

# COURSE PLANNER REQUIREMENT DOCUMENT

## 1.Introduction

### 1.1 Purpose

Course Planner is to be a website that will enable students to efficiently and easily schedule their courses and others tasks they created in the same calendar. Our scheduling system will generate helpful reminders directing you when to begin studying or working on certain tasks that you input to your personalized calendar. Course Planner is unique to other existing scheduling applications in that it allows UBC students specifically, to easily enter their registered courses and have their schedule automatically create entries for them with course information directly sourced from UBC's website. Our site also allows users to create unique calendar entries and have them displayed alongside their course schedule.

### 1.2 Background

The problem of	keeping track of important tasks at the same time as a course schedule
affects	UBC students
The impact of which is	forgetting to complete certain tasks before they are supposed to be completed.
A successful solution would be	a platform that is highly adaptable to each student's individual study habits.

For	UBC students
who	have a variety of courses conflicting with their personal schedule.
Our System	is all software
that	allows UBC students to easily view their course schedules and plan the rest of their schedule around them.
Unlike	existing reminder and scheduling applications that don't integrate with UBC courses
our product	utilises UBC course information to automatically create calendar entries

	for user courses and allows users to have reminders sent to them reminding them of tasks that they create and add to their calendars.
--	---

### **1.3 Scope:**

In our app, the major users we are facing are UBC students who will be taking courses and have multiple tasks to complete in one day or in a week.

## **2.Overall Description**

### **2.1 Use Cases**

- **Login and Create User Profile**

2.1.1 Use facebook to login to Course Planner account

2.1.2 Create a new course planner profile after first login in with Facebook

**Primary Actor:** UBC students.

**Stakeholders and Interests:** UBC students 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 account with a unique user ID and information of their registered courses (which will be displayed on user's calendar) and is logged in.

**Main Success Scenario:**

2.1.1 Use Facebook to login to Course Planner profile

1. User enters their Facebook account and password
2. User is redirected to the main page if they already have an account. Otherwise they are brought to the account information entry page.

2.1.2 Create a new course planner account after first login with Facebook

1. Users enter their Facebook account and password
2. User is brought to a first time login page
3. User fills in profile information and current courses
4. User is redirected to the home page
5. Calendar entries are created for the user's courses, which can be viewed on the calendar

**Error Scenario:**

2.1.1 Use Facebook to login to Course Planner profile

1. User enters wrong Facebook account or password
2. Message of incorrect account or password shown on the Facebook login page
3. The user is kept on the index page until they have successfully logged in.

● **Edit Course/Task Calendar**

2.2.1 Add new course

2.2.2 Delete course/task

2.2.3 Edit course/task

2.2.4 Add new task

**Primary Actor:** UBC students.

**Stakeholders and Interests:** UBC 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 with their Facebook account.

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

**Main Success Scenario:**

2.2.1 Add new course

1. Students navigate to the add course page
2. Students enter the course department, course number, course section
3. New entries appear in student's calendar

2.2.2 Delete course/task

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

2.2.3 Edit course/task

1. Student selects the course they want to edit
2. Student fills in the info they wish to change (title, date, start time, end time, information, location, background color, text color)
3. Updated course information now appears when viewing the Course page

2.2.4 Add new task

1. Student clicks on the add task button
2. Student enters the information that they wish to have indicated on the entry (title, date, start time, end time, information, location, background color, text color)
3. The new task appears on the calendar, at the time and day indicated when created

**Error Scenario:**

2.2.1 Add new course

1. Students navigate to the add course page
2. Students enter some course department, course number, course section which is not inside our database (i.e. not an offered UBC course)
3. Message showing the courses which could not be found in our database is displayed to the users
4. The user is returned to the page until they re-enter proper course information or they change pages.

2.2.3 Edit course/task (and 2.2.4 Add new task )

1. Student selects the course they want to edit ( or clicks on add tasks button )
2. Student fills in the info they wish to change with the start time later than the end time
3. An alert showing that the user should enter the start time earlier than the end time is displayed; the website stays in the same page

● **Reminder system**

2.3.1 View reminders for the day

2.3.2 View e-mail reminders

**Primary Actor:** UBC students

**Stakeholders and Interests:**

UBC students are the main stakeholders of this use case. Depending on the user's calendar, the webpage will automatically compare the date of the user's tasks and will display the current day tasks. It is a simply calendar reminder that users can check their scheduled tasks.

**Preconditions:** User has created an account with at least one course added.

**Postconditions:** The user's current tasks are displayed on the main page.

**Main Success Scenario:**

2.3.1 View reminders for the day

1. User navigates to the main page of the website
2. The user's courses and other tasks for the current day are displayed on the page

2.3.2 View e-mail reminders

1. User adds their e-mail and agrees to receive reminders by e-mail on their course information page
2. Any time after 4am daily, the user accesses their e-mail and views their tasks for the day

## **2.2 Non-Functional Requirements**

- **Safety Requirements**

A verification system must be implemented to ensure that only users who have logged in can be enrolled in courses are able to edit their courses. One possible solution is to implement a password system for adding a course to a user profile, and in this website we specifically use Facebook to verify users.

- **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. In order to get the permission of using facebook account, we also provided a privacy policy URL and Terms of Service URL which will be used to make sure users' information will remain with us.

- **Software Quality Requirements**

We want our product to be highly portable; we would like our users to be able to access our product from anywhere they have internet access and a browser. Therefore we have chosen our platform to be a website to allow cross-platform access. 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.

## **2.3 Operating Environment**

- **UI:** The user interface is built using HTML, CSS, PHP and Javascript libraries (Jquery, Ajax, MDN API).
- **DATABASE:** Our database is supplied by the Amazon relational database servers, and it is written using MySQL and we interact with it through PHP.

## **2.4 Design and Implementation Constraints**

For our website, during the design period we mostly focused on a website design which does not display very elegantly on mobile devices. So when users open this website with their phones, the page view can not be perfectly fit on their screens, however the functions of the website operate normally.

## **Changes and Rationale**

1. Our system no longer differentiates between students and administration. Instead all users will be considered as students. This is due to the difficulties experienced in entering user information into the database and providing features that would actually be unique between different types of users. 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 use cases as the TA's and professors do not have any special privileges, functions or responsibilities in our final design.
2. 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.
3. We will no longer be including the ability to create, rank and edit course features. This is because, based on the stage we are currently in with development, we believe that we should focus on improving the current implemented features to work at full and proper capacity by the demo day. Since administrator accounts are also no longer under our consideration, the ranking system's removal will not affect that feature. Additionally we will be simplifying our reminder system to send reminders for all tasks and courses in a user's calendar on a day-by-day basis and not only for specific course features such as exams and assignments.