markup Language
- CSS
    - Cascading Style Sheets
- AWS
    - Amazon web services

References

The document linked below outlines the requirement specifications of CSULB CECS WIKI:

CSULB CECS WIKI Requirement Specification Document

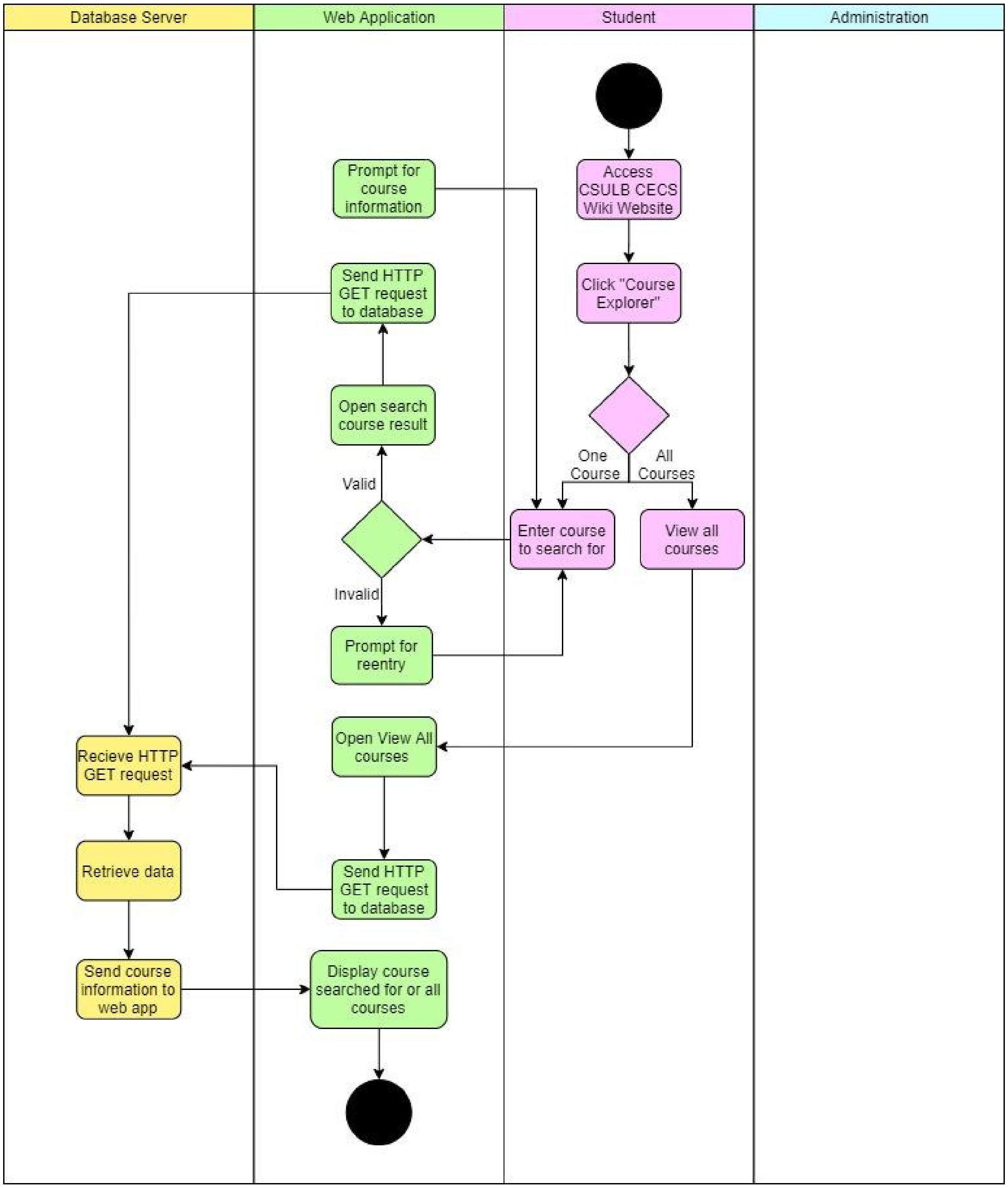
Technology Used

The web application will be developed with Node.js and React.js. The source code for the web application will consist of HTML/CSS for the front-end web development, and Express for the back-end web application. We will use cloud services like AWS Database for our application.

Behavior Specifications

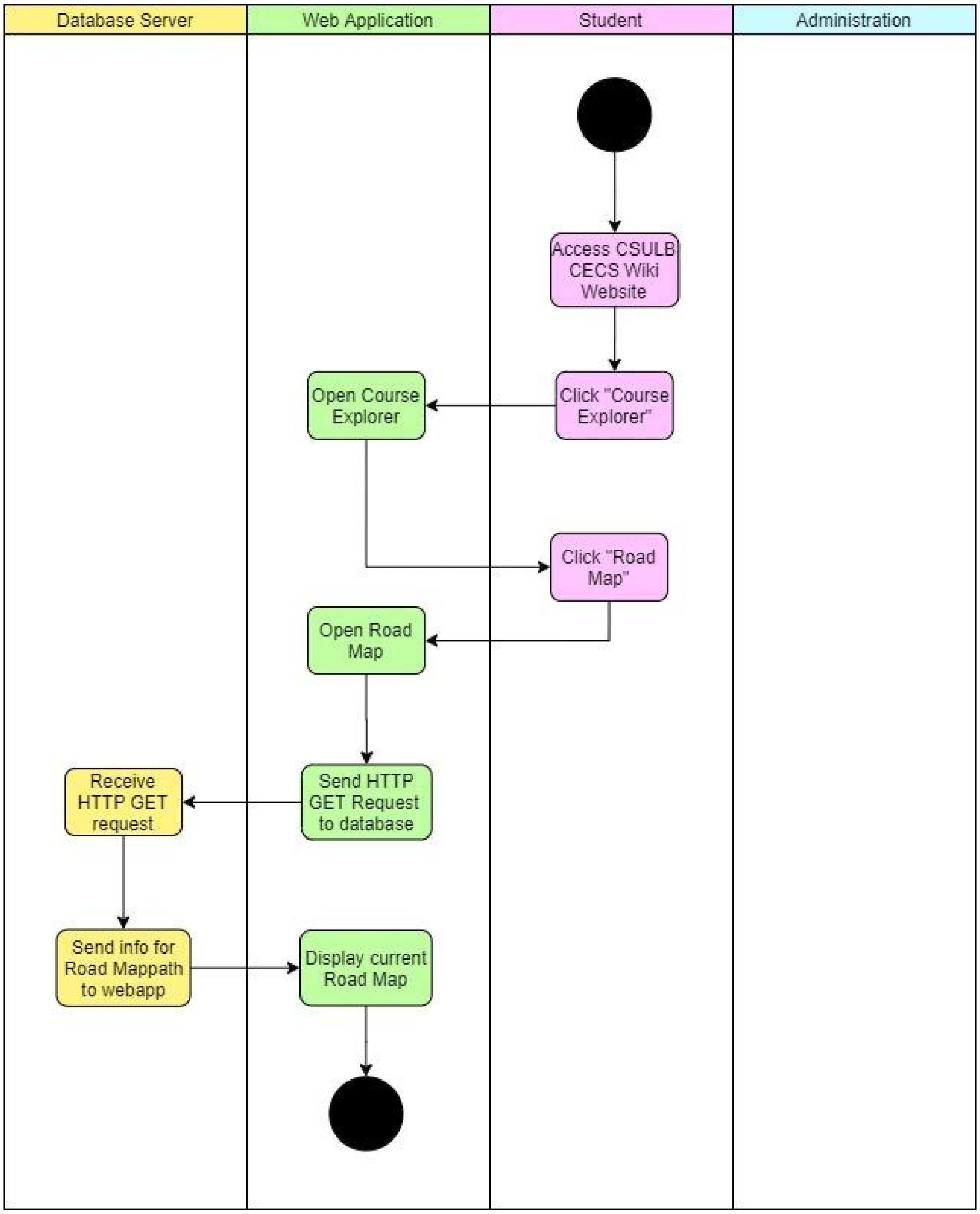
Feature #1 View Course

One of the key features is a view of important information regarding a course. Users will be able to search for a course within the CSULB Computer Science department, if the course is not in our database then an error message will be displayed to the user and will be able to search for another course . If the course is in our database, our website will display course name, course id, pre requisites if any and a brief description of the subject matter and course outline.


Database Server
Web Application
Student
Administration
Access CSULB CECS Wiki Website
Click "Course Explorer"
One Course
All Courses
Enter course to search for
View all courses
Prompt for course information
Send HTTP GET request to database
Open search course result
Valid
Invalid
Prompt for reentry
Open View All courses
Send HTTP GET request to database
Display course searched for or all courses
Recieve HTTP GET request
Retrieve data
Send course information to web app

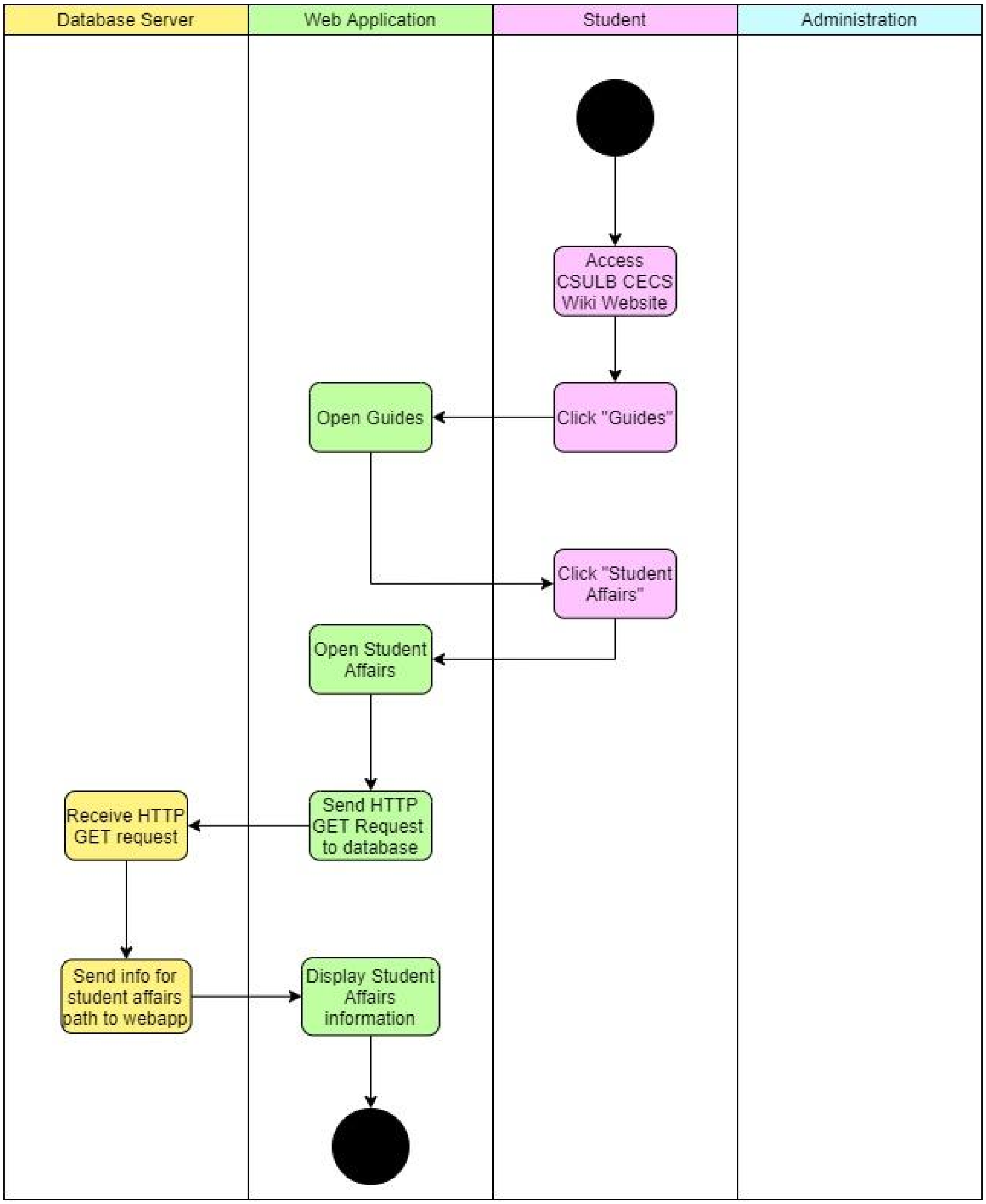
Feature #2 View Road Map

The website will allow users to view a road map displaying the courses required to attain a degree in Computer Science as well as a recommended order for taking these classes. This roadmap will give students an idea of which courses will serve as prerequisites for later courses, present students with instant information about specific courses by clicking on the course on the roadmap itself, and suggest courses for students to take in a given semester. The user will be able to access the roadmap from the home page by going to the course explorer, and then the Road Map page. The Road Map itself will be automatically displayed on the page, and the data presented will be based on current Computer Science degree information stored in the database.



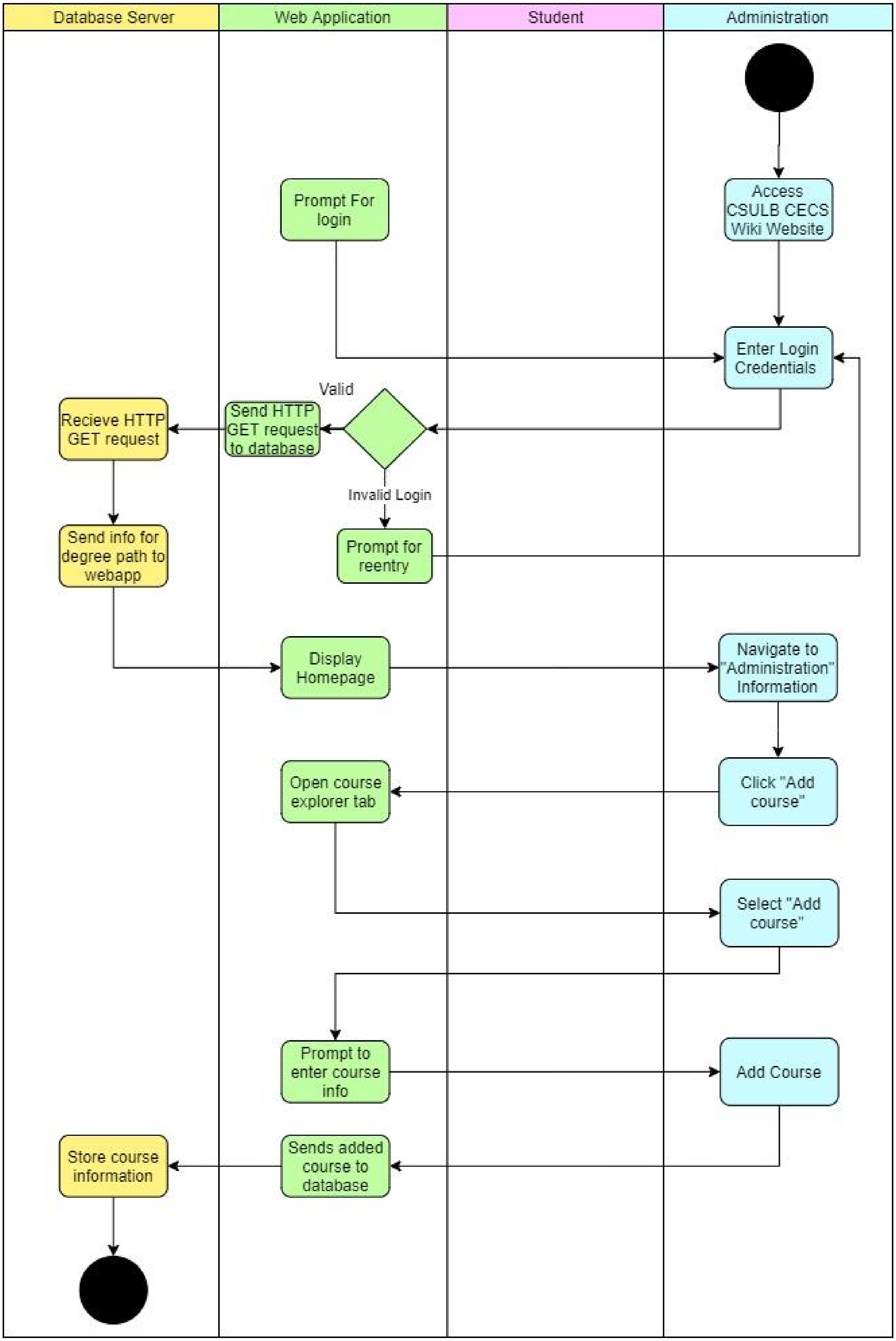
Feature #3 View Student Affairs

The web application will allow users to view a student affairs displaying computer science relative affairs for students. A student will be able to access the CSULB CECS Wiki home page. On the homepage they will have the ability of clicking on the guides icon, which will direct them to the guides web application within the website. Following the guides application will be on display on the web page, then they will be able to select the student affairs icon which will direct them to the student affairs web application within the website. The student affairs will be displayed on the web page, and the data presented will be based on information gathered and stored in the database for current Computer Science students.



Feature #4 Add Course

The administration will be able to add a course to the course explorer in which then the student will search. The administration will then access the web application by logging into their accounts. Once their credentials were successfully accepted the web application will display the home page. The administration will navigate the home page of the web application, which they will have access to navigate the administration information. They will have the capability of adding a course to the course explorer. On the web application they will select an add course on the course explorer. On the web application the administrator will be prompted to enter the course information, which the admin will need to manually input the course information and send it to be stored into the database.

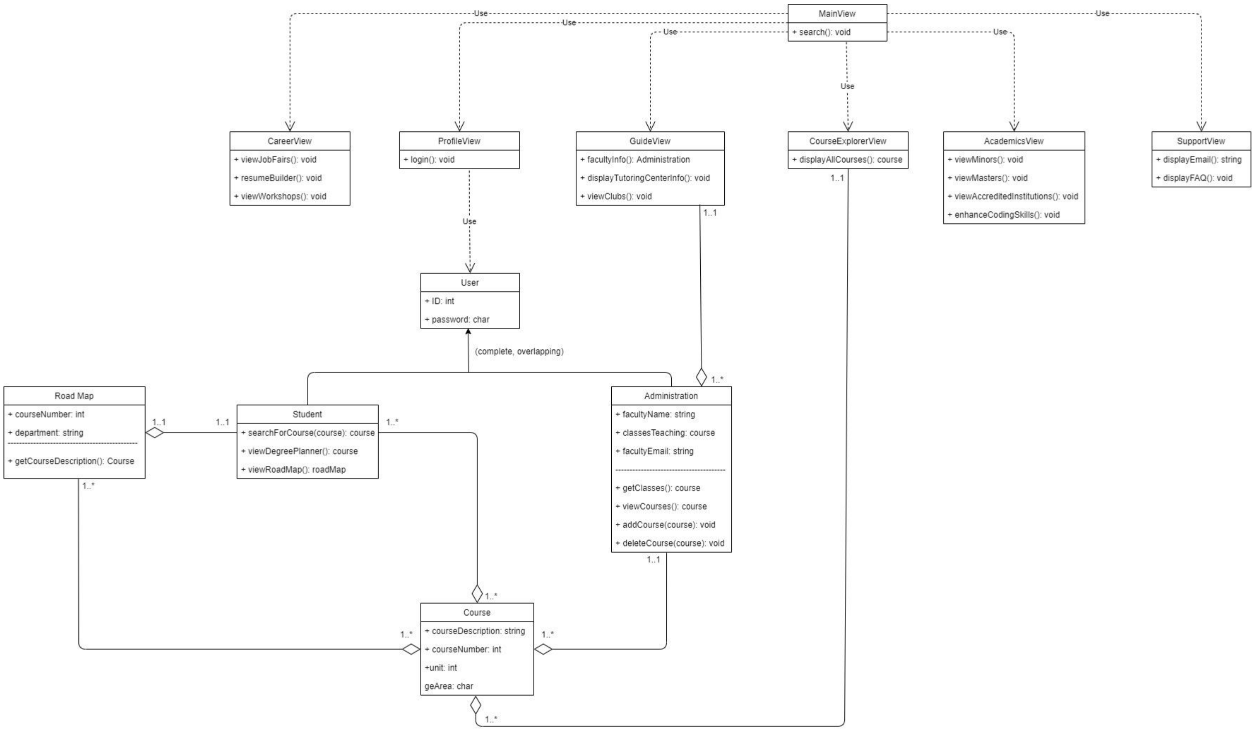

Database Server
Web Application
Student
Administration
Access CSULB CECS Wiki Website
Prompt For login
Enter Login Credentials
Valid
Recieve HTTP GET request
Send HTTP GET request to database
Invalid Login
Prompt for reentry
Send info for degree path to webapp
Display Homepage
Navigate to "Administration" Information
Click "Add course"
Open course explorer tab
Select "Add course"
Prompt to enter course info
Add Course
Store course information
Sends added course to database

Architecture Specifications

Class Diagram

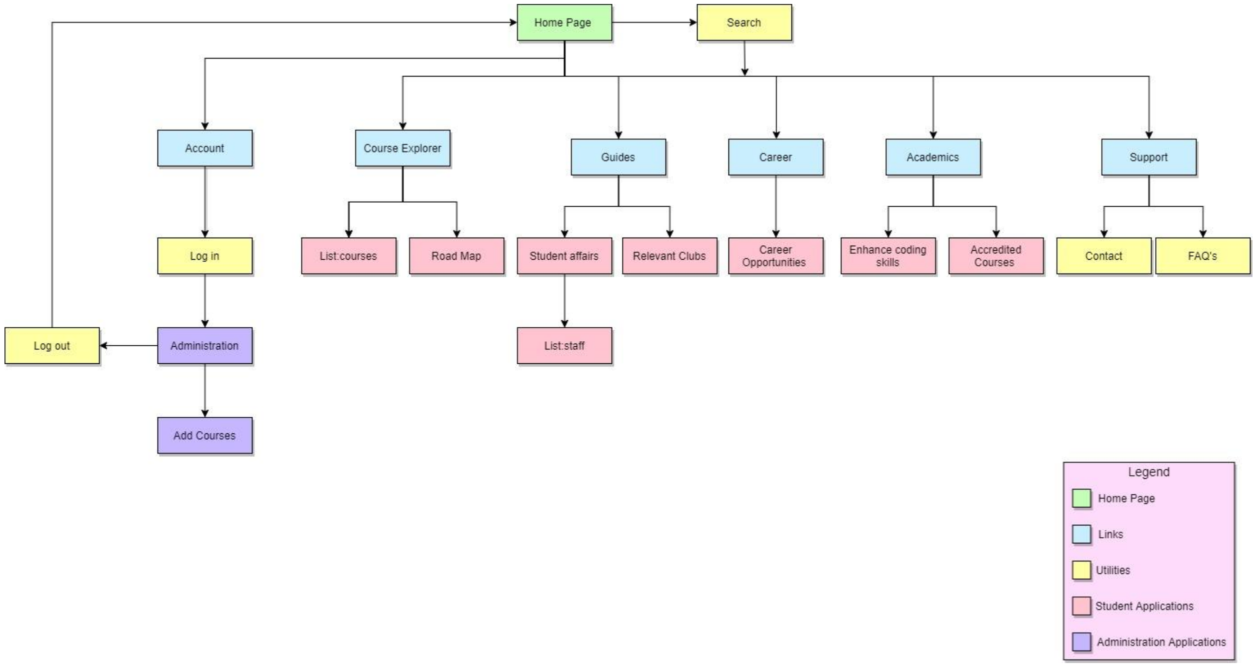
Below are the proposed UML CLass Diagrams for the CSULB CECS Wiki application. It details the structure of the classes needed to implement the web application. The diagram provides us with the relationship between the Model-View-Controller. The model is the central component of the pattern. It will directly manage the data, logic and rules of the application. The view obtains any representation of information in our case it will be used for the student application views. The controller itself will accept input and convert it to commands for the model or view. Linking the front-end to the back-end. Singleton pattern will be used for login credentials since we don’t want to create multiple instances of a single user all around our database that might cause log in errors and unnecessary use of memory. When creating a road map, the decorator pattern will be implemented, we will have our main class and through our decorator pattern we will add behaviors and states at runtime to our main class which in our case will be to add courses to link them together for our user to see his/hers created road map.

The MainView is the primary class from which all the other parts of the website are accessible. CareerView can then be used to track career growth specific information. ProfileView can be used to get more detailed information on a specific user’s profile. GuideView displays information on faculty and administration present for CSULB CECS. CourseExplorer is used to get detailed course information, including course ID, GE area covered, and course description. AcademicView gives degree information, including minors and Master’s degrees. Finally, SupportView assists users for any questions and technical support they require.



Site Maps

Below is the proposed web application map for CECS CSULB WIKI. It details the structure of the user interface, it’s client application and utilities, both student and Administration. The application is color coded depending on the functionality for either student or administration applications. The green element represents the initial home page of the web application. The pink element represents applications to link that students and administration have access. Yellow application represents utilities that both students and administration can use. Pale red are applications that students can access. Purple applications are used by administration applications.



Appendix

Amendment 1: Technology Used

- We no longer write HTML directly, but utilized React to create components

Amendment 2: Behavior Specifications - View Courses

- Activity diagram reflects that courses are now buttons that need to be clicked
- Two course cards are now present on the Course Explorer page

Amendment 3: Behavior Specifications - View Road Map

- Road map now also pulls from database
- Clicking a roadmap highlights courses you can take afterwards

Amendment 4: Site Maps

- Add Workfair Calendar
- Add "How to succeed on your job"
- Add Guide subpage 'Programming Guides'
- Add Guide subpage 'Ways to Study'
- Add Guide subpage 'Recommended IDE's'
- Add Guide subpage 'For Motivation'
- Add Career subpage 'Workshop & Workfair calendar'
- Add Career subpage 'Resume'
- Add Career subpage 'How to Succeed at your Job'
- Add Academics 'Applying to Graduate School (MS/PhD)
- Add Academics 'CECS Faculty'
- Add Academics 'Minor in Cyber Security Applications
- Add Academics 'Scheduling Classes'
- Add 'About'
- Add 'Contact'
- Add 'Feedback'
- Remove 'Account'
- Remove 'Login'
- Remove 'Logout'
- Remove 'Administration'
- Remove 'Add Courses'

Amendment 5: Technology Used

- We now use Amazon Elastic Cloud Computing (EC2) to host the website on the cloud

- We now use Amazon Relational Database Service (RDS) to store the MySQL database in the cloud

Amendment 6: Behavior Specifications - View Road Map

- Road map page now has a card that shows course and prerequisite information
- Clicking a course also sends data to the card and updates the card's state to display course information

Amendment 7: Architecture Specifications - Class Diagram

- Remove the following entities and their connections from the UML diagram:
    - ProfileView
    - User
    - Student
    - Administration
    - Student affairs
- Create the following entities:
    - ProgrammingLanguagesView
    - IDEsView
    - StudyView
    - MotivationView
    - InterviewQuestionsView
    - MastersView
    - MinorsView
    - FacultyView
    - SchedulerView
    - ResumeView
    - CalendarView
    - JobSuccessView
    - CareerFairView
- Update the name of the following entities:
    - SupportView → FeedbackView
- Create associations between the following entities:
    - CourseExplorerView → RoadMapView
    - AcademicsView → MastersView
    - AcademicsView → MinorsView
    - AcademicsView → FacultyView
    - AcademicsView → SchedulerView
    - CareerView → ResumeView
    - CareerView → CalendarView
    - CareerView → JobSuccessView
    - CareerView → CareerFairView

    - GuidesView → ProgrammingLanguagesView
    - GuidesView → IDEsView
    - GuidesView → StudyView
    - GuidesView → MotivationView
    - GuidesView → InterviewQuestionsView

Amendment 8: Behavior Specifications - Add Course

- Change this feature to ‘View Upcoming Events’
- The user can access the Calendar from the Career page and navigating to the ‘Work Fair Calendar’ page
- From here, the user can view the calendar in three different displays: by month, week, or day
- The calendar fetches data from the database as JSON objects, and displays the event in the corresponding day (if month or week view) or hour slot (if day view)
- The user can navigate from month to month, year to year, week to week, or day to day, and the database is updated regularly so that events are ensured to display in the calendar