

Release Planning Document

Course Title: COSC 4P02

Course Instructor: Naser Ezzati-Jivan

Due Date: January 19th, 2025

Team Name: The Mixers

Ashu Chauhan – 7001571

Avi Patel – 6741961

Fatima Abourida - 7119490

Jerome Uwaneme -7141270

Olaoluwa Akanji - 6908776

Oreoluwa Akanji - 6910483

Russell Salacup – 7177884

Table of Contents

Table of Contents.....	2
1. Introduction (Jerome).....	3
2. User Stories (Fatima).....	3
3. Product Backlog (Ola).....	5
Backlog screenshots:.....	7
4. Sprint Backlogs (Avi).....	8
5. Team Contributions.....	10
6. Challenges and Next Steps (Ashu).....	10
7. Database Research (Kitan).....	11
8. Appendices (Fatima).....	11

1. Introduction (Jerome)

Provide a brief overview of the project and its objectives. State the document's purpose (to outline user stories, backlogs, and the release plan).

Course Mix is an innovative application that allows students to achieve their academic goals and aspirations. By utilizing key metrics such as program duration, preferred graduation date, and target CGPA, the app generates a customized academic plan tailored to each student's unique preferences and requirements.

Purpose

The purpose of this document is to outline and review user stories, product and sprint backlogs, providing a concise plan for the software's release.

A note on IDs (Russell)

Jira is being used to track the information on this document. It has automatically created IDs for user stories and sub-tasks in the form "**SCRUM-#**", where the number "**#**" represents the items in chronological order of their creation (not necessarily indicating their priority or parent user story).

2. User Stories (Fatima)

User stories are high-level requirements written from the user's perspective. Each story should include the following details:

- ID
- Title
- Description
- Acceptance Criteria
- Priority (P)

ID	Title	Description	Acceptance Criteria	P
SCRUM-11	Create Account	As a student, I want to create an account so that I can save my academic preferences and progress.	Account creation form captures all required fields. Email verification process is functional. New account can be created and logged into without errors.	HIGH
SCRUM-131	Safety	As a student, I want to feel assured of safety and data protection.	All data transmissions are encrypted using SSL	HIGH
SCRUM-16	Course Conflict	As a student, I want to be able to easily identify course conflicts	The system alerts the user when they attempt to register for courses with overlapping times.	HIGH

SCRUM-17	Courses Required	As a student, I want to be able to easily see and plan the courses needed to pass my registered program	The system lists all courses required for the student's program. Visual indicators show completed and pending courses.	HIGH
SCRUM-20	Prerequisites	As a student, I want prerequisites for my courses to be auto-scheduled per semester so that my academic plan stays on track.	Prerequisite courses are automatically checked before allowing course registration. Provides a warning message if prerequisites are not met.	HIGH
SCRUM-21	Manual Adjustment	As a student, I want the ability to manually adjust the prerequisite schedule so that it fits my personal circumstances.	Allows user to manually adjust schedules. Changes are saved and reflected immediately in the user interface.	HIGH
SCRUM-8	Program Status	As a student, I want to be able to select my program status (coop, ...)	Drop-down menu allows selection of current program status. System updates user profile and permissions based on status selection.	HIGH
SCRUM-83	Past Grades	As a student, I want to be able to see my past grades as well as credits earned currently in the app	Accurately retrieves and displays past grades for all completed courses. Updates in real-time after grades are posted each semester.	HIGH
SCRUM-9	Create Profile	As a student, I want to be able create my profile (registration date, graduation date,...)	Profile creation captures and saves all relevant user information. Users can view and edit their profiles post-creation.	HIGH
SCRUM-23	Visualized Schedule	As a student, I want a timetable to be automatically generated based on my selected courses so I can visualize my schedule.	Timetable is generated and visually displayed based on user's course selections. Perhaps drag and drop functionality for manual schedule adjustments.	HIGH
SCRUM-12	Anonymous Reviews	As a student using the app, I want to be able to provide reviews and opinions regarding courses without being identified	Reviews are posted without revealing user identity. Only User's who registered into the course can add a review.	MED
SCRUM-71	Save Login	As a student, I want to be able to save my login after setting up my account	System remembers user login on the device unless manually logged out. Security measures to protect saved login information.	MED
SCRUM-19	See Reviews	As a student, I want to see reviews, failure rates, and other insights for each course so I can make informed decisions.	Reviews for courses are visible and include ratings and comments. Filter and sort functionality for reviews based on date, rating, etc.	MED
SCRUM-63	Recommend ed Electives	As a student i want the software to recommend suitable electives based on my interests and program requirements	System suggests electives based on user's past courses and interests. Provides elective options and quantity that match the program requirements.	MED

SCRUM-69	Update CPGA	As a student, I want access my updated CPGA at the end of every semester based off my recent grades.	CPGA is automatically calculated and updated each semester. Historical CPGA chart is available for review.	MED
SCRUM-121	Suggestions	As a student, I want to be able to access personalized suggestions and detailed statistics for my required courses.	System provides suggestions based on academic performance using reviews and other data. Customizable settings to refine the type of suggestions received.	LOW
SCRUM-72	Important Dates Notifications	As a student, I want to receive notifications for important course-related deadlines (e.g., add/drop dates, withdrawal deadlines) so that I can avoid missing critical actions	Notifications for important dates are sent via email and visible in the app. Customizable notification settings (enable/disable specific notifications).	LOW
SCRUM-24	See Course List	As a student, I want access to all Brock University courses so I can explore my options easily.	Comprehensive list of all courses offered is available. Courses can be filtered by department, availability, and other criteria.	LOW
SCRUM-70	Update Degree Audit	As a student, I want an updated degree audit worksheet of completed courses to show my current academic year	Automatically updates with course completions and requirements met. Provides a visual progress bar and detailed list of remaining requirements.	LOW
SCRUM-73	Career Paths	As a student, I want the app to provide insights into potential career paths or further education options based on the courses I've taken so that I can plan for life after graduation	Displays potential career paths based on the user's course history. Links to resources for further exploration of each career option.	LOW
SCRUM-89	Graduation Countdown	As a student, I want to be able to see how close I am to graduation	Shows a countdown to the user's expected graduation date. Alerts for any actions needed to ensure eligibility for graduation.	LOW

3. Product Backlog (Ola)

The product backlog is a prioritized list of tasks derived from user stories. Each task should include:

- Task

Add detailed explanations to the tasks if it is needed.

- Associated User Story









- Priority

Task	Description	Associated User Story	Priority
SCRUM-75	add selected program status schedule to user profile	SCRUM-8	High

SCRUM-74	link the drop down menu options to the database	SCRUM-8	Medium
SCRUM-14	add the schedules for each program status to the database	SCRUM-8	High
SCRUM-13	create a drop down option with the different program status options	SCRUM-8	Medium
SCRUM-84	add drop menu "Select Schedule Preference" we could provide few options (example: all my classes on same day to reduce days on campus)	SCRUM-9	Medium
SCRUM-82	add drop down menu "Select goal" options: graduate ASAP; highest gpa; and more	SCRUM-9	Medium
SCRUM-81	add desired graduation date, to determine the work load	SCRUM-9	Medium
SCRUM-80	add registration date (mm/yyyy), it is important because schedules may differ each year	SCRUM-9	High
SCRUM-79	add the drop down menu for program status as a field	SCRUM-9	High
SCRUM-78	add name field	SCRUM-9	High
SCRUM-77	automatically link email address used to create account to email field	SCRUM-9	Medium
SCRUM-76	create a profile form structure	SCRUM-9	Highest
SCRUM-31	Add input fields for target GPA (numerical input with range 0–4.0)	SCRUM-9	Medium
SCRUM-32	Validate inputs (e.g., ensuring GPA is within range and graduation date is realistic)	SCRUM-9	Low
SCRUM-33	Verify data persistence after updates and user logout/login.	SCRUM-9	Low

SCRUM-25	Design and implement a registration form with fields for name, email, password	SCRUM-11	High
SCRUM-26	Create a login page with a “Forgot Password” option.	SCRUM-11	High
SCRUM-29	Create a table to store users with fields for ID, name, email, hashed password, and profile preferences.	SCRUM-11	High
SCRUM-27	Set up user authentication using secure methods like JWT or OAuth	SCRUM-11	Medium
SCRUM-28	Hash passwords before storing them in the database.	SCRUM-11	Medium
SCRUM-30	Test registration and login with valid and invalid data inputs.	SCRUM-11	Low
SCRUM-117	Design a secure "Remember Me" option on the login page.	SCRUM-71	Low
SCRUM-118	Handle automatic login using saved credentials when the session is valid.	SCRUM-71	Low
SCRUM-120	Store encrypted user credentials or session tokens securely on the client side.	SCRUM-71	Medium

Backlog screenshots: (Russell)

<input type="checkbox"/>	Type	# Key	Summary	Sprint ↑
<input type="checkbox"/>	▼ 	SCRUM-8	As a student, I want to be able to select my program status (coop, ...)	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-75	add selected program status schedule to user profile	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-74	link the drop down menu options to the database	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-14	add the schedules for each program status to the database	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-13	create a drop down option with the different program status options	SCRUM Sprint 1
<input type="checkbox"/>	▼ 	SCRUM-9	As a student, I want to be able create my profile (registration date, graduation date,...)	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-84	add drop menu "Select Schedule Preference" we could provide few options (example: all my classes ...	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-82	add drop down menu "Select goal" options: graduate ASAP; highest gpa; and more	SCRUM Sprint 1

#84 = add drop menu "Select Schedule Preference" we could provide few options (example: all my classes on same day to reduce days on campus)

<input type="checkbox"/>	Type	# Key	Summary ↑	▼ Sprint ↑
<input type="checkbox"/>		SCRUM-82	add drop down menu "Select goal" options: graduate ASAP; highest gpa; and more	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-81	add desired graduation date, to determine the work load	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-80	add registration date (mm/yyyy), it is important because schedules may differ each year	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-79	add the drop down menu for program status as a field	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-78	add name field	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-77	automatically link email address used to create account to email field	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-76	create a profile form	SCRUM Sprint 1
<input type="checkbox"/>	▼	SCRUM-11	As a student, I want to create an account so that I can save my academic preferences and progress.	SCRUM Sprint 1
<input type="checkbox"/>	Type	# Key	≡ Summary	🔍 Sprint ↑
<input type="checkbox"/>	▼	SCRUM-11	As a student, I want to create an account so that I can save my academic preferences and progress.	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-25	Design and implement a registration form with fields for name, email, password	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-26	Create a login page with a "Forgot Password" option.	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-29	Create a table to store users with fields for ID, name, email, hashed password, and profile preferences.	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-27	Set up user authentication using secure methods like JWT or OAuth	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-28	Hash passwords before storing them in the database.	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-30	Test registration and login with valid and invalid data inputs.	SCRUM Sprint 1
<input type="checkbox"/>	➤	SCRUM-18	As a student, I want to be able to set my graduation date so that the web app can help me plan my ...	SCRUM Sprint 1

#18 = As a student, I want to be able to set my graduation date so that the web app can help me plan my courses accordingly

<input type="checkbox"/>	Type	# Key	≡ Summary	🔍 Sprint ↑
<input type="checkbox"/>	▼	SCRUM-18	As a student, I want to be able to set my graduation date so that the web app can help me plan my ...	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-31	Add input fields in the profile settings for graduation date (date picker) and target GPA (numerical i...	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-32	Validate inputs (e.g., ensuring GPA is within range and graduation date is realistic)	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-33	Verify data persistence after updates and user logout/login.	SCRUM Sprint 1
<input type="checkbox"/>	▼	SCRUM-71	As a student, I want to be able to save my login after setting up my account	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-117	Design a secure "Remember Me" option on the login page.	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-118	Handle automatic login using saved credentials when the session is valid.	SCRUM Sprint 1
<input type="checkbox"/>		SCRUM-120	Store encrypted user credentials or session tokens securely on the client side.	SCRUM Sprint 1

4. Sprint Backlogs (Avi)

Sprint backlogs are subsets of the product backlog, planned for specific sprints. Provide a breakdown of tasks for each sprint (do this for all sprints, understanding that tasks for later sprints can be revised as the project evolves).

Scrum Sprint 1	Tasks:
----------------	--------

	<ol style="list-style-type: none"> 1. Create a landing page 2. Add button functionality for signup and login 3. Create a new page for the signup and link it to landing page 4. Design and implement a signup form with fields for name, email, password 5. Design and implement only login function with forgot password functionality 6. Set up the database we will use for the project, most likely MySQL 7. Create a table to store users credentials(when signing up) with fields for ID, name, email, password 8. Set up user authentication using secure methods like JWT or OAuth 9. implement a remember me functionality 10. perform unit testing for the registration and login functionalities using edge cases and other scenarios 11. Design a page for profile setup, directly after signup 12. Add input fields in the profile setup for graduation date (date picker) and other relevant metrics 13. create a dropdown to select program(honours or coop) 14. Create a save profile button that saves graduation date, target gpa and program into MySQL db
Scrum Sprint 2	<ol style="list-style-type: none"> 1. Create a page for when the user logs in, it should display the users name, navigation bar with profile, logout button 2. Create a sidebar with menu options for, course registration, schedule display, course review 3. Create a page for course registration, 4. Build a comprehensive database of Brock University courses, including course codes, names, descriptions, prerequisites, and semester availability, either web scrape of manual entry 5. Design a searchable and filterable interface to browse courses, also implement 6. Include sorting options by semester, year, difficulty, or category. 7. Design the UI in a way that the courses are drag and drop, and prereqs/cs courses are automatically added to year number(year 1-5), also need to display failure rate of course 8. Create an algorithm based on computer science undergraduate course calendar, so that it automatically picks the required courses by year(cosc 1p02,1p03 etc) 9. Take into account custom schedules for electives and failed courses 10. Save schedule to a database and display courses taking for current year in schedule tab 11. Design the schedule display UI such that it shows 7 days a week and time to display courses 12. In the schedule display, implement some logic that flags courses which overlap either by unique colour or notification 13. On the home page add a table view for credits, the table must

	<p>have type(major,minor,social science, humanities), planned credits, completed credit</p> <p>14. Add another table to show planned and completed credits for ,numbered 1(alpha)00 to 1(alpha)99; at least three credits must be numbered 2(alpha)90 or above; at least three credits must be numbered 3(alpha)90 or above; and the remaining credits must be numbered 2(alpha)00 or above</p>
Scrum Sprint 3	<ol style="list-style-type: none"> 1. Create the landing page for the course review page 2. Implement functionality for students to post review for specific course, so essentially add a button first with add post 3. After add post button design a text box/field with course code/title and textbox for review 4. This page can be like a live feed section accessible to everyone except if someone wanted to post review they need account 5. Store the responses in a database and display them, add a timestamp to the post 6. implement functionality to filter by latest post, or course code 7. Add a checkbox so posts become anonymous if desired 8. Implement functionality to edit a post made by a user 9. Add a comment section to user posts so others can comment 10. implement functionality to store comments in the database 11. On the home landing page after login, implement functionality for brock account to send notifications for important course-related deadlines (e.g., add/drop dates, withdrawal deadlines)

5. Team Contributions

Provide a breakdown of each team member's contribution to the project and report.

Example Table:

Team Member	Contribution
Everyone	Creating user stories and their tasks
Ashu Chauhan	Challenges and Next Steps
Avi Patel	Sprint Backlog
Fatima Abourida	Meeting notes and User stories table
Jerome Uwaneme	Introduction
Olaoluwa Akanji	Product Backlog

Oreoluwa Akanji	Database research
Russell Salacup	Proofreading, miscellaneous additions, syncing to Jira, Screenshots

6. Challenges and Next Steps (Ashu)

Some challenges we have faced during the planning process is actually determining what we believe is a good end product, and being aligned on our goals with this project. I believe that we could have a few more discussions to discuss what we want the end product to look like so that we can better proceed when we begin developing it.

Another key concern is how we intend to design the system, a key hurdle we will face is how we wish to gather the data required for our application. Potential solutions could be scraping it from the Brock courses webpage, however, this link may change year by year, and thus we would need to update it when necessary. We will have to look at potential solutions for this, and possibly find a different way to gather the data necessary for our application. We also must decide how we wish to store the data, and how we wish to design our database to fit our web apps needs so that we can easily pull data as necessary with which our site will be populated.

As a team, everyone has their strengths and weaknesses and determining our tech stack is another challenge that we must tackle soon so that we can all learn frameworks / technologies that will be required for the project, and begin developing soon, as there may need to be time dedicated to learning tools that many have not used yet. This could be learning React, or JavaScript, and many other backend frameworks as well such as Django or Express. Moving forward, we must decide on an initial plan for the tech stack we wish to use, and everyone has to get comfortable with working within that stack with regards to what their role will be in the development team.

After that, we will begin with our sprints, and we will adjust the amount of tasks per sprint as we go through them to ensure we have adequate tasks in each spring, but not too much to the point where we are not progressing as much as we wish to in a sprint. All in all, our team is ambitious and I am positive that we will deliver a good final product, but challenges regarding system design, tech stack decisions, and features are things that we must tackle asap in order to meet our goals.

7. Database Research (Kitan)

For the Course Mix project, we evaluated several databases and narrowed our options to two: SQLite and MySQL. Below is a brief comparison of these databases, highlighting their

differences and why SQLite is the more suitable choice based on the functionality required for Course Mix.

	SQLite	MySQL
Type	Serverless, lightweight, and self-contained database	Server-based relational database management system (RDBMS)
Storage	Stores the entire database as a single file, making it portable and easy to manage	Requires a dedicated server to host and manage databases
Setup	Requires no installation or server setup, which simplifies deployment	Needs server installation, configuration, and management
Concurrency	Supports multiple read operations simultaneously but allows only one write operation at a time.	Handles multiple simultaneous read and write operations, designed for large-scale, multi-user environments
Scalability	Best suited for smaller-scale applications or local storage needs	Suitable for enterprise-level applications with complex queries and heavy traffic
Performance	Excels in simple, read-heavy operations due to its lightweight architecture	Optimized for large-scale, write-intensive, and complex transactional workloads

Why SQLite is Better for Course Mix Based on Functionality

Portability and Simplicity: SQLite stores the entire database in a single file, making it portable and easy to manage. This is especially useful for a project like Course Mix, where backups, transfers, and updates must be seamless.

Optimized for Read-Heavy Workloads: The Course Mix app primarily involves academic planning, querying course details, and displaying insights—operations that are heavily

read-oriented. SQLite's performance for read-heavy operations makes it a natural fit for this use case.

Cost-Effective and Maintenance-Free: SQLite does not require server administration, making it a cost-effective choice. It reduces the complexity and resources needed to manage the database, aligning with the goals of a streamlined application like Course Mix.

Lightweight and Easy Integration: SQLite's serverless and self-contained design eliminates the need for a separate database server. This simplicity makes it ideal for a lightweight application like Course Mix.

Simplicity for Small to Medium-Scale Applications: SQLite is best suited for small to medium-scale applications, matching the functional scope of Course Mix. Unlike MySQL, which is designed for larger, enterprise-level applications, SQLite offers simplicity and efficiency that better fit Course Mix's needs.

8. Appendices (Fatima)

Meeting notes:

User Stories Brain Storm

->GOAL: Think, discuss about the requirements (user stories)

- As a university, I want to be able to purchase a license for my organization.
 - Select student quantity
 - Add the email domain
 - Add all the emails until the threshold (student quantity selected) is reached

What is Course Mix?

- A personalized academic advising app
- A guide
- Metrics (Uses methods like calculating GPA to create a plan)
- Website
- Backend: create database of courses we pull uploaded from Brock
 - Current scale: "hard coded" (not directly from Brock servers)
 - Research if it's possible to get these requirements somewhere on Brock's sites
- Select uni
- Profile: target GPA, graduation date target
- Database: all Brock courses **offered?**
- Reviews: Course rating, text, failure rate (how accurate?)
 - Keep reviews small and quick to complete

Core Functionalities:

- User login /register page
- Profile creation with preferences(graduation date, target GPA)
- Pick University

- Pick Course/ degree pursuing
- Database with all Brock Courses
- Schedule for Coop and honour distinction
- Auto set Prereqs per semester and allow students to add courses
- When adding courses, students should see course reviews, failure rates etc..
- Timetable should be generated

Task: Add a minimum of 2 distinct user stories to Sprint 1, and add the tasks
List, go to plus, add child
Deadline: Thursday @7:00pm

web scraping to get data of courses ^automating the process,
links may change year by year / term by term
see if a similar system exists: course of action dashboard
review user stories sprint backlog: tasks
database management system: hardcoded database at current state electives? try to figure out
what knowledge you already have, and what you don't start learning now
how are we testing?

Excel sheet of User Stories:

ID	Title	Description	Acceptance Criteria	Priority
SCRUM-11	Create Account	As a student, I want to create an account so that I can save my academic preferences and progress.	Account creation form captures all required fields. Email verification process is functional. New account can be created and logged into without errors.	HIGH
SCRUM-131	Safety	As a student, I want to feel assured of safety and data protection.	All data transmissions are encrypted using SSL.	HIGH
SCRUM-16	Course Conflict	As a student, I want to be able to easily identify course conflicts	The system alerts the user when they attempt to register for courses with overlapping times.	HIGH
SCRUM-17	Courses Required	As a student, I want to be able to easily see and plan the courses needed to pass my registered program	The system lists all courses required for the student's program. Visual indicators show completed and pending courses.	HIGH
SCRUM-20	Prerequisites	As a student, I want prerequisites for my courses to be auto-scheduled per semester so that my academic plan stays on track.	Prerequisite courses are automatically checked before allowing course registration. Provides a warning message if prerequisites are not met.	HIGH
SCRUM-21	Manual Adjustment	As a student, I want the ability to manually adjust the prerequisite schedule so that it fits my personal circumstances.	Allows user to manually adjust schedules.	HIGH
SCRUM-8	Program Status	As a student, I want to be able to select my program status (coop,...)	Changes are saved and reflected immediately in the user interface.	HIGH
SCRUM-83	Past Grades	As a student, I want to be able to see my past grades as well as credits earned currently in the app	System updates user profile and permissions based on status selection. Accurately retrieves and displays past grades for all completed courses. Updates in real-time after grades are posted each semester.	HIGH
SCRUM-9	Create Profile	As a student, I want to be able create my profile (registration date, graduation date,...)	Profile creation captures and saves all relevant user information.	HIGH
SCRUM-23	Visualized Schedule	As a student, I want a timetable to be automatically generated based on my selected courses so I can visualize my schedule.	Users can view and edit their profiles post-creation. Timetable is generated and visually displayed based on user's course selections. Perhaps drag and drop functionality for manual schedule adjustments.	HIGH
SCRUM-12	Anonymous Reviews	As a student using the app, I want to be able to provide reviews and opinions regarding courses without being identified	Reviews are posted without revealing user identity. Only users who registered into the course can add a review.	MED
SCRUM-71	Save Login	As a student, I want to be able to save my login after setting up my account	System remembers user login on the device unless manually logged out. Security measures to protect saved login information.	MED
SCRUM-19	See Reviews	As a student, I want to see reviews, failure rates, and other insights for each course so I can make informed decisions.	Reviews for courses are visible and include ratings and comments.	MED
SCRUM-63	Recommended Electives	As a student I want the software to recommend suitable electives based on my interests and program requirements	Filter and sort functionality for reviews based on date, rating, etc. System suggests electives based on user's past courses and interests.	MED
SCRUM-69	Update CPGA	As a student, I want access my updated CPGA at the end of every semester based off my recent grades.	Provides elective options and quantity that match the program requirements. CPGA is automatically calculated and updated each semester. Historical CPGA chart is available for review.	MED
SCRUM-121	Suggestions	As a student, I want to be able to access personalized suggestions and detailed statistics for my required courses.	System provides suggestions based on academic performance using reviews and other data. Customizable settings to refine the type of suggestions received.	LOW
SCRUM-72	Important Dates Notifications	As a student, I want to receive notifications for important course-related deadlines (e.g., add/drop dates, withdrawal deadlines) so that I can avoid missing critical actions	Notifications for important dates are sent via email and visible in the app. Customizable notification settings (enable/disable specific notifications).	LOW
SCRUM-24	See Course List	As a student, I want access to all Brock University courses so I can explore my options easily.	Comprehensive list of all courses offered at the university is available. Courses can be filtered by department, availability, and other criteria.	LOW
SCRUM-70	Update Degree Audit	As a student, I want an updated degree audit worksheet of completed courses to show my current academic year	Automatically updates with course completions and requirements met. Provides a visual progress bar and detailed list of remaining requirements.	LOW
SCRUM-73	Career Paths	As a student, I want the app to provide insights into potential career paths or further education options based on the courses I've taken so that I can plan for life after graduation	Displays potential career paths based on the user's course history. Links to resources for further exploration of each career option.	LOW
SCRUM-89	Graduation Countdown	As a student, I want to be able to see how close I am to graduation	Shows a countdown to the user's expected graduation date. Alerts for any actions needed to ensure eligibility for graduation.	LOW

Excel sheet of Project Backlog:

Task	Description	Associated User Story	Priority	Scrum Sprint Number
SCRUM-75	add selected program status schedule to user profile	SCRUM-8	High	Sprint 1
SCRUM-74	link the drop down menu options to the database	SCRUM-8	Medium	Sprint 1
SCRUM-14	add the schedules for each program status to the database	SCRUM-8	High	Sprint 1
SCRUM-13	create a drop down option with the different program status options	SCRUM-8	Medium	Sprint 1
SCRUM-84	add drop menu "Select Schedule Preference" we could provide few options (example: all my classes on s	SCRUM-9	Medium	Sprint 1
SCRUM-82	add drop down menu "Select goal" options: graduate ASAP; highest gpa; and more	SCRUM-9	Medium	Sprint 1
SCRUM-81	add desired graduation date, to determine the work load	SCRUM-9	Medium	Sprint 1
SCRUM-80	add registration date (mm/yyyy), it is important because schedules may differ each year	SCRUM-9	High	Sprint 1
SCRUM-79	add the drop down menu for program status as a field	SCRUM-9	High	Sprint 1
SCRUM-78	add name field	SCRUM-9	High	Sprint 1
SCRUM-77	automatically link email address used to create account to email field	SCRUM-9	Medium	Sprint 1
SCRUM-76	create a profile form structure	SCRUM-9	Highest	Sprint 1
SCRUM-31	Add input fields for target GPA (numerical input with range 0–4.0)	SCRUM-9	Medium	Sprint 1
SCRUM-32	Validate inputs (e.g., ensuring GPA is within range and graduation date is realistic)	SCRUM-9	Low	Sprint 1
SCRUM-33	Verify data persistence after updates and user logout/login.	SCRUM-9	Low	Sprint 1
SCRUM-25	Design and implement a registration form with fields for name, email, password	SCRUM-11	High	Sprint 1
SCRUM-26	Create a login page with a "Forgot Password" option.	SCRUM-11	High	Sprint 1
SCRUM-29	Create a table to store users with fields for ID, name, email, hashed password, and profile preferences.	SCRUM-11	High	Sprint 1
SCRUM-27	Set up user authentication using secure methods like JWT or OAuth	SCRUM-11	Medium	Sprint 1
SCRUM-28	Hash passwords before storing them in the database.	SCRUM-11	Medium	Sprint 1
SCRUM-30	Test registration and login with valid and invalid data inputs.	SCRUM-11	Low	Sprint 1
SCRUM-117	Design a secure "Remember Me" option on the login page.	SCRUM-71	Low	Sprint 1
SCRUM-118	Handle automatic login using saved credentials when the session is valid.	SCRUM-71	Low	Sprint 1
SCRUM-120	Store encrypted user credentials or session tokens securely on the client side.	SCRUM-71	Medium	Sprint 1