Software Requirements Specification

for

4P02 Project (Course of Action Dashboard)

Prepared by Team Rocket:

Liam Yethon (6255384)
David Saldaña (6155964)
Jason Grightmire (5388327)
Katie Lee (6351696)
Jesse Masciarelli (6243109)
James Zhao (6240527)
Wenjie Lin (6135859)
Calvin Feng (6240790)

Brock University

February 7, 2021

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the system, Course of Action Dashboard (COAD), is to provide Brock University students with a user friendly platform and tool to make the course selection and scheduling process simpler. Currently, Brock University does not provide their students with a tool to help with planning out their course schedules. Students who struggle with tracking their progress towards their graduation requirements may experience stress, delayed graduation times, and wasted resources. COAD will solve these concerns by providing the students with a user friendly, course dashboard that they can rely on throughout their university career.

1.2 Product Scope

The system in its initial state is intended for use by undergraduate students in Brock University's Computer Science program. The system will provide users with an interactive undergraduate calendar/course scheduler/organizer. Students will be able to see their program details/requirements (i.e. required number of credits, required courses), an overview of courses that they've taken, a credit tracker, and etc. In short, we'd like to develop a user friendly dashboard and organizer which will allow students to easily plan their course selection and see all of their options, which Brock currently does not have a centralized location for.

2. Overall Description

2.1 Product Perspective

This product is to be a standalone product that can eventually be integrated into the Brock systems. The first iteration of this product will be a website that has no interactions with any part of Brock. Future versions will include a direct Brock login through Brock's current login portal system, and access to a dynamic list of courses pulled directly from Brock, which will allow the course listings to stay up to date without manual intervention.

2.2 Product Functions

- Allow users to view their credit overall status.
- Allow users to view how many courses they have planned or completed.
- Update and display the course schedule every time the user makes a change.
- Allow users to add, delete, and move course items on the schedule.
- Have various quick links to Brock official websites for users to read more detailed information when planning their course schedule.

2.3 User Classes and Characteristics

The most frequent user class will be undergraduate students enrolled in the Brock University Computer Science program. They will make use of all product functions including course manipulation and status tracking. This user class is required to be most satisfied with the system.

An infrequent user class will be Brock University academic advisors and other staff who are tasked with assisting students with scheduling concerns. They will make use of the course search and organization tools, without requiring the functionality of the status tracker.

2.4 Operating Environment

The system is intended to run on a web browser. The system will work on the following operating system: Windows, MacOS, and Linux. Recommended browsers: Google Chrome, Firefox, Microsoft Edge.

2.5 Design and Implementation Constraints

The primary constraint that may prove to be an obstacle is not having access to the Brock database for user logins or courses. If it is not possible to have access to these systems, the scope of the project will need to be restricted in order to compensate. The planner will need to be able to function without the user needing to be logged in, or a stand alone login system must be implemented and courses will need to be added in manually.

Another constraint related to the first concerns whether or not courses are to be offered in a given semester. If a course is no longer offered for a given semester but is not revealed to the students ahead of time, then the users could potentially plan their courses around the one that is now invalid.

A minor constraint for this product is the maintenance required to keep courses up to date. Without access to the Brock database, a lot of work will be required if there are multiple changes to the course descriptions.

2.6 Assumptions and Dependencies

Assumptions:

- Since the application is a web based application there is a need for the internet browser. It will be assumed that the users will possess active internet connectivity.

Dependencies:

- Amazon AWS - a free relational database system that COAD will initially depend on to manage student and course data.
- Course listings/credit offerings provided by Brock University's website - course listings and credit offerings that COAD will pull its searches from.

3. Functional Requirements

3.1 High Priority

- The system shall allow users to view their credit overall status.
- The system shall allow users to view how many courses that they have planned and how many courses that they have completed.
- The system shall display the course schedule that users have planned.
- The system shall allow users to add, delete, and move course items on the schedule.
- The system shall have quick links to Brock official websites so that users can read more detailed information when planning their course schedule.

3.2 Medium Priority

- The system shall allow users to view description and prerequisites of courses.
- The system shall allow users to download their planned schedule as a PDF file.

3.3 Low Priority

- The system shall allow the user's status to be stored for the next time he returns to the web site. This will save the user x minutes per visit by not having to re-enter already supplied data.
- The system shall notify users at the beginning of each semester so they can update their schedules if needed.

4. Non-Functional Requirements

4.1 Reliability

- The system shall be completely operational at least 90%+ of the time.
- Down time after a failure shall not exceed 5 minutes.

4.2 Usability

- A user should be able to figure out how to use the system by himself/herself.
- Courses list should include all courses provided by Brock University.
- The system should warn users when certain planning is not feasible, such as planning 10 courses in one semester.

4.3 Performance

- The system should be able to support at least 200 simultaneous users.

- The mean time to view a web page over a 56Kbps modem connection shall not exceed 20 seconds.
- The mean time to download and view the course schedule in PDF format for a 56Kbps modem connection shall not exceed 20 seconds.

4.4 Security

- The system shall provide password protected access to web pages that are to be viewed only by users themselves.
- The system shall filter text input into any text entry fields.

4.5 Supportability

- The system should be able to accommodate changes of available courses without major reengineering.
- The system web site shall be viewable from Internet Explorer 4.0 or later, Netscape Navigator/Communicator 3.0 or later and the America Online web browser version 3.0 or later.

4.6 Online User Documentation and Help

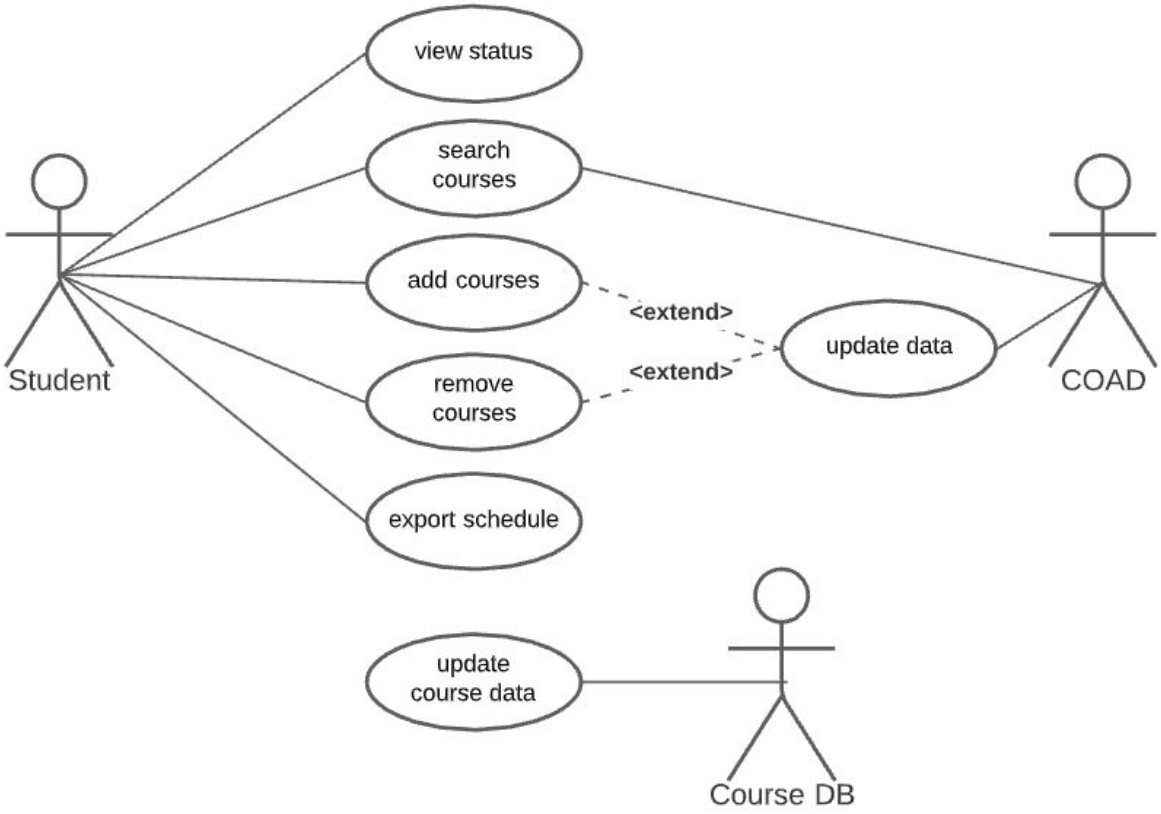
- The system shall provide a web page that explains how to navigate the site. This page should be customized based on what pages that user is allowed to access.
- This help page should be accessible from all other pages.

4.7 Interfaces

- Components of different functionalities should be colored differently.

5. Use Case Diagram

5.1 Diagram

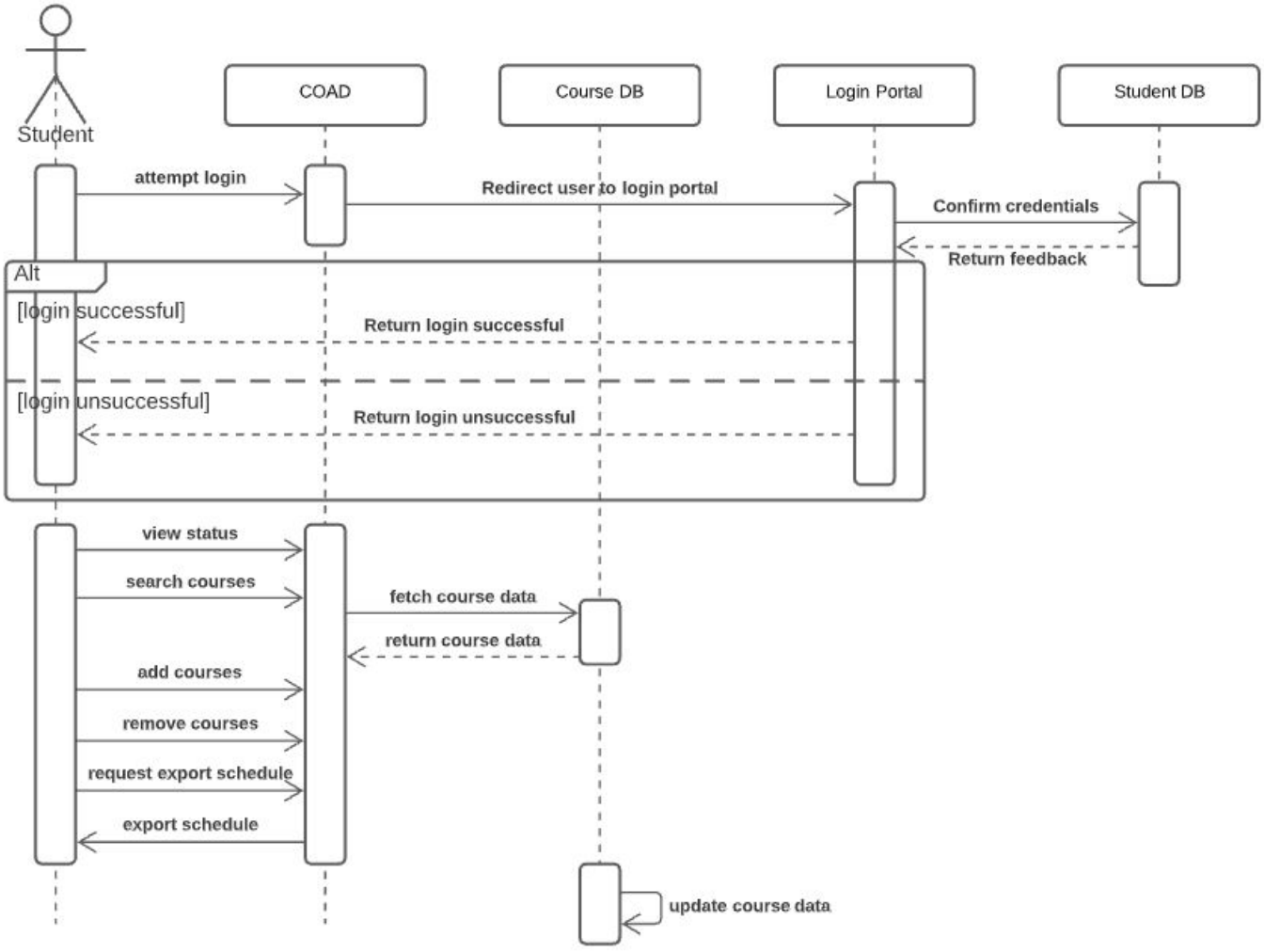


5.2 Diagram Notes

In order for the student to perform any of the listed use cases, they must first successfully log-in to the COAD system. COAD fetches data from the course database at regular intervals to ensure that scheduled and searched courses are up-to-date.

6. Sequence Diagram

6.1 Diagram

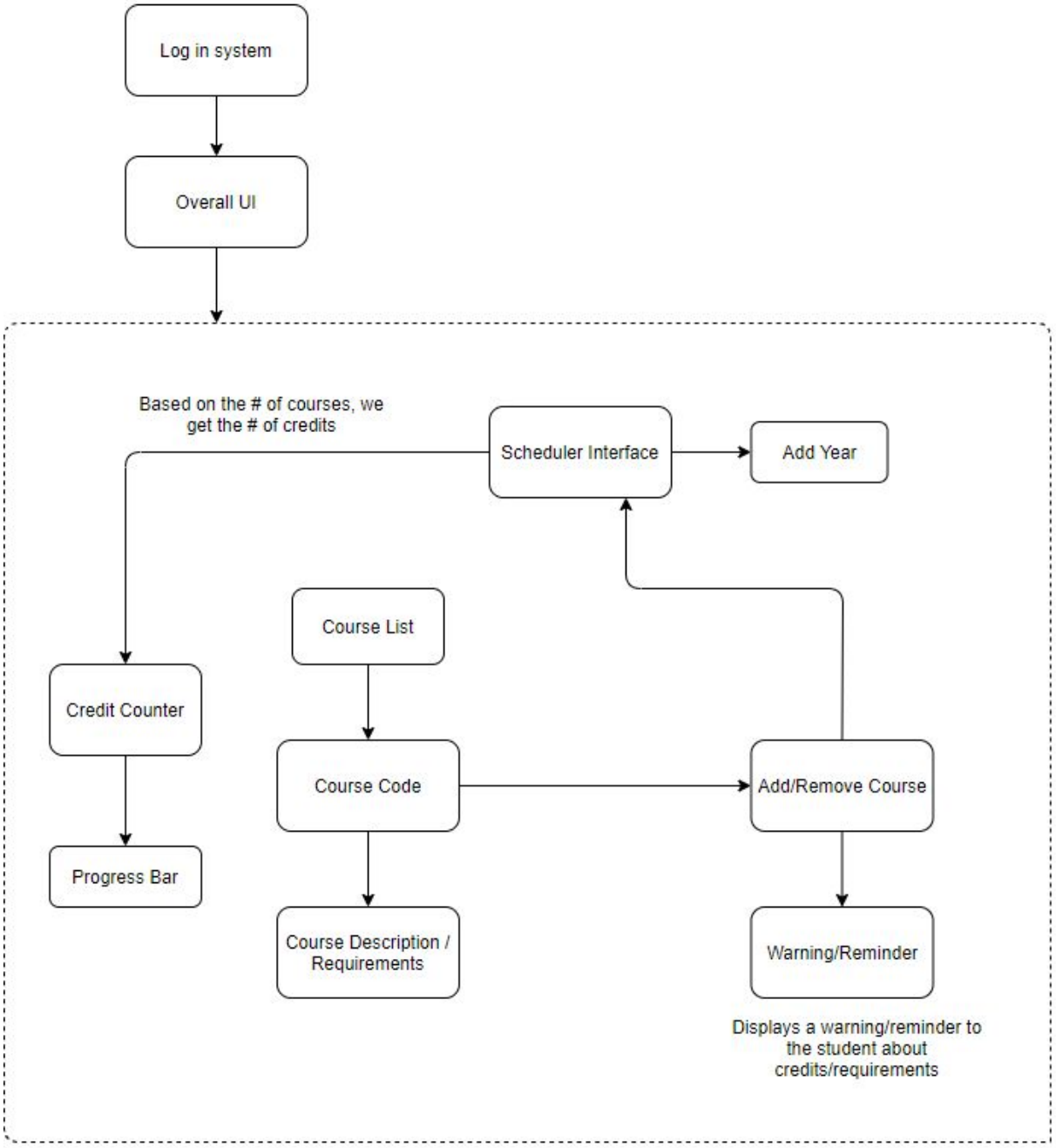


6.2 Diagram Notes

An unsuccessful log-in attempt will not allow the student to interact with the COAD system in any way other than another log-in attempt. Each of the following actions are independent of one another: view status, search courses, add courses, remove courses, and request export schedule.

7. System Architecture Diagrams

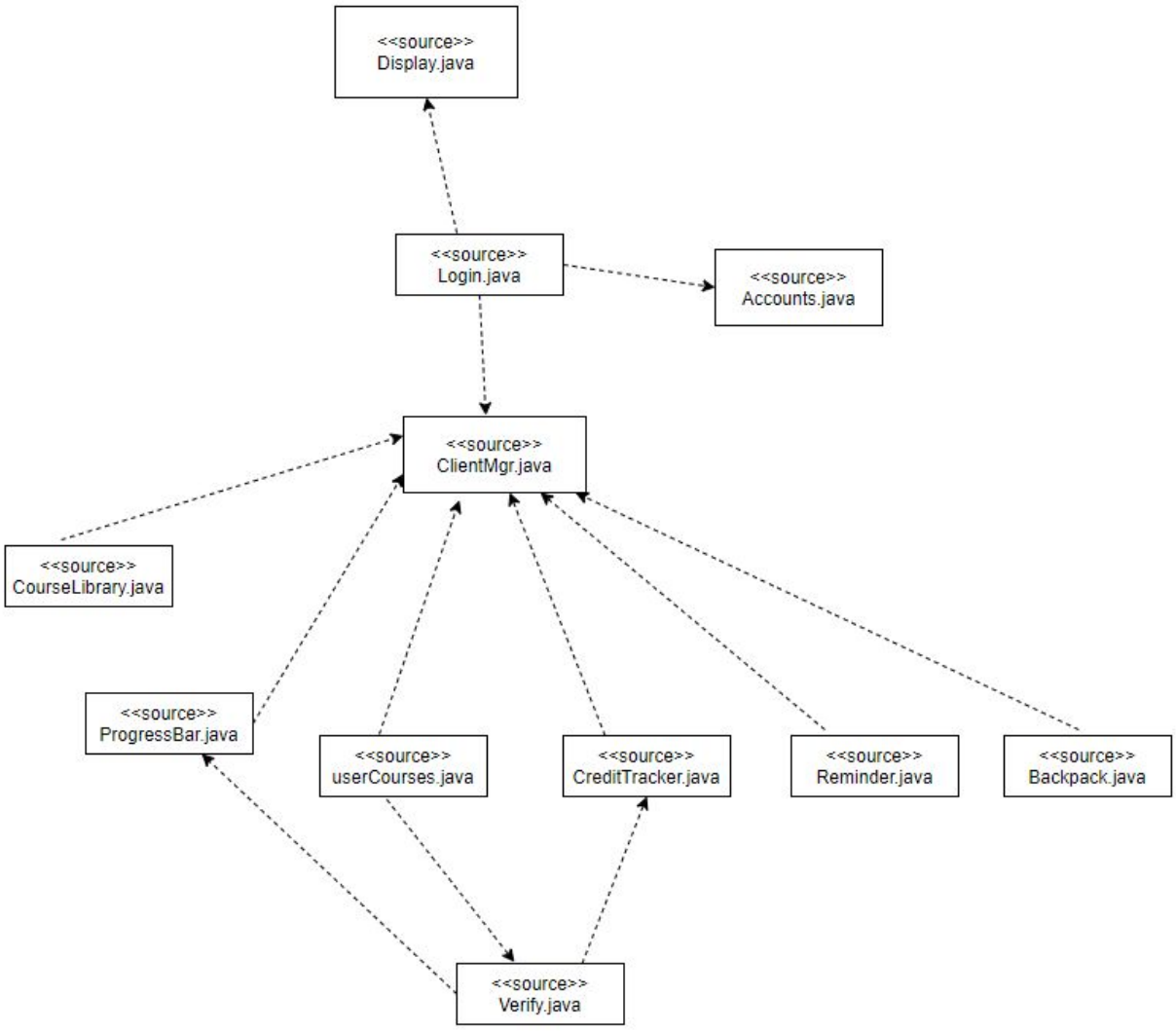
7.1 Logical View



7.2 Logical View Explanation

The purpose of this diagram is to show how each component of the system is connected to each other. Depending on the action, other components of the system will come into play. For example, the course list box connects to the course code box since that function will provide a list of course codes.

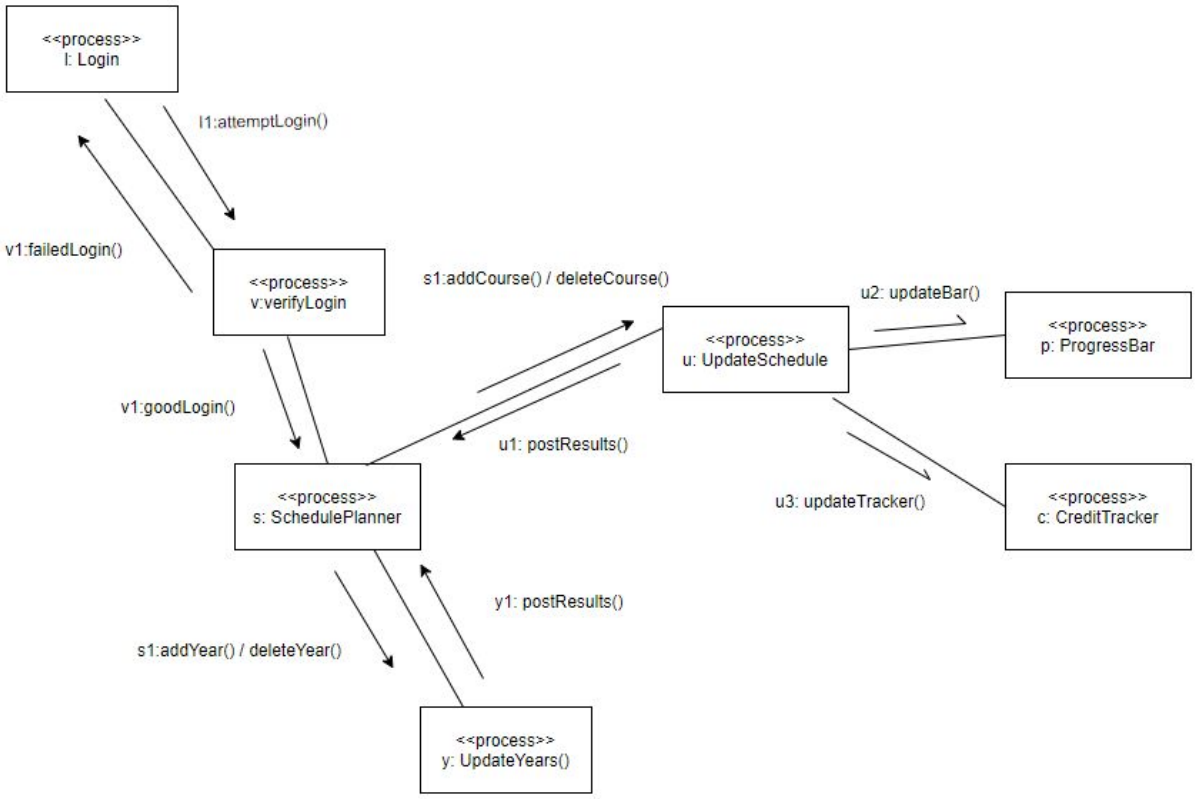
7.3 Development View



7.4 Development View Explanation

The purpose of this diagram is to show the organization of the software modules in relation to each other. The arrows represent which file is imported into which. For example, the CourseLibrary.java is imported into the ClientMgr.java.

7.5 Process View



7.6 Process View Explanation

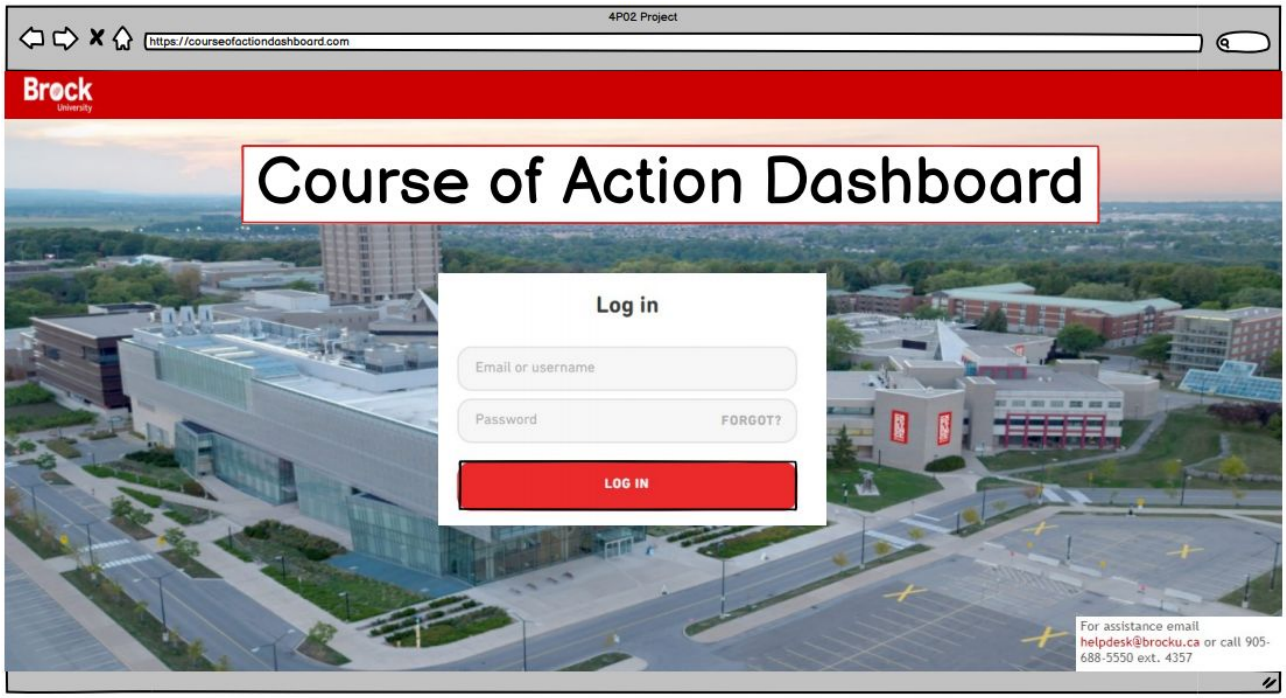
The purpose of this diagram is to show the processes and threads and their interactions when forming the concurrency and synchronization mechanics within the system. The arrows are represented below as:

Synchronous communication 

Asynchronous communication 

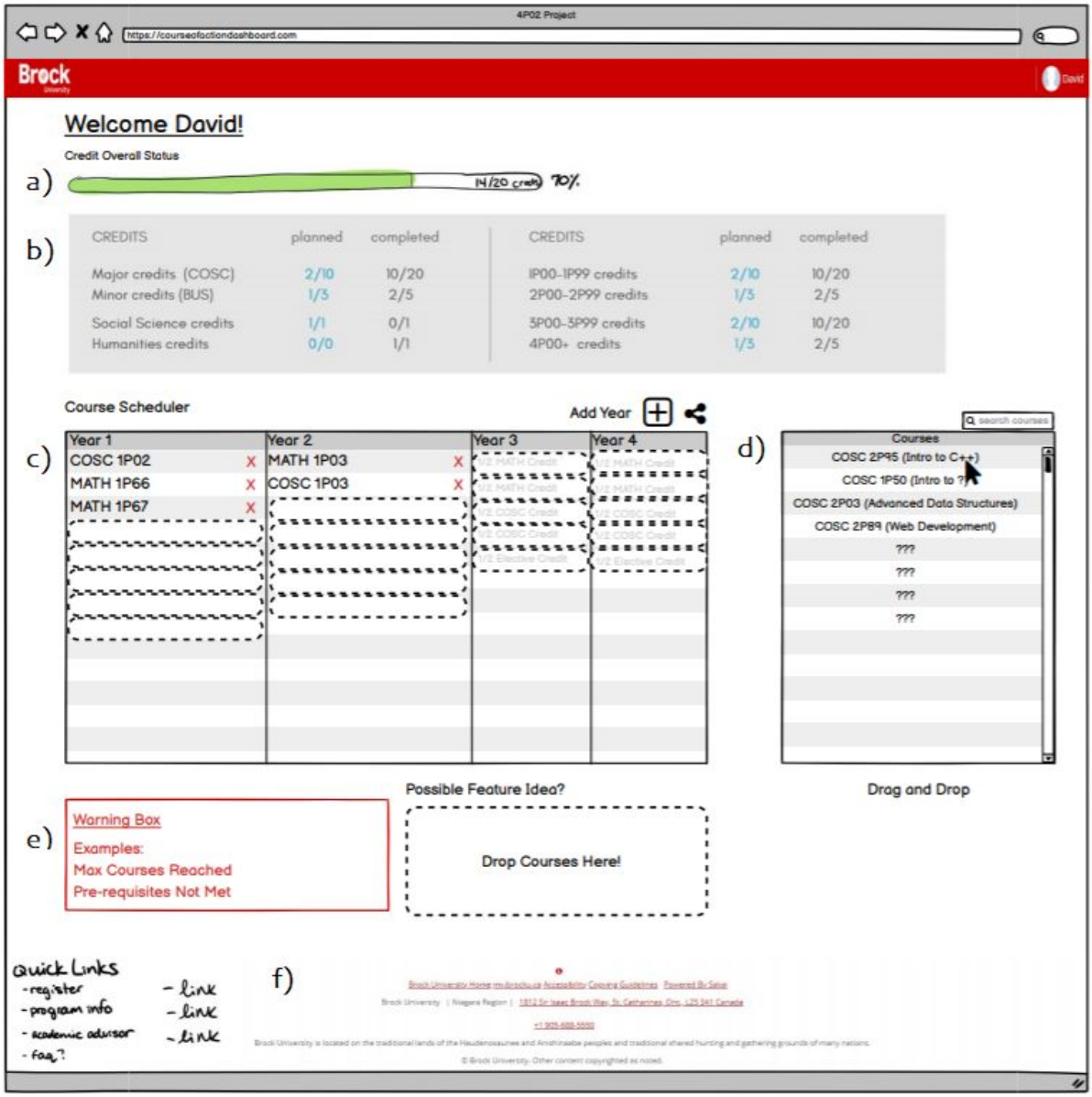
8. UI Design

8.1 Log-in Page



a) Landing log-in page where the user will use their Brock credentials to gain access to the Course of Action Dashboard.

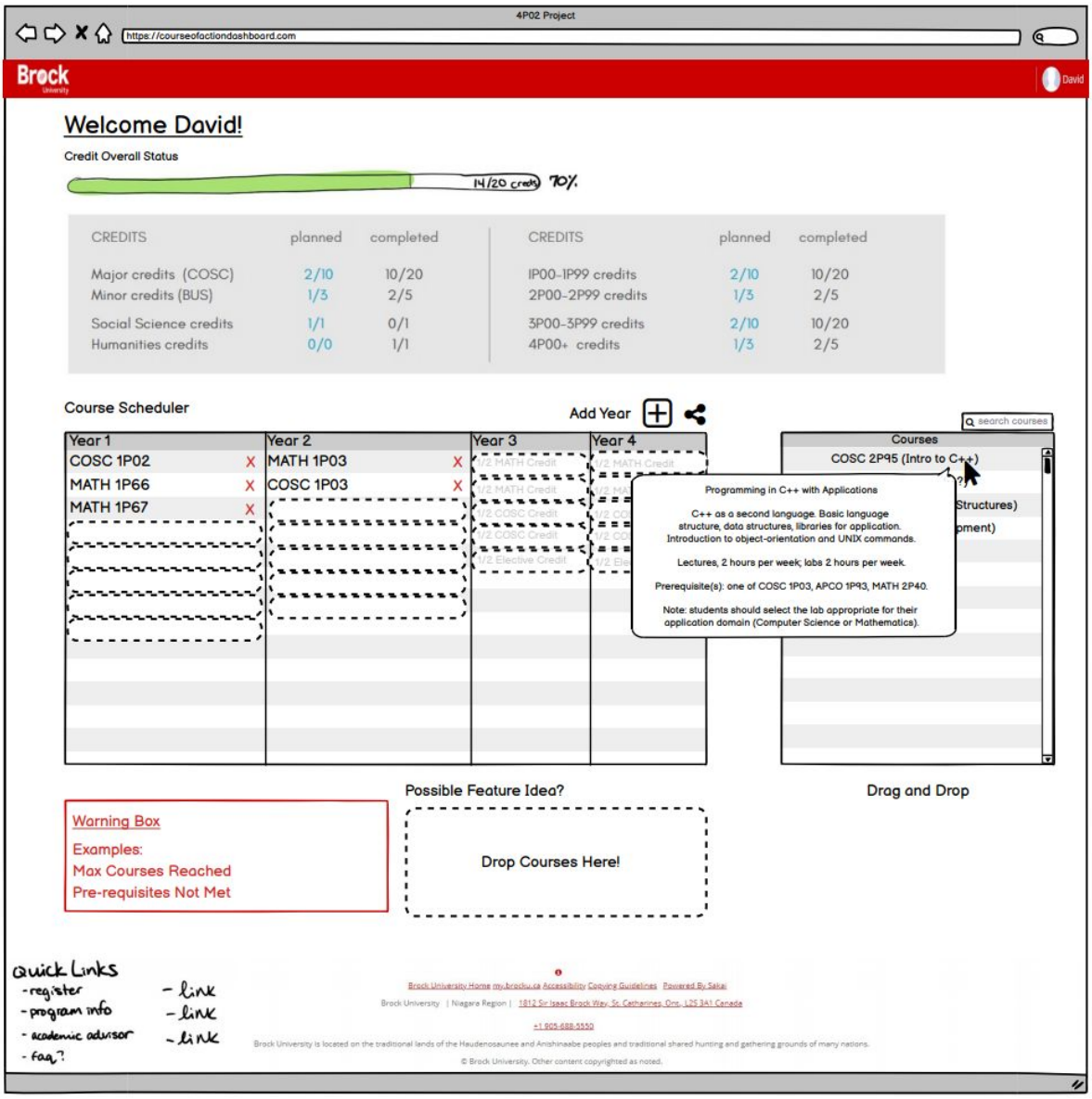
8.2 Main Schedule Page



a) Credit progress bar: describes the total number of overall credits that the student has taken out of the total number of credits that are required to graduate.
b) Credit overview: describes how many credits of each credit type the student has completed. Planned credits refers to the expected number of credits that the student is planning to take based on their course scheduler (part c).
c) Course scheduler: the interactive tool that enables the student to flexibly plan out their future course schedules. Features include: adding/removing/re-arranging courses and selecting elective credits.
d) Course listings: students may search and select from a list of elective credits.

e) Warning box: alerts the user of any issues regarding their course schedule (i.e. too many credits are planned for a semester)
f) Quick links: external resources the user may want to access (i.e. academic advisor quick links, registration links, program information links)

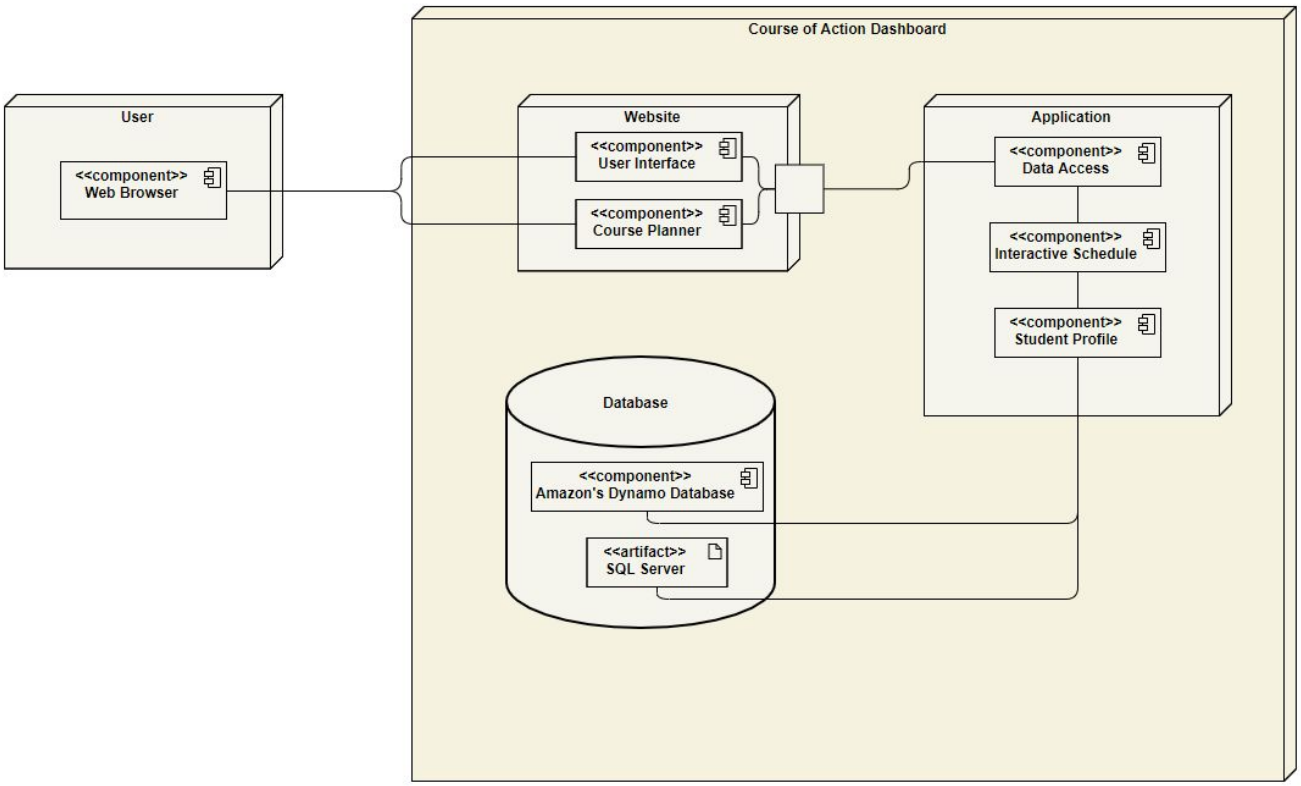
8.3 Course Description Pop Up



Course titles, descriptions, prerequisites, and any notes are accessible by hovering over the credit containers (as shown above).

9. Deployment Diagram

9.1 Diagram



9.2 Diagram Explanation

The purpose of this deployment diagram is to show the structure of the run time system our website application will follow.It consists of the following nodes: “user”, “website”, “application” and “database”. Each of these nodes contains certain components that allow them to function properly and allow for the web app to run.The user will access the web application through a web browser, which will use the internet to connect to the website we created for the course planner.This website will then have access to the application we created where the user can access their own profile which allows for an overview of their degree progress at Brock. This application also contains an interactive planner that the user can utilize to plan their future years at Brock.The application will have access to an external database provided by Amazon’s Web Service which will contain the API calls necessary for this application to function properly.