Vision

Course Planner is to be a website that will enable students to efficiently and easily schedule reminders for all of their coursework. Using information collected from your peers, through a comprehensive course element rating system, as well as custom settings collected from the user, a scheduling system will generate helpful reminders directing you when to begin studying or working on certain assignments. Course Planner is unique to other existing scheduling applications in that it automatically creates and schedules reminders using both user's preferences and information sourced from other students to allow easier study scheduling.

The problem of	scheduling study and homework sessions
affects	university students.
The impact of which is	spending unnecessary time on easier tasks and not enough time on harder tasks.
A successful solution would be	a platform that is highly adaptable to each student's individual study habits but also tabulates information from a variety of other sources to advise students on when to study.

For	students
who	have a variety of courses with important assignments and exams to study for.
Our System	is all software
that	tabulates ratings for specific course features based on peer ratings and personal preferences and uses them to assign dates for study reminders.
Unlike	existing reminder and scheduling applications that need to be manually set and do not reference the difficulty of specific tasks,
our product	crowdsources and collects difficulty ratings from fellow students and automatically uses this information to create reminders.

Use Cases

1. Create or login to user profile

1.1 Use facebook to login to Course Planner profile
1.2 Create a new course planner profile after first login in with Facebook

Primary Actor: University students, TAs, professor

Stakeholders and Interests: University students and professors/TAs are the main stakeholders of this use case. They should be able to create and login to their account.

Preconditions: User has a Facebook account.
Postconditions: User has a Course Planner profile and is logged in.

Main Success Scenario:

1.1 Use Facebook to login to Course Planner profile
1. Users enter their Facebook account and password
2. User is brought to their Course Planner user profile

1.2 Create a new course planner profile after first login with Facebook
1. Users enter their Facebook account and password
2. User is brought to an empty Course Planner profile page
3. User fills in profile information and current courses
4. A course schedule is generated and displayed on User's profile

Open Issues:
1. The source of a student's initial schedule is undecided. The courses could be added manually by the student or imported from the student's CWL.

2. Edit course schedule

2.1 Add course
2.2 Delete course
2.3 Edit course

Primary Actor: University students

Stakeholders and Interests: University students are the main stakeholder of this use case. They should be able to create a course schedule and edit it and their courses easily.

Preconditions: Student has a Course Planner profile and has logged in.

Postconditions: A course has been added, removed or modified in the student's calendar.

Main Success Scenario:

2.1 Add course
1. Students navigate to the add course page
2. Students fill in the name, time, location and other information, and confirm the change
3. Course appears in student's calendar in their profile

2.2 Delete course
1. Students select course they want to remove, and confirm the change
2. Course is no longer present in student's calendar in their profile

2.3 Edit course
1. Students select the course they want to edit
2. Students fill in the name, time, location and other information, and confirm the change
3. Updated course information now appears when viewing the Course page

Open Issues:
1. The source of the courses information is undecided. All courses information could either be added by students individually or be exported from UBC websites.

3.   Reminder system

3.1 Set a manual reminder
3.2 Set a rank-based reminder

Primary Actor: University students, TAs/Profs

Stakeholders and Interests:
University students are the main stakeholders of this use case. They could set their own schedule, and tell the system to produce reminders for upcoming assignments. University TAs/Profs are the other stakeholders of this use case. They could teach students more efficiently and connect more to students.

Preconditions: User has created an account with at least one course added.
Postconditions: Users can login their account and post on the discussion board.

Main Success Scenario:

3.1 Set a manual reminder
1. User selects the class for which they want a reminder set
2. User navigates to the reminders section of the course profile
3. User creates a new reminder of a manual type

       4.   User inputs reminder criteria (when, how, what task to be reminded of)

3.2 Set a rank-based reminder
      1.   User selects the class for which they want a reminder set
      2.   User navigates to the reminders section of the course profile
      3.   User creates a new reminder of a rank-based type
      4.   User selects the task of which to be reminded
      5.   Reminder criteria is pre-filled and user accepts new reminder

4. <u>Ranking and modifying course features</u>

4.1 Rank course feature
4.2 Add new course feature
4.3 Add information to existing feature

Primary Actor: University students, TA/Prof

Stakeholders and Interests: University students are the main stakeholder of this use case. They could rank the difficulty of quizzes, homeworks and labs so that other students could manage their time more efficiently.
University TAs/Profs are the other stakeholders of this use case. They could give their input on the difficulty of the quizzes, homework and labs to help students schedule their time. In this way, they making their teaching more efficient.

Preconditions: User has created an account with at least one course added and has logged in.
Postconditions: A course feature has been added or had its attributes modified and ranked.

Main Success Scenario:
4.1 Rank course feature
      1.   User navigates to course page
      2.   User selects the feature they wish to rank
      3.   User inputs their ranking in the features ranking section
4.2 Add course feature
      1.   User navigates to course page
      2.   User selects to add a feature under its proper category (textbook, homework etc.)
      3.   User inputs feature attributes (description, due date etc.)
      4.   User confirms and adds the completed feature to the course page
4.3 Add information to existing feature
      1.   User navigates to course page
      2.   User selects feature to add to
      3.   User inputs updated feature attribute

4. Attribute appears on course page with newest information displayed first and with previous information being displayed underneath.

Open Issues:
1. What scale to use for rankings (5 stars, sliding scale etc.)

Non-Functional Requirements

Safety Requirements

A verification system must be implemented to ensure that only students and TA/professors enrolled in the courses are able to generate assignments and rate them. One possible solution is to implement a password system for adding a course to a user profile. Also to further prevent improper posting of assignments we will add a feature to report an inaccurate assignment submission.

Security Requirements

Because we are asking for the facebook accounts of our users we must take steps to ensure that this information remains with us and cannot be used for any illicit purposes. This security need will be met by using Facebook's login system.

Software Quality Requirements

We want our product to be highly portable; we would like our customers to access our service both on mobile and on desktop devices. Our next priority is the reliability of our product; ideally we would find a way to ensure that every assignment and course only gets added to our database once and will not be lost once entered.

Changes and Rationale

Differentiation on our system between TA's/professors will no longer be supported instead all user profiles will be considered students. This is due to the difficulties experienced in entering user information into the database. Another contributing factor is that there is no method by which to verify a user is in fact either a TA or professor for the course automatically. This does not invalidate any of our user cases as the TA's and professors do not have any special privileges, functions or responsibilities.

We will no longer be focusing on making our website mobile friendly. We are removing this requirement to instead better direct our efforts into ensuring that the rest of the project

proceeds as scheduled. Our efforts will be better spent implementing our core features than on this requirement.