Requirements Specification Document

Online Course Registration Update Plan

Version Final
25/11/2019

Engineering School Solutions (ESS)

Team 1

Dendy Addison, Sujay Damani, Evan Dungate, Maisha Mir, Samarth Roy, YongQi (John) Shan, Xiaobin Yang

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

The requirements specification document "Online Course Registration Update Plan" version Final describes the requirements of the product, the Course Registrator, as specified by the University of Learning, otherwise known as UL, in meetings held on September 24th, October 3rd, October 23rd, 2019, and November 5th. The purpose of the requirements specification document is to analyze and provide details on what functionalities the Course Registrator will have. The Course Registrator will replace the old registration system, the OCRS, and provide an improved online course registration system for every student at UL. The Course Registrator will also improve the experience of each professor. Furthermore, the Course Registrator will improve the administrative capabilities for the UL's administrators. Through improving the course registration system, the UL will retain more students on a semester to semester basis.

1.2 Project Scope

The Course Registrator will be a web service that provides students at the UL with the tools and information to register for courses and subsequently plan their degree progression. Also, the Course Registrator will reduce the workload of administrators. Additionally, the Course Registrator provides administrators at UL with tools to manage the courses offered in a given semester and assist students with the registration process. Furthermore, the Course Registrator will benefit students at UL by reducing the time taken to register for courses, provide students with more clarity on what courses are required for their program, and reducing their dependency on administrators for assistance. As a result of making the registration process easier for students, we expect the following improvements:

- Increased student retention at UL from semester to semester.
- More students will complete their degree in accordance with the UL-recommended schedule.
- Administrators will spend less time aiding students registering for courses.
- Administrators will have more time to focus on tasks that do not involve course registration for the university.

These improvements enabled by the Course Registrator will benefit UL and satisfy the UL's goal of "providing a streamlined, efficient, and enjoyable experience so that students can truly focus their energies towards learning, instead of spending time on confusing and error-prone administrative processes"[4] as specified in their request for proposal.

1.3 Glossary of Terms

Term	Description
Administrators	Academic advisors and department heads.
Administrative Privilege	Functions that only administrators have.
Administrative Change	An action made by an administrator including adding new course sections or modifying course section capacities.
Alterable Details	The professor, section number, date and time, course capacity, and waitlist capacity of the course.
ASMP	Assumption.
Basic Details	The time, days of the week, and location of a course.
Code Component	A small section of code that performs one action. This could be a singular function in a piece of code.
Code Module	A section of code that performs actions based on one feature. This could be a single class in the overall application.
CON	Constraint.
Course Components	A lecture, tutorial, and/or lab for a given course.
Course Details	The number of students currently enrolled, professor, section number, date and time, course location, course capacity, waitlist capacity, and current number of students on the waitlist of the course.
Course Registrator	The new system that will replace the OCRS.
Defensive Software	Software used to repel an attack on a computer system. Attacks such as DoS attacks.
Department	The field of study a course belongs to. One example of a department is the Math department.
DDoS Attack	Distributed Denial of Service Attack. A DoS Attack from many different sources at the same time.

DoS Attack	Denial of Service Attack. Disrupting a system with large amounts of requests from a single source causing the system to be overwhelmed and stop working.
Drop	Withdrawing registration.
FAQ	Frequently Asked Questions.
General Privileges	Functions that students and professors have.
Hamburger menu	The name given to the menu icon found in programs and websites that hides the traditional file menu.
Help Page question	A question on the help page that has an answer written for it. These are often the frequently asked questions that users ask.
Help Page topic	A topic on the help page that gives a user information about a given topic. Topics could include how to register for courses or how to use the program progress checker.
Integration Test	A software test to confirm that the combination of two or more code components or the combination of two or more code modules causes no errors.
Must	Within the project's scope.
OCRS	Online Course Registration System. The registration system currently in use by UL.
Oracle	A database software.
ORM	Object Relational Mappings are frameworks that translate SQL result sets into code object. This makes development safer and more readable.
Pagination	A design element on tables of data to reduce the size of a table by breaking the table into multiple pages of the same size to allow the table to still display all the data but in smaller sets.
Placeholders	A placeholder is used in sql statements to hold the place of a value before it is replaced by user input. It enhances database security.
Regression Test	A software test to confirm that a recent code change will not adversely affect existing

	features.
REQ	Requirement.
RR	Related Requirements
Schema	Relational Database: How the data is organized in the database. A blueprint of how the data is constructed with its attributes.
Semester	The months that a course is available in. The semesters are either Spring, Summer, or Fall.
SQL	Structured Query Language.
SQL Injection	Attackers write a malicious SQL query on user input to access database information which they have no right to access. An SQL injection attack may damage or compromise the database by modifying, deleting, adding or creating data in database.
SSD	System sequence diagram.
UL	The University of Learning.
User	All user groups that will use the Course Registrator. Students, professors, and administrators.

1.4 References

[1] Hacksplaining, “Protecting Against SQL Injection,” *Hacksplaining*. [Online]. Available: https://www.hacksplaining.com/prevention/sql-injection [Accessed: 30-Sep-2019].

[2] T. Keary, “DoS vs DDoS Attacks: The Differences and How To Prevent Them,” *Comparitech*, 21-Nov-2018. [Online]. Available: https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention/ [Accessed: 30-Sep-2019].

[3] University of Learning, “Google Doc.”17-Sep-2019. [Online]. Available: https://docs.google.com/document/d/1Ygj7CGOqTqQd6M-qyQOcHNJmPF4ZAaMsuoH5bIsBLT0/edit [Accessed: 20-Sep-2019]. Used as a reference. Never directly quoted the RFP.

[4] University of Learning, “University of Learning,” *University of Learning*, 17-Sep-2019. [Online]. Available: https://sites.google.com/view/seng321university-of-learning/mission-statement?authuser=0 [Accessed: 29-Sep-2019].

1.5 Overview

The requirements specification document “Online Course Registration Update Plan” version Final describes the requirements of the product as specified by the University of Learning in meetings held on September 24th, October 3rd, October 23rd, and November 5th 2019. Section 2 contains a general view of the proposed replacement product, the Course Registrator, including the user classes, operating environment, constraints, and assumptions. After that, section 3 specifies the planned features and their corresponding requirements. Section 4 details what external interfaces are required in the sub-categories including user interfaces, hardware interfaces, software interfaces, and communications interfaces. Similarly, section 5 details other non-functional requirements such as performance and safety requirements. Section 6 describes other requirements such as database requirements and student privacy requirements. Added in version 2.0, section 7 includes each use case and the use case diagram which shows the relationship between each use case. For more models, section 8 includes the Entity Relationship Diagram (ERD) for each user class of the Course Registrator. Section 9 includes the Data Flow Diagram 0, Data Flow Diagram 1, and Data Flow Diagram 2 where each data flow diagram shows the relationship between each user class and the Course Registrator in varying levels of specificity. In section 10 includes storyboards added in RSD 3.0 that shows how the product will work. Added for RSD Final, section 11 includes test scenarios that cover all included requirements. Finally, section 12 includes the traceability matrix for the entire list of requirements in RSD Final.

2 Overall Description

2.1 Product Perspective

The Course Registrator will be an improvement to UL’s website which will allow students to sign-up for courses in a more efficient and intuitive manner than the current OCRS allows. The Course Registrator will replace the OCRS which is the system currently in use by UL. The Course Registrator will be available on UL’s website in the same location as the current OCRS. The Course Registrator will aid each student in making better decisions through program progress monitoring, providing answers for frequently asked questions, and giving each student more information on their registration status within each course. With these features, UL will have a more professional and intuitive online presence.

2.2 Product Features

The Course Registrator will have 5 features which are Search, Timetable Builder, Trackable Waitlist, Help Page, and Program Progress Checker.

The Timetable Builder, used during course registration, allows each student to create an overview of their entire semester. In addition, each student can use the timetable builder to create a visual representation of their weekly schedule.

In order to make the Course Registrator easy to use, each user class (student, professor, and administrator) will have their own separate user interface.

Each professor will have a user interface similar to the student version but with additional information for each course that professor is scheduled to teach. Each administrator will have access to administrative privileges in the user interface compared to the student and professor interface.

2.3 User Classes and Characteristics

The stakeholders of the Course Registrator are students, professors, administrators, UL, and ESS. The user group that will interact the most with the Course Registrator is the students so satisfied students is a desired outcome of implementing the Course Registrator. Similarly, professors will be using the Course Registrator and a satisfactory course registration system can provide professors with a better experience at UL. Administrators currently use the OCRS every semester so the Course Registrator will improve the administrators' experience and reduce the amount of work that administrators do. UL is a stakeholder that will not be a primary user of the Course Registrator but an improved student experience can improve the reputation and student retention of UL. ESS is invested in the success of the Course Registrator as satisfied customers improve the reputation of ESS and can lead to future projects being acquired.

Each user of the Course Registrator will have access to information regarding course registration. The users of the Course Registrator are students, professors, and administrators. There are two different user privileges of the Course Registrator: general privileges and administrative privileges. Professors have general privileges. Each student has access to general privileges while each administrator has access to functions that require administrative privileges. Each user will be provided with a different view after login based on which user privilege they belong to. Each feature that a student or professor can use is also accessible by a user with administrative privileges.

The following is an overview of the goals for each user class.

With the Course Registrator:

Students:

- Each student must be able to register for an open course.
- Each student must be able to join the waitlist for a course that is full.
- With the integrated timetable builder:
    - Each student must be able to create a view of the entire semester's schedule.
    - Each student must be able to create a view of their daily schedule.
- Each student must be able to view their current position on the waitlist.
- Each student must be able to view the information of each course including:
    - The number of students currently enrolled.

    - The professor.
    - The section number.
    - The date and time.
    - The course capacity.
    - The waitlist capacity.
    - The current number of students on the waitlist.

Professors:

- With the integrated timetable builder:
    - Each professor must be able to view their entire semester's schedule.
- Each professor must be able to view the information of each course that they are teaching including:
    - The number of students currently enrolled.
    - The section number.
    - The date and time.
    - The course capacity.
    - The waitlist capacity.
    - The current number of students on the waitlist.

Administrators:

- Each administrator must be able to view each student's timetable by semester.
- Each administrator must be able to view the information of each course including:
    - The number of students currently enrolled.
    - The professor.
    - The section number.
    - The date and time.
    - The course capacity.
    - The waitlist capacity.
    - The current number of students on the waitlist.
- Each administrator must be able to edit the information of each course including:
    - The section number.
    - The date and time.
    - The course capacity.
    - The waitlist capacity.
    - The professor.
- Each administrator must be able to enroll any student into a course.
- Each administrator must be able to drop any student from a course the student is enrolled in.
- Each administrator must be able to waive any restriction including year and prerequisite restriction for any student.
- Each administrator must be able to change the professor of each course.

2.4 Operating Environment

- The Course Registrator must work on Google Chrome, Firefox, Edge, and Safari.

- The Course Registrator must use an Oracle database.

2.5 Design and Implementation Constraints

CON-1: Each student can only be waitlisted in a maximum of 10 courses.

CON-2: SQL must be used for fetching data from the Oracle database.

CON-3: The Course Registrator must be available on computers and mobile devices

2.6 Assumptions and dependencies

ASMP-1: Each feature must be designed to exclusively support English.

ASMP-2: The Course Registrator must run on UL servers.

ASMP-3: Each course component at UL will take place between 8am and 10pm each day, Monday to Friday.

ASMP-4: Each student will create their credentials on an existing sign up page on UL website.

ASMP-5: Each student may be declared in up to 2 majors.

ASMP-6: Each user is assigned a user privilege when their account is created.

ASMP-7: Each user must be logged in to use the Course Registrator using UL's pre-existing login interface.

ASMP-8: The maximum number of students on the waitlist should be equal to the class's capacity.

ASMP-9: The Course Registrator must be accessible from UL's webpage.

3 System Features

The following system features have been listed in order of priority, starting with the highest priority feature. Each feature is discussed in terms of the primary user.

3.1 Search

The Search feature is split into two sub-features with similar functionality but different uses. One sub-feature is used for searching for courses while the other sub-feature is used exclusively by administrators to search for students.

3.1a Course Search

3.1.1a Description and Priority

The Course Search sub-feature of the Search feature provides each student with the ability to search from the entire list of available courses and provides information on each course related to their search. Course search is a high priority feature as it allows each student to search for courses.

3.1.1a Functional Requirements

REQ-3-1-2a-1: Each user must be able to search for any course based on the subject of the course.

Initial Feature: Course Search
Rationale: In order for students to register for courses online, they must be able to search through all courses offered by UL.

REQ-3-1-2a-2: Each user must be able to search for any course based on the time that the course takes place.

Initial Feature: Course Search
Rationale: Each student may be busy or seeking a course that does not conflict with their current schedule in order to fill an empty time slot on their schedule. As a result, it is important to UL that each student is able to search for any course based on the time and weekday of each course that is offered.

REQ-3-1-2a-3: Each user must be able to search for any course based on the name of the course.

Initial Feature: Course Search
Rationale: A student may already know the name of a course they want to take based on information they have gathered elsewhere. To accommodate for this, the Course Registrator should allow a student to apply previous information to search for each course by its name.

REQ-3-1-2a-4: Each user must be provided each searched course's course details.

Initial Feature: Course Search
Rationale: It is important that each student has access to course details when searching for a course to register in, as course details will enable them to make more educated decisions on which course they would like to register for.

REQ-3-1-2a-5: Each administrator must be able to edit the alterable details of each searched course.

Initial Feature: Course Search
Rationale: Since courses at UL are subject to their location, professor, time,

REQ-3-1-2a-6: Each user must be able to search for any course based on the department that offers the course.

Initial Feature: Course Search
Rationale: A student may know what department they would like to register for a course with, but not know exactly what course they want to take. Accordingly, a student should be able to narrow their search to show only the courses offered by a specific faculty.

3.1b Student Search

3.1.1b Description and Priority

The Student Search sub-feature of the Search feature provides each administrator the ability to search from the entire list of currently enrolled students and provides information on each student. Student Search is a high priority feature that allows each administrator to search for students that they want to add or drop from classes.

3.1.2b Functional Requirements

REQ-3-1-2b-1: Each administrator must be able to search for each student based on that student's student number.

Initial Feature: Student Search
Rationale: Each student at UL has a unique student ID and an administrator will need to search for a student in the Course Registrator that they can help a student with their academic needs. Each administrator should be able to find each student based on their unique student ID.

REQ-3-1-2b-2: Each administrator must be able to search for each student based on that student's declared program.

Initial Feature: Student Search
Rationale: Each administrator at UL provides advice for any student in a certain program. As a result, an administrator at UL may need to view each student that has declared that program. Therefore, an administrator should be able to search for a student based on the declared program of each student.

REQ-3-1-2b-3: Each administrator must be provided details of each student that was searched.

Initial Feature: Student Search
Rationale: One responsibility of each administrator at UL is to advise students on their academic goal and course selection. In order to fulfil the responsibility of giving advice, each administrator should be able to view the information of each student.

REQ-3-1-2b-4: Each administrator must be provided the timetable of each student that was searched.

Initial Feature: Student Search
Rationale: Each administrator may want to view a student's timetable to assess a student's courses in order to provide advice.

3.2 Timetable Builder

3.2.1 Description and Priority

The Timetable Builder provides each student with an interactive and updateable timetable in the form of a weekly schedule. The Timetable Builder allows each student to select courses for registration. Also, the Timetable Builder will update in real time to provide each student with their daily schedule. Each student will be able to see each course's information that the student is registered in such as a course's location and time slot. Each professor will be able to see each course that they are teaching. Each administrator can use the Timetable Builder to see each student's or professor's timetable. In addition, each administrator can use the Timetable Builder to add or drop a student from a course or change the professor teaching a course. The Timetable Builder is a high priority feature.

3.2.2 Functional Requirements

REQ-3-2-2-1: After searching for any course, each student must be able to register for each course that is open for registration.

Initial Feature: Timetable Builder
Rationale: The aim of the Course registrator is to improve the registration process of courses for each student at UL. Therefore, if a student can find a course offered by UL, they should be able to register for it using the course registrator.

REQ-3-2-2-2: After searching for any student, each administrator must be able to add that searched student to a course.

Initial Feature: Timetable Builder
Rationale: Each student may visit an administrator in the administrator's office in hopes of having an administrator register them for a course. In order for each administrator to help each student that visits register in a course, each administrator should be able to register any student in a course.

REQ-3-2-2-3: After searching for any student, each administrator must be able to drop that searched student from a course.

Initial Feature: Timetable Builder
Rationale: A student may come and visit an administrator in their office for advice which can include an administrator dropping a course for a student. As a result, the ability to drop a student from any number of their registered courses is important for administrators.

REQ-3-2-2-4: Each administrator must be able to override any restrictions including year and prerequisite course restrictions for any student.

Initial Feature: Timetable Builder
Rationale: Each student at UL can take a course that they are restricted from registering for if any administrator gives that student approval to take the course. Once any administrator has given permission to a student, there needs to be a way for that administrator to override the restriction for that student in the Course Registrator.

REQ-3-2-2-5: Each administrator must be able to change each course's professor.

Initial Feature: Timetable Builder
Rationale: Since the professor of a course is can change due to various circumstances, each administrator should be able to edit the professor of a course.

REQ-3-2-2-6: Each student must be able to see each course that they are registered for on their timetable.

Initial Feature:
Rationale: Each student needs to know the time and place of each of their registered classes so that they can attend them. Furthermore, each student should be able to see which courses they are enrolled in for a given semester and a timetable is a visual representation to assist students.

REQ-3-2-2-7: Each student must be provided the basic details on their timetable about each registered course:

Initial Feature: Timetable Builder

Rationale: The basic details are the most important pieces of information that each student needs to find and attend their classes. The timetable is the main view for each student so the basic details of each course should be on their timetable.

REQ-3-2-2-8: Each student must be able to drop any course that they are currently registered in.

Initial Feature: Timetable Builder

Rationale: Each student's circumstances may change which could require them to drop themselves from a course. An administrator should not have to be involved every time a student wishes to drop a course. Therefore, each student should be able to drop any course that they are currently registered in.

REQ-3-2-2-9: Each user must be able to switch between each week on their timetable.

Initial Feature: Timetable Builder

Rationale: There are some weeks that have different schedules so each student should be able to view the different weeks on their schedule by switching between the weeks.

REQ-3-2-2-10: Each professor must be able to see each course that they are teaching on their timetable.

Initial Feature: Timetable Builder

Rationale: Each professor should be able to view each course they are teaching so that they can know where and at what time to go to each course.

REQ-3-2-2-11: Each professor must be able to see the basic details on their timetable about each course they are teaching:

Initial Feature: Timetable Builder

Rationale: The basic details are the most important pieces of information that a professor needs to find and attend their classes. The timetable is the main view for each professor so the basic details of each course they are teaching should be on their timetable.

3.2.3 Interface Requirements

REQ-3-2-3-1: The timetable must display time slots from 8am to 10pm, Monday to Friday.

Initial Feature: Timetable Builder
Rationale: UL offers classes from 8am to 10pm Monday to Friday therefore the timetable should be able to display classes from 8am - 10pm Monday to Friday.

REQ-3-2-3-2: Each day on the timetable must have 28 time slots starting at 8am where 1 time slot represents 30 minutes.

Initial Feature: Timetable Builder
Rationale: UL offers classes of varying length which are mostly comprised of 30 minute chunks. Also, the timetable should have a uniform division so that it is easily understandable. Therefore, the timetable should be broken up into 30 minute intervals.

REQ-3-2-3-3: The timetable must show the current week as the default layout.

Initial Feature: Timetable Builder
Rationale When a user is looking at their timetable they will usually be concerned times proximate to the current date. For example, what class they need to go to next, or what does their schedule look like tomorrow. As a result, the timetable should default to the current week.

3.3 Trackable Waitlist

3.3.1 Description and Priority

The Trackable Waitlist provides each student with information about each waitlist they are on. If there is an open position in a course, the student who receives the notification must also be able to leave the waitlist and join that course. The Trackable Waitlist is a medium priority feature.

3.3.2 Functional Requirements

REQ-3-3-2-1: If a course is full, each student must be able to join that course's waitlist.

Initial Feature: Trackable Waitlist

Rationale: Each course has a limited capacity so each student should be given the opportunity to register for a queue in the form of a waitlist to try to register for a course that is full.

REQ-3-3-2-2: Each student must be able to see their current position on each course's waitlist that they are waitlisted for.

Initial Feature: Trackable Waitlist

Rationale: If each student is able to view their current position on a course's waitlist, each student can guess their likelihood of being able to register for a waitlisted course. For example, if a student's position is too far down the waitlist, they may wish to enroll for another course.

REQ-3-3-2-3: When there is an open space in a course, the student with the lowest position in that course's waitlist, position 1, must receive a notification.

Initial Feature: Trackable Waitlist

Rationale: If a student is in position 1 on a course's waitlist, that student should be informed right away so that they can accept the offer to register for that course.

REQ-3-3-2-4: Each student must be able to drop any course they are waitlisted for at any time.

Initial Feature: Trackable Waitlist

Rationale: Each student may wish to drop a course they are waitlisted for if there is a time conflict with another class they are enrolled in or if that student finds that the course is a better fit for a different time or semester.

REQ-3-3-2-5: Each student that drops a course from their waitlist must also be removed from the dropped course's waitlist.

Initial Feature: Trackable Waitlist

Rationale: The waitlist acts as a queue for each student so when a student on the waitlist drop a course, that course's waitlist should update to reflect that change.

REQ-3-3-2-6: Each student must be able to see the current number of students on each course's waitlist that they are waitlisted for.

Initial Feature: Trackable Waitlist

Rationale: Each student on the waitlist may want to see the total number of students currently waitlisted for a course to determine their chances of being able to register for the course.

REQ-3-3-2-7: Each student must be able to see the maximum number of students allowed on the waitlist.

Initial Feature: Trackable Waitlist
Rationale: Each student must be able to see each course's maximum waitlist capacity to determine if there is enough space for them to be able to join a course's waitlist.

REQ-3-3-2-8: When there is an open space in a course, the student with the lowest position in that course's waitlist, position 1, must be able to enroll in that course.

Initial Feature: Trackable Waitlist
Rationale: The student at position 1 on a course's waitlist must be given top priority over all other students to enroll for the course if there is an open space.

REQ-3-3-2-9: Each student that joins each course's waitlist must have that course added to their Trackable Waitlist.

Initial Feature: Trackable Waitlist
Rationale: A student must be able to view the courses they are waitlisted for on their Trackable Waitlist in order to keep track of their courses.

3.4 Help Page

3.4.1 Description and Priority

The Help Page is a page dedicated to frequently asked questions and tutorials to clarify errors and any common questions that a user may have. The Help Page is a medium priority feature.

3.4.2 Functional Requirements

REQ-3-4-2-1: Each user must be able to access the Help Page from anywhere in the Course Registrator.

Initial Feature: Help Page
Rationale: Any user could be unaware of what they need to do in any part of the Course Registrator. Each user could be assisted if each user is able to get help from each page that they are on.

REQ-3-4-2-2: Each administrator's contact information must be listed on the Help Page.

Initial Feature: Help Page
Rationale: In the case where a student cannot find a complete answer to their question on the Help Page, they must be able to contact an administrator for further information. To make this easier for the student, each administrator's contact information must be clearly listed on the Help Page.

REQ-3-4-2-3: Each administrator's work hours must be listed on the Help Page.

Initial Feature: Help Page
Rationale: If a student needs to contact an administrator for further information, the student must know the availability of the administrator. In order for a student to know the administrator's availability, the administrator's work hours must be listed clearly on the Help Page.

REQ-3-4-2-4: On the Help Page, each student must be provided with a tutorial on how to use the Course Registrator.

Initial Feature: Help Page
Rationale: When a student first goes to use the Course Registrator, they may not know how to use the Course Registrator. In order to ensure that each student knows how to use the Course Registrator, the Help Page must provide a tutorial on how to use the Course Registrator.

3.5 Program Progress Checker

3.5.1 Description and Priority

Each student must be able to check their progress in their degree using the Program Progress Checker. Each student must be able to use the Program Progress Checker to see the list of courses completed and the list of courses still required to be completed for the student. The Program Progress Checker is a medium priority feature.

3.5.2 Functional Requirements

REQ-3-5-2-1: Each student must be able to view a list of every course that they have completed.

Initial Feature: Program Progress Checker
Rationale: In order for each student to see their overall progress on their declared degree, each student must be able to see the courses they have already completed in the past.

REQ-3-5-2-2: Each student that has declared a program must be able to view a list of every course that they need to complete to finish their declared program.

Initial Feature: Program Progress Checker
Rationale: If a student wishes to see what courses they need in order to finish their declared program, the Program Progress Checker must show a list of every course they need to complete.

REQ-3-5-2-3: Each administrator must be able to access each student's Program Progress Checker.

Initial Feature: Program Progress Checker
Rationale: If a student has questions regarding their progress in their declared program, an administrator must be able to access the student's Program Progress Checker to gather information to assist the student.

REQ-3-5-2-4: For each course displayed on a student's Program Progress Checker, that student must have a link to access each course's details.

Initial Feature: Program Progress Checker
Rationale: For a student to easily access the information of a course when looking at their Program Progress Checker, the Program Progress Checker must display a clickable link that takes the student to a clear view of the course's course details.

4 External Interface Requirements

4.1 User Interface

This section lists and details all requirements related to the user interfaces. The user interface is the interface through which the user of the Course Registrator interacts with the Course Registrator. Each Course Registrator user will have a different user interface based on their user privilege level.

4.1.1 Screen Layout Constraints

The following requirements in section 4.1.1 define the constraints the user interface must be designed to satisfy.

REQ-4-1-1-1: Each web page of the Course Registrator must be scalable to fit on common desktop screen resolutions and mobile device screen resolutions.

Initial Feature: User Interface
Rationale: Making the Course Registrator scale for both desktop and mobile device screen resolutions means that each user will not have to manually resize the web page they are viewing which will ensure that each user has a seamless experience when using the Course Registrator.

REQ-4-1-1-2: Each web page of the Course Registrator must use design elements that follow common web page design formats. These elements include but are not limited to:

- Hamburger menus.
- Pagination on tables.
- Dynamically updated website elements such as tables and graphs.

Initial Feature: User Interface
Rationale: Making sure that each web page of the Course Registrator follows common design elements found on other web pages lowers the likelihood that the user will struggle using the Course Registrator.

4.1.2 Error Message Display

Section 4.1.2 defines the requirements any error message display must satisfy.

REQ-4-1-2-1: Each error's message must be no more than 10 words long.

Initial Feature: User Interface
Rationale: Having more than 10 words in an error message may introduce ambiguity for a user. Hence, to be concise and unambiguous, each error message must contain less than 11 words.

REQ-4-1-2-2: Each error message must be distinct from the rest of the Course Registrator's elements.

Initial Feature: User Interface
Rationale: Each user must be able to read error messages easily. If an error message is distinct compared to the rest of the Course Registrator, it is much more likely that each user will be able to see it conveniently.

REQ-4-1-2-3: Each error message must be displayed near the element which caused the aforementioned error.

Initial Feature: User Interface
Rationale: Each error message must be displayed near the element that causes the error so that the element on which an error occurs is obvious to each user.

4.2 Hardware Interfaces

As of the writing of the RSD Final, there is no planned requirements for hardware interfaces. The current OCRS is entirely software-based so there will be no addition of hardware interfaces for the new Course Registrator.

4.3 Software Interfaces

4.3.1 Database

Section 4.3.1 contains requirements related to the database that will be used by the Course Registrator.

REQ-4-3-1-1: The Course Registrator database must save each student's GPA, courses, previous grades, and declared program which are migrated from existing databases.

Initial Feature: Software Interfaces
Rationale: Each student's data such as GPA, courses, previous grades, etc. must be preserved when the UL migrates its users to the Course Registrator. This data migration is necessary for the functioning of features such as the Program Progress Checker.

REQ-4-3-1-2: The Course Registrator's database must store updated information from the tasks performed by each user of the Course Registrator such as adding a new course.

Initial Feature: Software Interfaces
Rationale: Each action performed by each user that causes a change in data stored on the Course Registrator must be reflected in the database. The absence of this database feature requirement will make the Course Registrator highly likely to display incorrect information to each user and might even make the Course Registrator unusable in some cases.

4.4 Communications Interfaces

4.4.1 Web browser

Section 4.4.1 contains requirements related to any web browser used to interact with the Course Registrator.

REQ-4-4-1-1: Each user must be able to access the Course Registrator using any of the following browsers:

- Chrome.
- Firefox.
- Edge.
- Safari.

Initial Feature: Communication Interfaces
Rationale: Since the Course Registrator is accessed via the UL's website, the accessibility of the Course Registrator via different browsers is directly dependent on the accessibility of the UL website. Nonetheless, this is a requirement that the Course Registrator must fulfill as each user is likely to use various browsers to access the Course Registrator.

REQ-4-4-1-2: Each user must be able to access the Course Registrator from a mobile device.

Initial Feature: Communication Interfaces
Rationale: Each user may want to access the Course Registrator via a mobile device for a number of reasons such as easy accessibility, remote accessibility, or a temporary lack of other options. Therefore, the Course Registrator must be accessible via a mobile device.

4.4.2 Email

Section 4.4.2 contains requirements related to any email interaction with a user of the Course Registrator.

REQ-4-4-2-1: Each student must receive an email to notify them of their incomplete registration status every day until either:

- that student enrolls in the missing course component.
- that student drops the course.

Initial Feature: Communication Interfaces
Rationale: In order to successfully complete a course, each student must be registered in and pass all components of that course (lecture, lab, tutorial). Therefore, if a student has an incomplete registration for a course, the Course Registrator sends a daily reminder to that student until that student either drops or registers in all components of that course.

REQ-4-4-2-2: Each student must receive an email to notify them of any change in their waitlist status.

Initial Feature: Communication Interfaces
Rationale: A change in the waitlist status of each student warrants an action by that student. That student is required to make a time-sensitive decision to either accept or reject a registration offer. Therefore, the Course Registrator must send an email to each student regarding any change in their waitlist status and provide them adequate time for each student to make a decision.

REQ-4-4-2-3: Each student must receive an email to notify them of any change in the course details of each registered course.

Initial Feature: Communication Interfaces
Rationale: Course details for any course each student is registered for are relevant to each student. Therefore, each change made to a registered course's course details must be conveyed to each registered student via email immediately.

5 Other Non-Functional Requirements

5.1 Software Quality Attributes

5.1.1 Performance Requirements

REQ-5-1-1-1: When each user queries the server, a page must be served to the user in at most 1 second.

Initial Feature: Software Quality Attributes
Rationale: UL wants to provide the best user experience and they have a standard response time of 1s or less for every query executed and hence queries executed on the Course Registrator will follow the same standards.
Test Scenario 33: Set filters or type name for a course in the search bar. Start a timer when the search button is clicked. As soon as the search results are displayed, stop the timer. Verify if the action performed is executed in 1s or less.

REQ-5-1-1-2: Each user logging in to the Course Registrator must not cause the Course Registrator's response time to increase above 1 second for any other user who is currently using the Course Registrator.

Initial Feature: Software Quality Attributes

Rationale: UL wants to allow multiple users to be logged in the Course Registrator and not have each new user affect the performance speed for each user already logged in.
Test Scenario 34: Have at least two users log into the Course Registrator as any and perform a course search. Set filters or type name for a course in the search bar. Start a timer when the search button is clicked. As soon as the search results as displayed, stop the timer. Verify if actions executed are performed in 1s or less for each user performing this test scenario.

REQ-5-1-1-3: Any changes made to the course registration of each student must be updated on that student's timetable in the Timetable Builder in at most 1 second.

Initial Feature: Software Quality Attributes
Rationale: UL would like to have timetable changes being displayed in at most 1s after the change is made by the Student/Admin to the timetable of the Student.
Test Scenario 35: When a user makes a change to the timetable such as adding or removing a course from the timetable, as the button for adding a course is clicked or when a button for removing a course is clicked, start the timer. As soon as the action (adding of course or removing of the course) is executed, stop the timer. Verify that the action executed is performed in at most 1s.

REQ-5-1-1-4: An error message must appear near the source of the error caused by the user in less than 1 second.

Initial Feature: Software Quality Attributes
Rationale: The UL would like the users of the Course Registrator to be notified in less than 1 second if an error is made.

5.1.2 Security Requirements

The Course Registrator is a web application so it must be able to repel the following web-based attacks:

- DOS/DDoS attacks:

REQ-5-1-2-1: Bandwidth must be increased to ensure that increased traffic can be handled.

Initial Feature: Software Quality Attributes
Rationale: UL must increase its bandwidth in order to handle increased traffic so that a website crash can be avoided.

REQ-5-1-2-2: An IP address must be blocked if it is detected to be sending a large amount of requests or causing “[unusually high] traffic levels”[2].

Initial Feature: Software Quality Attributes
Rationale: UL would like to block IP addresses that are trying to spam bandwidth server causing “[unusually high] traffic levels”[2].

REQ-5-1-2-3: Defensive software must be deployed on the server hosting the Course Registrator to protect against attacks.

Initial Feature: Software Quality Attributes
Rationale: UL would like to have defensive software for the Course Registrator to protect their website against attacks.

- SQL Injection attacks:

REQ-5-1-2-4: The Course Registrator must use “parameterized statements”[1].

Initial Feature: Software Quality Attributes
Rationale: The Course Registrator must avoid any SQL injection attacks and “parameterized statements”[1] help avoid them.

REQ-5-1-2-5: SQL queries must be constructed with placeholders.

Initial Feature: Software Quality Attributes
Rationale: SQL queries sent to the database must have placeholders while creating prepared statements instead of exact information to avoid sql injection attacks.

REQ-5-1-2-6: Each parameter value must fill a placeholder to query the Course Registrator’s database at execution time.

Initial Feature: Software Quality Attributes
Rationale: For safety and privacy concerns, all prepared query placeholders must only be replaced at execution time.

REQ-5-1-2-7: “ORM(Object Relational Mapping) frameworks”[1] must be used.

Initial Feature: Software Quality Attributes
Rationale: The Course Registrator will use ORM to decouple the domain model from the database schema.

REQ-5-1-2-8: Each user not active for 5 minutes must be logged out.

Initial Feature: Software Quality Attributes
Rationale: An inactive User should be logged out to reduce traffic on Course Registrator servers.

REQ-5-1-2-9: Each student must not have administrative privileges for the Course Registrator.

Initial Feature: Software Quality Attributes
Rationale: UL does not want any Student User to have administrative privileges.

REQ-5-1-2-10: All data transmitted between the Course Registrator and each user must be encrypted.

Initial Feature: Software Quality Attributes
Rationale: The data transfer must be encrypted to protect each user from potential hackers stealing each user’s information.

5.1.3 Scalability

REQ-5-1-3-1: If the number of students enrolled at UL increases, the new user traffic must not cause the Course Registrator to crash.

Initial Feature: Software Quality Attributes
Rationale: UL would like to have new students use Course Registrator without Course Registrator crashing.

5.1.4 Testability

REQ-5-1-4-1: The Course Registrator must be split into code components and code modules.

Initial Feature: Software Quality Attributes
Rationale: Code components and modules makes it easier to alter the Course Registrator in the

future and for traceability purposes.

REQ-5-1-4-2: Each code component must have unit tests.

Initial Feature: Software Quality Attributes
Rationale: Unit tests are needed to test that each code component has been correctly implemented.

REQ-5-1-4-3: Each code module must have unit tests.

Initial Feature: Software Quality Attributes
Rationale: Unit tests is a way to verify that each code module has been correctly implemented.

REQ-5-1-4-4: After each code component is added or changed, integration tests must verify all possible cases and produce no errors.

Initial Feature: Software Quality Attributes
Rationale: ESS does not want any errors to be produced if any code component is added or changed.

REQ-5-1-4-5: After each code module is added or changed, integration tests must verify all possible cases and produce no errors.

Initial Feature: Software Quality Attributes
Rationale: ESS does not want any errors to be produced if any code module is added or changed.

REQ-5-1-4-6: All Regression Tests must ensure that future updates do not affect existing features.

Initial Feature: Software Quality Attributes
Rationale: Future updates can interfere with the already existing features so Regression Tests is one method of mitigating incompatibility from new updates.

5.1.5 Compatibility

REQ-5-1-5-1: Each Course Registrator web page must be compatible with both desktop computers as well as mobile devices.

Initial Feature: Software Quality Attributes
Rationale: Making web page responsive gives each user a consistent experience no matter which device the user access to Course Registrator. With a responsive design, each user does not need to resize the page manually to navigate or use links, button, or images on the Course Registrator.

REQ-5-1-5-2: Interacting with the Course Registrator on a smartphone or tablet must be as intuitive as interacting with the Course Registrator on a desktop computer.

Initial Feature: Software Quality Attributes
Rationale: No matter what device is used to access the Course Registrator, the Course Registrator should contain all of the necessary components and be used as easy as on a desktop computer. Each user should immediately know how to navigate and use the Course Registrator.

5.1.6 Availability

REQ-5-1-6-1: The Course Registrator must not go offline for more than 6 minutes per year without manual intervention.

Initial Feature: Software Quality Attributes
Rationale: This requirement guarantees each user the accessibility to Course Registration. It is essential for UL to work normally. Any unplanned downtime of the Course Registrator brings each user inconvenience which hinders the course registration process.

5.1.7 Privacy

REQ-5-1-7-1: Each student must not be allowed access to another student's personal information.

Initial Feature: Software Quality Attributes
Rationale: Each student's personal information must be protected and kept private from every other student.

REQ-5-1-7-2: Each student's personal information must not be available to the public.

Initial Feature: Software Quality Attributes
Rationale: No institution or individual is allowed to expose any student's private information according to the law. As a result, each student's information should not be visible or accessible from the public.

5.1.8 Reliability

REQ-5-1-8-1: The Course Registrator must support up to 26,000 users concurrently without crashing.

Initial Feature: Software Quality Attributes
Rationale: The Course Registrator will serve approximately 20,000 users. However, growth of UL may increase the number of users in the future so the Course Registrator should support up to 26000 users.

5.2 Trackable Waitlist

REQ-5-2-1: Each student's waitlist page must be updated to display each required course component of each waitlisted course for that student.

Initial Feature: Trackable Waitlist
Rationale: The Waitlist page must always display each component of a course that a student wants to register in on the waitlist. Each student must see the course on their waitlist after adding a course to their waitlist.

5.3 Help Page

REQ-5-3-1: The Course Registrator must record in the database the number of clicks received by each Help Page topic on the help page.

Initial Feature: Help Page
Rationale: Using statistics on the Help Page informs UL on what questions are popular when using the Course Registrator. This is helpful because each administrator is provided with insight on each problem with using the Course Registrator that can further future development of the Course Registrator.

REQ-5-3-2: The Course Registrator must record in the database the number of clicks received by each Help Page question on the help page.

Initial Feature: Help Page

Rationale: This requirement is a method to record the number of clicks for each question on help page so that the Course Registrator can sort each question based on the number of clicks.

REQ-5-3-3: The 10 most frequently clicked topics or questions must be displayed at the top of the help page.

Initial Feature: Help Page
Rationale: The questions with the most clicks are the most common problems that each user is facing. Putting these popular questions at the top of the Help Page can lead to each user finding their answers immediately.

REQ-5-3-4: The Help Page must have a topic describing the steps a student should take to build their timetable.

Initial Feature: Help Page
Rationale: The timetable is a vital part in the Course Registrator. Therefore, providing the information on how to build a timetable will also let each user familiarize themselves with the Course Registrator.

5.4 Immediate Error Checking

REQ-5-4-1: When registering for a course, each student must be immediately notified of a year restriction.

Initial Feature: Immediate Error Checking
Rationale: A year restriction notification immediately reminds each student that can not register due to their current year standing why they can not register for that course.

REQ-5-4-2: When registering for a course, each student must be immediately notified of a degree restriction.

Initial Feature: Immediate Error Checking
Rationale: A degree restriction notification informs each student that can not register due to their degree that a course is restricted to each student that has particular degree

REQ-5-4-3: When registering for a course, each student must be immediately notified of a missing prerequisite course.

Initial Feature: Immediate Error Checking
Rationale: The missing prerequisite course message notifies each student that can not register due to missing prerequisite courses what courses need to be taken before registering.

For the following requirement, a time conflict between two courses happens when a student tries to register for a course that is scheduled for the same time as another course the student has already registered for.

REQ-5-4-4: When registering for a course, each student must be immediately notified of a time conflict between two courses.

Initial Feature: Immediate Error Checking
Rationale: The requirement clearly tells each student the course has time conflict with the existing added courses. It is impossible for each student to take two courses at the same time.

6 Other Requirements

6.1 Database Requirements

REQ-6-1-1: Each administrator must be able to modify the Course Registrator's data in the Course Registrator's database using the Course Registrator.

Initial Feature: Database Requirements
Rationale: Each administrator has privileges to modify data of the Course Registrator. This allows each administrator the ability to change inaccurate data, such as course details and student registration information.

REQ-6-1-2: Each administrator must be able to add new Course Registrator data to the Course Registrator's database using the Course Registrator.

Initial Feature: Database Requirements
Rationale: Each administrator has privileges to add data to the Course Registrator. The goal of this is administrator can help a student add or drop a course.

REQ-6-1-3: Each query that updates data in the Course Registrator's database must update the Course Registrator's database in less than 0.3 seconds.

Initial Feature: Database Requirements
Rationale: The delay for updating the data for Course Registrator's database should be as short as possible. The delay will also influence how fast each user can see the error or success message after each action. If the update is within 0.3s, each user of the Course Registrator will have an overall good experience using the Course Registrator.

7 Use Cases

7.1 Use Case Diagram

The use case diagram details the interactions between each of the different use case definitions from the perspective of the different user privileges that are defined above.

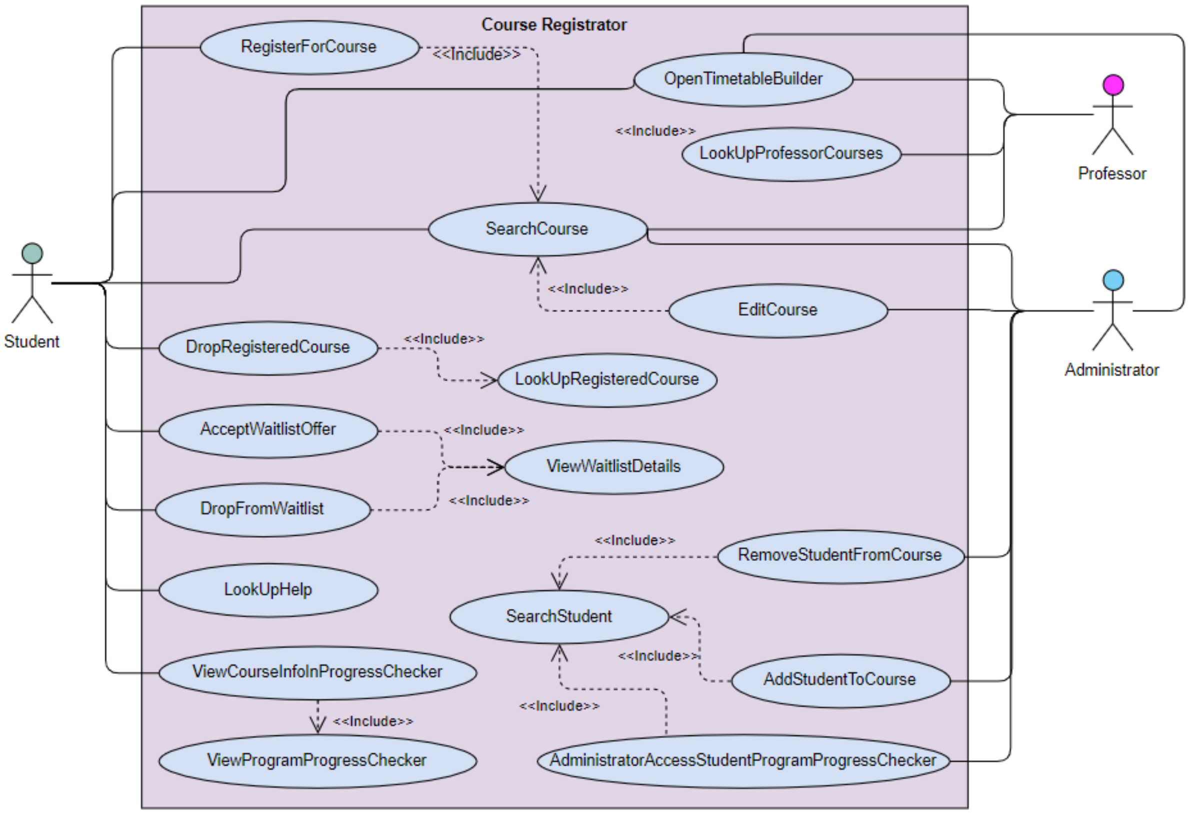


Figure 1. Use case diagram

7.2 Use Case Definitions and System Sequence Diagrams (SSD)

The following tables detail each interaction a user might have with the Course Registrator. This includes use cases for all User Classes as well as branching flow due to a possible error. Included in each section is the sequence diagram associated with the given use case.

7.2.1 Search Course

Use Case: SearchCourse
ID: UC-1
Brief description: The User searches for courses.
Actor(s): User (Administrator, Professor, Student)
Preconditions: 1. The User must be logged in to the Course Registrator.
Main flow: 1. The User selects the course search. 2. While the User searches for a course with filters: 2.1. If the User searches by using filters: 2.1.1. The User selects any combination of the following filters for search: 2.1.1.1. Semester. 2.1.1.2. Department. 2.1.1.3. Day. 2.1.1.4. Time. 2.2. Else: 2.2.1. The User enters text into the course search bar. 2.3. If a course matches (semester, department, course name, required course) in the database: 2.3.1. The User views all matching courses and each of the matching course's info. 2.4. Else: 2.4.1. The User sees a "No courses matched your search" message.
Postconditions: None.
Alternative Flow(s): None.

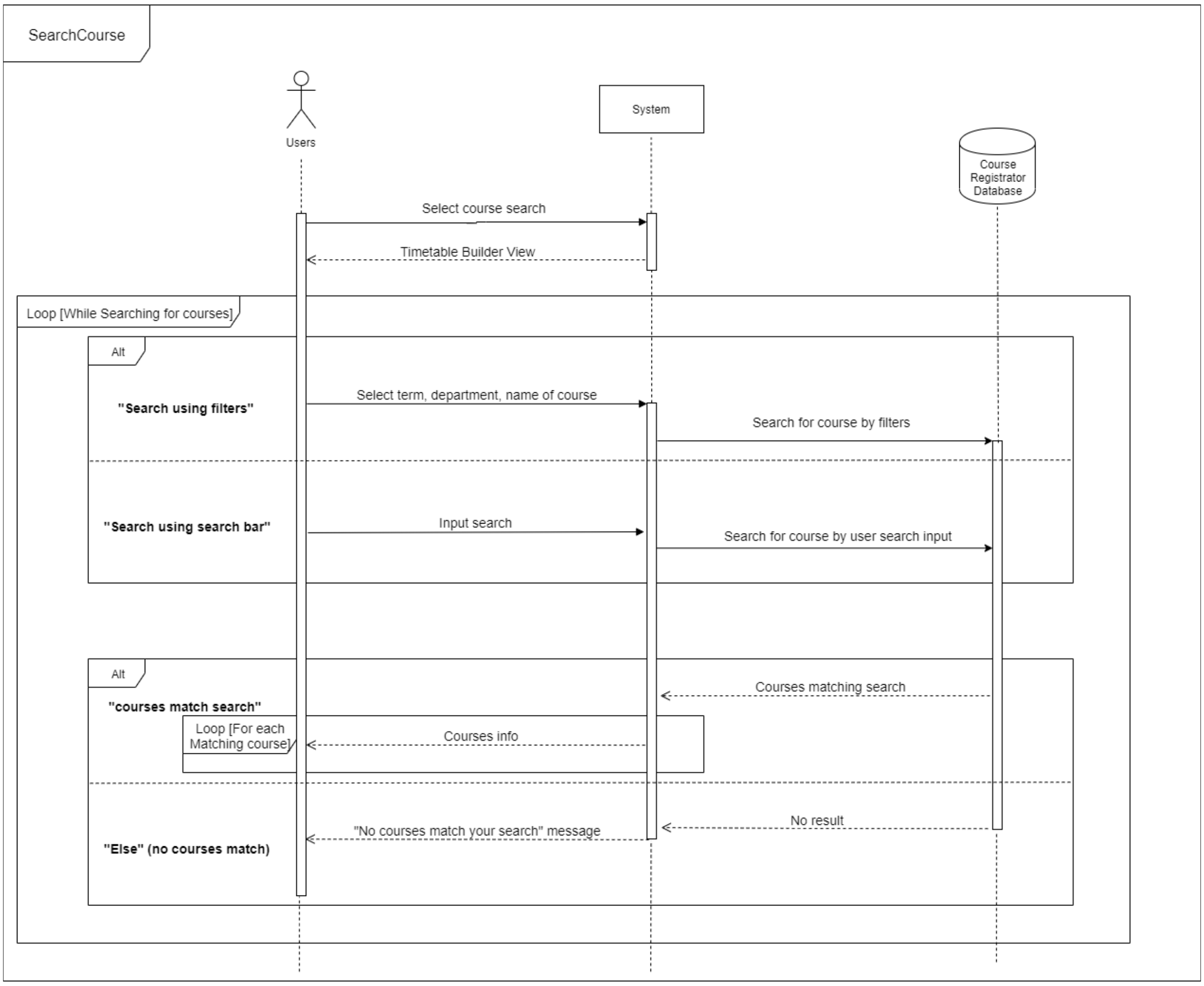

SearchCourse
Users
System
Course Registrator Database
Select course search
Timetable Builder View
Loop [While Searching for courses]
Alt
"Search using filters"
Select term, department, name of course
Search for course by filters
"Search using search bar"
Input search
Search for course by user search input
Alt
"courses match search"
Courses matching search
Loop [For each Matching course]
Courses info
"Else" (no courses match)
No result
"No courses match your search" message

Figure 2. SearchCourse SSD

7.2.2 Register For Course

Use Case: RegisterForCourse
ID: UC-2
Brief description: The Student registers for a course.
Actor(s): Student
Preconditions: 1. The Student must be logged in to the Course Registrator. 2. The course must exist in the system database. 3. The course must have an open section or an open waitlist. 4. The Student must be eligible to take the course (no restrictions).
Main flow: 1. include(SearchCourse) 2. If the course section is not full: 2.1. The Student selects “Add Course”.

2.2. The Student has the course added to their Timetable Builder automatically. 2.3. The Student is registered for the course automatically. 3. If course is full and the waitlist has empty spots: 3.1. The Student is prompted to add the course to their Trackable Waitlist. 3.2. The Student confirms the prompt. 3.3. The Student sees the course on their Trackable Waitlist.
Postconditions: 1. The Student is either registered or waitlisted for a course.
Alternative Flow(s): None

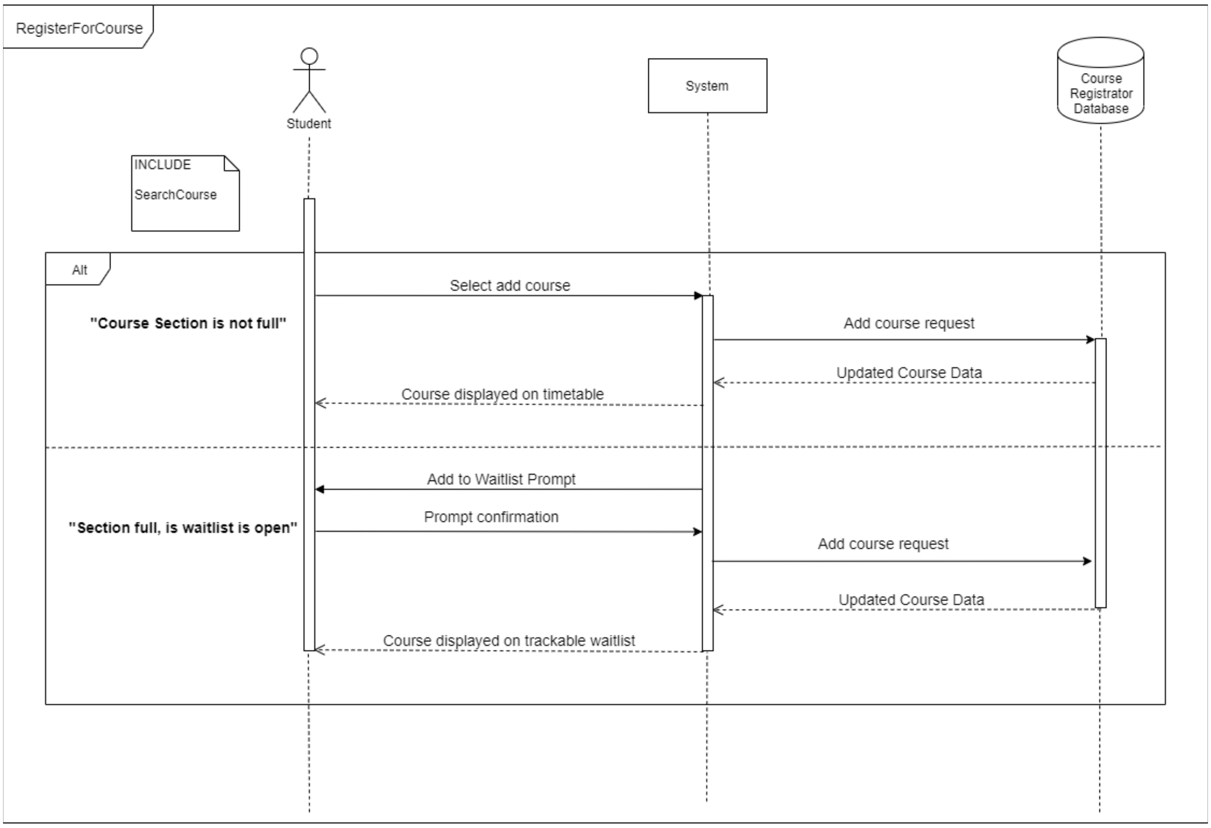


Figure 3. RegisterForCourse SSD

7.2.3 Look Up Professor Courses

Use Case: LookUpProfessorCourses
ID: UC-3
Brief description: A Professor views courses they are teaching.
Actor(s): Professor
Preconditions: 1. The Professor must be logged in to the Course Registrator. 2. The Professor must be teaching at least one course.

Main flow:

1. The Professor selects Course List.
2. The Professor views the courses they are teaching.
3. For each course the Professor teaches, the professor sees each course's course details.

Postconditions: None

Alternative flow(s): None

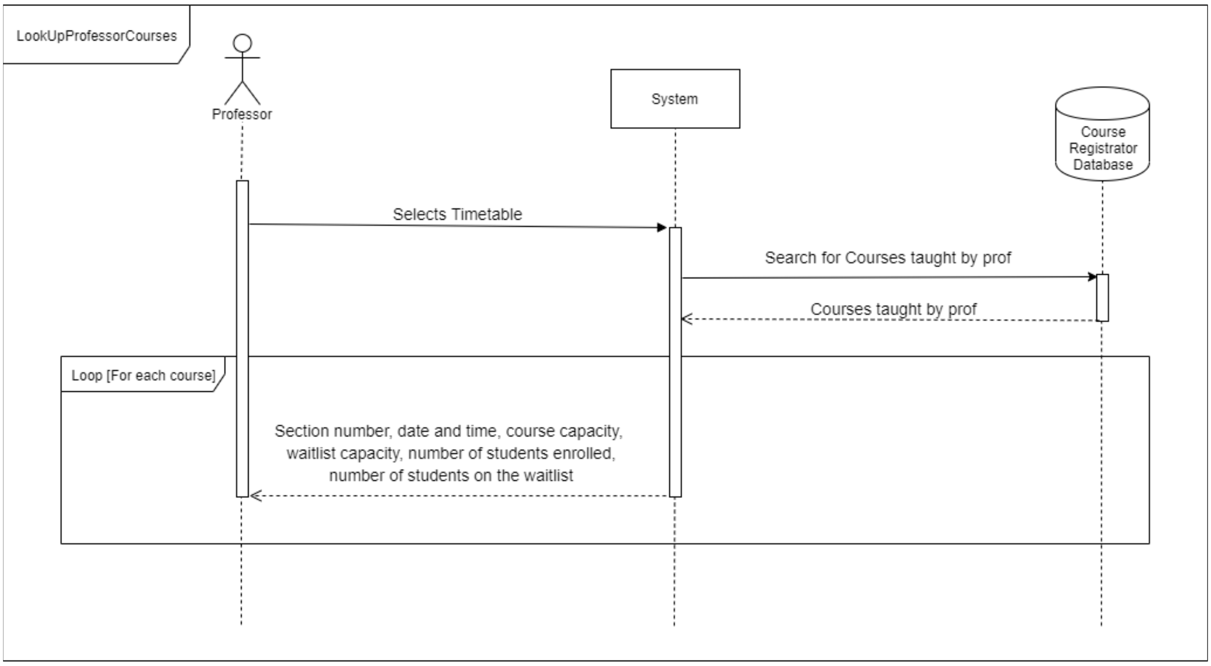

LookUpProfessorCourses
Professor
System
Course Registrator Database
Selects Timetable
Search for Courses taught by prof
Courses taught by prof
Loop [For each course]
Section number, date and time, course capacity, waitlist capacity, number of students enrolled, number of students on the waitlist

Figure 4. LookUpProfessorCourses SSD

7.2.4 Open Timetable Builder

Use Case: OpenTimetableBuilder

ID: UC-4

Brief description: The User navigates to the Timetable Builder.

Actor(s): User (Administrator, Student)

Preconditions:

1. The Student must be registered for the Course.
2. The Student must be logged in to Course Registrator.

Main flow:

1. The User navigates to the Timetable Builder.
2. For each course in timetable builder:
    - 2.1. The User views the course's name.
    - 2.2. The User views the Professor's name.
    - 2.3. The User views the course's location.

2.4. The User views the course’s day and time.
Postconditions: None
Alternate Flow(s): None

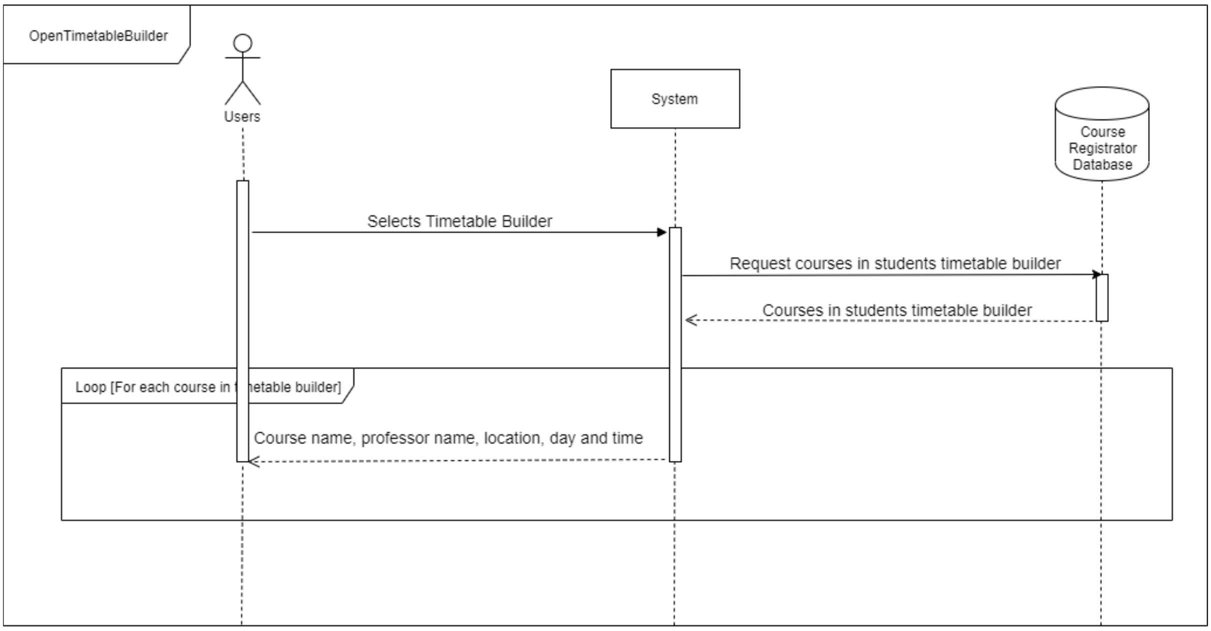

OpenTimetableBuilder
Users
System
Course Registrator Database
Selects Timetable Builder
Request courses in students timetable builder
Courses in students timetable builder
Loop [For each course in t etable builder]
Course name, professor name, location, day and time

Figure 5. OpenTimetableBuilder SSD

7.2.5 Drop Registered Course

Use Case: DropRegisteredCourse
ID: UC-5
Brief description: The Student drops a registered course.
Actor(s): Student
Preconditions: 1. The Student must be logged in to the Course Registrator. 2. The Student must be registered in at least one course.
Main flow: 1. The Student selects Timetable Builder. 2. The Student selects “Drop Course”. 3. The Student is prompted to confirm their choice. 4. If the Student confirms their choice: 4.1. The Student will see confirmation that they have been removed from the course. 4.2. The Student will be removed from the course. 5. If the Student does not confirm the choice: 5.1. The Student will see no change to the Timetable Builder.
Postconditions:

1. The Student drops a registered course.
Alternative flow(s): None

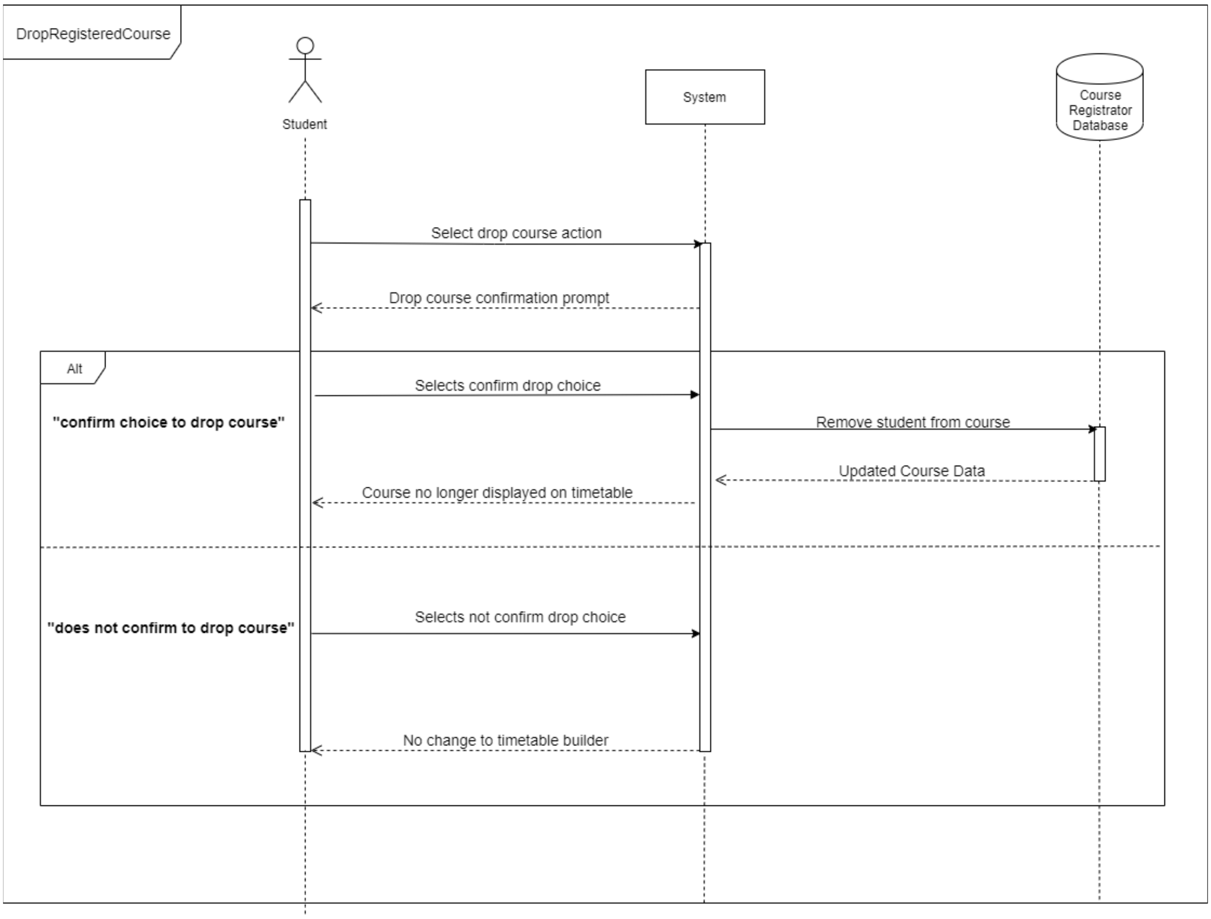

DropRegisteredCourse
Student
System
Course Registrator Database
Select drop course action
Drop course confirmation prompt
Alt
"confirm choice to drop course"
Selects confirm drop choice
Remove student from course
Updated Course Data
Course no longer displayed on timetable
"does not confirm to drop course"
Selects not confirm drop choice
No change to timetable builder

Figure 6. DropRegisteredCourse SSD

7.2.6 View Waitlist Details

Use Case: ViewWaitlistDetails
ID: UC-6
Brief description: The Student views waitlist details for each course on the Trackable Waitlist.
Actor(s): Student
Preconditions: 1. The Student must be logged in to the Course Registrator. 2. The Student is on the waitlist of at least one course.
Main flow: 1. The Student navigates to the Trackable Waitlist. 2. For each course on the Trackable Waitlist: 2.1. The Student sees the waitlist's maximum capacity. 2.2. The Student sees the current number of people on the waitlist. 2.3. The Student sees their position on the waitlist.

2.4. If the Student has a spot offered for this course. 2.4.1. The Student sees an option to register for a course. 2.5. Else 2.5.1. The Student sees an option to remove themselves from the waitlist.
Postconditions: None.
Alternative flow(s): None.

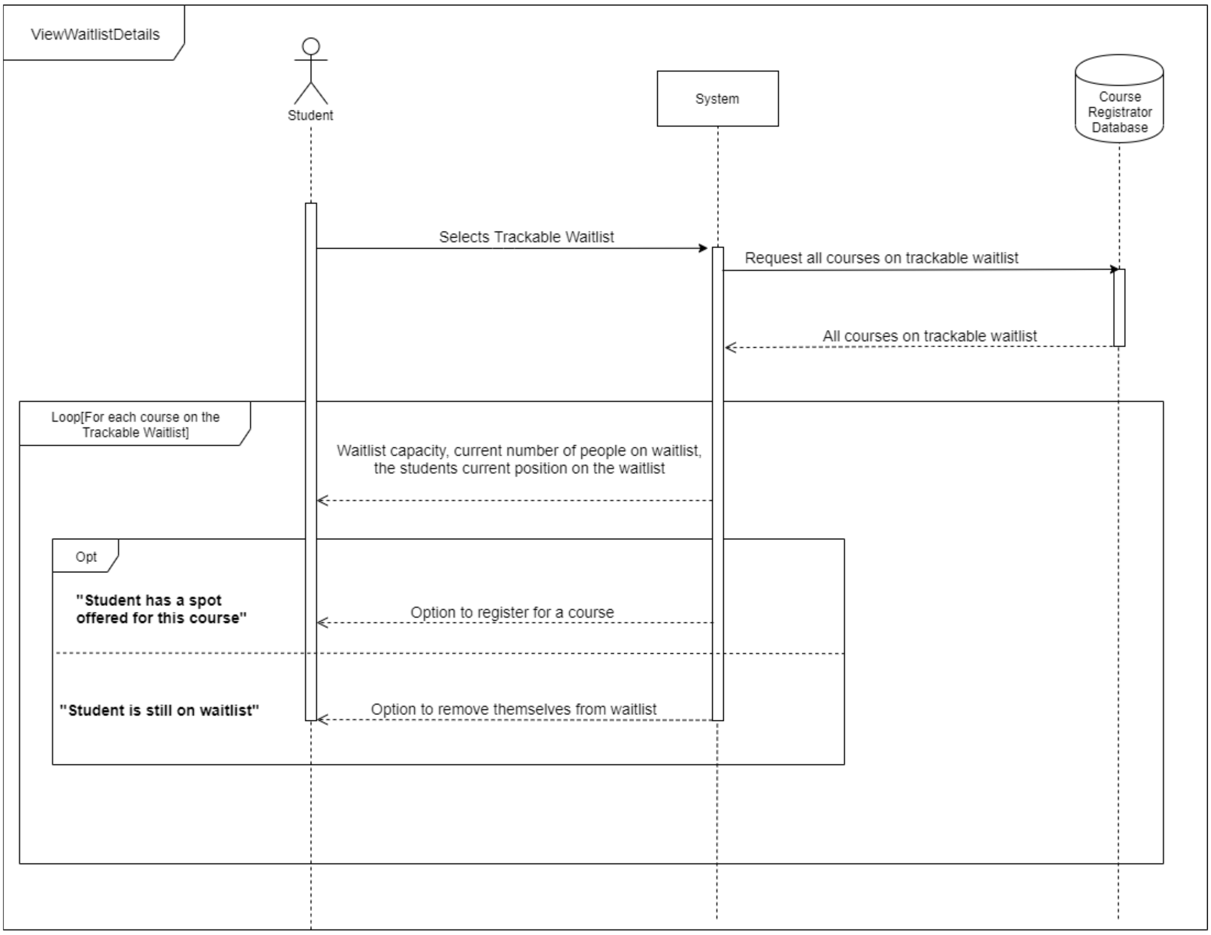

ViewWaitlistDetails
Student
System
Course Registrator Database
Selects Trackable Waitlist
Request all courses on trackable waitlist
All courses on trackable waitlist
Loop[For each course on the Trackable Waitlist]
Waitlist capacity, current number of people on waitlist, the students current position on the waitlist
Opt
"Student has a spot offered for this course"
Option to register for a course
"Student is still on waitlist"
Option to remove themselves from waitlist

Figure 7. ViewWaitlistDetails SSD

7.2.7 Accept Waitlist Offer

Use Case: AcceptWaitlistOffer
ID: UC-7
Brief description: The Student accepts an offer for course registration from the Trackable Waitlist.
Actor(s): Student

Preconditions:

1. The course must be on the Trackable Waitlist.
2. The Student must be logged in to the Course Registrator.
3. The Student must be at the top of the waitlist.
4. The course must have an open spot.

Main flow:

1. The Student receives a notification about a course registration offer from the Trackable Waitlist.
2. include(ViewWaitlistDetails)
3. The Student views the course offer specified in the notification.
4. The Student accepts the offer to register for the course.
5. The Student is given confirmation that they have been accepted into the course.
6. The Student sees their timetable display the accepted course.

Postconditions:

1. The Student is added to the course.
2. The Student is removed from the course's waitlist.
3. The course is removed from the Trackable Waitlist.

Alternative flow(s): None

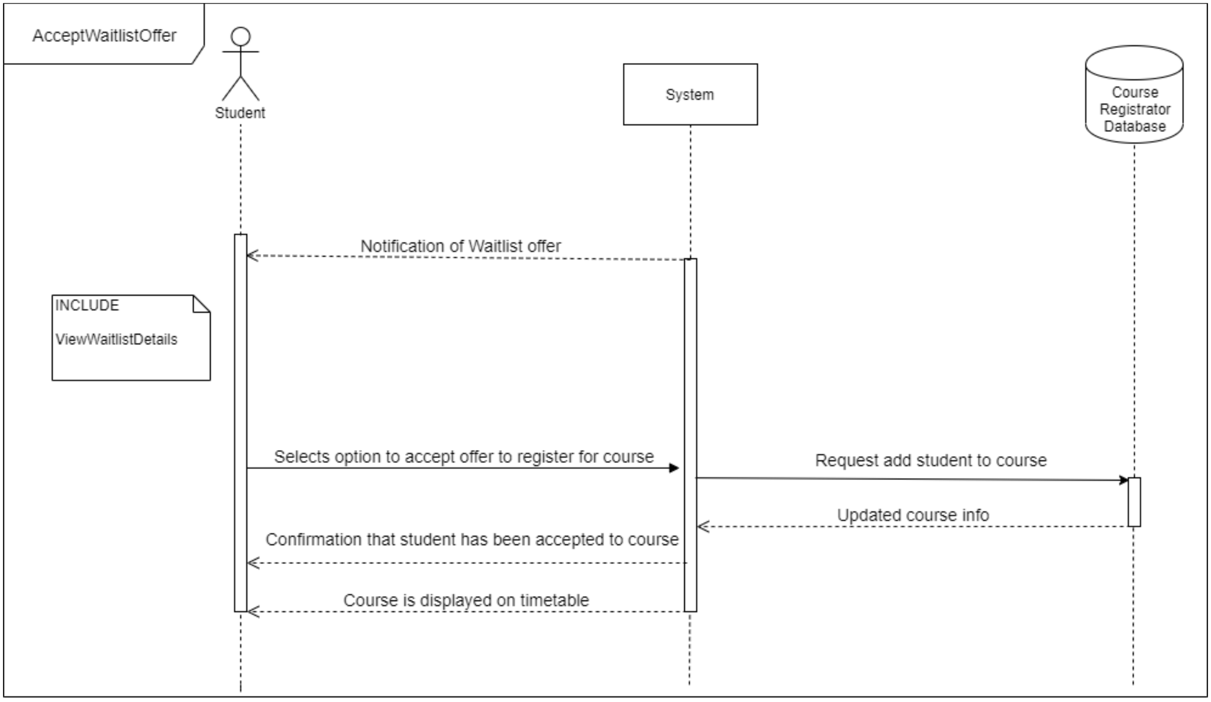

AcceptWaitlistOffer
Student
System
Course Registrator Database
Notification of Waitlist offer
INCLUDE
ViewWaitlistDetails
Selects option to accept offer to register for course
Request add student to course
Updated course info
Confirmation that student has been accepted to course
Course is displayed on timetable

Figure 8. AcceptWaitlistOffer SSD

7.2.8 Drop From Waitlist

Use Case: DropFromWaitlist

ID: UC-8

Brief description: The Student drops themselves from a course's waitlist.

Actor(s): Student

Preconditions:

1. The Student is logged in to the Course Registrator.
2. The course the Student wants to drop is on the Trackable Waitlist.

Main flow:

1. include(ViewWaitlistDetails).
2. The Student sees the course they would like to drop.
3. The Student drops from the course.
4. The Student is shown a prompt by the Trackable Waitlist to confirm dropping the course.
5. If the Student submits their decision to be removed from the waitlist.
   5.1. The Student is given confirmation that they have been dropped from the waitlist.
6. Else
   6.1. The Student remains on the waitlist

Postconditions:

1. The Student is removed from the course's waitlist.
2. The course is removed from the Trackable Waitlist.

Alternative flow(s): None

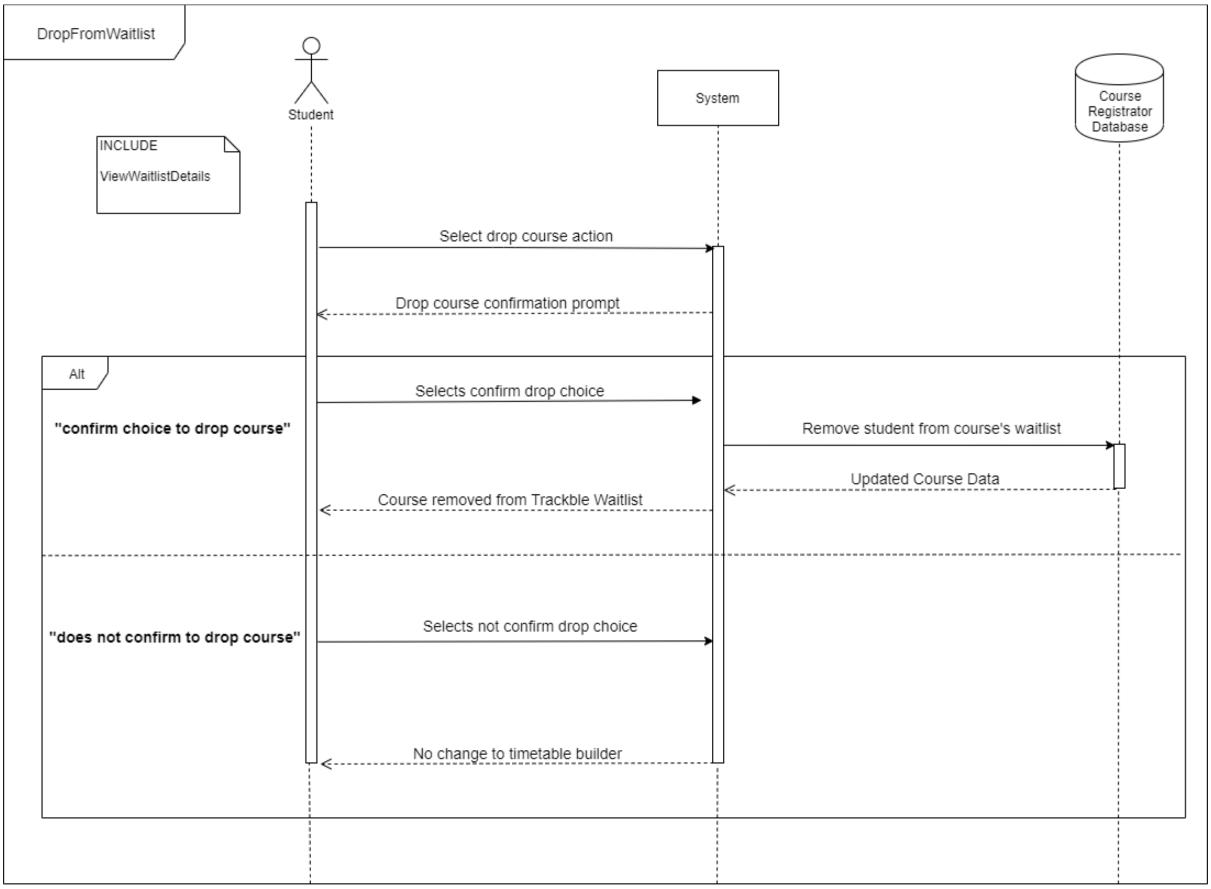

DropFromWaitlist
Student
System
Course Registrator Database
INCLUDE
ViewWaitlistDetails
Select drop course action
Drop course confirmation prompt
Alt
"confirm choice to drop course"
Selects confirm drop choice
Remove student from course's waitlist
Updated Course Data
Course removed from Trackble Waitlist
"does not confirm to drop course"
Selects not confirm drop choice
No change to timetable builder

Figure 9. DropFromWaitlist SSD

7.2.9 Look Up Help

Use Case: LookUpHelp
ID: UC-9
Brief description: The Student searches for answers on the Help Page.
Actor(s): Student
Preconditions: 1. The Student must be logged in on the Course Registrator.
Main flow: 1. The Student selects “Help”. 2. The Student sees the Help Page. 3. The Student searches for a resource (FAQ or Tutorial) to answer their question. 4. If the Student finds a resource: 4.1. The Student selects the resource 4.2. The Student views common solutions to problems with the selected resource.
Postconditions: None
Alternative flow(s): 3b). The Student views contact details for an Administrator. 3.1b). For each Administrator, the Student sees: 3.1.1b). The Administrator’s name, email, phone number, location and work hours.

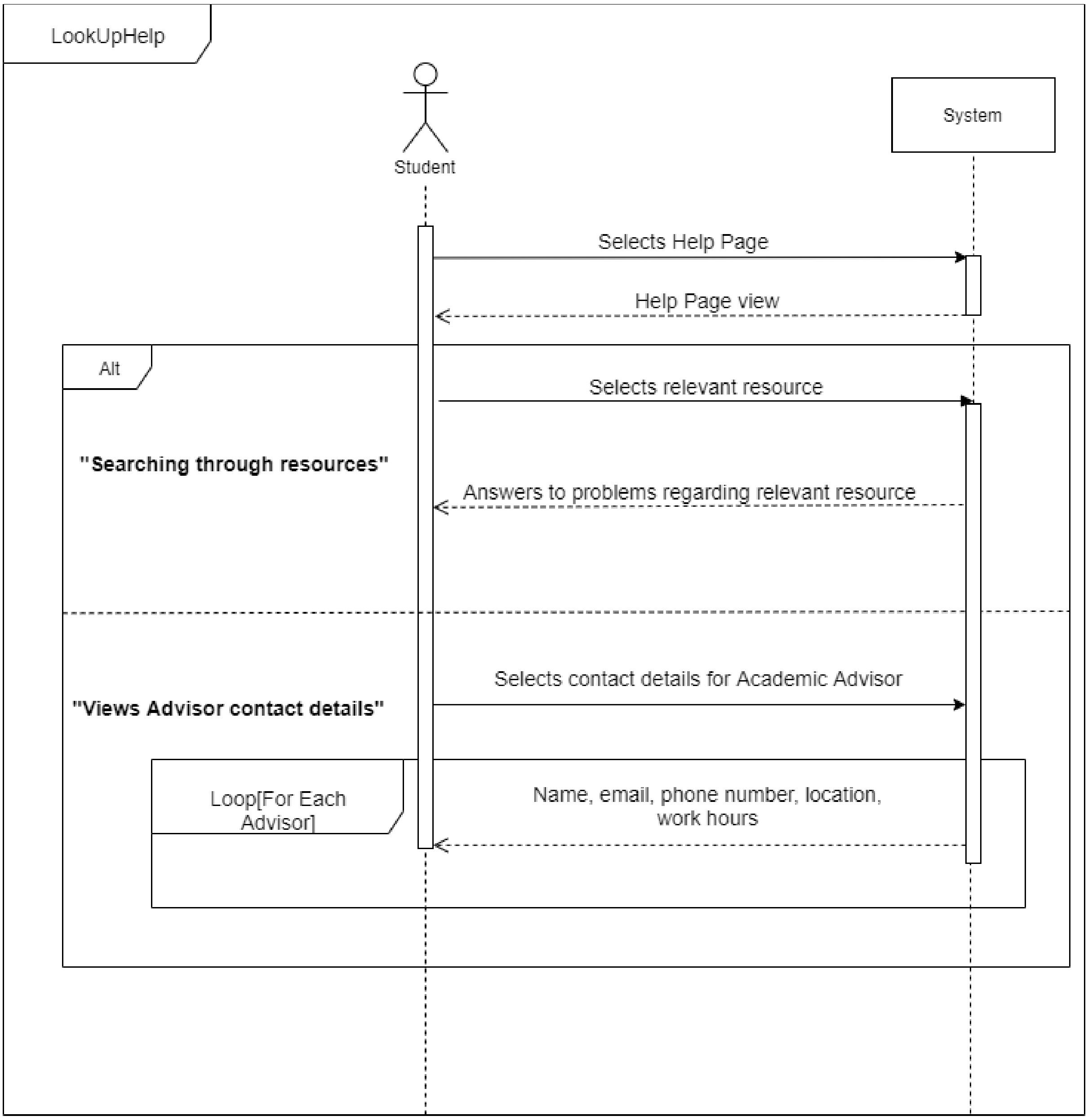

LookUpHelp
Student
System
Selects Help Page
Help Page view
Alt
Selects relevant resource
"Searching through resources"
Answers to problems regarding relevant resource
Selects contact details for Academic Advisor
"Views Advisor contact details"
Loop[For Each Advisor]
Name, email, phone number, location, work hours

Figure 10. LookUpHelp SSD

7.2.10 View Program Progress Checker

Use Case: ViewProgramProgressChecker
ID: UC-10
Brief description: The Student navigates to the Program Progress Checker.
Actor(s): Student
Preconditions: 1. The Student must be logged into Course Registrator. 2. UL's program declaration page exists.

Main flow:

1. The Student navigates to the Program Progress Checker section of the Course Registrator.
2. The Student observes a list of all courses they have completed and a list of all courses they must complete to graduate.

Postconditions: None

Alternative flow(s): None

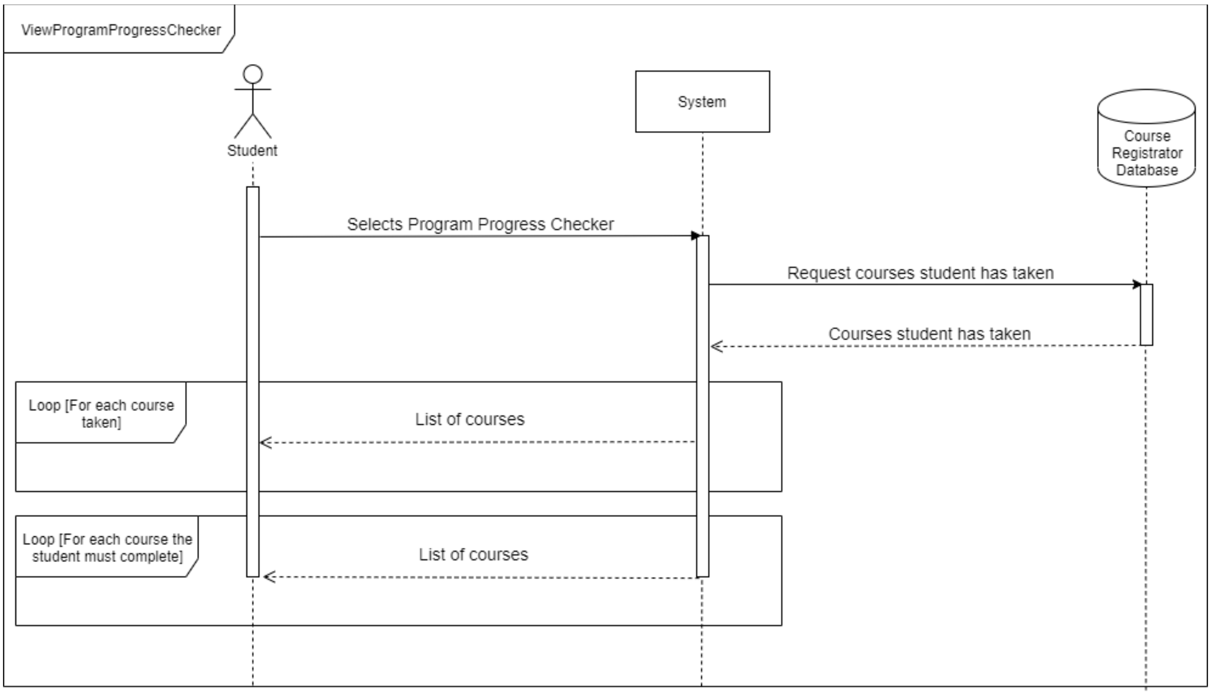

ViewProgramProgressChecker
Student
System
Course Registrator Database
Selects Program Progress Checker
Request courses student has taken
Courses student has taken
Loop [For each course taken]
List of courses
Loop [For each course the student must complete]
List of courses

Figure 11. ViewProgramProgressChecker SSD

7.2.11 Search Student

Use Case: SearchStudent

ID: UC-11

Brief description: An administrative user searches for a Student.

Actor(s): Administrator

Preconditions:

1. The Administrator must be logged in to the Course Registrator.

Main flow:

1. The Administrator selects the student search.
2. While the Administrator searches for a Student:
    - 2.1. Input search by name or studentID:
    - 2.2. If a Student is matching name, declared program, or studentID in the database:
        - 2.2.1. The Administrator views all matching Students.
    - 2.3. Else:

2.3.1. The Administrator sees a “No Students matched your search” message.
Postconditions: The Administrator is provided with the Student’s details.
Alternative flow(s): None

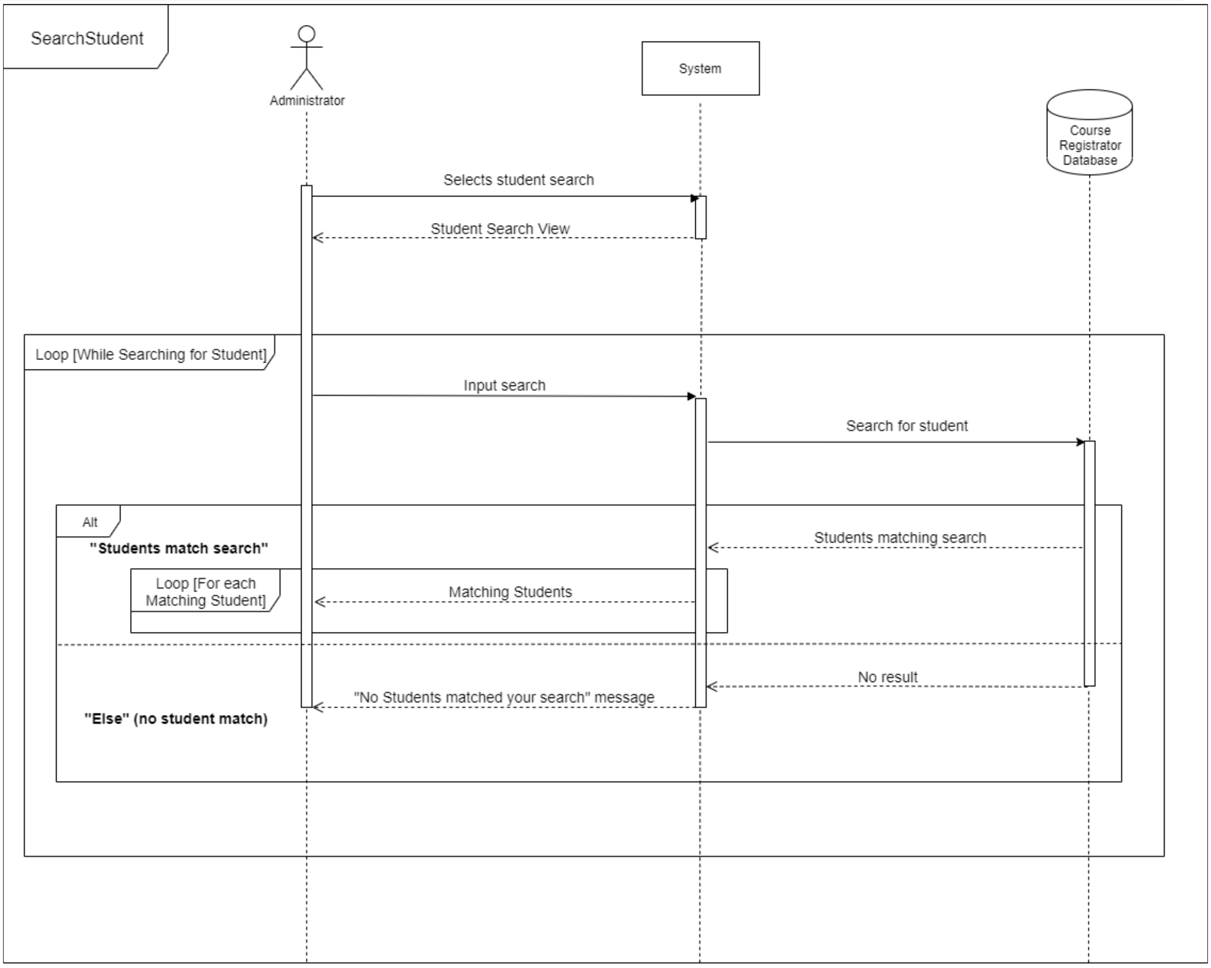


Figure 12. Search Student SSD

7.2.12 Administrator Access Student Program Progress Checker

Use Case: AdministratorAccessStudentProgramProgressChecker
ID: UC-12
Brief description: An administrator accesses a student’s Program Progress Checker.
Actor(s): Administrator
Preconditions: None
Main flow: 1. include(SearchStudent) 2. The Administrator selects the Student they seek to aid. 3. The Administrator navigates to the Student’s Program Progress Checker.

4. The Administrator sees the Student’s Program Progress Checker.
Postconditions: None
Alternative flow(s): None

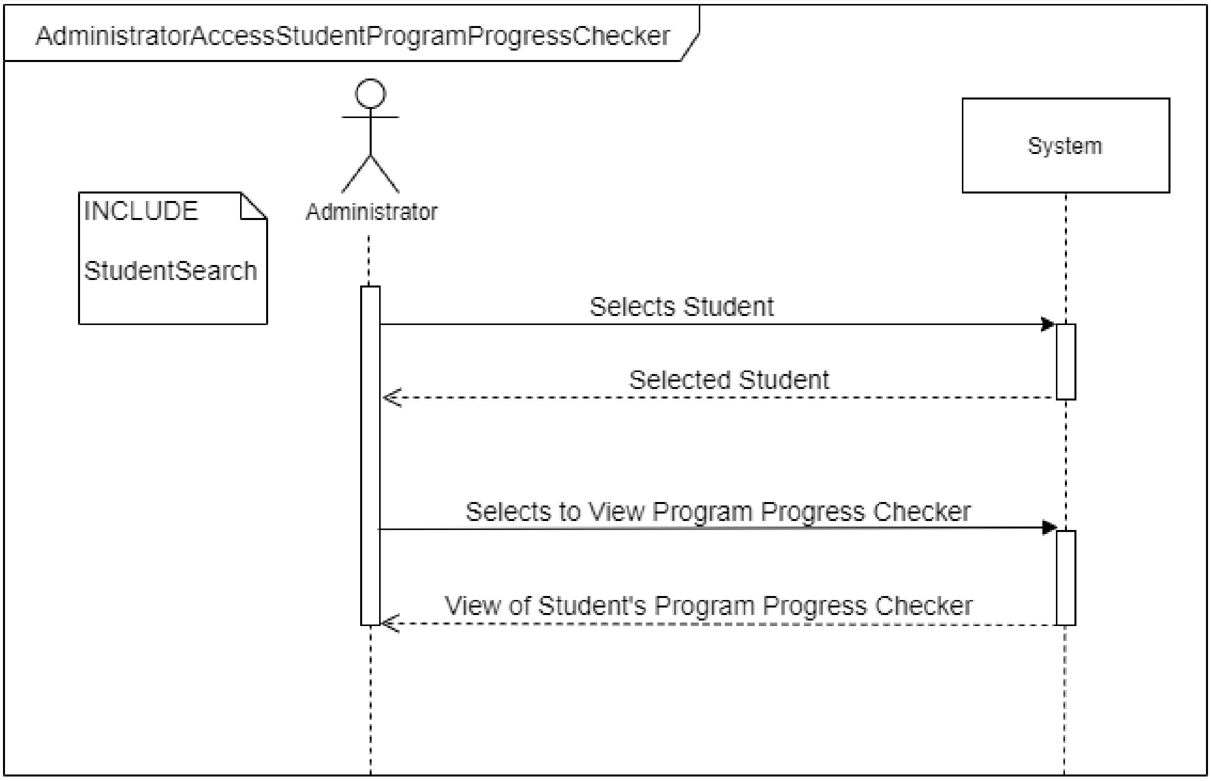

AdministratorAccessStudentProgramProgressChecker
INCLUDE
StudentSearch
Administrator
System
Selects Student
Selected Student
Selects to View Program Progress Checker
View of Student's Program Progress Checker

Figure 13. AdministratorAccessStudentProgramProgressChecker SSD

7.2.13 View Course Info In Program Progress Checker

Use Case: ViewCourseInfoInProgressChecker
ID: UC-13
Brief description: The Student is on the Program Progress Checker and is trying to learn more about a course they must take.
Actor(s): Student
Preconditions: 1. The Student is logged in on the Course Registrator. 2. UL has a course description page for all courses offered. 3. The Student has declared their program.
Main flow: 1. include(ViewProgramProgressChecker). 2. The Student selects the course that they want to learn more about. 3. The Student sees the course details as published by UL.

Postconditions: None
Alternative flow(s): None

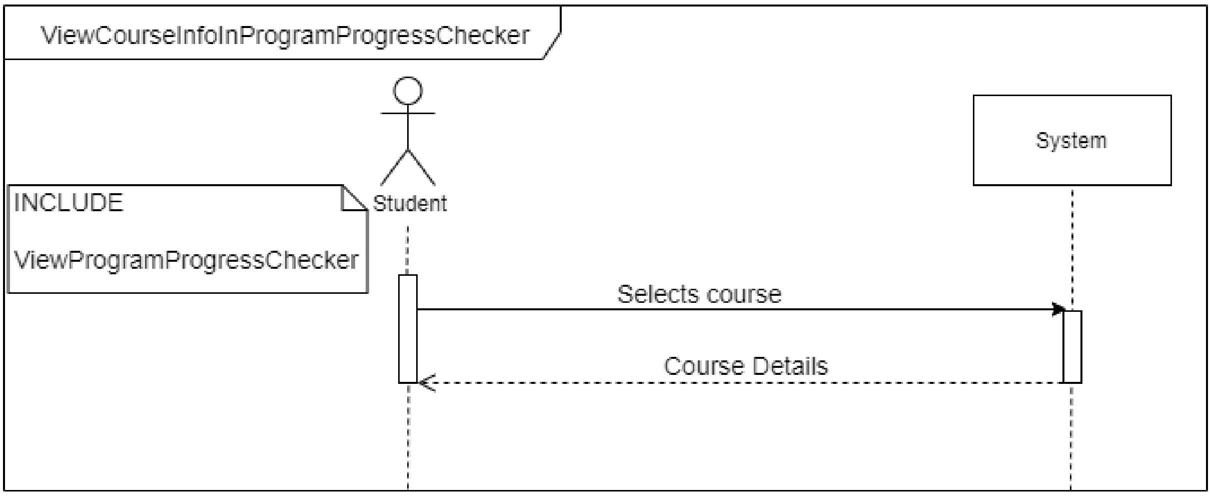

ViewCourseInfoInProgramProgressChecker
INCLUDE
ViewProgramProgressChecker
Student
System
Selects course
Course Details

Figure 14. ViewCourseInfoInProgramProgressChecker SSD

7.2.14 Add Student To Course

Use Case: AddStudentToCourse
ID: UC-14
Brief description: An Administrator adds a Student to a course.
Actor(s): Administrator
Preconditions: 1. The Student must be registered on the Course Registrator. 2. The Administrator must be logged in to the Course Registrator. 3. The Administrator must know the course name to be added to the students timetable.
Main flow: 1. include(SearchStudent). 2. The Administrator selects a student. 3. The Administrator navigates to the Student's Timetable Builder. 4. The Administrator enters the course name of the course to be added to the Student's timetable. 5. The Administrator adds the course. 6. The Administrator is prompted to confirm adding the course. 7. The Administrator confirms the prompt. 8. The Administrator sees the course on the Student's Timetable Builder. 9. The Administrator notifies the Student that the course has been added to their Timetable Builder.
Postconditions:

1. The Student is added to the course.
Alternative flow(s):

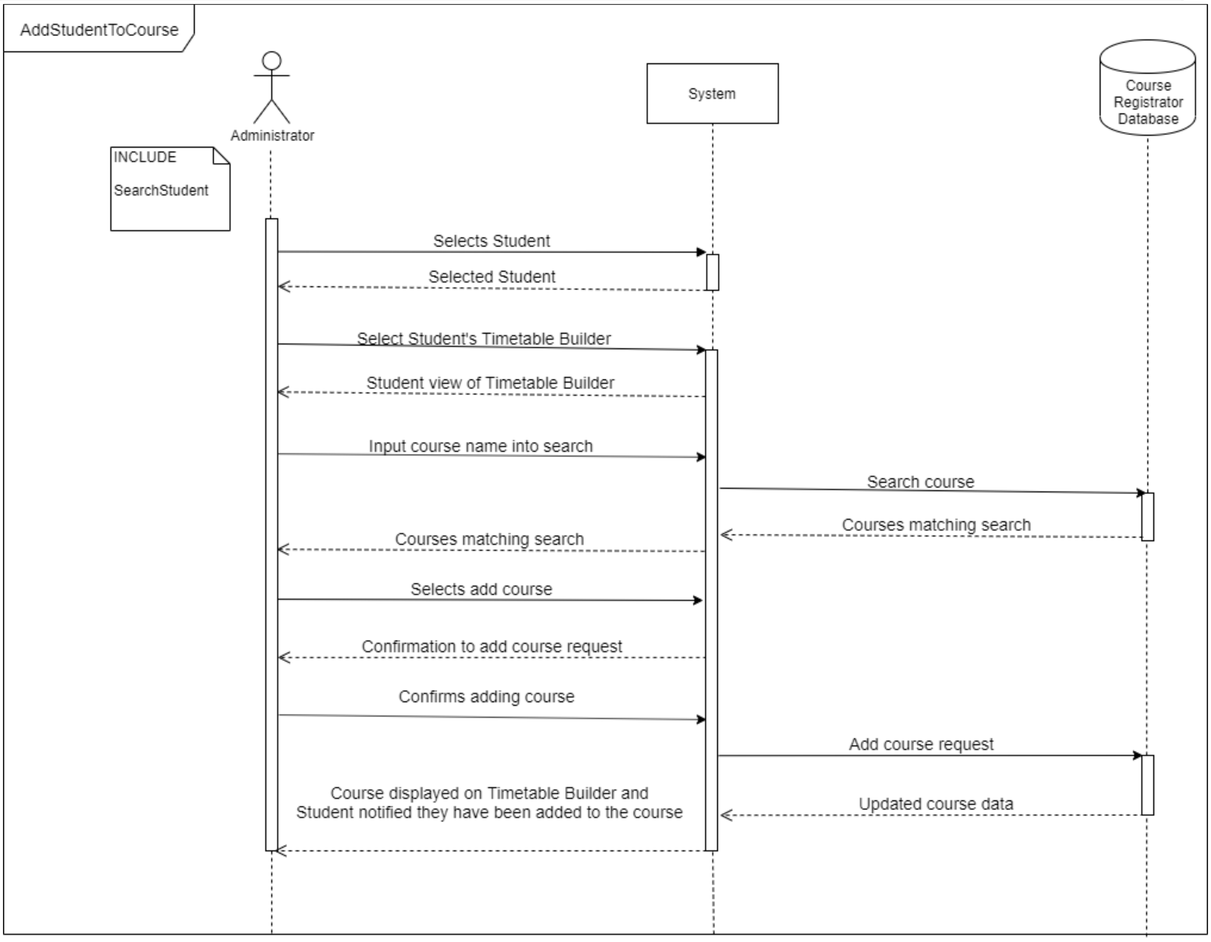

AddStudentToCourse
Administrator
System
Course Registrator Database
INCLUDE
SearchStudent
Selects Student
Selected Student
Select Student's Timetable Builder
Student view of Timetable Builder
Input course name into search
Search course
Courses matching search
Courses matching search
Selects add course
Confirmation to add course request
Confirms adding course
Add course request
Course displayed on Timetable Builder and Student notified they have been added to the course
Updated course data

Figure 15. AddStudentToCourse SSD

7.2.15 Remove Student From Course

Use Case: RemoveStudentFromCourse
ID: UC-15
Brief description: An Administrator removes a Student from a course.
Actor(s): Administrator
Preconditions: 1. The Student must be registered in a course. 2. The Administrator must be logged in to the Course Registrator.
Main flow: 1. include(StudentSearch). 2. The Administrator navigates to the Student's Timetable Builder. 3. The Administrator selects the course to be removed in the Student's Timetable Builder. 4. The Administrator is prompted to confirm the removal of the student from the course.

5. The Administrator sees the course is no longer on the Student’s Timetable Builder.
6. The Administrator notifies the Student that the course has been removed from their Timetable Builder.

Postconditions:

1. The Student is removed from the course.

Alternative flow(s): None

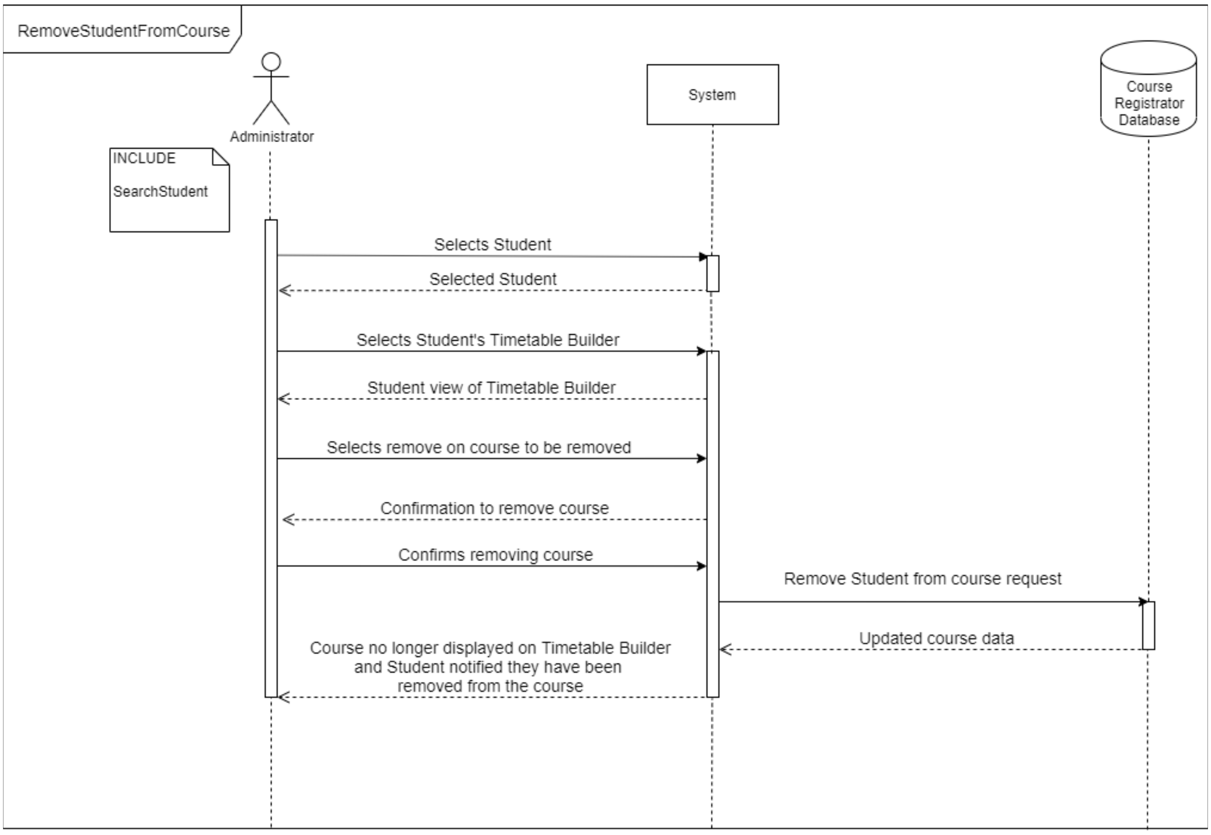

RemoveStudentFromCourse
Administrator
System
Course Registrator Database
INCLUDE
SearchStudent
Selects Student
Selected Student
Selects Student's Timetable Builder
Student view of Timetable Builder
Selects remove on course to be removed
Confirmation to remove course
Confirms removing course
Remove Student from course request
Updated course data
Course no longer displayed on Timetable Builder and Student notified they have been removed from the course

Figure 16. RemoveStudentFromCourse SSD

7.2.16 Edit Course

Use Case: EditCourse

ID: UC-16

Brief description: Administrator adds a new section to a course.

Actor(s): Administrator

Preconditions:

1. The Administrator must be logged in to the Course Registrator.

Main flow:

1. include(SearchCourse).
2. The Administrator selects to edit course details option.
3. While editing, for the selected course:
    3.1. The Administrator may edit the days and times for the course.

3.2. The Administrator may edit the location for the course.
3.3. The Administrator may edit the professor for the course.
3.4. The Administrator may edit the course size.
3.5. The Administrator may edit the waitlist size.
4. Administrator confirms the changes.

Postconditions:

1. The Administrator’s changes are reflected in the database.

Alternative flow(s): None

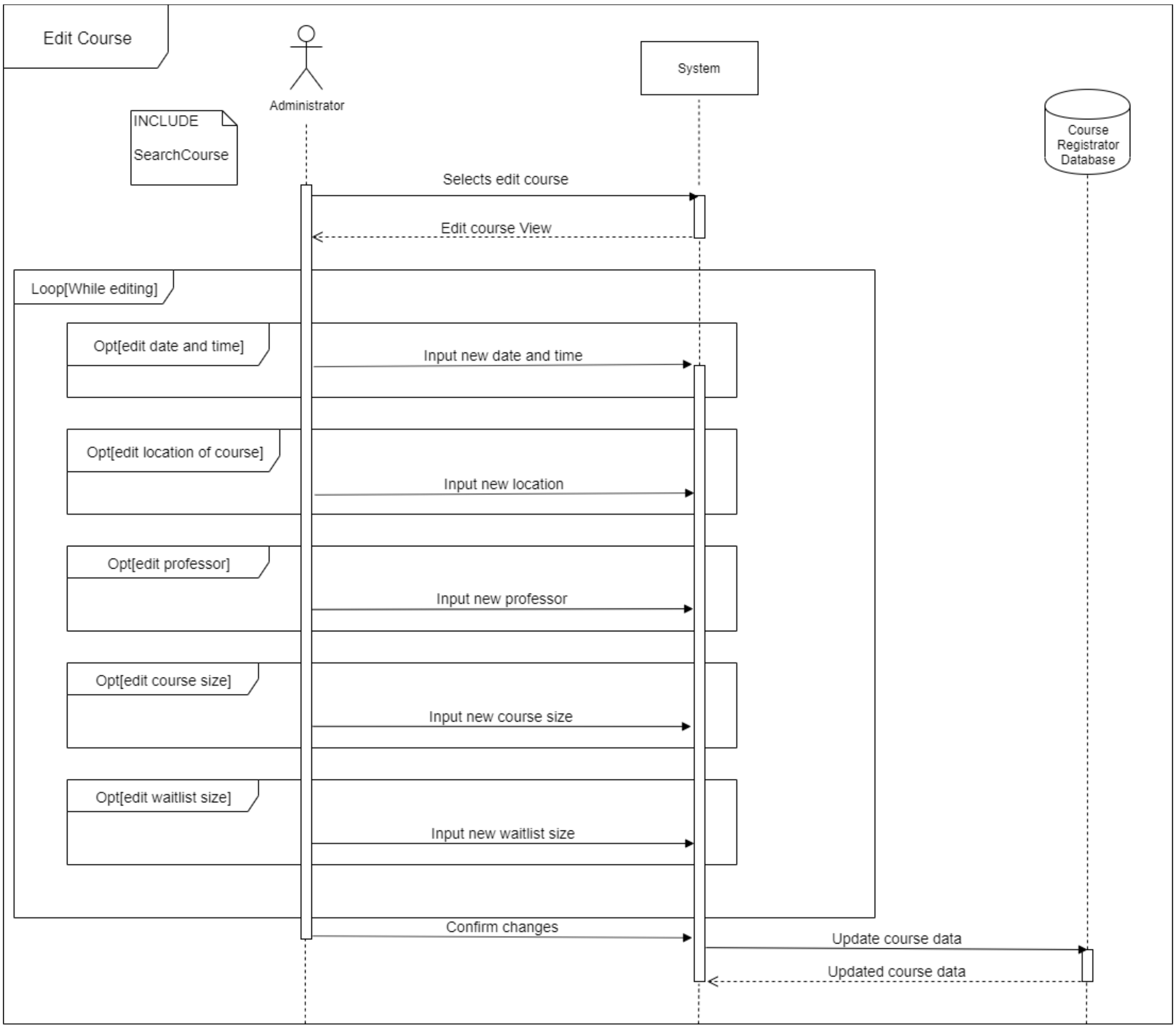


Figure 17. EditCourse SSD

8 Entity Relationship Diagram (ERD)

The following diagram details the relationship between each entity that exists in the project. In addition, each entity has details on what information they contain.

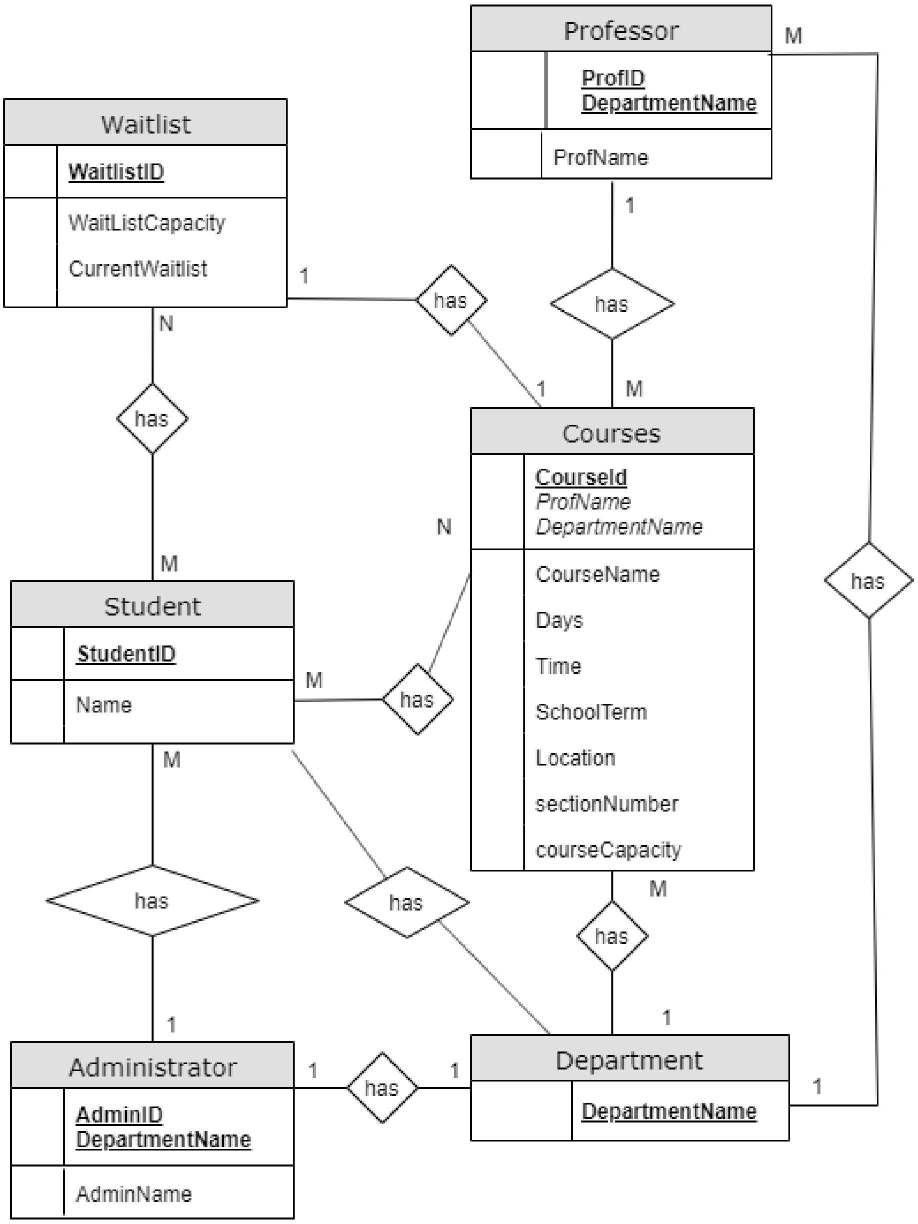


Figure 18. Entity Relationship Diagram

9 Data Flow Diagrams (DFDs)

Each of the diagrams below show how the data flows between users interacting with the Course Registrator and how the data flows between different processes within the system itself. The first diagram, DFD 0, shows a high level interaction where the system is shown as one process and not broken down into separate processes. DFD 1 is where the system is broken down into its component processes and DFD 2 is where one process is broken down and described in detail even further. The data dictionary describes the terms used in the diagrams.

9.1 DFD 0 (Context Diagram)

The DFD 0, or Context Diagram, describes the interaction between the Course Registrator and the three user classes at the most general level. The Course Registrator is shown as one process and all data flows in and out of the system are included regardless of the process they may belong to. The system is broken down into more specific processes in DFD 1 and DFD 2.

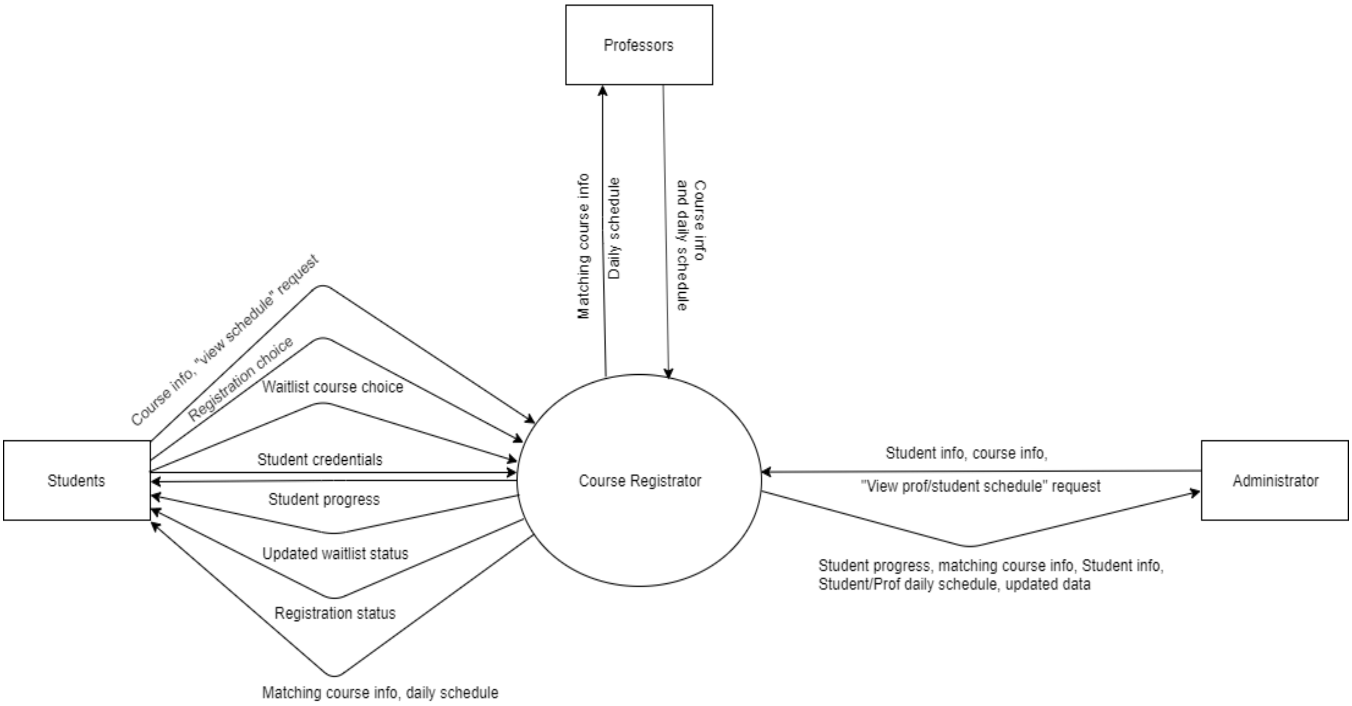


Figure 19. Data Flow Diagram 0 (Context Diagram)

9.1.1 Data Dictionary

Term	Information included
Course Info	Includes course subject, name, date, or time.

Matching Course Info	Includes course subject, name, date and time, location, and professor.
Registration Choice	Registration Choice can be adding a course or dropping a course.
Student Info	Includes student name, student number, student's declared program and courses a student is taking.
Student Progress	A list of all required courses a student needs to take based on their program. Completed courses are differentiated from courses a student is yet to complete.
Updated Data	Includes any data corresponding to the change to a student being added or dropped to a course. Data like new list of registered courses for the given student, or student was unable to be added to the course for x reason.
Waitlist Course Choice	Waitlist Course Choice for a student can be adding or dropping themselves from that course's waitlist, and accepting or rejecting a waitlisted course offer.

9.2 DFD 1

The DFD 1 describes the interaction between the system and the three user classes. The system contains view schedule, search, add or drop, change waitlist status, view program progress and course registration database.

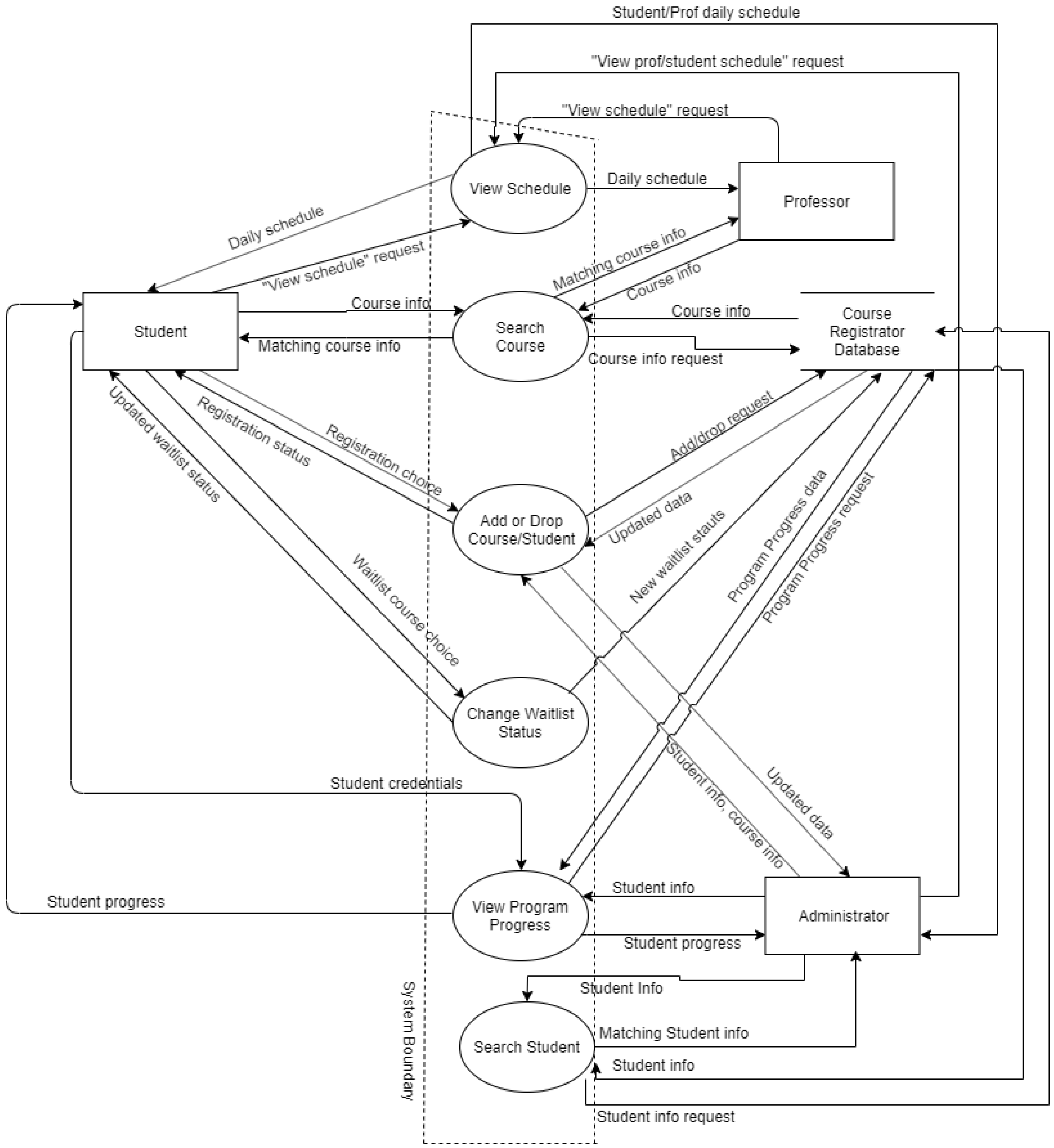


Figure 20. Data Flow Diagram 1

9.3 DFD 2

The DFD 2 describes how the Administrator and the Student add or drop courses and how that influences the database.

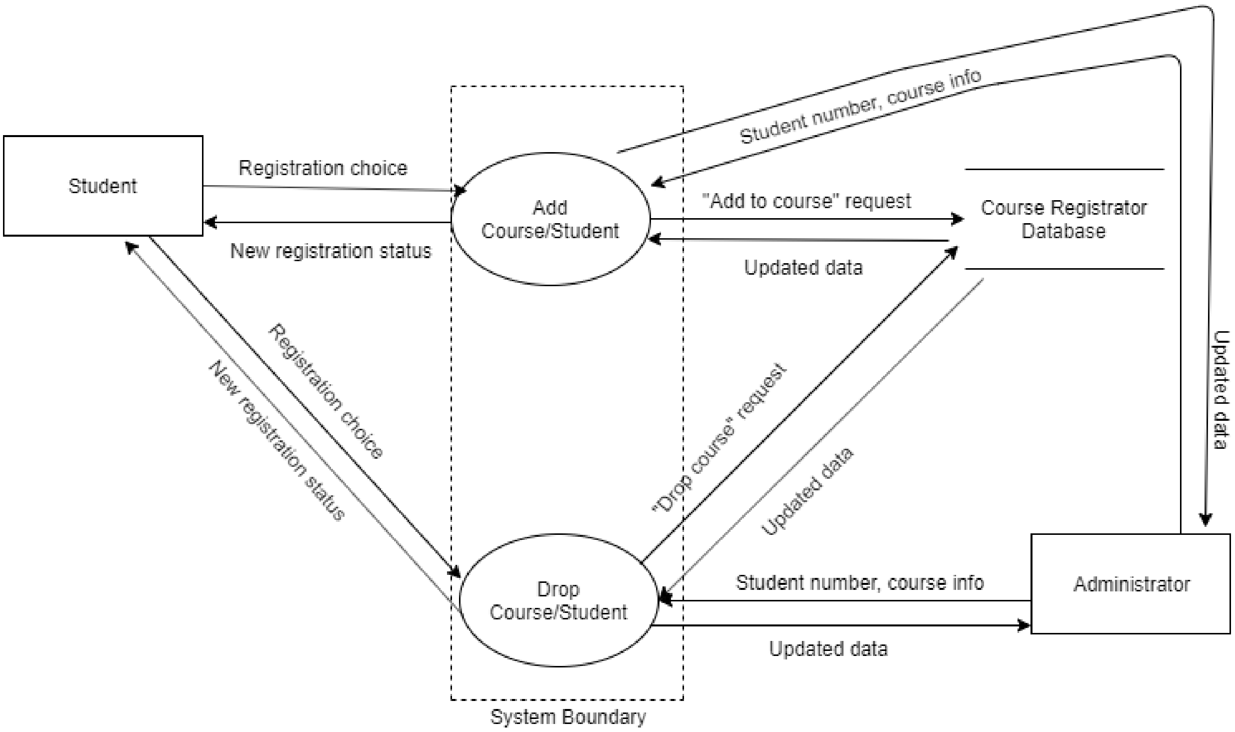


Figure 21. Data Flow Diagram 2

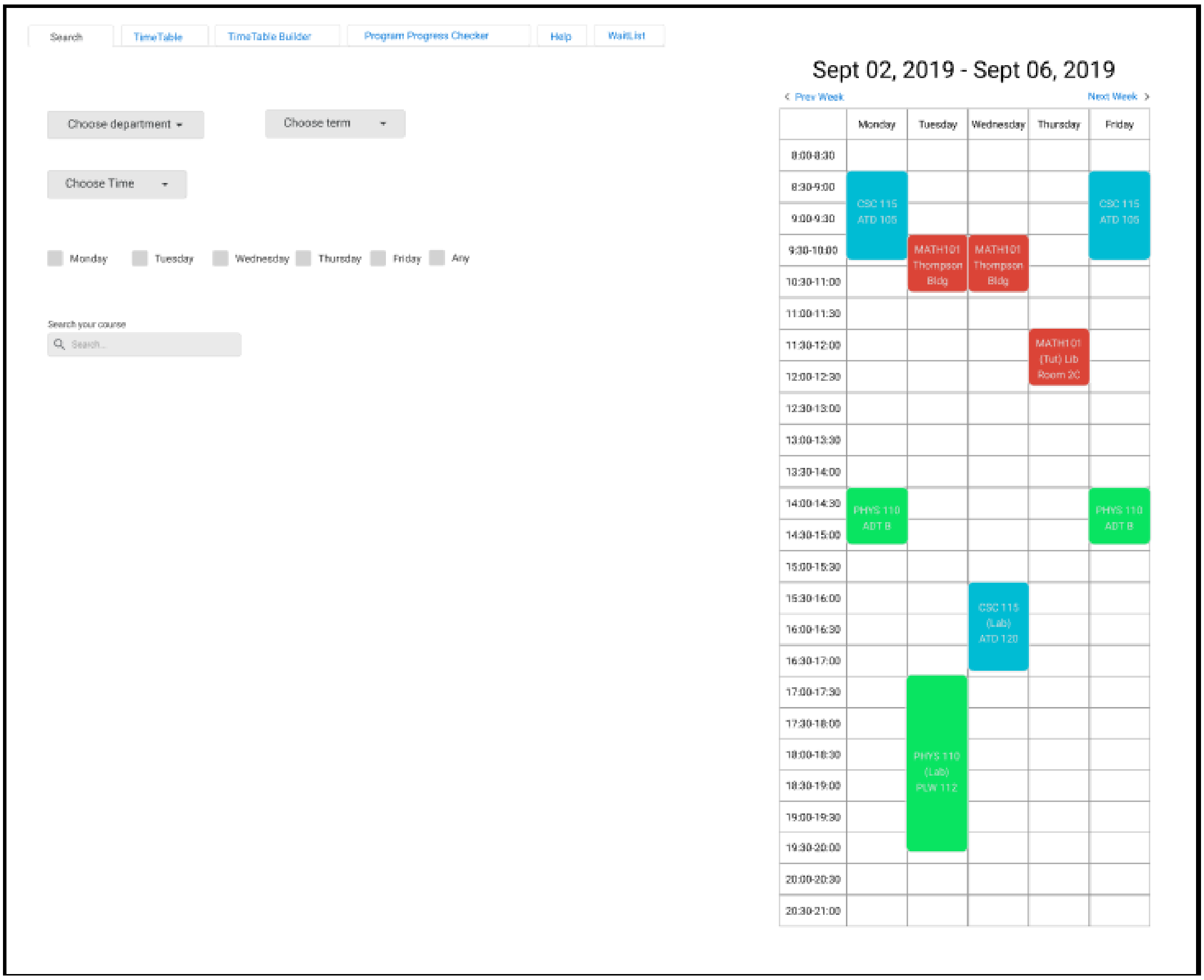
10 Storyboards

The following section details the storyboards for each of the main provided use cases. Included in each storyboard is a walkthrough of the different UI prototype screens.

10.1 User searches for a course and registers for a course or adds a course to waitlist (UC-2)

On the search screen, the user can input the necessary filters they wish to search with. Once the user clicks on search, they will be given a list of matching courses according to their search criteria. After that, each course will have an "Add" button available for the user to click that will add the user to that course if there are no errors. If the selected course has room, then the user will be added to the course when they select "Add". If there is no room in the selected course but room on the waitlist, when the user selects "Add" they will be given a prompt. The prompt will state that the user is going to be added to the waitlist for the selected course. The user then accepts the prompt for the changes to be saved.

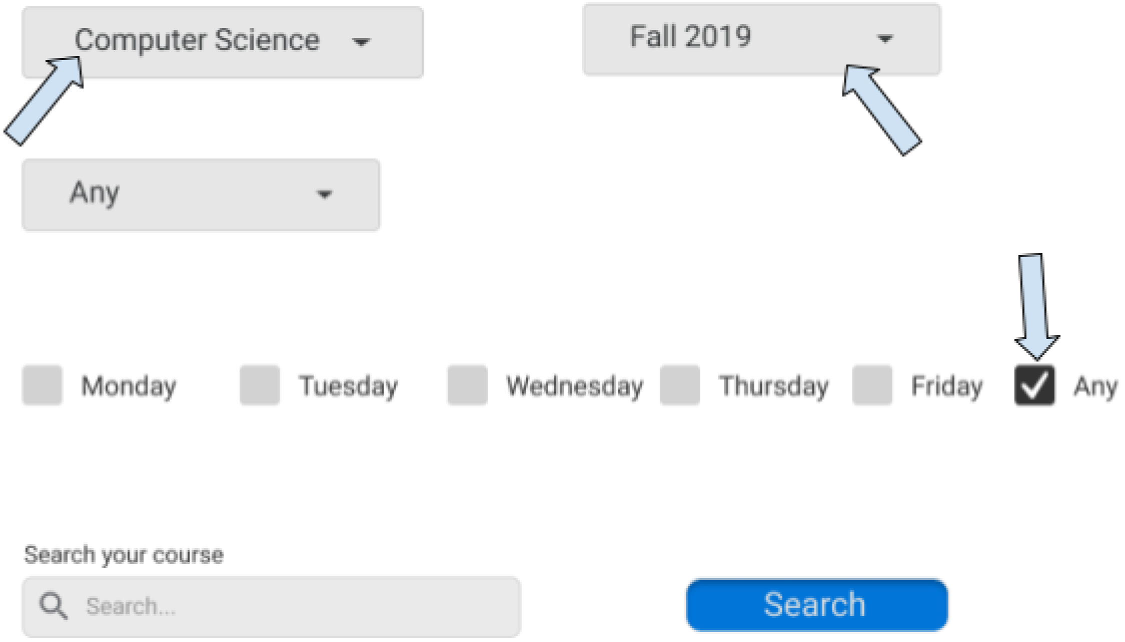
1. The user sees the course search screen.



Sept 02, 2019 - Sept 06, 2019

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					
17:00-17:30		PHYS 110 (Lab) PLW 112			
17:30-18:00					
18:00-18:30					
18:30-19:00					
19:00-19:30					
19:30-20:00					
20:00-20:30					
20:30-21:00					

2. The user uses filters such as the department, semester, and time for courses that they are searching for.



Computer Science ▾

Fall 2019 ▾

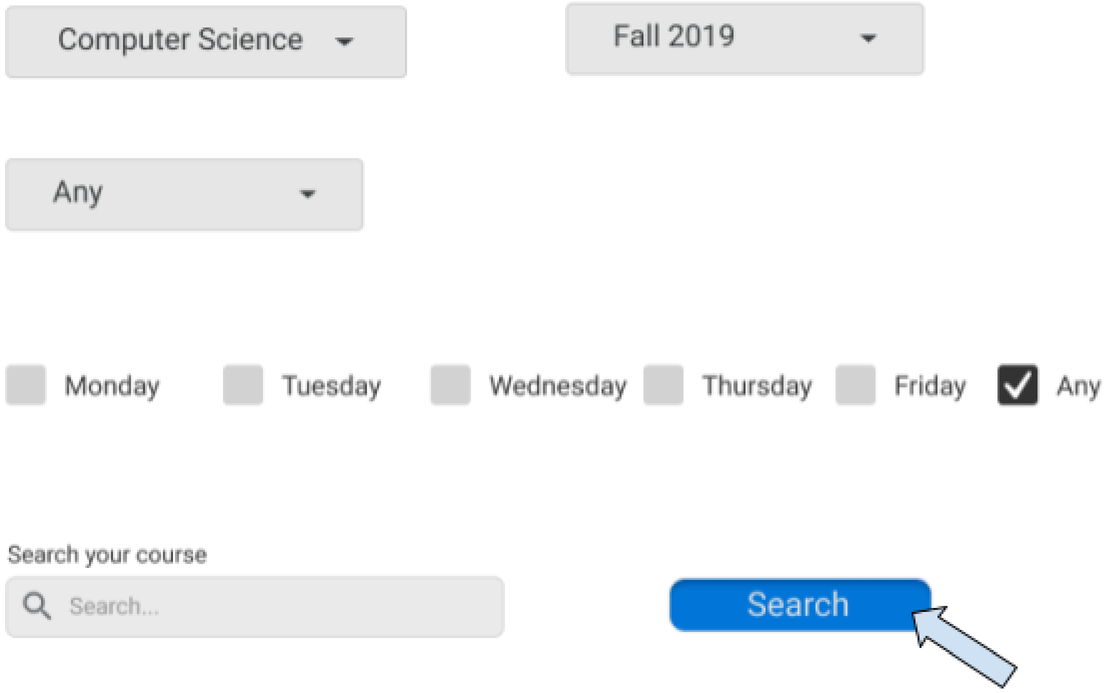
Any ▾

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday ☑ Any

Search your course

Search...

3. The user clicks on search after selecting filters.



Computer Science ▾

Fall 2019 ▾

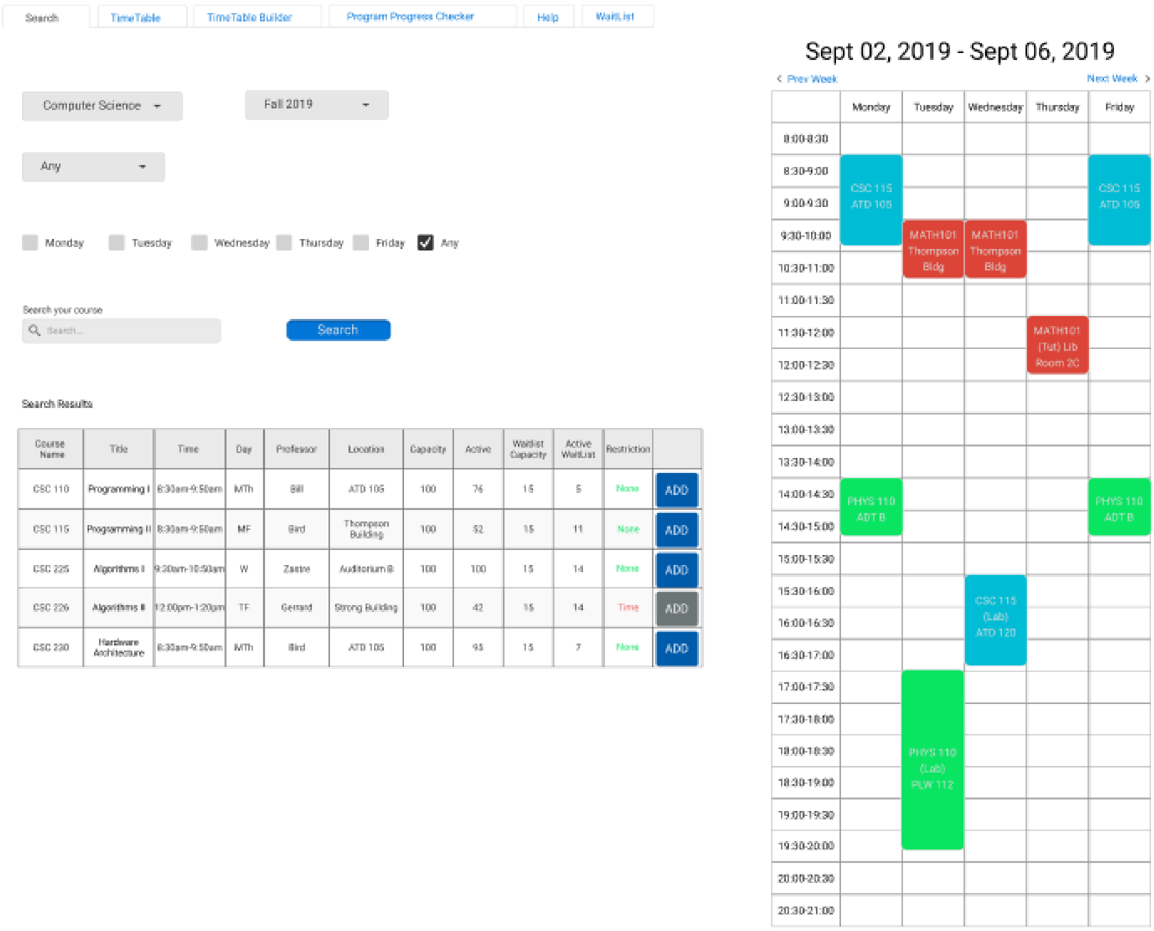
Any ▾

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday ☑ Any

Search your course

Search...

4. Each matching course and their respective course details are displayed.



Computer Science ▾

Fall 2019 ▾

Any ▾

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday ☑ Any

Search your course

Search...

Search Results

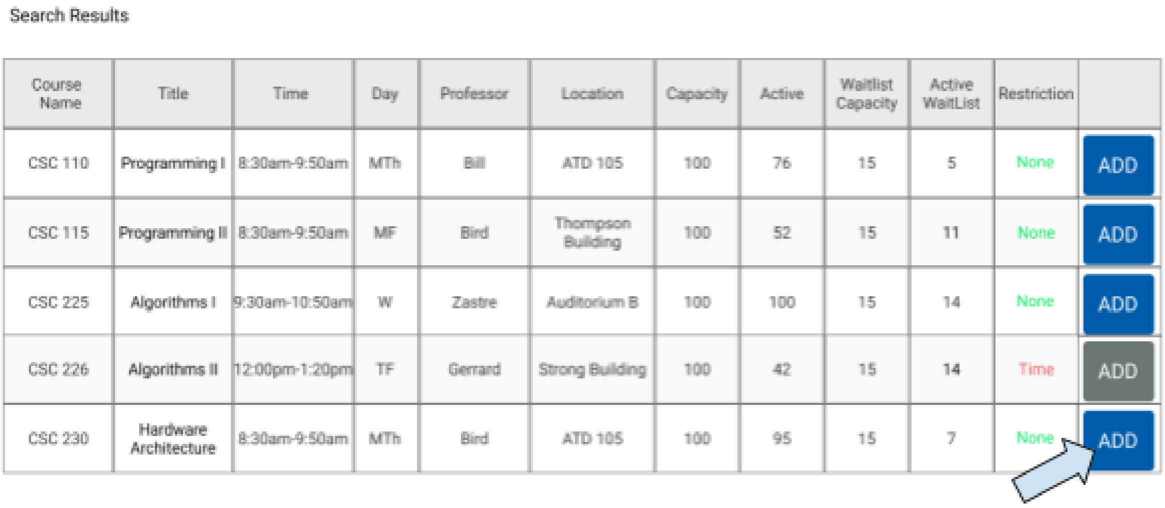
Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	ADD
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	ADD
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	14	None	ADD
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	Time	ADD
CSC 230	Hardware Architecture	8:30am-9:50am	MTh	Bird	ATD 105	100	95	15	7	None	ADD

Sept 02, 2019 - Sept 06, 2019

< Prev Week   Next Week >

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					
17:00-17:30		PHYS 110 (Lab) PLW 112			
17:30-18:00					
18:00-18:30					
18:30-19:00					
19:00-19:30					
19:30-20:00					
20:00-20:30					
20:30-21:00					

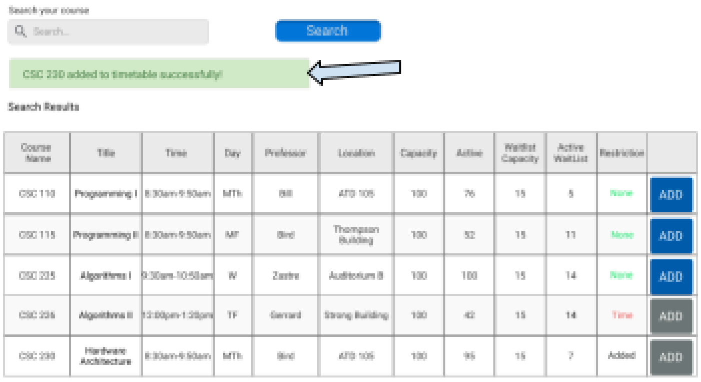
5. If the course section is not full:
   a. The user clicks the “Add” button. In this case, the user adds the course csc 230.



Search Results

Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	ADD
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	ADD
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	14	None	ADD
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	Time	ADD
CSC 230	Hardware Architecture	8:30am-9:50am	MTh	Bird	ATD 105	100	95	15	7	None	ADD

b. The user sees confirmation that the course was added and that the course was added to the timetable.



Search your course

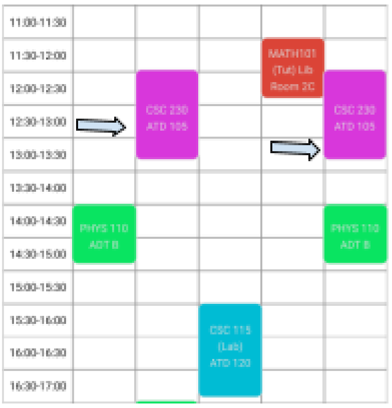
Search...

Search

CSC 230 added to timetable successfully!

Search Results

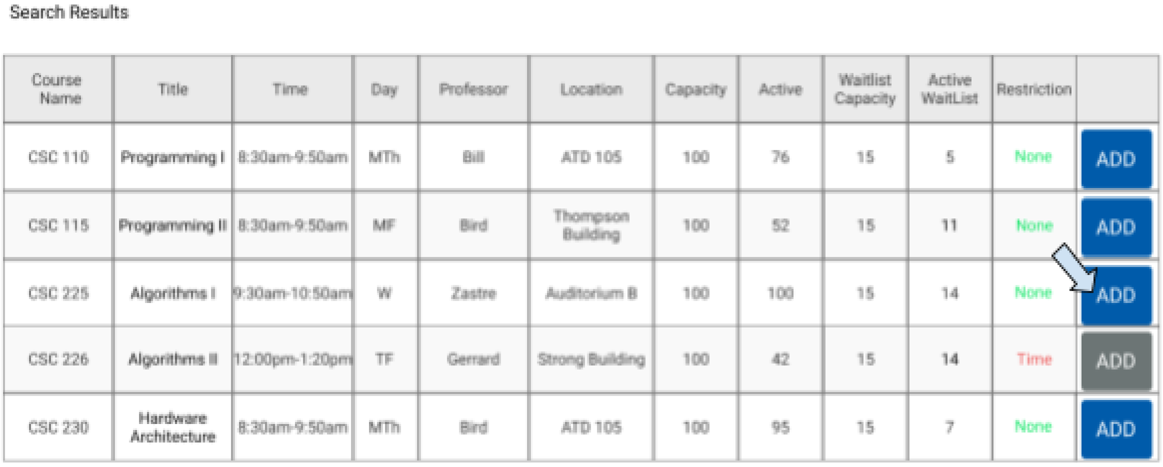
Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	ADD
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	ADD
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	14	None	ADD
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	Time	ADD
CSC 230	Hardware Architecture	8:30am-9:50am	MTh	Bird	ATD 105	100	95	15	7	Added	ADD



11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30		CSC 230 ATD 105			CSC 230 ATD 105
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					

6. If the course section is full and the active waitlist is not full:

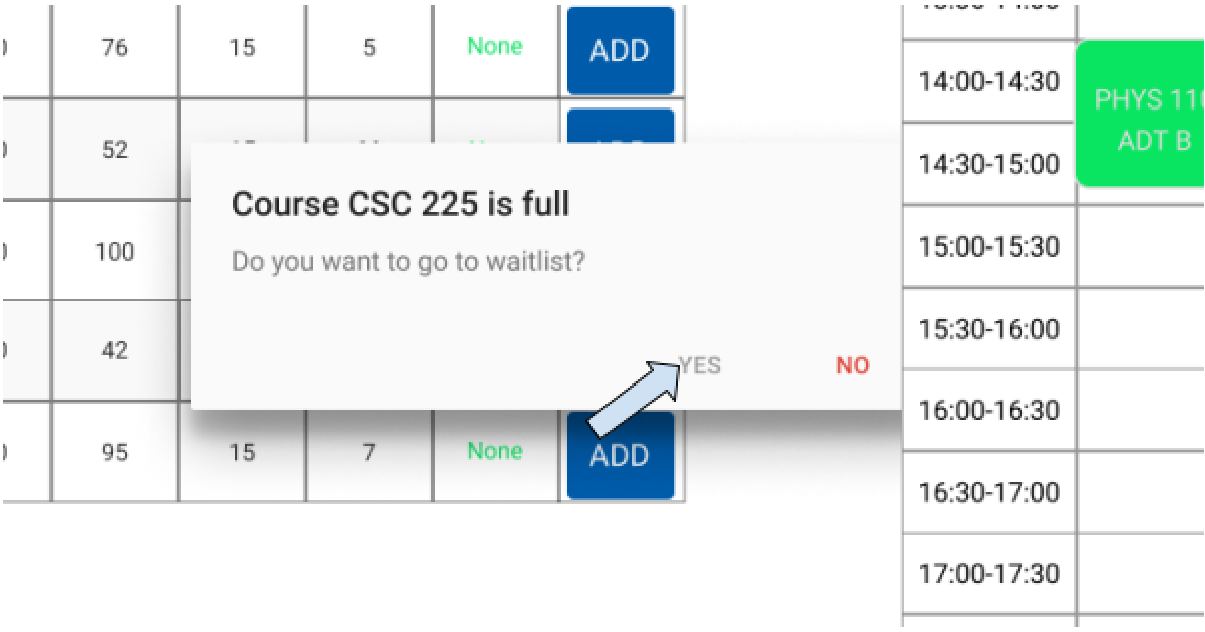
a. The user clicks the "add" button.



Search Results

Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	ADD
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	ADD
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	14	None	ADD
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	Time	ADD
CSC 230	Hardware Architecture	8:30am-9:50am	MTh	Bird	ATD 105	100	95	15	7	None	ADD

b. If the course is full, the user is prompted to add the course to the waitlist if the active waitlist is not full as shown in step 5. The case below shows the user clicking "Yes" and adding the course to waitlist.

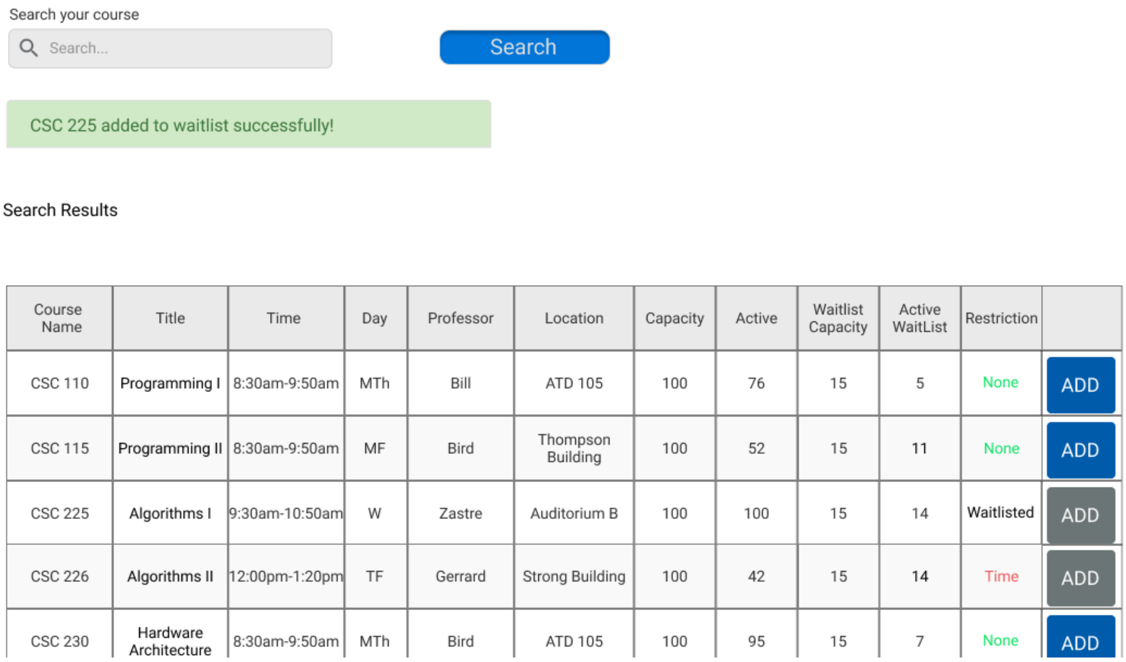


76	15	5	None	ADD
52				
100				
42				
95	15	7	None	ADD

Course CSC 225 is full

Do you want to go to waitlist?

c. A confirmation is displayed for a course that is added to the user’s waitlist.



CSC 225 added to waitlist successfully!

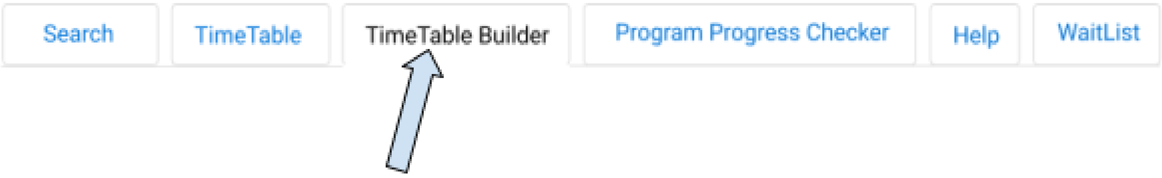
Search Results

Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	ADD
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	ADD
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	14	Waitlisted	ADD
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	Time	ADD
CSC 230	Hardware Architecture	8:30am-9:50am	MTh	Bird	ATD 105	100	95	15	7	None	ADD

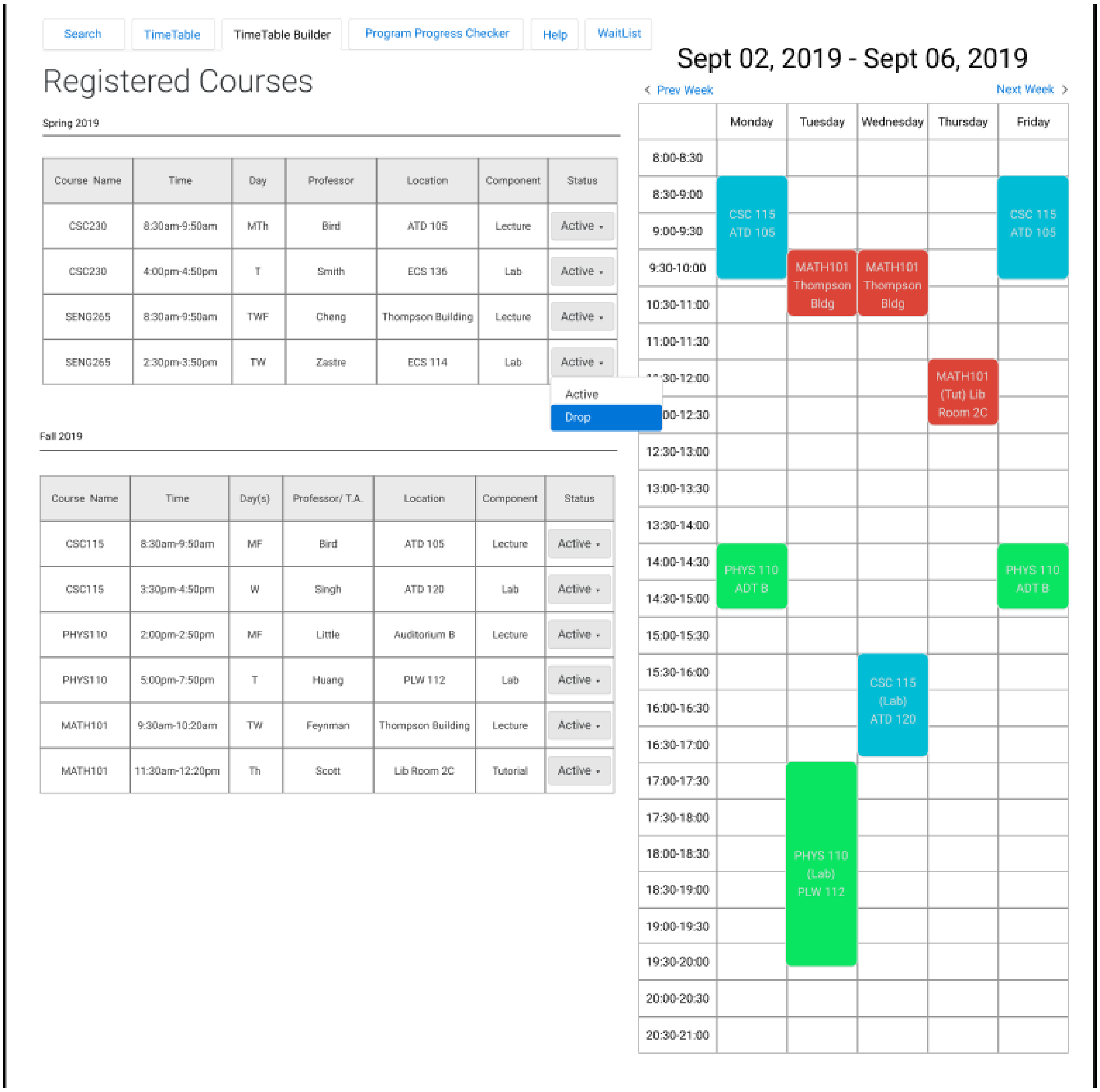
10.2 Dropping a registered course from Timetable Builder (UC-5)

The student selects the timetable builder tab which allows them to see each course that they are registered in along with the timetable. The student can then change the status of each course to “Drop” from “Active” and then will be given a prompt to confirm their decision. Selecting “Yes” on the prompt will drop that course from the student’s timetable builder.

1. The student selects timetable builder tab.



2. The student sees the timetable builder with the option for dropping a course under the status column.



Registered Courses

Spring 2019

Course Name	Time	Day	Professor	Location	Component	Status
CSC230	8:30am-9:50am	MTh	Bird	ATD 105	Lecture	Active ▾
CSC230	4:00pm-4:50pm	T	Smith	ECS 136	Lab	Active ▾
SENG265	8:30am-9:50am	TWF	Cheng	Thompson Building	Lecture	Active ▾
SENG265	2:30pm-3:50pm	TW	Zastre	ECS 114	Lab	Active ▾

Active
Drop

Fall 2019

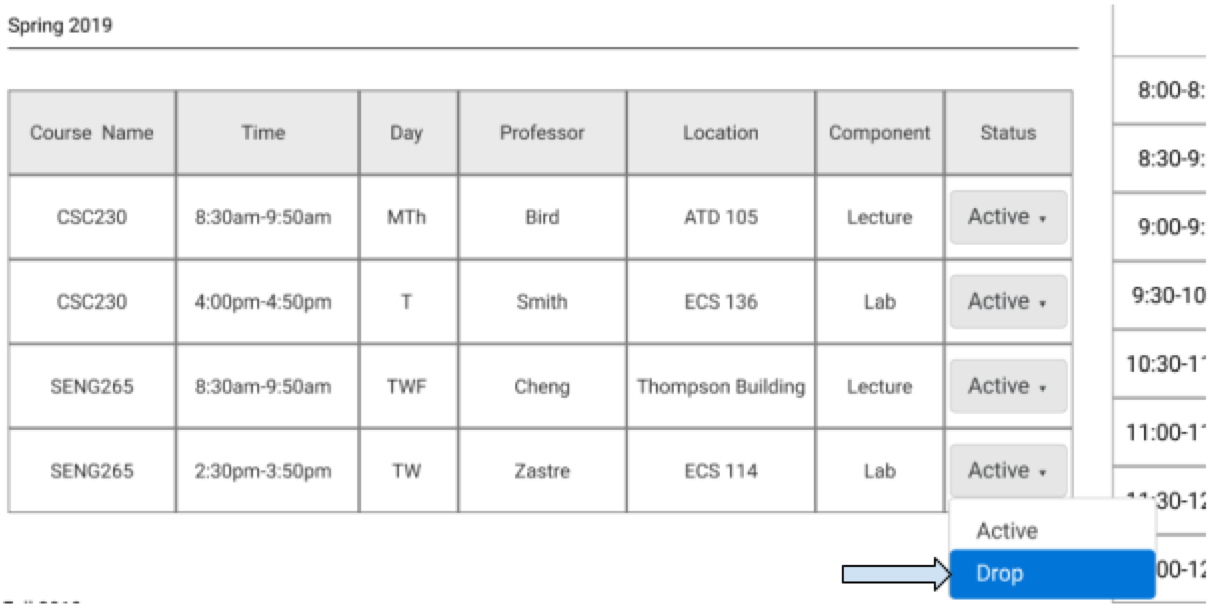
Course Name	Time	Day(s)	Professor/ T.A.	Location	Component	Status
CSC115	8:30am-9:50am	MF	Bird	ATD 105	Lecture	Active ▾
CSC115	3:30pm-4:50pm	W	Singh	ATD 120	Lab	Active ▾
PHYS110	2:00pm-2:50pm	MF	Little	Auditorium B	Lecture	Active ▾
PHYS110	5:00pm-7:50pm	T	Huang	PLW 112	Lab	Active ▾
MATH101	9:30am-10:20am	TW	Feynman	Thompson Building	Lecture	Active ▾
MATH101	11:30am-12:20pm	Th	Scott	Lib Room 2C	Tutorial	Active ▾

Sept 02, 2019 - Sept 06, 2019

‹ Prev Week   Next Week ›

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
30-12:00				MATH101 (Tut) Lib Room 2C	
00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					
17:00-17:30		PHYS 110 (Lab) PLW 112			
17:30-18:00					
18:00-18:30					
18:30-19:00					
19:00-19:30					
19:30-20:00					
20:00-20:30					
20:30-21:00					

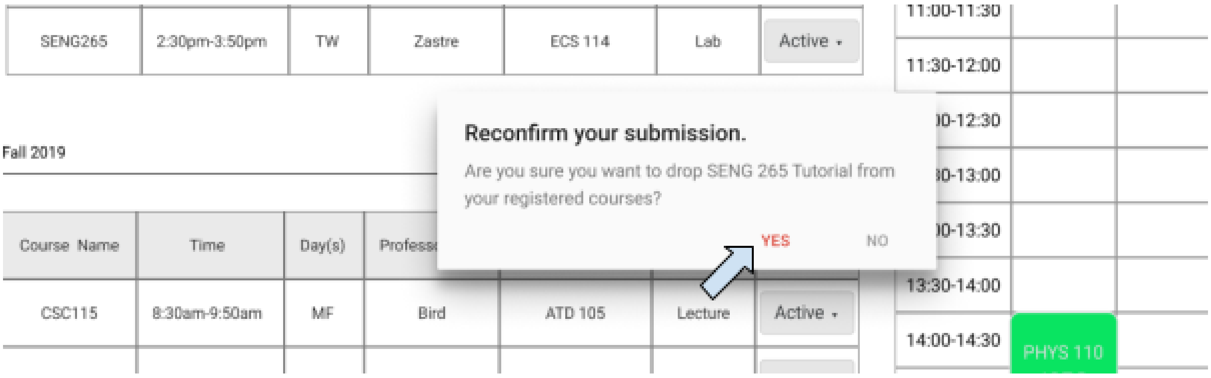
3. The student clicks on drop.



Spring 2019

Course Name	Time	Day	Professor	Location	Component	Status
CSC230	8:30am-9:50am	MTh	Bird	ATD 105	Lecture	Active ▾
CSC230	4:00pm-4:50pm	T	Smith	ECS 136	Lab	Active ▾
SENG265	8:30am-9:50am	TWF	Cheng	Thompson Building	Lecture	Active ▾
SENG265	2:30pm-3:50pm	TW	Zastre	ECS 114	Lab	Active ▾

4. The student is prompted to confirm the selection for dropping a course. The student clicks "Yes" for this use case of dropping a registered course.



SENG265	2:30pm-3:50pm	TW	Zastre	ECS 114	Lab	Active ▾

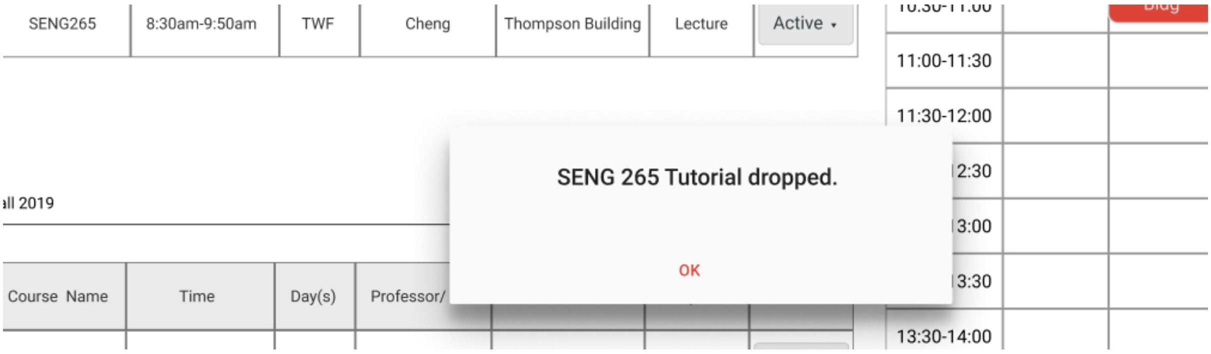
Fall 2019

Course Name	Time	Day(s)	Profess
CSC115	8:30am-9:50am	MF	Bird

Reconfirm your submission.

Are you sure you want to drop SENG 265 Tutorial from your registered courses?

5. The student sees a prompt confirms that the selected registered course is dropped.



SENG265	8:30am-9:50am	TWF	Cheng	Thompson Building	Lecture	Active ▾

ıll 2019

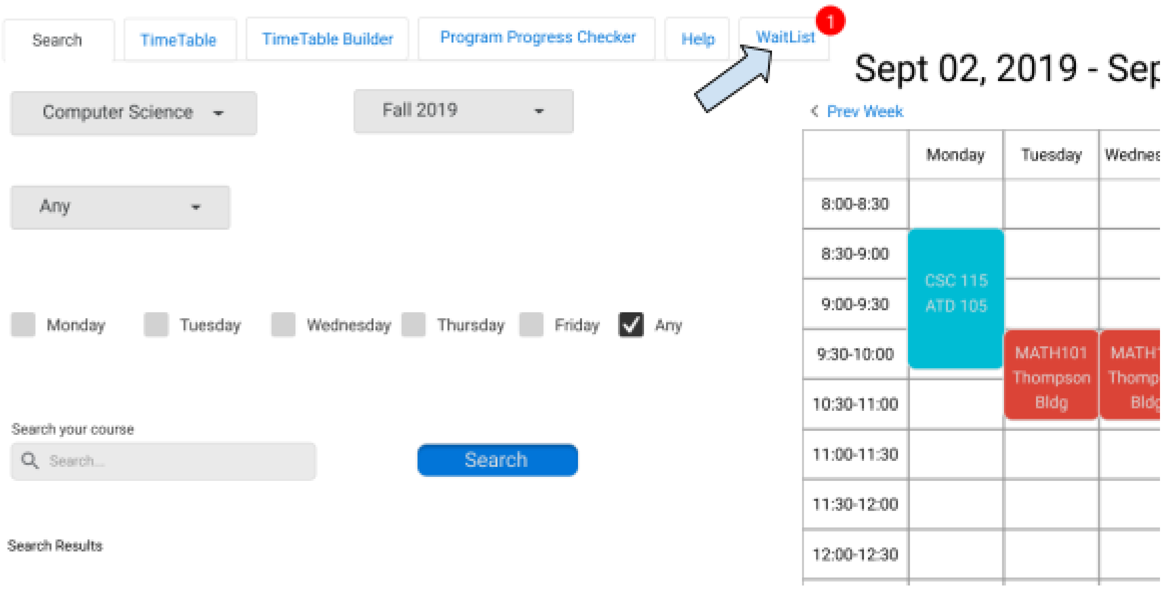
Course Name	Time	Day(s)	Professor/

SENG 265 Tutorial dropped.

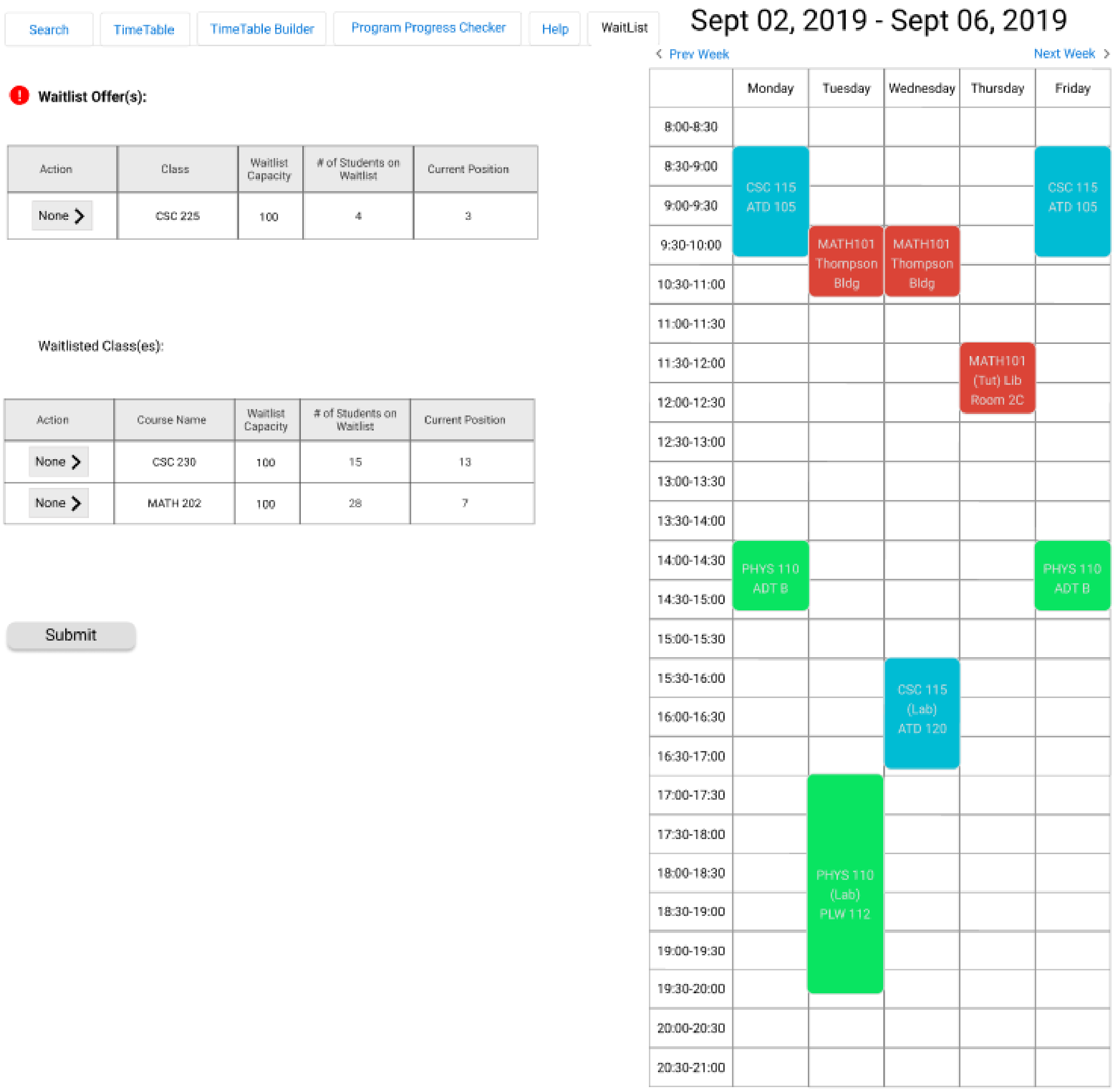
10.3 Accept a Waitlist Offer (UC-7)

The student is shown a notification that they have a waitlist offer. The notification is shown as a red circle with a number inside indicating the number of notifications. After that, the student selects the "Waitist" tab and is shown all of their waitlist offers, each waitlist the student is a part of, and the student's timetable. The student then selects an action and selects the "Accept" action if they wish to join the course or the student would select the "Drop" action if they want to drop their status in the waitlist. Then, the student selects "Submit" and is given a confirmation that their changes were saved.

1. The Student select the "Waitist" tab showing a notification for a new offer.



2. The student sees the “Waitist” page.



Waitlist Offer(s):

Action	Class	Waitlist Capacity	# of Students on Waitlist	Current Position
None >	CSC 225	100	4	3

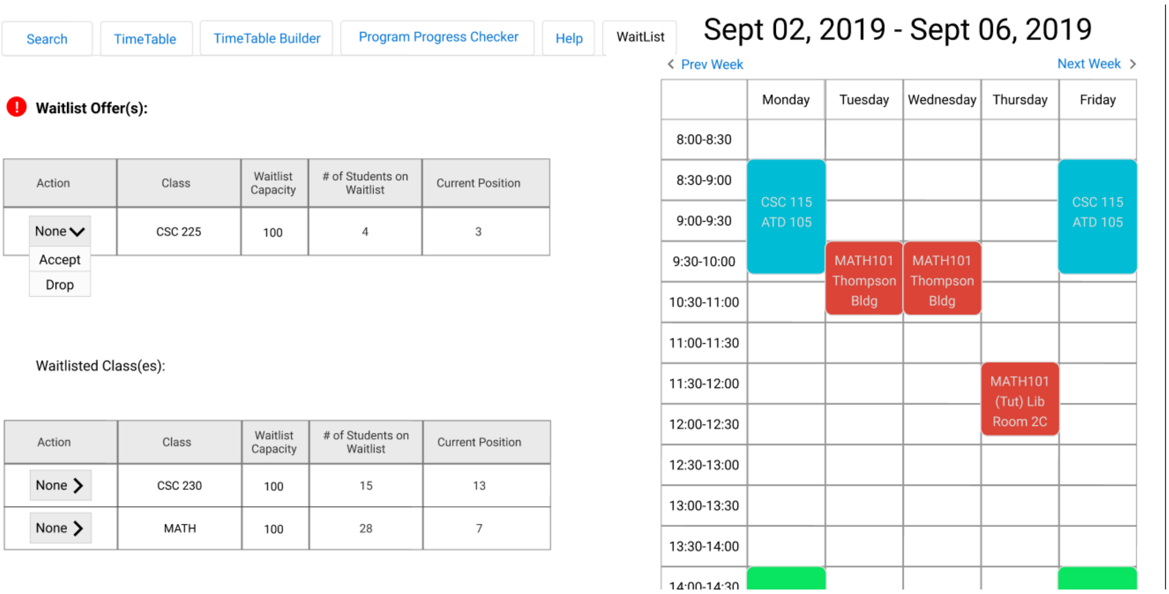
Waitlisted Class(es):

Action	Course Name	Waitlist Capacity	# of Students on Waitlist	Current Position
None >	CSC 230	100	15	13
None >	MATH 202	100	28	7

Sept 02, 2019 - Sept 06, 2019

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					
17:00-17:30		PHYS 110 (Lab) PLW 112			
17:30-18:00					
18:00-18:30					
18:30-19:00					
19:00-19:30					
19:30-20:00					
20:00-20:30					
20:30-21:00					

3. The student selects the course in the “Waitlist Offer” section.



Waitlist Offer(s):

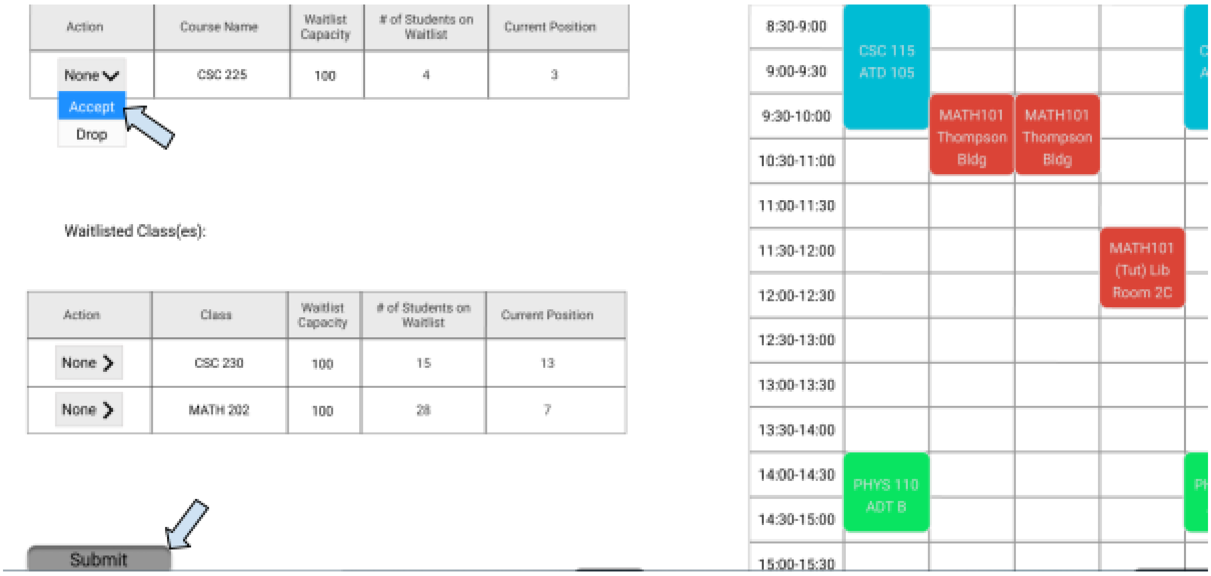
Action	Class	Waitlist Capacity	# of Students on Waitlist	Current Position
None Accept Drop	CSC 225	100	4	3

Waitlisted Class(es):

Action	Class	Waitlist Capacity	# of Students on Waitlist	Current Position
None	CSC 230	100	15	13
None	MATH	100	28	7

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30					

4. The Student selects the “Accept” option in the dropdown list in the action column for the desired waitlisted course offer and clicks submit.



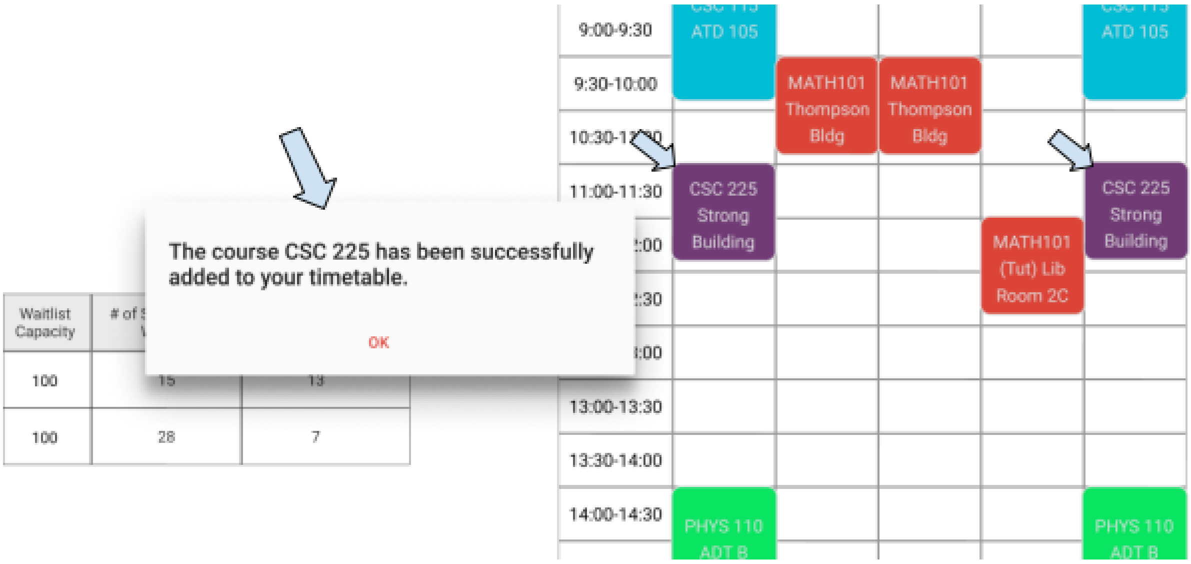
Action	Course Name	Waitlist Capacity	# of Students on Waitlist	Current Position
None Accept Drop	CSC 225	100	4	3

Waitlisted Class(es):

Action	Class	Waitlist Capacity	# of Students on Waitlist	Current Position
None	CSC 230	100	15	13
None	MATH 202	100	28	7

8:30-9:00	CSC 115 ATD 105				C A
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				Pl
14:30-15:00					
15:00-15:30					

5. The student sees a notification that the course is added to their timetable and sees the course in the updated timetable.



The course CSC 225 has been successfully added to your timetable.

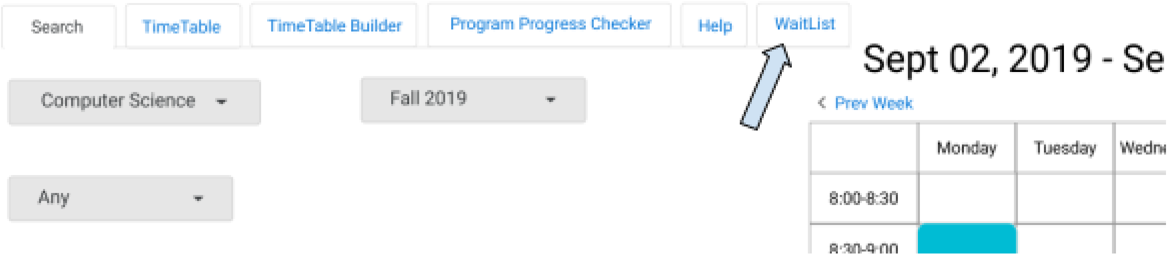
Waitlist Capacity	# of	
100	15	13
100	28	7

9:00-9:30	ATD 105				ATD 105
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-1[illegible]0					
11:00-11:30	CSC 225 Strong Building				CSC 225 Strong Building
:00				MATH101 (Tut) Lib Room 2C	
:30					
:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B

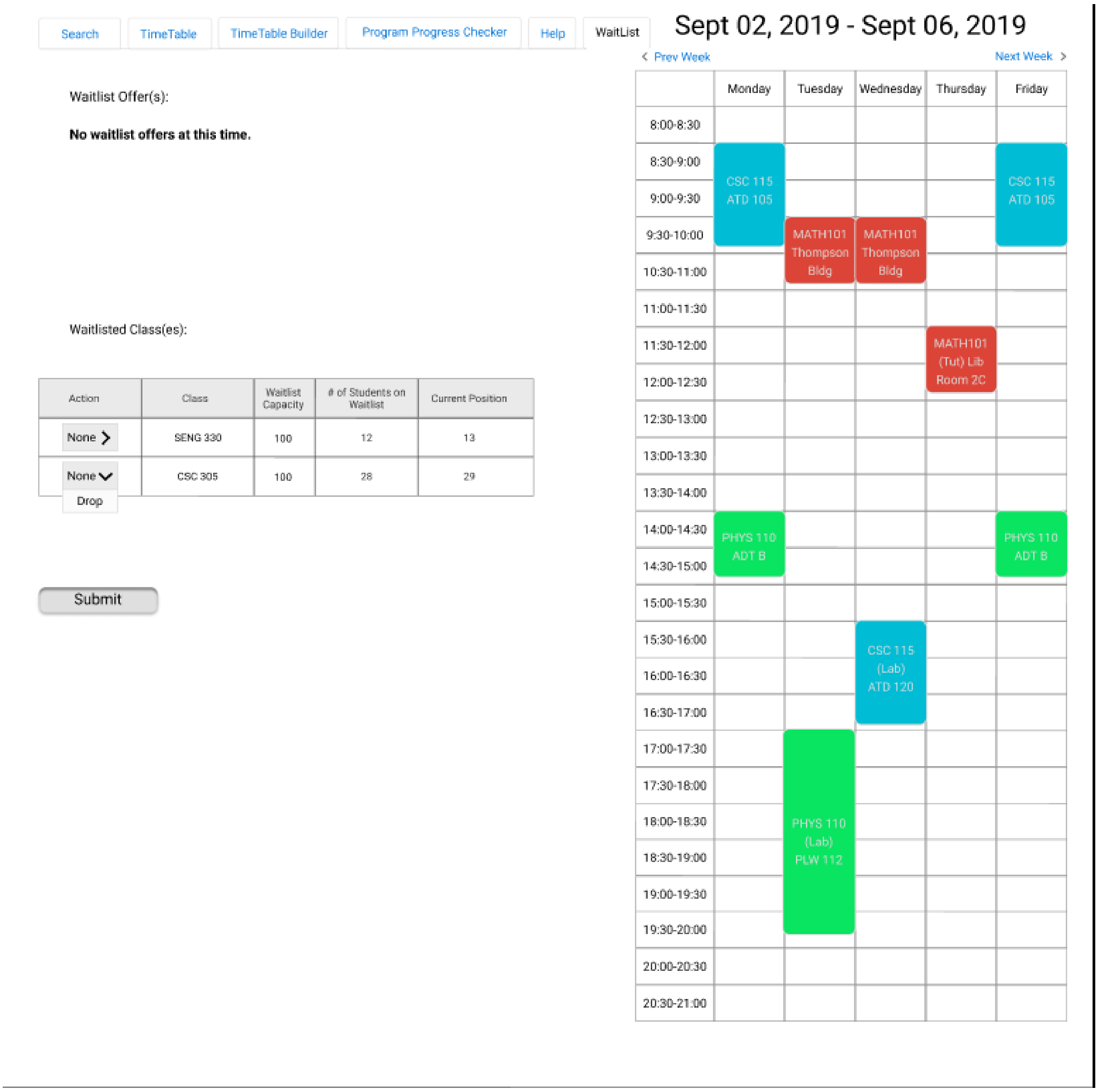
10.4 Dropping a Course from Waitlist (UC-8)

The student selects the “Waitist” tab. The student is then shown each waitlist offer, each of their currently waitlisted classes, and their timetable. After that, the student uses the action menu and selects “drop”. The student then clicks submit and is given a prompt asking to confirm their choice. When the student confirms their choice, they are given a confirmation that their changes have been saved.

1. The student clicks on the “WaitList” tab.



2. The student waitlist view after clicking on “Waitlist” tab.



Waitlist Offer(s):

No waitlist offers at this time.

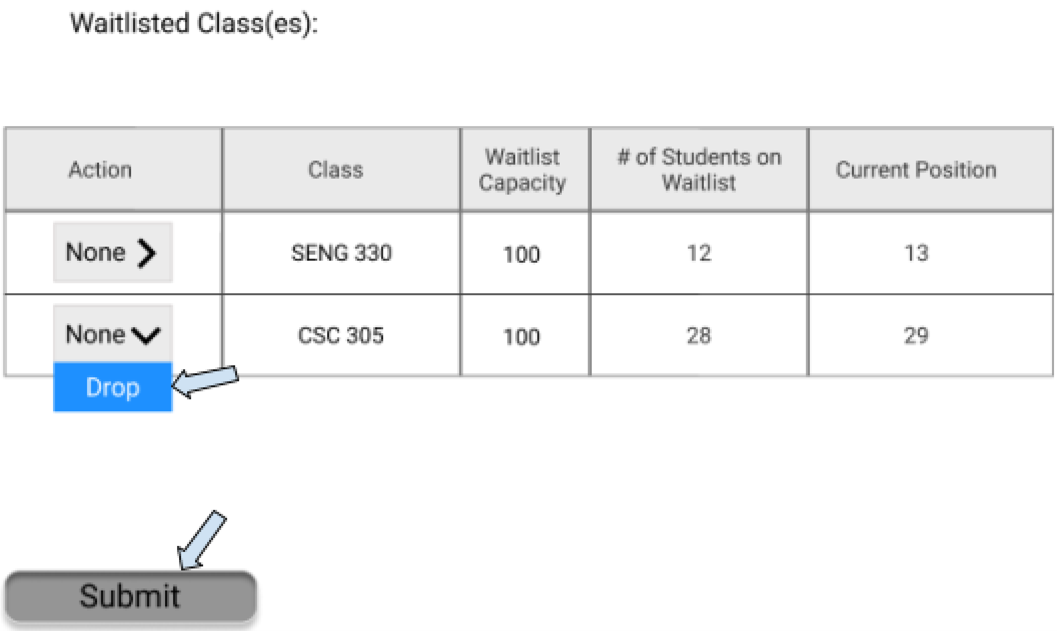
Waitlisted Class(es):

Action	Class	Waitlist Capacity	# of Students on Waitlist	Current Position
None ❯	SENG 330	100	12	13
None ⌄ Drop	CSC 305	100	28	29

Sept 02, 2019 - Sept 06, 2019

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					
17:00-17:30		PHYS 110 (Lab) PLW 112			
17:30-18:00					
18:00-18:30					
18:30-19:00					
19:00-19:30					
19:30-20:00					
20:00-20:30					
20:30-21:00					

3. The Student selects the “Drop” option in the dropdown list in the action column for the desired waitlisted course and clicks submit.

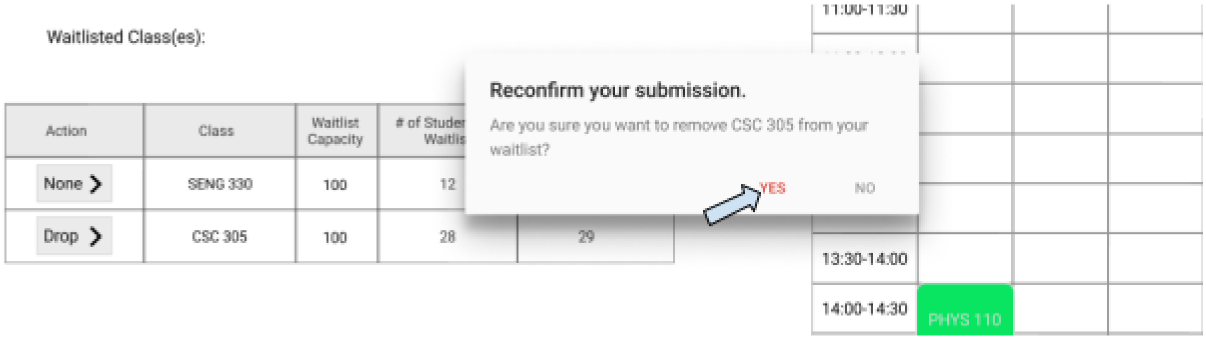


Waitlisted Class(es):

Action	Class	Waitlist Capacity	# of Students on Waitlist	Current Position
None ❯	SENG 330	100	12	13
None ⌄ Drop	CSC 305	100	28	29

Submit

4. The Student sees a pop up asking for confirmation to drop the selected course from their waitlisted courses. For this use case, the student selects “Yes”.



Waitlisted Class(es):

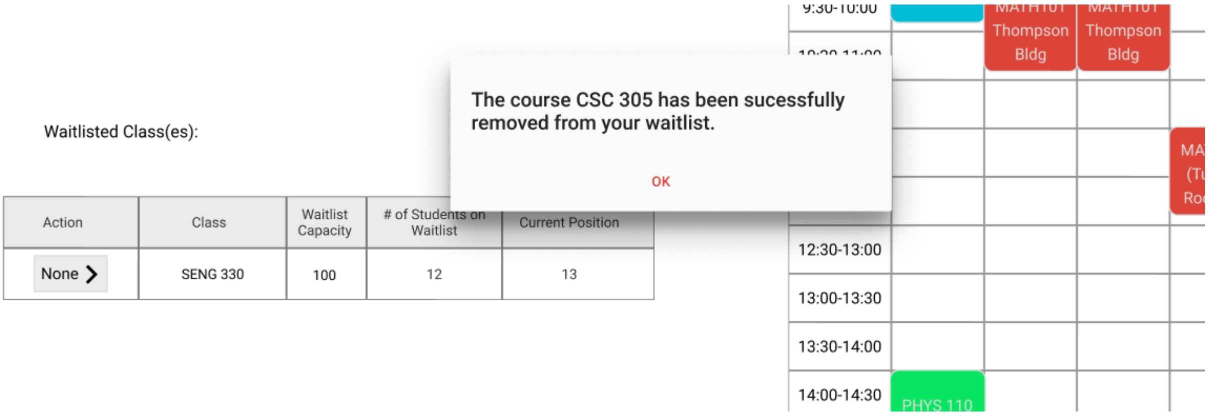
Action	Class	Waitlist Capacity	# of Studer Waitlis	
None ❯	SENG 330	100	12	
Drop ❯	CSC 305	100	28	29

Reconfirm your submission.

Are you sure you want to remove CSC 305 from your waitlist?

YES NO

5. The student sees the notification that the course is dropped from the waitlisted course.



Waitlisted Class(es):

Action	Class	Waitlist Capacity	# of Students on Waitlist	Current Position
None ❯	SENG 330	100	12	13

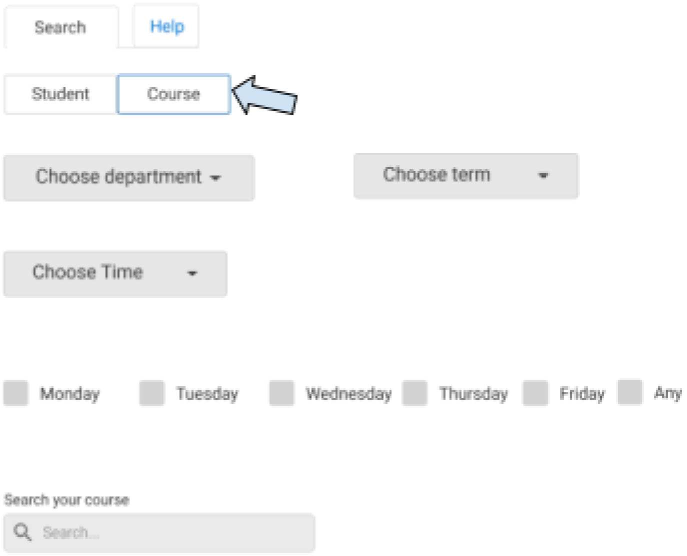
The course CSC 305 has been sucessfully removed from your waitlist.

OK

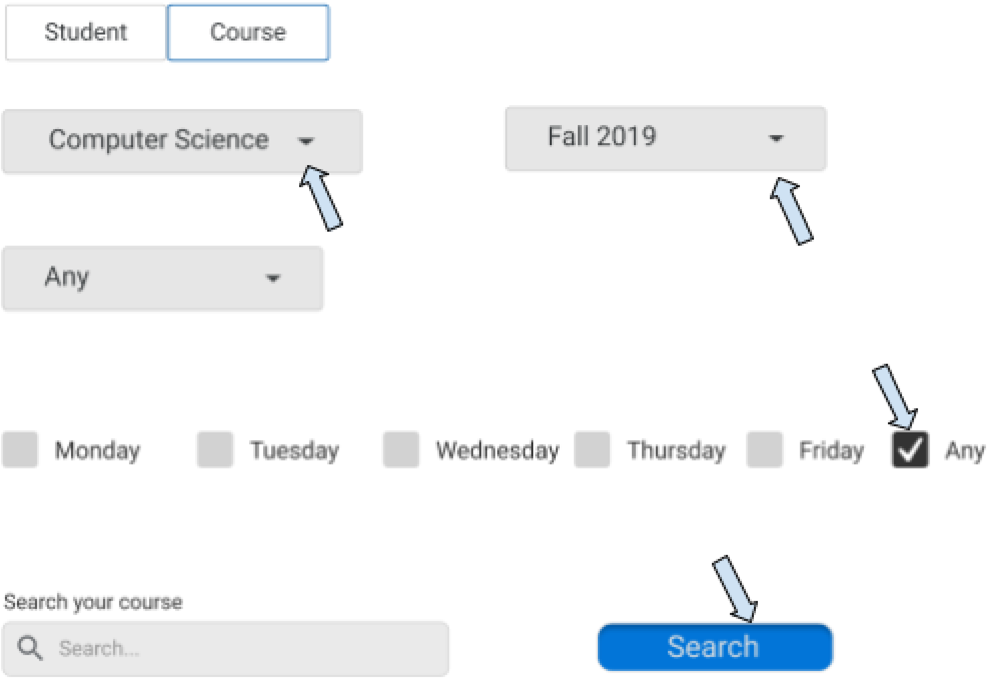
10.5 Administrator searches for a course and edits the course (UC-16)

The administrator selects the course toggle selector in the search feature. This allows the administrator to search for a course rather than a student. After that, the administrator will input the necessary filters for a course they are looking for. The administrator will then be provided with a list of courses that match the search criteria. Subsequently, the administrator can select to edit a course by clicking on the "Edit" button for each course. The administrator will be provided a view of each course's course details and will be able to change each course's alterable details. The administrator will then be given a prompt before their changes are saved.

1. The administrator selects "Course" in the "Search" tab and sees course search filters.



2. The administrator selects filters and clicks on search.



Student | Course

Computer Science ▾

Fall 2019 ▾

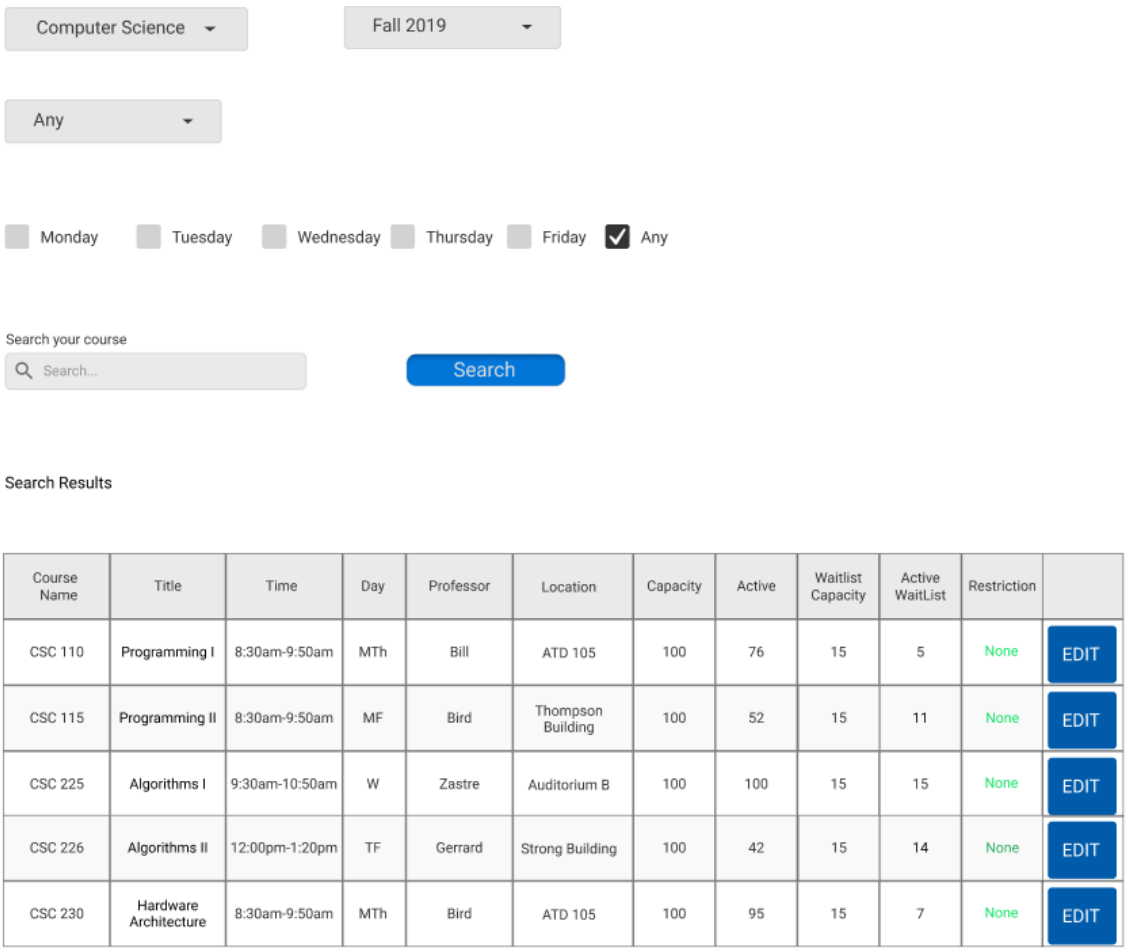
Any ▾

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday ☑ Any

Search your course

Search...

3. The administrator sees the matched results with an option to edit each course.



Computer Science ▾

Fall 2019 ▾

Any ▾

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday ☑ Any

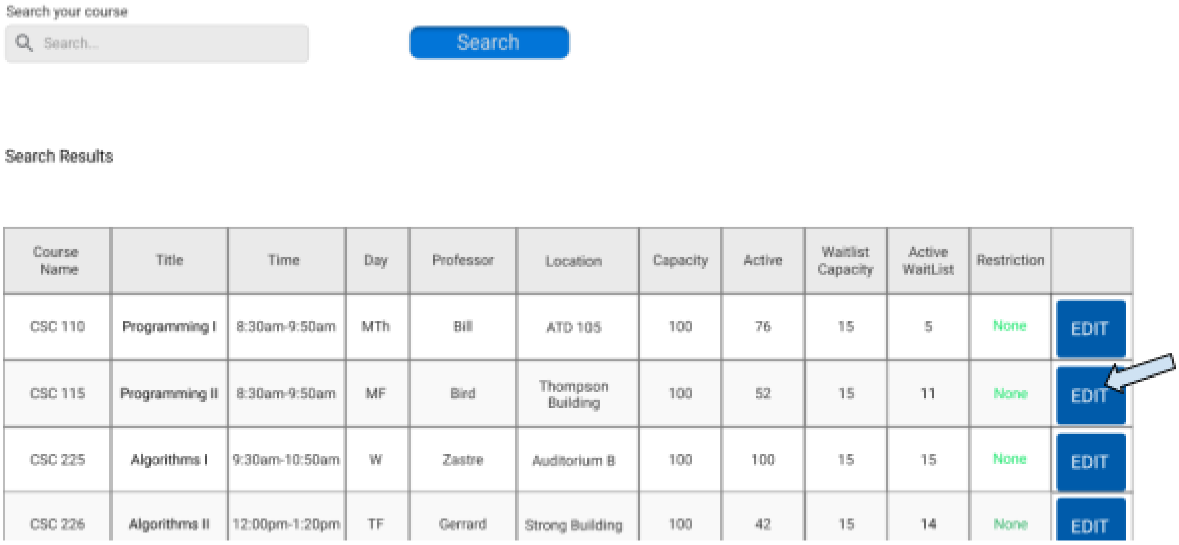
Search your course

Search...

Search Results

Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	EDIT
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	EDIT
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	15	None	EDIT
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	None	EDIT
CSC 230	Hardware Architecture	8:30am-9:50am	MTh	Bird	ATD 105	100	95	15	7	None	EDIT

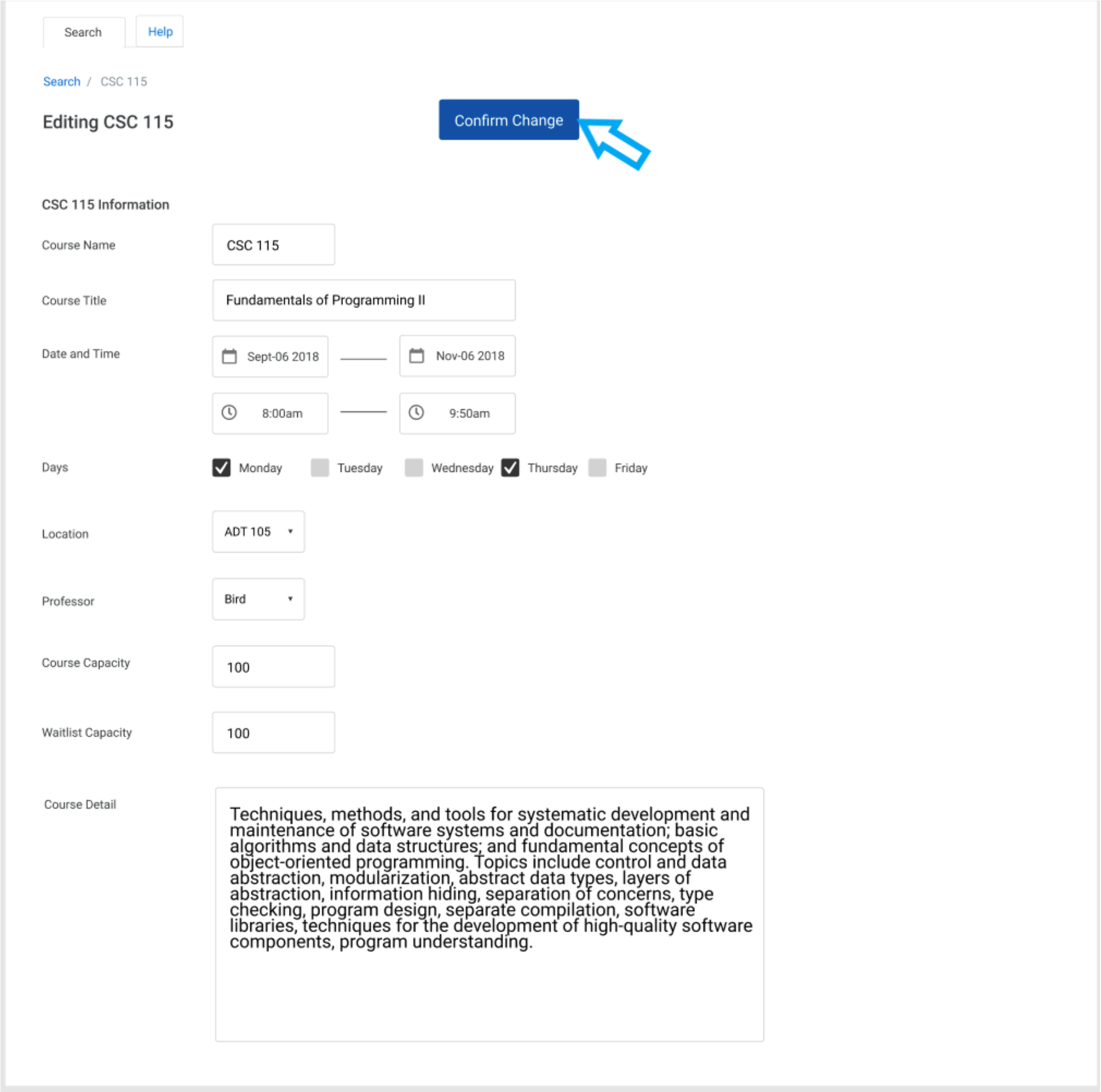
4. The administrator clicks on “Edit” for a course.



Search Results

Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	EDIT
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	EDIT
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	15	None	EDIT
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	None	EDIT

5. The administrator is taken to a new page with the course’s alterable details that can be edited.

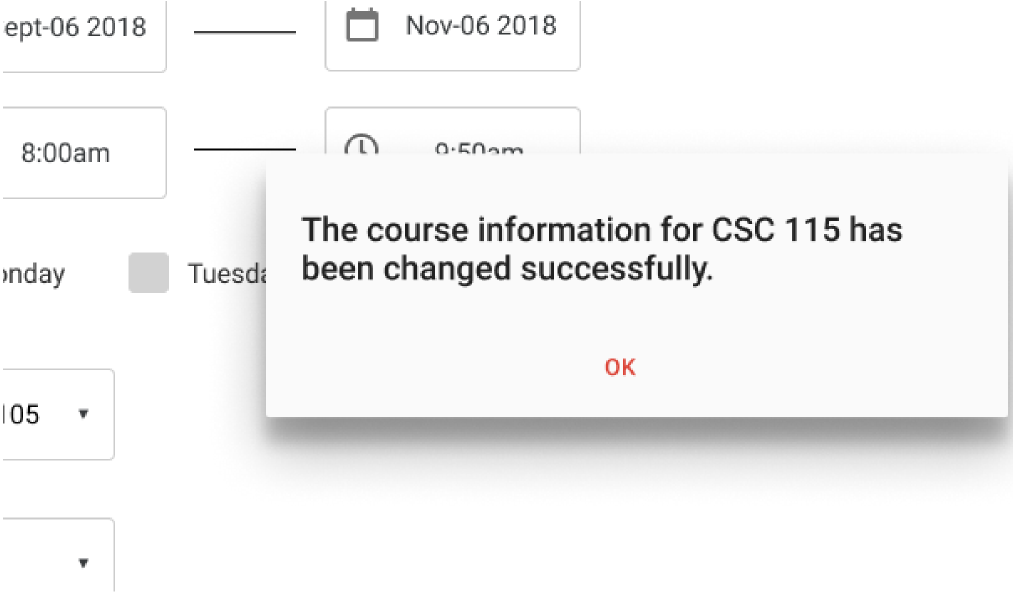


Editing CSC 115

CSC 115 Information

Course Name	CSC 115
Course Title	Fundamentals of Programming II
Date and Time	Sept-06 2018 — Nov-06 2018 8:00am — 9:50am
Days	☑ Monday □ Tuesday □ Wednesday ☑ Thursday □ Friday
Location	ADT 105 ▾
Professor	Bird ▾
Course Capacity	100
Waitlist Capacity	100
Course Detail	Techniques, methods, and tools for systematic development and maintenance of software systems and documentation; basic algorithms and data structures; and fundamental concepts of object-oriented programming. Topics include control and data abstraction, modularization, abstract data types, layers of abstraction, information hiding, separation of concerns, type checking, program design, separate compilation, software libraries, techniques for the development of high-quality software components, program understanding.

6. The Administrator is provided with the notification that they have made changes.

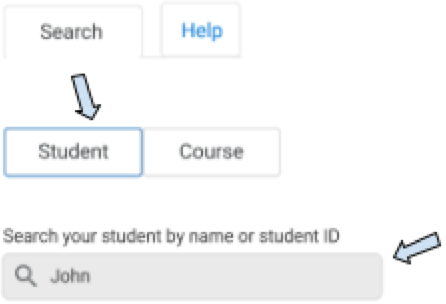


The course information for CSC 115 has been changed successfully.

10.6 Administrator searches a student and adds the student to a course (UC-14)

The administrator selects the “Student” tab in the search feature. The “Student” tab enables the administrator to search for a student rather than a course. The administrator inputs a student’s name or studentID and is shown all matching students. The administrator can then choose the “Edit” option for a student. Afterwards, in the course list for a given student, the administrator can search for a course that the student wants to be added to. Then, the administrator will search for the course requested by the student and be given a list of matching courses. Subsequently, the administrator will be able to add the course to the student’s course list. The administrator will be given a prompt before the changes are saved.

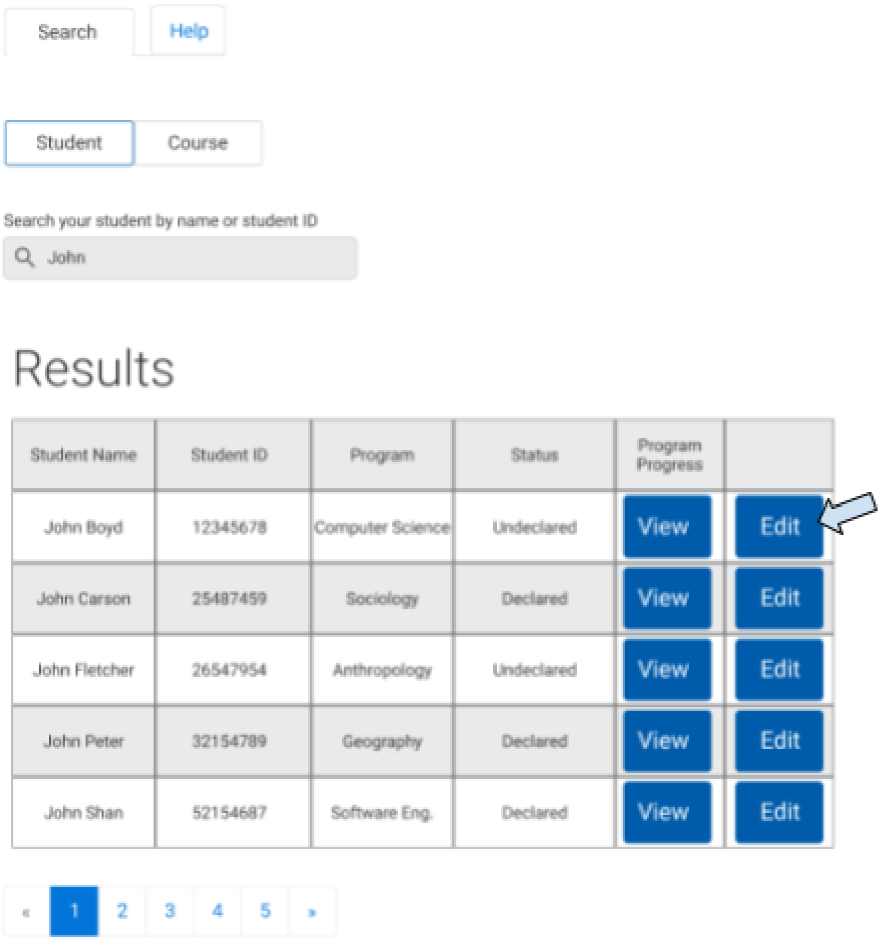
1. The administrator selects the “Student” tab and enters a student’s name, declared program or studentID in the search bar.



Search your student by name or student ID

John

2. The administrator is provided with all matching students and clicks on “Edit” for the desired student.



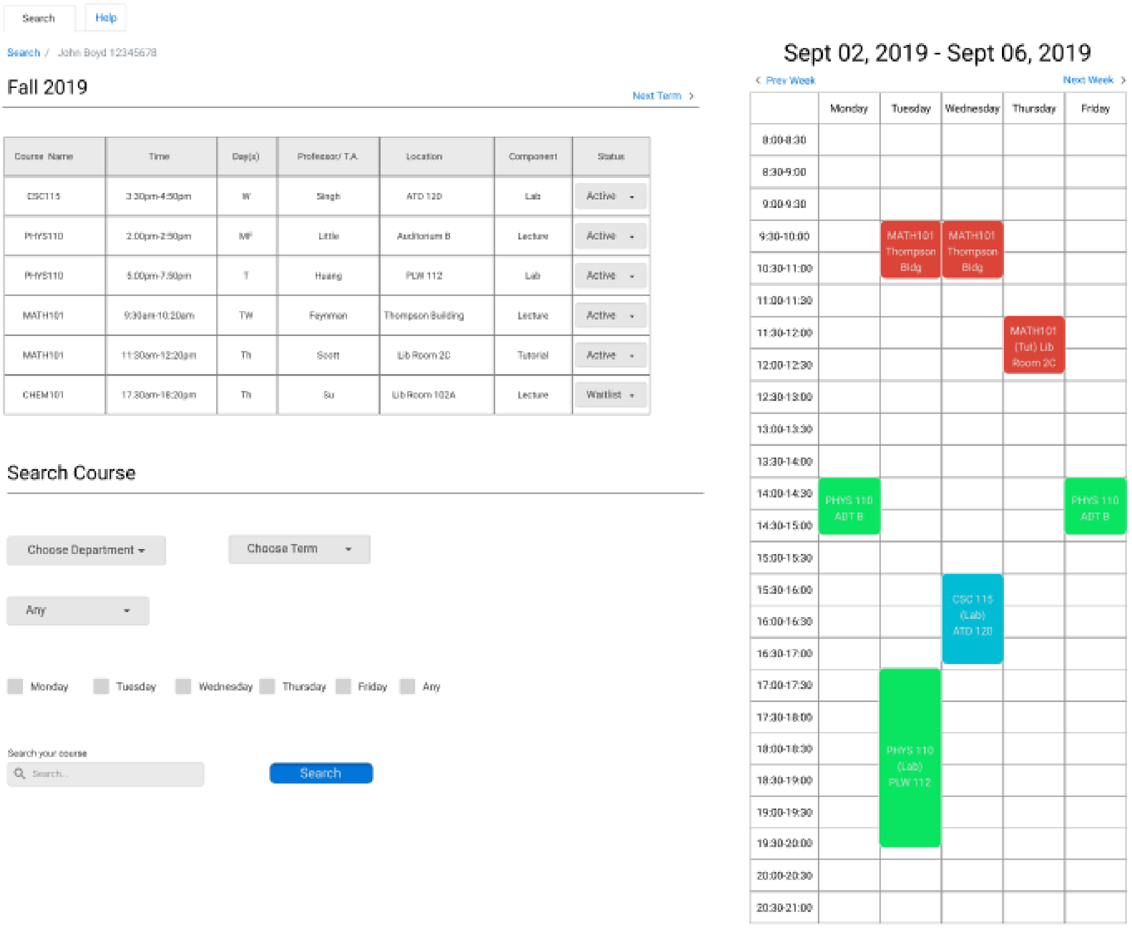
Search your student by name or student ID

John

Results

Student Name	Student ID	Program	Status	Program Progress	
John Boyd	12345678	Computer Science	Undeclared	View	Edit
John Carson	25487459	Sociology	Declared	View	Edit
John Fletcher	26547954	Anthropology	Undeclared	View	Edit
John Peter	32154789	Geography	Declared	View	Edit
John Shan	52154687	Software Eng.	Declared	View	Edit

3. The administrator is taken to a new page with a view of the student’s timetable builder.



Fall 2019

Course Name	Time	Day(s)	Professor/ T.A	Location	Component	Status
CSC115	3:30pm-4:50pm	W	Singh	ATD 120	Lab	Active ▾
PHYS110	2:00pm-2:50pm	MF	Little	Auditorium B	Lecture	Active ▾
PHYS110	5:00pm-7:50pm	T	Huang	PLW 112	Lab	Active ▾
MATH101	9:30am-10:20am	TW	Feynman	Thompson Building	Lecture	Active ▾
MATH101	11:30am-12:20pm	Th	Scott	Lib Room 2C	Tutorial	Active ▾
CHEM101	17:30am-18:20pm	Th	Su	Lib Room 102A	Lecture	Waitlist ▾

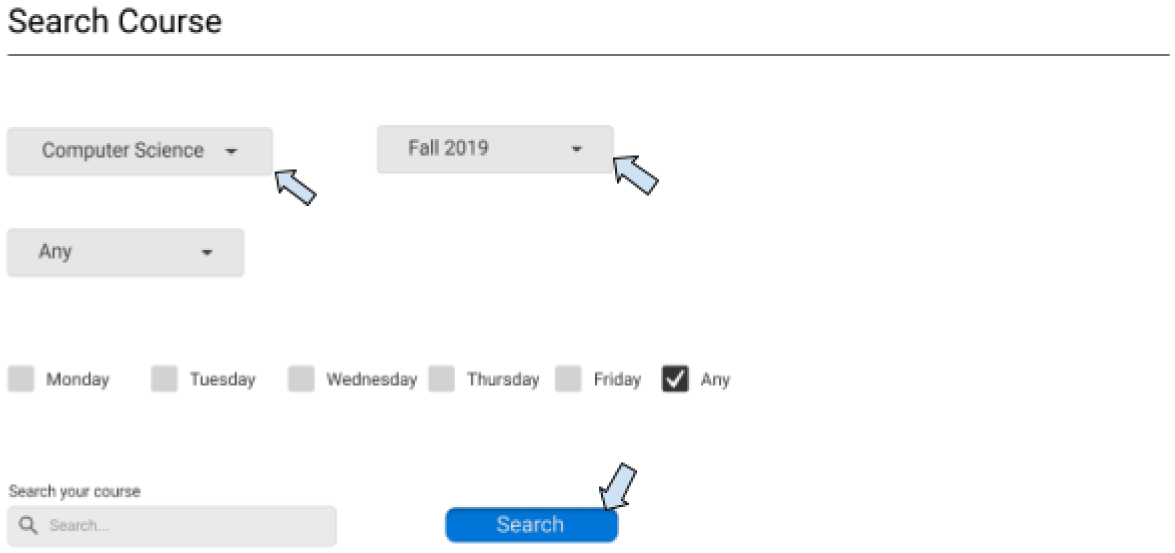
Search Course

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday □ Any

Sept 02, 2019 - Sept 06, 2019

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00					
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					
17:00-17:30		PHYS 110 (Lab) PLW 112			
17:30-18:00					
18:00-18:30					
18:30-19:00					
19:00-19:30					
19:30-20:00					
20:00-20:30					
20:30-21:00					

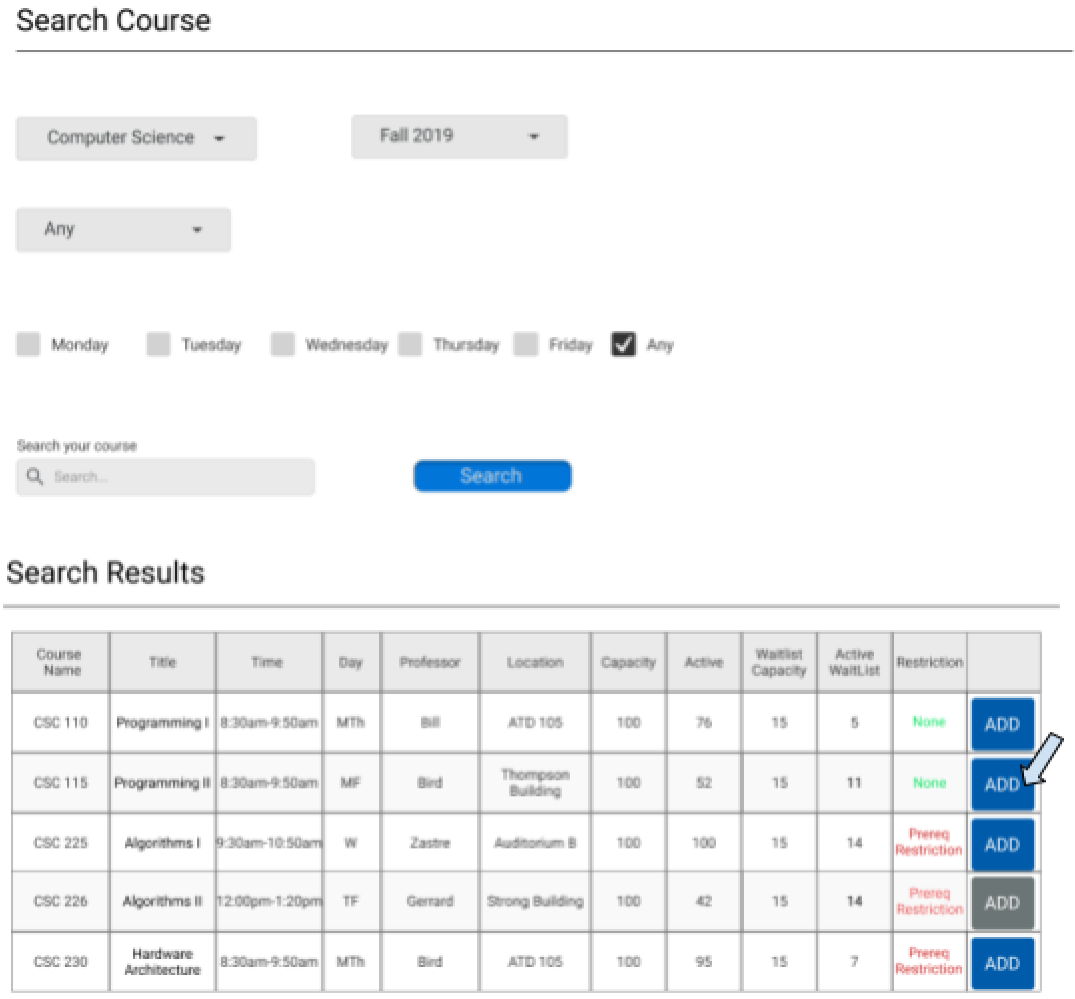
4. The Administrator selects filters for the course that needs to be added for the student.



Search Course

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday ☑ Any

5. The search results are displayed to the administrator and the administrator selects “Add” to add the course to the student’s timetable.



Search Course

Computer Science ▾ Fall 2019 ▾

Any ▾

□ Monday □ Tuesday □ Wednesday □ Thursday □ Friday ☑ Any

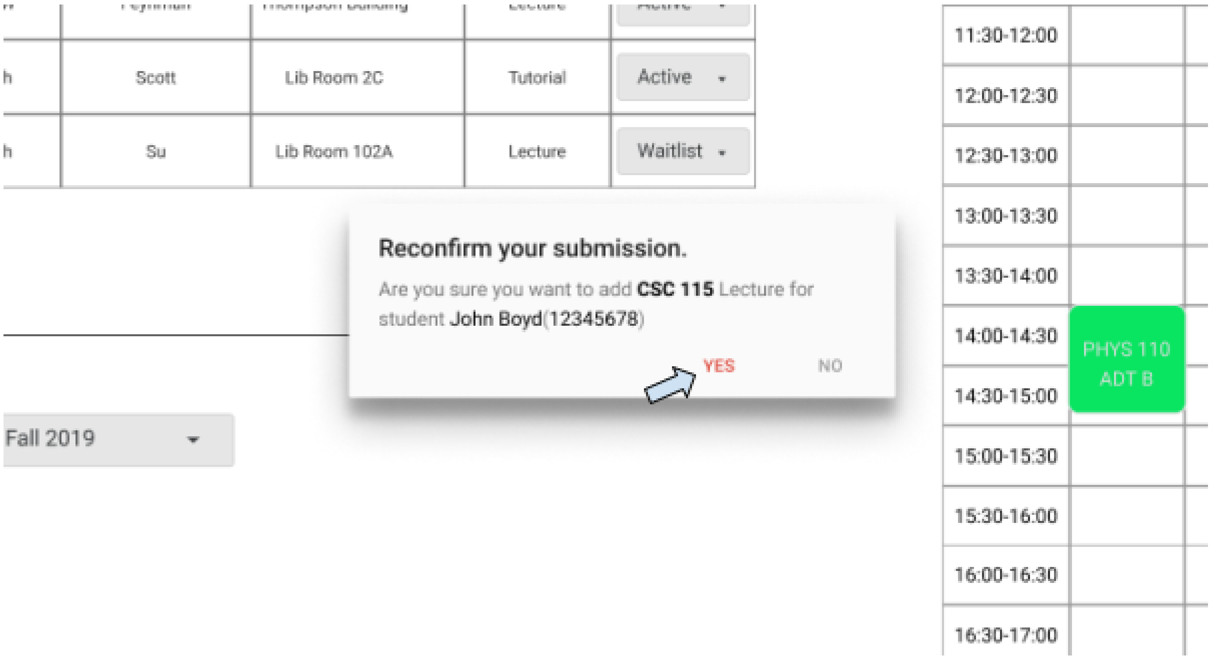
Search your course

Search...

Search Results

Course Name	Title	Time	Day	Professor	Location	Capacity	Active	Waitlist Capacity	Active WaitList	Restriction	
CSC 110	Programming I	8:30am-9:50am	MTh	Bill	ATD 105	100	76	15	5	None	ADD
CSC 115	Programming II	8:30am-9:50am	MF	Bird	Thompson Building	100	52	15	11	None	ADD
CSC 225	Algorithms I	9:30am-10:50am	W	Zastre	Auditorium B	100	100	15	14	Prereq Restriction	ADD
CSC 226	Algorithms II	12:00pm-1:20pm	TF	Gerrard	Strong Building	100	42	15	14	Prereq Restriction	ADD
CSC 230	Hardware Architecture	8:30am-9:50am	MTh	Bird	ATD 105	100	95	15	7	Prereq Restriction	ADD

6. A pop up is displayed to confirm the administrator's choice for adding the student to the selected course. In this case the administrator clicks "Yes".



h	Scott	Lib Room 2C	Tutorial	Active ▾
h	Su	Lib Room 102A	Lecture	Waitlist ▾

Reconfirm your submission.

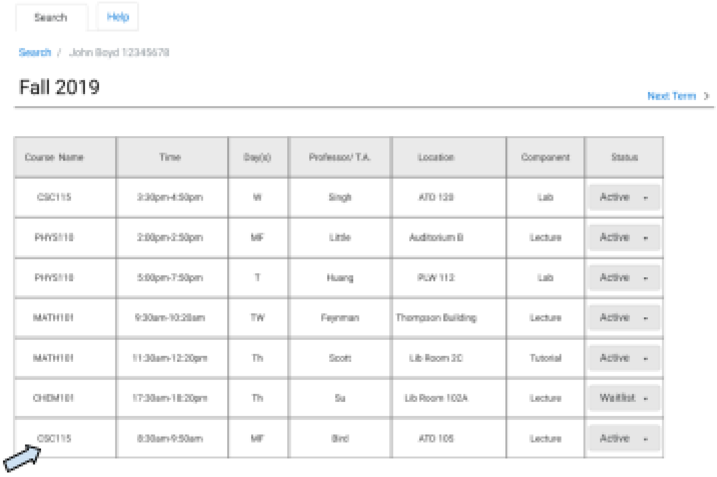
Are you sure you want to add **CSC 115** Lecture for student **John Boyd**(12345678)

YES NO

Fall 2019 ▾

11:30-12:00	
12:00-12:30	
12:30-13:00	
13:00-13:30	
13:30-14:00	
14:00-14:30	PHYS 110 ADT B
14:30-15:00	
15:00-15:30	
15:30-16:00	
16:00-16:30	
16:30-17:00	

7. The course is added to the timetable of the student and the student's time table is updated.

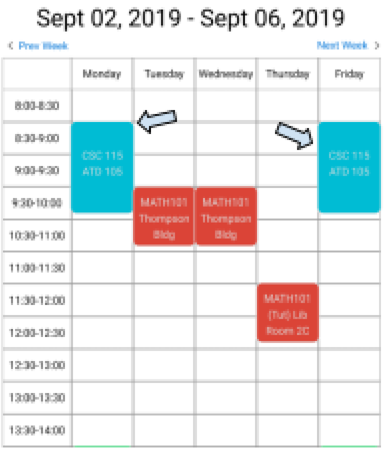


Search / John Boyd 12345678

Fall 2019

Next Term ›

Course Name	Time	Day(s)	Professor/ T.A.	Location	Component	Status
CSC115	3:30pm-4:50pm	W	Singh	ATD 120	Lab	Active ▾
PHYS110	2:00pm-2:50pm	MF	Little	Auditorium B	Lecture	Active ▾
PHYS110	5:00pm-7:50pm	T	Huang	PLW 112	Lab	Active ▾
MATH101	9:30am-10:20am	TW	Feynman	Thompson Building	Lecture	Active ▾
MATH101	11:30am-12:20pm	Th	Scott	Lib Room 2C	Tutorial	Active ▾
CHEM101	17:30am-18:20pm	Th	Su	Lib Room 102A	Lecture	Waitlist ▾
CSC115	8:30am-9:50am	MF	Bird	ATD 105	Lecture	Active ▾



Sept 02, 2019 - Sept 06, 2019

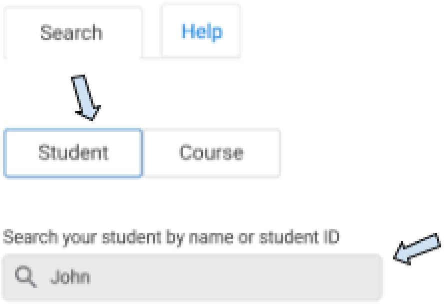
‹ Prev Week Next Week ›

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					

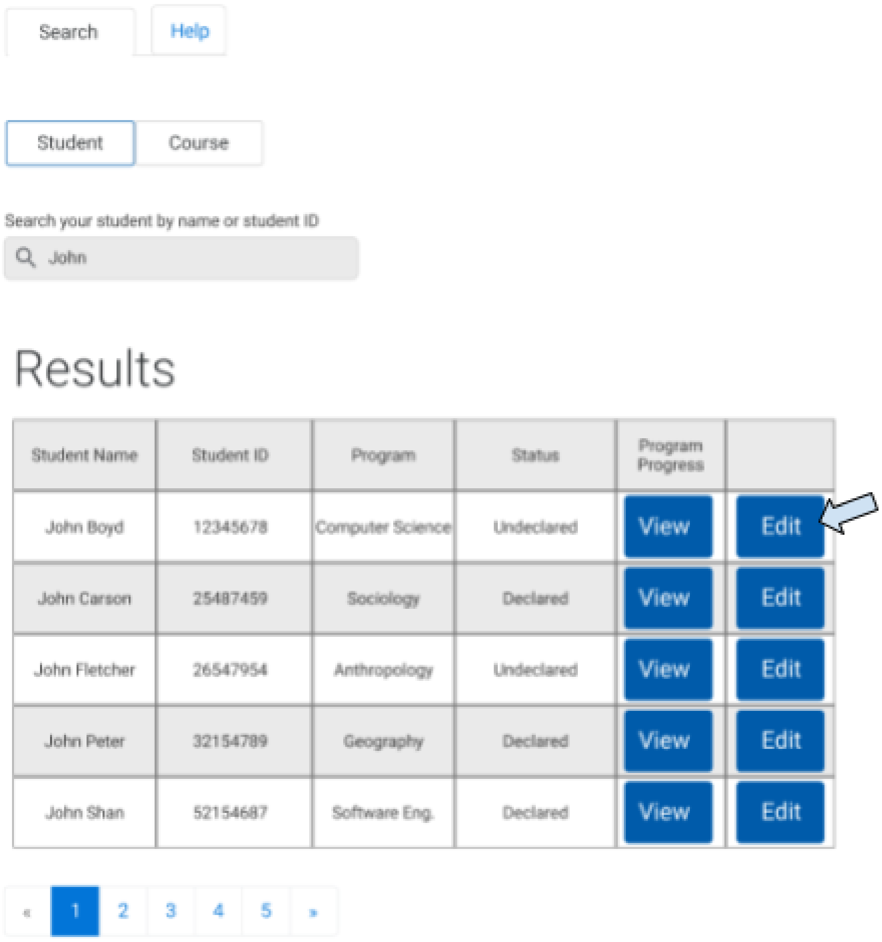
10.7 Administrator searches a student and removes the student from a course (UC-15)

The administrator selects the “Student” tab in the search feature. This enables the administrator to search for a student rather than a course. The administrator inputs the student’s name or studentID and is shown all matching students. The administrator can then choose the “Edit” option for the student which will take them to the student’s timetable builder. Subsequently, the administrator can see each course the student is registered or waitlisted in and can remove the student from the course or waitlist for that course. The administrator will be given a prompt before the changes are saved.

1. In the search tab the administrator selects the “Student” tab and enters a students name in the search bar.



2. The administrator is provided with all matching students and clicks on “Edit” for the desired student.



Search | Help

Student | Course

Search your student by name or student ID

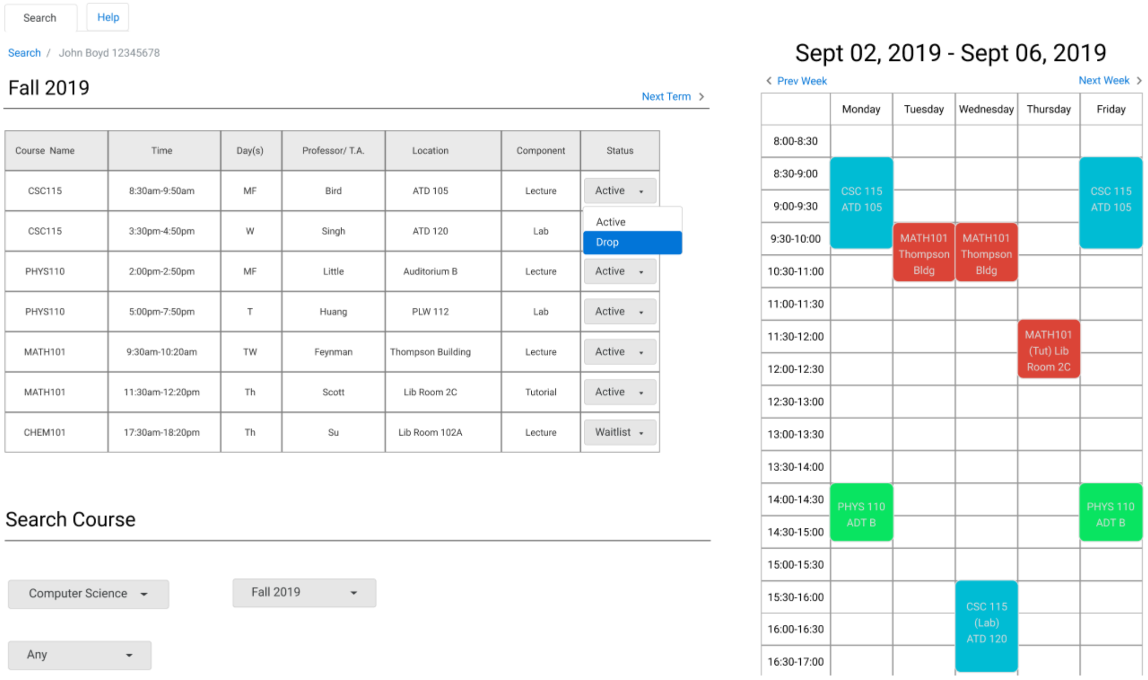
John

Results

Student Name	Student ID	Program	Status	Program Progress	
John Boyd	12345678	Computer Science	Undeclared	View	Edit
John Carson	25487459	Sociology	Declared	View	Edit
John Fletcher	26547954	Anthropology	Undeclared	View	Edit
John Peter	32154789	Geography	Declared	View	Edit
John Shan	52154687	Software Eng.	Declared	View	Edit

« 1 2 3 4 5 »

3. The administrator is taken to a new page with a view of the student’s timetable builder.



Search | Help

Search / John Boyd 12345678

Fall 2019 — Next Term >

Course Name	Time	Day(s)	Professor/ T.A.	Location	Component	Status
CSC115	8:30am-9:50am	MF	Bird	ATD 105	Lecture	Active
CSC115	3:30pm-4:50pm	W	Singh	ATD 120	Lab	Active Drop
PHYS110	2:00pm-2:50pm	MF	Little	Auditorium B	Lecture	Active
PHYS110	5:00pm-7:50pm	T	Huang	PLW 112	Lab	Active
MATH101	9:30am-10:20am	TW	Feynman	Thompson Building	Lecture	Active
MATH101	11:30am-12:20pm	Th	Scott	Lib Room 2C	Tutorial	Active
CHEM101	17:30am-18:20pm	Th	Su	Lib Room 102A	Lecture	Waitlist

Search Course

Computer Science | Fall 2019

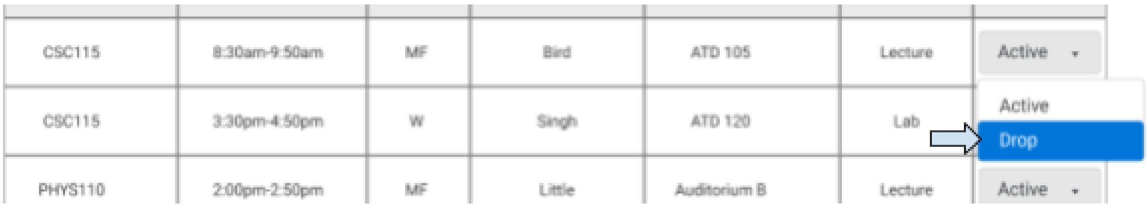
Any

Sept 02, 2019 - Sept 06, 2019

< Prev Week — Next Week >

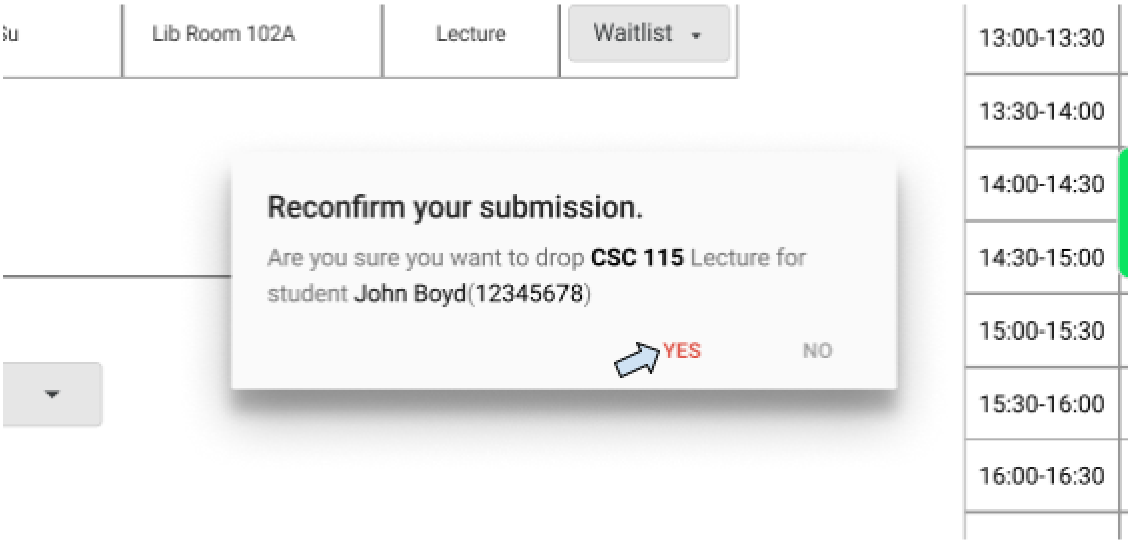
	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00	CSC 115 ATD 105				CSC 115 ATD 105
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115 (Lab) ATD 120		
16:00-16:30					
16:30-17:00					

4. The administrator selects “Drop” for the desired course that needs to be dropped.



CSC115	8:30am-9:50am	MF	Bird	ATD 105	Lecture	Active
CSC115	3:30pm-4:50pm	W	Singh	ATD 120	Lab	
PHYS110	2:00pm-2:50pm	MF	Little	Auditorium B	Lecture	Active

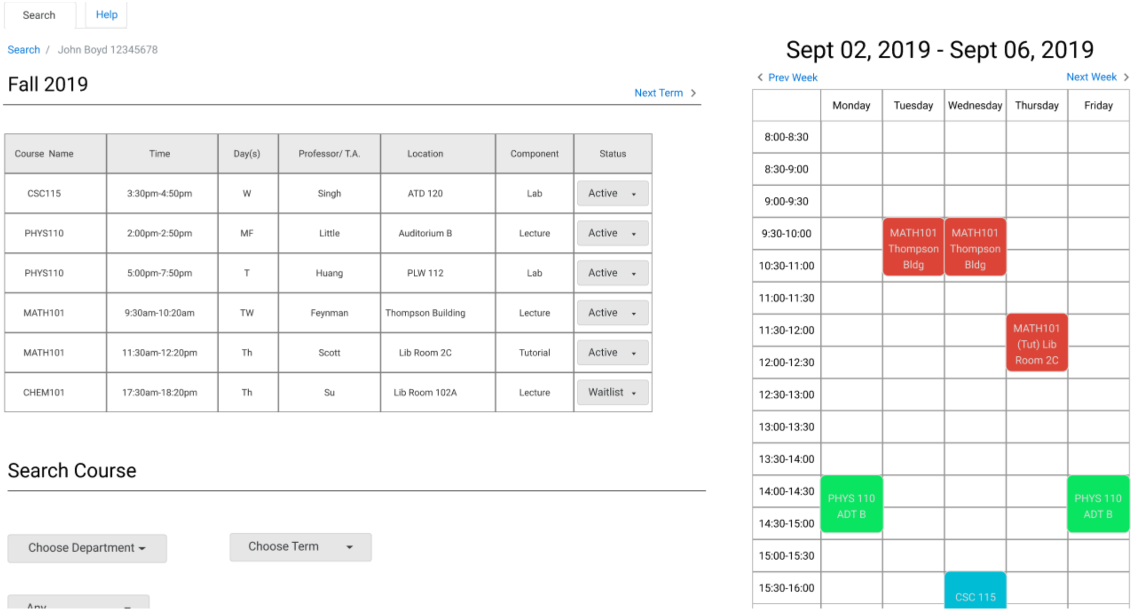
5. A pop up is displayed asking the administrator to confirm the choice of dropping the student from the selected course. In this case the administrator clicks “Yes”.



Reconfirm your submission.

Are you sure you want to drop **CSC 115** Lecture for student **John Boyd**(12345678)

6. The course is removed from the student’s timetable and the student’s timetable is updated to reflect this.



Fall 2019

Course Name	Time	Day(s)	Professor/ T.A.	Location	Component	Status
CSC115	3:30pm-4:50pm	W	Singh	ATD 120	Lab	Active
PHYS110	2:00pm-2:50pm	MF	Little	Auditorium B	Lecture	Active
PHYS110	5:00pm-7:50pm	T	Huang	PLW 112	Lab	Active
MATH101	9:30am-10:20am	TW	Feynman	Thompson Building	Lecture	Active
MATH101	11:30am-12:20pm	Th	Scott	Lib Room 2C	Tutorial	Active
CHEM101	17:30am-18:20pm	Th	Su	Lib Room 102A	Lecture	Waitlist

Search Course

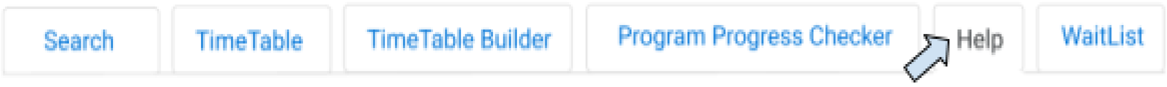
Sept 02, 2019 - Sept 06, 2019

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-8:30					
8:30-9:00					
9:00-9:30					
9:30-10:00		MATH101 Thompson Bldg	MATH101 Thompson Bldg		
10:30-11:00					
11:00-11:30					
11:30-12:00				MATH101 (Tut) Lib Room 2C	
12:00-12:30					
12:30-13:00					
13:00-13:30					
13:30-14:00					
14:00-14:30	PHYS 110 ADT B				PHYS 110 ADT B
14:30-15:00					
15:00-15:30					
15:30-16:00			CSC 115		

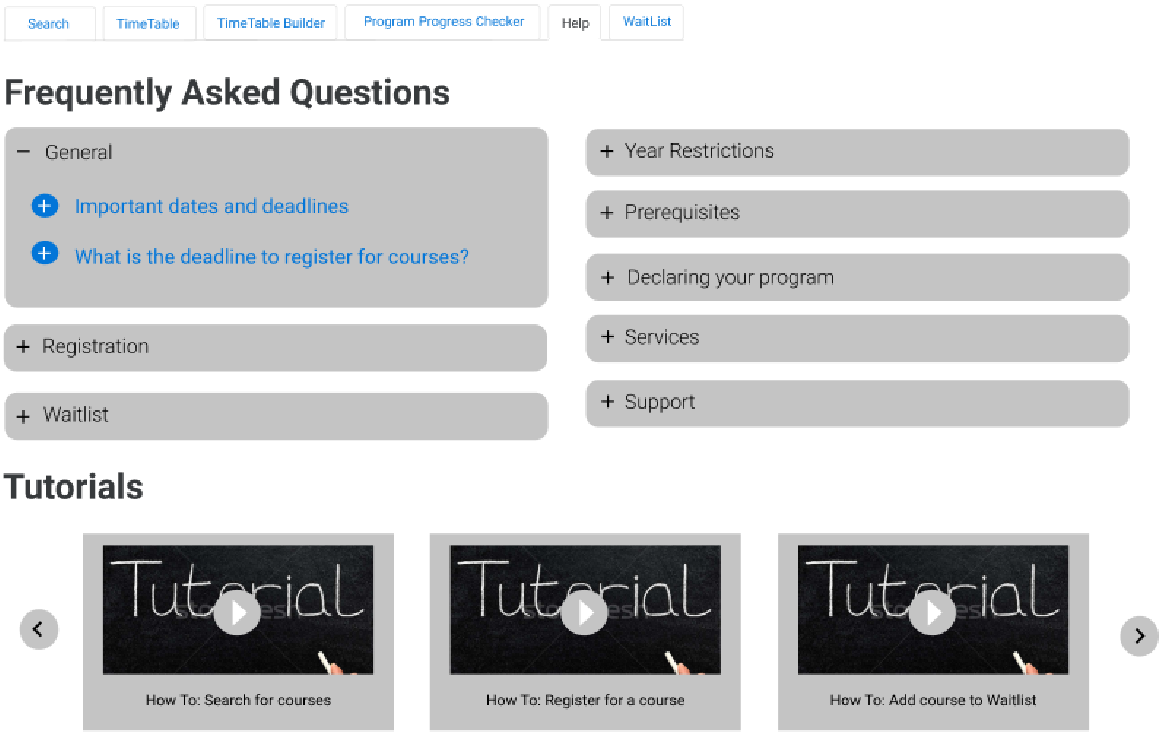
10.8 The user looks up help page (UC-9)

The user clicks on the “Help” tab along the top of the page which takes the user to the “Help” page. The “Help” page has different sections that lead to relevant information on UL’s website that will aid each user with their problems. In addition, there are multiple tutorials regarding different features of the Course Registrator.

1. The user clicks on the “Help” tab.



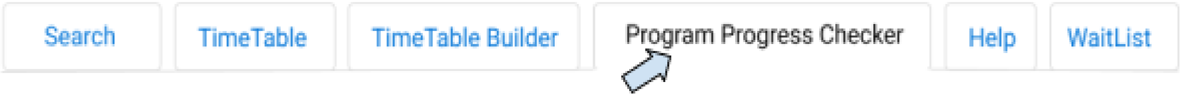
2. The user is taken to the “Help” page.



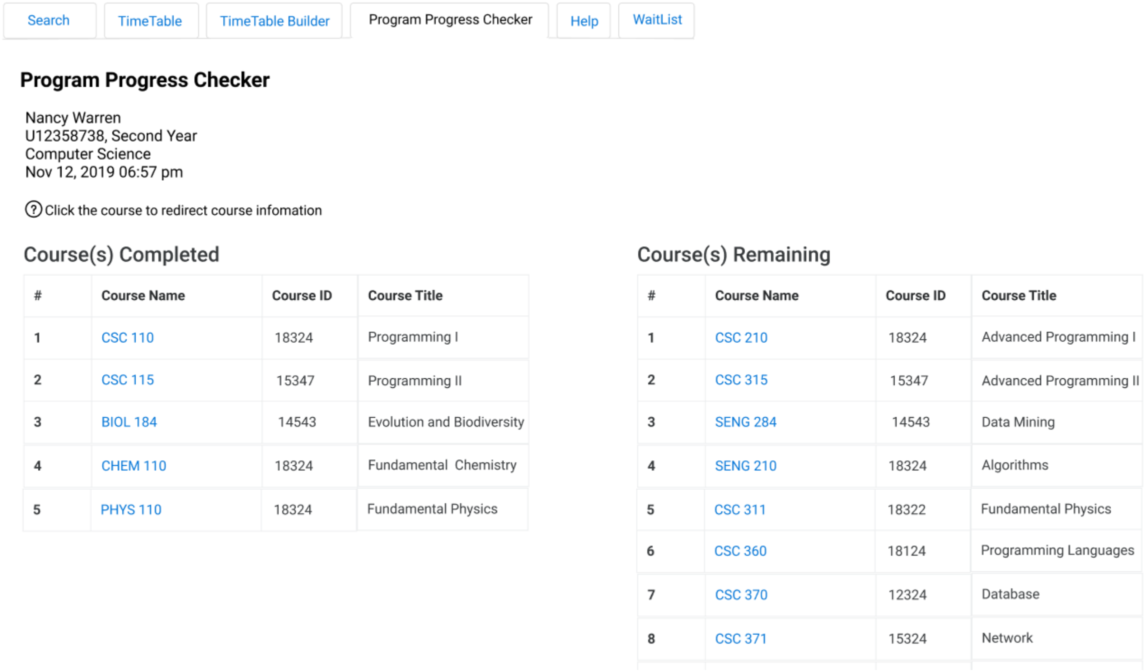
10.9 The Student accesses Program Progress Checker(UC-10)

This storyboard shows the student navigating to the “Program Progress Checker” tab, and then shows the view the student will have of the “Program Progress Checker”. This view will display a list of courses the student has completed as well as a list of the courses that still need to be completed for the student to graduate.

1. The student navigates to the “Program Progress Checker” section of the Course Registrator.



2. The student observes a list of each course that they have completed and a list of each course they must complete to graduate.



Program Progress Checker

Nancy Warren
U12358738, Second Year
Computer Science
Nov 12, 2019 06:57 pm

ⓘ Click the course to redirect course infomation

Course(s) Completed

#	Course Name	Course ID	Course Title
1	CSC 110	18324	Programming I
2	CSC 115	15347	Programming II
3	BIOL 184	14543	Evolution and Biodiversity
4	CHEM 110	18324	Fundamental Chemistry
5	PHYS 110	18324	Fundamental Physics

Course(s) Remaining

#	Course Name	Course ID	Course Title
1	CSC 210	18324	Advanced Programming I
2	CSC 315	15347	Advanced Programming II
3	SENG 284	14543	Data Mining
4	SENG 210	18324	Algorithms
5	CSC 311	18322	Fundamental Physics
6	CSC 360	18124	Programming Languages
7	CSC 370	12324	Database
8	CSC 371	15324	Network

Section 11 Test Scenarios

This section contains test scenarios to ensure that all requirements outlined in this document are covered and testable

TEST-1: Enter the timetable builder. Ensure that there is a department filter and time filter. Also, ensure that there are check boxes for each day of the week. Now, select department and choose an option. Ensure that each course returned is under that selected department. Register for one of the courses on this list that has no restrictions. Verify that the course now shows up on the timetable. Remove the selection of the department filter back to the default. Now, select a time from the time filter. Ensure that each course returned occur during the time selected. Remove the time selection filter back to the default. Now, in the search box, enter the name of a course to register for (for example “seng 321”). Ensure that each course in the Course Registrator database that match the name that was searched is returned. Register for one of the courses returned that has no restrictions. Ensure

that this course is also displayed on the timetable. Now, take a look at the timetable and ensure that the basic details of each course is displayed. Next, select a different week for the timetable view. If the registered courses are still offered during this week, ensure that they are still displayed. If the courses are not supposed to be displayed during the selected week, ensure they are not displayed. Then, in the registered course section, ensure that the course details for each course registered in this test are displayed. After that, select one of the registered courses and drop that course. Verify that the student has been removed from that course and that the dropped course is no longer displayed on the timetable or registered courses section.
RR: 3-1-2a-1, 3-1-2a-2, 3-1-2a-3, 3-1-2a-4, 3-1-2a-6, 3-2-2-1, 3-2-2-6, 3-2-2-7, 3-2-2-8, 3-2-2-9

TEST-2: Enter the administrator view and select course search. Then, using the filters or the search bar. search for a course. Once a course has been found, select the edit button for that course and ensure that a new window that displays all alterable details about the course pops up. Under the professor section of details, you should see the name of the current professor as well as a field to input the name of a new professor. Input the name of a new professor and save your changes. Now return to search and find the course with the changed professor again. Verify that the name of the new professor is displayed. Repeat this test for all other alterable class details.
RR: 3-1-2a-5, 3-2-2-5

TEST-3: Enter the administrator view, and select student search. In the search bar, input a declared program name and select search. Verify that the returned results are every student that has declared that major and that for each student, you can see their name, student ID, and declared program. Now select one student from the returned list and take note of their name and student ID. Next, input this student ID into the search bar, run the search and verify that the returned list has only one student as the ID should be unique and that the returned student name matches the note taken earlier. Now, select to edit this student and verify that you are taken to this student's Timetable Builder. In the student's Timetable Builder, ensure that you can see the timetable. Next, using the search features, find a course to add and register the student in the course. Verify that the student is now in the course. Next, using search, find a course that the student is restricted from taking. Register the student in the course. Verify that there is no problem adding this student to the restricted course and ensure that the student is now registered for the course. Once you have ensured the student is in the course, select drop to drop that course. Verify that the student has been removed from the course.
RR: 3-1-2b-1,3-1-2b-2, 3-1-2b-3, 3-1-2b-4, 3-2-2-2, 3-2-2-3, 3-2-2-4

TEST-4: Enter the view of a professor. Observe the timetable. Verify that each course the professor used for this test is teaching is displayed on the timetable. In addition, Verify that the start time, end time and location information of each class is displayed as well.
RR: 3-2-2-10, 3-2-2-11

TEST-5: Enter the Course Registrator and navigate to the timetable. Verify that the timetable displays shows the current week. Verify that the timetable displays 5 columns in order: Monday, Tuesday, Wednesday, Thursday, Friday. Verify that there are 28 columns displaying the time from 8am to 10pm divided by 30 minute time slots.
RR: 3-2-3-1, 3-2-3-2, 3-2-3-3

TEST-6: When trying to register for a course, if the Course Registrator displays a prompt saying that the course is full and the prompt asks if the course should be added to the waitlist, select "Yes" to add the course to the Trackable Waitlist.
RR: 3-3-2-1

TEST-7: Click on the tab titled “Waitlist”. Either within “Waitlist offers” or “Waitlisted Classes”, observe the waitlist information of each course, including the current position on the course’s waitlist under the column titled “Current Position”.
RR: 3-3-2-2

TEST-8: Open the Course Registrator. If there is an open space in a course, the student at position 1 will be able to see a notification symbol on their “Waitlist” tab. Upon clicking the “Waitlist” tab, a student will be able to see the offered course under “Waitlist Offers”.
RR: 3-3-2-3

TEST-9: Click on the tab titled “Waitlist”. Navigate to the bottom of the screen to the table titled “Waitlisted Classes” and click on the drop down menu beside the class that is to be dropped. Select “Drop” and when asked to confirm, reselect drop to successfully drop the course.
RR: 3-3-2-4

TEST-10: Click on the tab titled “Waitlist”. Navigate to the bottom of the screen and observe that the dropped waitlisted course is no longer displayed under “Waitlisted Classes”.
RR: 3-3-2-5

TEST-11: Click on the tab titled “Waitlist”. Either within “Waitlist offers” or “Waitlisted Classes”, observe the waitlist information of each course, including the number of students currently waitlisted for a course under the column titled “# of Students on Waitlist”.
RR: 3-3-2-6

TEST-12: Click on the tab titled “Waitlist”. Either within “Waitlist offers” or “Waitlisted Classes”, observe the waitlist information of each course, including the maximum number of students allowed on the waitlist under the column titled “Waitlist Capacity”.
RR:3-3-2-7

TEST-13: Click on the tab titled “Waitlist”. Navigate to the table titled “Waitlist Offers”. Beside the desired course, click on the drop down menu and select “Accept” to accept the offer.
RR: 3-3-2-8

TEST-14: In student mode, click on the tab titled “Waitlist” and navigate to the table titled “Waitlisted Class(es)”. This table shows all the classes a student is waitlisted for.
RR: 3-3-2-9

TEST-15: Within any page of the Course Registrator, scroll to the top of the page and click on the “Help” tab to navigate to the Help page.
RR:3-4-2-1

TEST-16: Click on the tab titled Help on the Course Registrator homepage. Once in the Help page, click on the drop down menu titled “Support” and find an administrator’s contact information.
RR: 3-4-2-2

TEST-17: Click on the tab titled Help on the Course Registrator homepage. Once in the Help page, click on the drop down menu titled “Support” and find an administrator’s work hours information.
RR: 3-4-2-3

TEST-18: In the Course Registrator homepage, click on the tab titled “Help”. Once on the Help page, a student navigates to the bottom of the page, to the “Tutorial” section and clicks on the tutorial

related to their question.
RR: 3-4-2-4

TEST-19: Enter the Program Progress Checker and observe the list titled “Courses Complete”. This list shows the courses the student has already completed in the past.
RR: 3-5-2-1

TEST-20: Enter the Program Progress Checker and observe the list titled “Courses Remaining”. This list shows the courses the student needs to complete in order to finish their declared program.
RR: 3-5-2-2

TEST-21: Enter the administrator view and select the student view. Once within the student view, enter a student’s student number in the student search and select the matching result. After clicking on the desired student’s name, click on the button labeled “View”. Upon clicking “View”, click on the Program Progress Checker and view the displayed details.
RR: 3-5-2-3

TEST-22: Enter the Program Progress Checker and click on the desired course’s name. The name of the course is a clickable link which takes a student to the page displaying the course’s details.
RR: 3-5-2-4

TEST-23: Open the same web page of the Course Registrator on both desktop and mobile device screens. Ensure that the webpage is displayed clearly on both screens.
RR 4-1-1-1

TEST-24: To ensure that each web page of the Course Registrator follows common design elements, such as the aforementioned hamburger menus, access the Course Registrator from a mobile device and confirm that menus are displayed as hamburger menus. To confirm the pagination of tables, navigate to a page which displays search results, for example “Course Search”, and confirm that the table information is shown in multiple pages. Also, to confirm that website elements such as tables and graphs are dynamically updated, resize the search result window on a desktop screen and see if the display changes accordingly.
RR: 4-1-1-2

TEST-25: Perform the following actions that result in an error being displayed. Then evaluate the error message displayed based on three criteria: conciseness (must be 10 words or less), distinctness (error message must stand out when error is thrown), and intuitiveness (if it is obvious which element caused the error). First, search for a course and register for it if the course is not full. Then search for a course with a time conflict and try to add it ensure the error message displayed adheres to the three criteria above. Similarly, search for a course with a degree/ year restriction and try to add it, ensure message adheres to the criteria above. Finally, search for a course that has already been registered in. Try to add it again. Ensure message adheres to the criteria above.
RR: 4-1-2-1, 4-1-2-2, 4-1-2-3

TEST-26: Randomly select a small sample of student numbers from UL’s students. As an administrator, search for each student in the random sample using their student number and check if details such as their program and declaration status are accurate. Then, click on ‘View’ to view each student’s Program Progress Checker and verify if details such as the courses completed or courses needed are accurate.
RR: 4-3-1-1

TEST-27: This requirement can be tested by performing a variety of actions that cause changes in the Course Registrator database and then checking if the changes are retained after loading a new session. Some of the actions that can be used are as follows: First, Search for a course and register for it. Start a new session and check if the course has been added to the list of registered courses. Second, Drop a registered course. Start a new session and check if the course has been removed from the list of registered courses. Third, As a student user, accept a waitlisted course offer. Start a new session and check if the course has been added to the list of registered courses and removed from the list of waitlisted courses. Fourth As a student user. reject a waitlisted course offer. Start a new session and check if the course has been removed from the list of waitlisted courses. Finally, As an administrative user, search for a course and edit the course details. Start a new session and check if the changes in details for the course are reflected when the course is searched and/ or viewed.
RR: 4-3-1-2

TEST-28: Try to access the Course Registrator using each of the browsers listed above. Using each browser, try to use a few features of the Course Registrator.
RR 4-4-1-1

TEST-29: Try to access the Course Registrator using a mobile device via one of the supported browsers (Chrome, Firefox, Edge, Safari). Try to use each feature of the Course Registrator and verify if they work as expected.
RR 4-4-1-2

TEST-30: As a student, register for less than the required number of components for a course. Check to see if the Course Registrator sends daily email reminders to register for the missing components.
RR: 4-4-2-1

TEST-31: As an administrator, change the size of a class that has an active waitlist such that the status of the first student on the waitlist changes. Check if that student receives an email notification.
RR: 4-4-2-2

TEST-32: As an administrator, change the details of a course. For any student registered for that course, check if they receive an email notification regarding the change.
RR: 4-4-2-3

TEST-33: Set filters or type name for a course in the search bar. Start a timer when the search button is clicked. As soon as the search results are displayed, stop the timer. Verify if the action performed is executed in 1s or less.
RR: 5-1-1-1

TEST-34: Have at least two users log into the Course Registrator as any and perform a course search. Set filters or type name for a course in the search bar. Start a timer when the search button is clicked. As soon as the search results as displayed, stop the timer. Verify if actions executed are performed in 1s or less for each user performing this test scenario.
RR: 5-1-1-2

TEST-35: When a user makes a change to the timetable such as adding or removing a course from the timetable, as the button for adding a course is clicked or when a button for removing a course is clicked, start the timer. As soon as the action (adding of course or removing of the course) is executed, stop the timer. Verify that the action executed is performed in at most 1s.
RR: 5-1-1-3

TEST-36: For each scenario that can generate an error message, generate an error message and verify that it appears in less than 1 second by timing the execution time, where the start time is when the error is made and the end time is when the error message is displayed.
R: 5-1-1-4

TEST-37: Send the course registrator 10,000 search queries and verify that bandwidth is increased and queries are answered.
RR 5-1-2-1

TEST-38: Have one user send 10,000 queries to Course Registrator in 1s. Verify if the user is blocked from the Course Registrator and the IP Address has been blocked.
RR 5-1-2-2

TEST-39: Execute a cyber attack on the Course Registrator and verify if the defensive software, that is hosted on the server hosting the Course Registrator, prevents the cyber attack.
RR 5-1-2-3

TEST-40: Perform a SQL injection on the Course Registrator. Verify if the attack fails as “prepared and parameterized statements” help in preventing such attacks.
RR: 5-1-2-4, 5-1-2-5

TEST-41: Print the prepared query on the console just before execution of that query. Verify that placeholders have not been filled. As the query is executed, verify if the placeholders have been replaced by data by printing the query on the terminal/console.
RR: 5-1-2-6

TEST-42: Write a query that would implement the ORM framework. Execute the same query on the Course Registrator database and verify if the results match.
RR: **5-1-2-7**

TEST-43: Log in to the Course Registrator. As the search button is clicked start the timer. Verify if the login session has expired as soon as the timer reaches 5 minutes.
RR: 5-1-2-8

TEST-44: Log in to the Course Registrator as a Student. Try to execute any of the administrative privileges listed above. Verify that they cannot be executed by a student User
RR: 5-1-2-9

TEST-45: Enter/Update personal data as any of the User listed above. Print the data to the browser console and verify the printed data cannot be parsed.
RR: 5-1-2-10

TEST-46: Create new login information of a student on UL student database. Log in to the Course Registrator using new Student login information. Verify if the Course Registrator has not crashed..
RR: 5-1-3-1

TEST-47: Perform code coupling/cohesion test and a line count vs component/module count. Verify that both counts match.
RR: 5-1-4-1

TEST-48: Perform all unit tests for code components and verify the output passes all tests and coverage reports show 100% coverage.
RR: 5-1-4-2

TEST-49: Perform all unit tests for code modules and verify the output passes all tests and coverage reports show 100% coverage.
RR: 5-1-4-3

TEST-50: Perform all integration tests for code components and verify that all tests execute without failure.
RR: 5-1-4-4

TEST-51: Perform all integration tests for code modules and verify that tests executed execute without failure.
RR: 5-1-4-5

TEST-52: Before an update to Course Registrator perform all Regression Tests and record the results. Execute the update for Course Registrator. Once again perform Regression tests for features existing before the update and verify that the results of this test match the recorded results from Regression Tests performed before the update to Course Registrator
RR: 5-1-4-6

TEST-53: Using popular screen resolutions of desktop computer, including 640×480, 800×600, 960×720, 1024×768, 1280×960, 1400×1050, 1440×1080, 1600×1200, 1856×1392, 1920×1440, and 2048×1536 and popular phone screen resolutions, including 480×800, 640×1136, 720×1280, 750×1334, 1080×1920, and 1440×2560, to test the Course Registrator. For each resolution, verify if each web page of the Course Registrator has any missing part or incorrect css style. Check if the timetable is usable on a phone screen so that each user can use the Course Registrator from a phone.
RR: 5-1-5-1

TEST-54: Using popular resolutions used by tablets and smartphones, including 640×480, 800×600, 960×720, 1024×768, 1280×960, 1400×1050, 1440×1080, 1600×1200, 1856×1392, 1920×1440, 720×1280, 750×1334, 1080×1920, and 1440×2560 to test the Course Registrator. For each resolution, go through every use cases on the Course Registrator and make sure that each user knows where to find the button, menu, and links to redirect other page and ensure that all of the components of the Course Registrator on the desktop computer will also be used on this resolution.
RR: 5-1-5-2

TEST-55: Monitor if the Course Registrator is able to run and serve for an entire year.
Some accidents like lack of electricity, hack attacks and natural disasters must be considered.
RR: 5-1-6-1

TEST-56: Login as a student and check if any part of the Course Registrator displays any other student’s personal information.
RR: 5-1-7-1

TEST-57: Verify that the system adheres to Canadian privacy regulations.
RR: 5-1-7-2

TEST-58: Open 26,000 instances of the Course Registrator at the same time. Verify that the Course Registrator can function without crashing for thirty minutes.

RR: 5-1-8-1

TEST-59: Finding a course which is full and clicking "Add" button to add the course. Click the "Yes" button add a course to waitlist. Verify if the coursed is added. Also, ensure that a course will disappear if a user removes or accepts the course on the waitlist page.
RR: 5-2-1

TEST-60: For each topic, click a different number of times and record the number. Go to the database and ensure that the number of clicks of each topic increases correctly in the database.
RR: 5-3-1

TEST-61: For each question, click a number of times and record that number. Ensure that the number of clicks of each question recorded in the database has the correct values.
RR:5-3-2

TEST-62: Login as a student or an administrator and click "Help Page" to verify there are 10 questions on the top of help page. Afterwards, go to the database and check if these 10 questions are most frequently clicked.
RR: 5-3-3

TEST-63: Login as a student and click the "Help Page" on the top. Verify that there is a topic and describe how a student build timetable step by step.
RR: 5-3-4

TEST-64: Login as a student and find a high-level course that is relative to the student's major but the student can not take this year. Put the course on the search bar and the web page will return the course. Verify that the course shows"Year Restriction".
RR: 5-4-1

TEST-65: Login as a student and find a course beyond the student's program. Put the course on the search bar and the web page will return the course. Verify that the course shows "Degree Restriction".
RR: 5-4-2

TEST-66: Login as a student and find a course with a prerequisite course but not take yet . Put the course on the search bar and the web page will return the course results. Verify that the results shows "Missing Prerequisite".
RR: 5-4-3

TEST-67: Login as a student and find a course having a time conflict with a registered course. Put the course on the course search bar and the web page will return the course results. Verify that the results shows "Time Conflict".
RR: 5-4-4

TEST-68: After an administrator modifies Course Registrator's data, ensure that a message appears and tells the administrator that action succeeds. Then refresh the page or go to database to see if the data is modified.
RR: 6-1-1

TEST-69: After an administrator add data to Course Registrator's data, ensure that a message appears and tells the administrator that action succeeds. Then refresh the page or go to database to see if the data is added.

RR: 6-1-2

TEST-70: Based on the number of users, the Course Registrator's must shoulder the pressure of 100,000 queries and update the data. Verify if query can be completed in 0.3 seconds and ensure both server and the database have no problems during tests.
RR: 6-1-3

Section 12 Traceability Matrix

This section features the traceability matrix for our document which links each requirement to its Use Case ID, Associated Storyboard and Test Case Scenario #.

Feature/ Requirement Name	**Requirement ID**	**Use Case ID**	**Associated Storyboard**	**Test Case Scenario #**
Course Search	3-1-2a-1	UC-1	10.1	TEST-1
	3-1-2a-2	UC-1	10.1	TEST-1
	3-1-2a-3	UC-1	10.1	TEST-1
	3-1-2a-4	UC-1	10.1	TEST-1
	3-1-2a-5	UC-16	10.5	TEST-2
	3-1-2a-6	UC-1	10.1	TEST-1
Student Search	3-1-2b-1	UC-13	10.6	TEST-3
	3-1-2b-2	UC-13	10.6	TEST-3
	3-1-2b-3	UC-13	10.6	TEST-3
	3-1-2b-4	UC-14	10.6	TEST-3
Timetable Builder	3-2-2-1	UC-2	10.1	TEST-1
	3-2-2-2	UC-14	10.6	TEST-3

	3-2-2-3	UC-15	10.7	TEST-3
	3-2-2-4	UC-14	10.6	TEST-3
	3-2-2-5	UC-16	10.5	TEST-2
	3-2-2-6	UC-4	10.2	TEST-1
	3-2-2-7	UC-4	10.2	TEST-1
	3-2-2-8	UC-5	10.2	TEST-1
	3-2-2-9	UC-4	10.2	TEST-1
	3-2-2-10	UC-3		TEST-4
	3-2-2-11	UC-3		TEST-4
	3-2-3-1	UC-4	10.2	TEST-5
	3-2-3-2	UC-4	10.2	TEST-5
	3-2-3-3	UC-4	10.2	TEST-5
Trackable Waitlist	3-3-2-1	UC-2	10.2	TEST-6
	3-3-2-2	UC-6	10.3	TEST-7
	3-3-2-3	UC-7	10.3	TEST-8
	3-3-2-4	UC-8	10.4	TEST-9
	3-3-2-5	UC-8	10.4	TEST-10
	3-3-2-6	UC-6	10.3	TEST-11

	3-3-2-7	UC-6	10.3	TEST-12
	3-3-2-8	UC-7	10.3	TEST-13
	3-3-2-9	UC-2	10.1	TEST-14
Help Page	3-4-2-1	UC-9	10.8	TEST-15
	3-4-2-2	UC-9	10.8	TEST-16
	3-4-2-3	UC-9	10.8	TEST-17
	3-4-2-4	UC-9	10.8	TEST-18
Program Progress Checker	3-5-2-1	UC-10	10.9	TEST-19
	3-5-2-2	UC-10	10.9	TEST-20
	3-5-2-3	UC-12		TEST-21
	3-5-2-4	UC-13	10.9	TEST-22
User Interface	4-1-1-1	N/A	N/A	TEST-23
	4-1-1-2	N/A	N/A	TEST-24
	4-1-2-1	N/A	N/A	TEST-25
	4-1-2-2	N/A	N/A	TEST-25
	4-1-2-3	N/A	N/A	TEST-25
Software Interfaces	4-3-1-1	N/A	N/A	TEST-26
	4-3-1-2	N/A	N/A	TEST-27

Communications Interfaces	4-4-1-1	N/A	N/A	TEST-28
	4-4-1-2	N/A	N/A	TEST-29
	4-4-2-1	N/A	N/A	TEST-30
	4-4-2-2	N/A	N/A	TEST-31
	4-4-2-3	N/A	N/A	TEST-32
Software Quality Attributes	5-1-1-1	N/A	N/A	TEST-33
	5-1-1-2	N/A	N/A	TEST-34
	5-1-1-3	UC-2, UC-5, UC-7	N/A	TEST-35
	5-1-1-4	UC-2	N/A	TEST-36
	5-1-2-1	N/A	N/A	TEST-37
	5-1-2-2	N/A	N/A	TEST-38
	5-1-2-3	N/A	N/A	TEST-39
	5-1-2-4	N/A	N/A	TEST-40
	5-1-2-5	N/AN/A	N/A	TEST-40
	5-1-2-6	N/A	N/A	TEST-41
	5-1-2-7	N/A	N/A	TEST-42
	5-1-2-8	N/A	N/A	TEST-43
	5-1-2-9	N/A	N/A	TEST-44

	5-1-2-10	N/A	N/A	TEST-45
	5-1-3-1	N/A	N/A	TEST-46
	5-1-4-1	N/A	N/A	TEST-47
	5-1-4-2	N/A	N/A	TEST-48
	5-1-4-3	N/A	N/A	TEST-49
	5-1-4-4	N/A	N/A	TEST-50
	5-1-4-5	N/A	N/A	TEST-51
	5-1-4-6	N/A	N/A	TEST-52
	5-1-5-1	N/A	N/A	TEST-53
	5-1-5-2	N/A	N/A	TEST-54
	5-1-6-1	N/A	N/A	TEST-55
	5-1-7-1	N/A	N/A	TEST-56
	5-1-7-2	N/A	N/A	TEST-57
	5-1-8-1	N/A	N/A	TEST-58
Trackable Waitlist (Non Functional Requirements)	5-2-1	N/A	N/A	TEST-59
Help Page (Non Functional Requirements)	5-3-1	N/A	N/A	TEST-60
	5-3-2	N/A	N/A	TEST-61

	5-3-3	UC-9	10.3	TEST-62
	5-3-4	UC-9	10.3	TEST-63
Immediate Error Checking	5-4-1	UC-2	10.1	TEST-64
	5-4-2	UC-2	10.1	TEST-65
	5-4-3	UC-2	10.1	TEST-66
	5-4-4	UC-2	10.1	TEST-67
Database Requirements	6-1-1	N/A	N/A	TEST-68
	6-1-2	N/A	N/A	TEST-69
	6-1-3	N/A	N/A	TEST-70

Appendix: Issues List

Negotiation of Scope - Meeting took place on October 9th 2019.

Negotiated Administrator functional requirements.

Academic advisors and department heads are administrators and the department head’s interactions are removed.

Recruiters are no longer a user class. Will be replaced by more information on the help page.