Software Requirements Specification

for

Academic Planner Program

Version 1.0 approved

Mike Dai 400188179 daiy29
Camisha Mortensen 400215847 mortensc
Yifan Jiang 400151015 jiany25

McMaster University 4ZP6 Group 13

2022-10-31

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
G13	2022-10-31	Requirement Spec	1.0
Yifan Jiang	2022-11-29	Table of Contents	1.1

1. Introduction

1.1 Purpose

An academic planner that allows students to keep track of tests and assignments in their courses as well as plan study and lecture time around their schedules.

1.2 Document Conventions

- Mandatory core feature
- Lower priority or expandable feature, time permitting
- to be revised/re-evaluated at date specified

1.3 Intended Audience and Reading Suggestions

This document is primarily intended for the developers of the project in order to describe in detail what the project will entail (and what it won't), why it is being created, what will be used to create it, the features it will offer and who it is being created for. The requirements in this document are measurable and specific and will be referred back to upon completion of the project in order to gauge the success of the team.

This document also gives professors, teaching assistants and potential judges of the development team's work an in-depth view of what the team intends to create and their motivations. A quick overview of the technical stack used is available for easy reference.

Although not the primary audience, users of the application may also view this document in order to better understand the application and what unique features it might offer them compared to similar options on the market

For developers it is suggested to focus most on section 4.0 of the document and pay close attention to the details of each requirement when implementing. Section 5.2 will be extremely important in protecting user data and must be considered at every step. Developers should also be aware that section 3.3 may be more flexible than others, should the team discover while working that certain technologies, tools or frameworks are not suiting their needs as originally expected, other options may be considered at that time.

For other audiences it is suggested to get an understanding of the exact niche this application fills by learning about it's purpose and scope in section 1.1 and 1.4, how it stands apart from the competition in section 2.1 and the features it offers in less technical terms in section 2.2. This should help any user know what to expect from the product and determine if it is something that can be helpful to them.

1.4 Product Scope

This application will allow students to plan both their upcoming semesters and overall degree. It will offer features for users to input details about each of their courses, assignments, tests and deadlines and calculate grades. Users will be able to view their courses and goals in an overall dashboard view as well as a calendar view by month, week, day, etc. and check them off a To Do List, tracking their grades and progress as they go. Additionally, students will be able to generate a weekly study plan based off of their schedule and assignment priority/difficulty. The strategy for this product is to combine and expand upon the best features of similar Calendar, Time Management and Academic Planning applications in once place and for students to be able to use the application throughout their entire degree. The app will be created using the MongoDB, Express, React, Node (MERN), stack.

1.5 References

Web Content Accessibility Guidelines
https://www.w3.org/WAI/standards-guidelines/wcag/

2. Overall Description

2.1 Product Perspective

The application is a new and self-contained academic planner that helps students to plan their upcoming semester and manage their time. Similar applications include, Google Calendar, Outlook Calendar, Google Keep and TickTick. There are many other generalized planners, To-Do list organizers and habit trackers available online as well as internal academic planners such McMaster's MyTimeTable or Avenue to Learn. However, there are few, free solutions available for students to manually track their progress in classes and in their degrees and organize their study time around the rest of their life. This application aims to provide a solution to this problem.

2.2 Product Functions

- Users will be able to add the details for each of their current courses including, lecture schedule, assignments, tests, and syllabus information.
- Added assignments, labs, and tests will include weights, deadlines, and relevant links
- The application will generate corresponding tasks based on course information such as a lecture review task, a reading task, or a note-organizing task
- Students can return to the course homepages they create throughout the semester to record assessment marks, monitor in-progress grades and calculate the minimum mark requirements for outstanding assessments depending on their goals
- Course details and study tasks will be viewable on the Calendar
- The same tasks will also be viewable on the To-Do list in more detail where users can check them off
- Each course will be clearly highlighted in a unique color across the application
- Users will be able to add non-academic tasks to the calendar and To-Do list

- Task on the calendar and To-Do list will be taggable and filterable
- The application will generate a suggested weekly study plan in order to help students decide what they should study, when and for how long based on their schedule and assignment priority/difficulty. This can be automatically added to the calendar and To-Do list.
- Users will also be able to access degree planning section from the Dashboard where they can track and view the progress of their entire degree

2.3 User Classes and Characteristics

- University students are the primary audience of the application. They are aged 18-25 and come from a diverse background, they may be international students from anywhere in the world, they may be studying any major, they may be any gender, English may not be their first language however they are very tech savvy and familiar with similar applications. They may be juggling multiple major responsibilities such as multiple classes, work, extracurriculars and social events and use the application up to multiple times daily in order to keep track of their lives.
- Non university students are also an important user class. They are younger than the primary audience at around 12-17 (middle and high school aged). This group is equally diverse and tech savvy as the university students however they may not have quite as many responsibilities in life to keep track of and the degree planning section is likely to be of no use to them for now. However the application can still be a powerful tool for those in this group who want to learn to plan and track their academics and build good time management skills for the future.
- Parents of students may also be users of the product when helping their child organize their time. This is the oldest user class at 30+ and likely the least tech savvy however they are still assumably familiar with similar applications and should have no issues guiding someone through the application

2.4 Operating Environment

The application will be available on all modern web browsers (chrome, firefox) and will be optimized for both a desktop and mobile view, therefore there are minimal hardware limitations for the project. As a standard web application, the HTTP protocol will be used for communication and sensitive user data will be encrypted.

2.5 Design and Implementation Constraints

- The application will not interact with any other application to automatically fill out any of the course information, users will be required to do this themselves. This keeps the product accessible to students from any institution.
- The application will only be available in English.
- The application will be developed using the most up to date version of all technologies in the MERN stack.

2.6 User Documentation

- Homepage: Users who are not signed into any account will land on a homepage which gives an overview of the application including an about section and highlights of key features
- New user pop-up guides: As a new user navigates the application for the first time they will be prompted with small dismissible pop-ups that guide them through the flow of the application and all of the major features in the expected order

2.7 Assumptions and Dependencies

- We assume that the user is enrolled in classes where they have access to all of the details of the course and will be able to provide these to the application
- We assume that the user speaks English (ESL will be sufficient)
- We assume that the user has an email address
- We assume that the user is familiar with how to use a computer and navigate a web browser

3. External Interface Requirements

3.1 User Interfaces

All major functionality of the application will require the user to interact with the interface. The application will be designed responsively to ensure a pleasant viewing experience on both mobile and desktop browsers. Users can always expect a standard navigation bar across the top of the website with links to the Homepage, Dashboard, Calendar, To-Do List, and an Account button which upon being clicked reveals a Settings Page, Account options and a Sign In/Out option. The GUI will take accessibility into account ensuring all pages are in line with Web Content Accessibility Guidelines.

3.2 Hardware Interfaces

This product should be able to run on as many types of devices as possible, and thus hardware should not be a limitation. There is very minimal hardware usage for this program outside of the basic components of a computer or handheld device. At the bare minimum, the supported device types will be mobile phones, tablets, as well as most computers.

3.3 Software Interfaces

This project will run as a web application using the MERN stack. Most of the application should be written in the most up-to-date version of Javascript. [The development team reserves the right to re-evaluate the technologies, libraries and frameworks used to create this application during implementation in January 2023 should the following not suit their needs.]

- MongoDB: This is a cross-platform document-oriented NoSQL database program. It uses JSON-like objects with optional schemas. This will be used to store and retrieve information needed for the application; an example would be course information.
- Express.js: Express.js is to be used alongside Node.js as a framework. As we are using HTTP, it will be used as middleware, and assist in routing and specifying HTTP verbs and URL patterns for the web application.
- Node.js: Node.js is used for server development and can be run on many types of operating systems, such as Windows, Linux, Unix, Mac OS, making this ideal for allowing our application to be flexible and usable on most devices. It allows for dynamic web content to be displayed in the browser running on Javascript. Some of the key features are that it's asynchronous, has no buffering, and is scalable.
- React.js: React will be used for designing the user interface for the web application. It is dynamic, simple to use, and scalable to various devices.

We will host this application on Amazon Web Services.

3.4 Communications Interfaces

This application will use the HTTP protocol for communication, as it will be a web application. This application will feature functions such as forms, email authentication, electronic forms, and web navigation which will be handled by Node.js through HTTP request methods. Sensitive information such as passwords, will be encrypted and stored in our database using a verifiable standard such as AES.

4. System Features

4.1 Allow user to sign up

Allow user to sign up.

4.1.1 Description and Priority

When a user opens the web application, they should be able to sign up for an account. This is of high priority.

4.1.2 Stimulus/Response Sequences

User access the web application through their web browser, the deployed front-end website sends website to user and show the the homepage also known as the log-in page , then the user click the "Sign Up" button, then the application redirects the user to the sign-up page to collect information,

then the user can input their information, then the user clicks on confirmation, then the application adds the new user account with collected information to the database and redirects the user back to the login page.

4.1.3 Functional Requirements

SIGN-1: When a user clicks on the “Sign Up” button on the homepage also know as the log-in page, the system shall redirect the user to the sign up page that has fields for collecting the account information.

SIGN-2: The system shall require the user to input their account information of name, email address, password, re-enter password.

SIGN-3: When a user clicks on the “Sign Up” button below the fields of account information on the sign up page, the system shall add to the database a new account entry with unique id, name, email address, password.

SIGN-4: When the database completes adding the new account entry, the sign-up is completed and the system shall redirect the user back to the homepage also known as the log-in page.

4.2 Allow user to login

Allow user to login.

4.2.1 Description and Priority

When a user opens the web application, they should be able to log in to their account. This is of high priority.

4.2.2 Stimulus/Response Sequences

A user access the web application through their web browser, the deployed front-end website sends website to user and show the the homepage also known as the log-in page , then the user input their email address and password and click on the “Log In” button, then the database retrieve the user’s account information and send to the front-end website, then the front-end website displays the dashboard page with the user’s planner information.

4.2.3 Functional Requirements

LOGIN-1: When a user is intended to log in to their account, the system shall require them to input their email address and password.

LOGIN-2: When a user click on the “Log In” button with input email address and password, the system shall verify the account information.

LOGIN-3: If input email address has not been recorded with an account entry in the database, the system shall prompts that the account does not exit and inform the user to sign up.

LOGIN-3: If input email address has not been recorded in the database and the email address and password pair does not match any account entry in the database, the system shall prompts that the input password is incorrect and ask the user to change the input and log in again.

LOGIN-4: If input the email address and password pair matches an account entry in the database, the database shall retrieve the account entry and sends the information to the front-end, then the deployed front-end website sends website to user and show the dashboard page with the user’s planner information.

4.3 Provide a navigation bar

Provide a navigation bar.

4.3.1 Description and Priority

The system shall provide a navigation bar that is always shown on the top of any webpage in the front-end website, which includes buttons of “Dashboard”, “Timetable”, “Calendar”, “To-do List”, avatar. This is of high priority.

4.3.2 Stimulus/Response Sequences

A user is at a webpage of the front-end website and intended to navigate to another webpage or change the view of their planner, then the user clicks on a button and the system redirects the user to the corresponding webpage linked to the button clicked.

4.3.3 Functional Requirements

NAV-1: When a user clicks on the “Dashboard” button at the left of the navigation bar, the system shall redirect the user to the dashboard page.

NAV-2: When a user clicks on the “Timetable” button at the centre region of the navigation bar, the system shall redirect the user to the timetable page and show the planner in a weekly timetable view.

NAV-3: When a user clicks on the “Calendar” button at the center region of the navigation bar, the system shall redirect the user to the calendar page and show the planner in a monthly calendar view.

NAV-4: When a user clicks on the avatar button at the right of the navigation bar, the system shall pop up a pull-down menu below the avatar, which includes buttons of “Account”, “Settings”, “Log Out”

NAV-5: When a user clicks on the “Account” button on the avatar’s pull-down menu, the system shall redirect the user to the account information page.

NAV-6: When a user clicks on the “Settings” button on the avatar’s pull-down menu, the system shall redirect the user to the general system settings page.

NAV-7: When a user clicks on the “Log Out” button on the avatar’s pull-down menu, the system shall out the user out and redirect the user to the homepage also know as the log-in page.

4.4 Provide a dashboard page

Provide a dashboard page.

4.4.1 Description and Priority

The system shall provide a dashboard page, which contains blocks of program information, annual plan, ongoing term, current weekly timetable, current monthly calendar and latest to-do list, to allow user to entirely monitor their planner; the display scope of the content in the planner is constrained by the webpage space. This is of high priority.

4.4.2 Stimulus/Response Sequences

User has logged in their account, and then the system shows the dashboard page.

User is at a webpage of the front-end website and intended to navigate to the dashboard page, and then they click on Dashboard on the navigation bar, and the system redirects them to the dashboard page.

User is at the dashboard page and intended to go to another page or a view of the planner, and they click on a block on the dashboard page, and the system redirects them to the page linked to the block.

4.4.3 Functional Requirements

DASH-1: The degree information block shall display a degree progress bar, degree information and current GPA based on the user's data.

DASH-2: When a user clicks on the degree information block, the system shall redirect the user to the degree information page.

DASH-3: The system shall allow the user to input course history with grades on the degree information page.

DASH-4: The system shall calculate the current GPA based on the history GPA and in-progress marks, and display the mark on the degree information page.

DASH-5: The annual plan block shall display the annual timeline of ongoing term and upcoming term with registered course, and non-academic plan base on the user's data.

DASH-6: When a user clicks on the annual plan block, the system shall redirect the user to the annual plan page.

DASH-7: The system shall allow user to input annual plan in a timeline on the annual plan page.

DASH-8: The ongoing term block shall display the term information, a list of course registration and in-progress marks for each course.

DASH-9: When a user clicks on the ongoing term block, the system shall redirect the user to the ongoing term page with the list of course registration.

DASH-10: The current weekly timetable block shall display the current weekly timetable information with periodic schedule of each course and ongoing tasks of the current week.

DASH-11: When a user clicks on the current weekly timetable block, the system shall redirect the user to the planner in the weekly timetable view.

DASH-12: The current monthly calendar block shall display the current monthly calendar information with all ongoing tasks distributed in the current month.

DASH-13: When a user clicks on the current monthly calendar block, the system shall redirect the user to the planner in monthly timetable view.

DASH-14: The latest to-do list block shall display the to-do list with the latest ongoing tasks.

DASH-15: When a user clicks on the latest to-do list block, the system shall redirect the user to the planner to the to-do list page.

4.5 View in to-do list view

View in to-do list view.

4.5.1 Description and Priority

The system shall provide a to-do list page and allow the user to view all tasks in a to-do list view, with the information of scheduled date, due time, course name, task title, and highlight font and weight if it is an assessment task. This is of high priority.

4.5.2 Stimulus/Response Sequences

User is viewing their planner and intended to view their tasks in a to-do list view, and they click on the “To-do List” button on the navigation bar and the system shall display the planner in a to-do list view with all tasks.

4.5.3 Functional Requirements

TODO-1: The to-do list view of planner shall list all tasks in chronological order by default.

TODO-2: The to-do list view of planner shall list all sessional tasks with the information of scheduled time, course name, task title.

TODO-3: The to-do list view of planner shall list all assessment tasks with the information of due time, course name, task title, weight, in highlighted font.

TODO-4: The to-do list view of the planner shall have a check box in front of all tasks.

TODO-5: When user click on a task in the to-do list, the system shall pop up a bubble from the task with all information of the task following by the button of “complete”, “abandon”, “view in timetable”, “view in calendar”, “modify”, “delete”.

4.6 View in a weekly timetable view

View in a weekly timetable view.

4.6.1 Description and Priority

The system shall provide a weekly timetable page and allow the user to view all tasks in a weekly timetable view, where each task is a block inserted in a time slot with the information of duration, due time, course name, task title, and highlight font and weight if it is an assessment task. This is of high priority.

4.6.2 Stimulus/Response Sequences

User is viewing their planner and intended to view their tasks in a weekly timetable view, and they click on the “Timetable” button on the navigation bar and the system shall display the planner in a weekly timetable view with all tasks.

4.6.3 Functional Requirements

TTABLE-1: In the weekly timetable of planner, each task is a block inserted in a time slot under a day in chronological order.

TTABLE-2: In the weekly timetable of planner, each sessional task is a block inserted in a time slot with information of course name, task title, location, duration.

TTABLE-3: In the weekly timetable of planner, each assessment task that requires submission is a block with information of course name, task title, due time, weight.

TTABLE-4: In the weekly timetable of planner, each assessment task of quiz, exam or test is a block with information of course name, task title, schedule time, duration, weight.

TTABLE-5: When user click on a task block in the weekly timetable of planner, the system shall pop-up a bubble from the block with all information of the task following by the button of “complete”, “abandon”, “view in to-do-list”, “view in timetable”, “modify”, “delete”.

4.7 View in a monthly calendar view

View in a monthly calendar view.

4.7.1 Description and Priority

The system shall provide a weekly timetable page and allow the user to view all tasks in a weekly timetable view, where each task is a block inserted in a time slot with the information of scheduled time, due time, course name, task title, and highlight font and weight if it is an assessment task. This is of high priority.

4.7.2 Stimulus/Response Sequences

User is viewing their planner and intended to view their tasks in a monthly calendar view, and they click on the “Calendar” button on the navigation bar and the system shall display the planner in a monthly calendar view with all tasks.

4.7.3 Functional Requirements

CAL-1: In the monthly calendar of the planner, each task is a block inserted in a time slot under a day in chronological order.

CAL-2: In the monthly calendar of the planner, each sessional task is a block inserted in a time slot with information of course name, task title.

CAL-3: In the monthly calendar of the planner, each assessment task that requires submission is a block with information of course name, task title, due time, weight.

CAL-4: In the monthly calendar of the planner, each assessment task of quiz, exam or test is a block with information of course name, task title, schedule time, duration, weight.

CAL-5: When user click on a task block in the monthly calendar of planner, the system shall pop-up a bubble from the block with all information of the task following by the button of “complete”, “abandon”, “view in to-do-list”, “view in timetable”, “modify”, “delete”.

4.8 Allow user to input course registration

Allow user to input course registration.

4.8.1 Description and Priority

The system shall allow user to input course registration information on the course list page.

4.8.2 Stimulus/Response Sequences

User is starting their term and intended to input course course, and the user click on the course list page and the system redirects the user to the course list page, and the use add a course with course name and assign a unique color scheme specific to the course.

4.8.3 Functional Requirements

COURSE-1: The system shall allow user to input course entry with course name in the course list.

COURSE-2: The system shall require user to assign a unique color scheme to the course.

COURSE-3: The system shall allow user to input periodic schedules of course session including lectures, tutorials, labs, office hours in the course list page.

4.9 Automatically generate tasks from periodic schedule

Automatically generate tasks from periodic schedule.

4.9.1 Description and Priority

The system shall automatically generate sessions from periodic schedule into tasks, such that the user does not have to repeatly add similar tasks manually.

4.9.2 Stimulus/Response Sequences

A user is intended to have all periodical sessions, including weekly sessions or every-other-week sessions, into the system, and the system has automatically generated these tasks, and the user and view them in the to-do list view, the timetable view and the calendar view.

4.9.3 Functional Requirements

GENETASK-1: The system shall automatically add session of courses as entries to the database, and display all entries as tasks in the system.

4.10 Allow user to add, modify and delete tasks

Allow user to add, modify and delete tasks.

4.10.1 Description and Priority

The system shall allow user to add, modify and delete tasks to the system.

4.10.2 Stimulus/Response Sequences

User intended to add a new task into the system, and the user click on the "+" button on the planner, and the system pop up a space with fields to collect information, and the user click save after inputing all necessary information, and the system add a new task entry to the database and display the task in the system's webpage.

4.10.3 Functional Requirements

TASK-1: When the user click on the "+" button on the planner, and the system pop up a space with fields to collect information including course name, task title, scheduled time, due time, weight, priorty, linked tasks.

TASK-2: All ongoning tasks under a course shall display in the assigned color scheme of that course.

TASK-3: All assessment tasks including assignments are in highlited font.

TASK-4: The system shall allows user to modify, delete, check off, mark as completed or abandon tasks.

TASK-5: All complete tasks are display in gray color.

TASK-6: All abandoned tasks are display in gray color with a delete line.

4.11 Provide an overall Course List page

4.11.1 Description and Priority

The application will provide an overall course list page that allows the user to quickly see all of the courses they have added to the application and a small snapshot of their current grade in each course. Users will also be able to add new courses from this page. This is of high priority and critical to the functioning of the application.

4.11.2 Stimulus/Response Sequences

User clicks the “Courses” button in the top navigation bar.

4.11.3 Functional Requirements

COURSELIST-1: The program shall always display a button titled “Add Course” that allows the user to create a new course and store it in the database

COURSELIST-2: The program shall display a list of all added courses showing the name and course code with a small amount of details on the users current grades.

COURSELIST-3: Clicking on the course takes the user to an inner page with the full details they have entered for that course

4.12 Provide individual course pages

4.12.1 Description and Priority

The application will allow the user to create individual course pages where they can input details about the course such as faculty contacts, syllabus information, weekly schedules, assignments, tests, grades and view tasks. This is a high priority feature.

4.12.2 Stimulus/Response Sequences

User clicks on the “Courses” button in the top navigation bar and then selects the “Add Course” button.

4.12.3 Functional Requirements

COURSEPAGE-1: The user shall be able to input professor and teaching assistant contact details such as email and phone number and view it displayed with hyperlinks on the page

COURSEPAGE-2: The user shall be able to manually upload their weekly timetable. This will be added as tasks to both the overall calendar and overall To-Do list pages.

COURSEPAGE-3: The user will see a weekly calendar view of just this course's tasks on this page.

COURSEPAGE-4: The user will see a todo list view of just this course's tasks on this page.

COURSEPAGE-5: Upon completion of an assignment, users can enter the grades that they have earned. A table of all grades earned for the course will be visible along with a summary of their overall mark.

COURSEPAGE-6: User shall be able to assign a unique color to each course which will be used throughout the application

COURSEPAGE-7: Users shall be able to re-visit and re-edit inputted information anytime

4.13 Record course grades and subordinate assessments marks

4.13.1 Description and Priority

The system shall allow users to input their overall grade for a specific course, as well as on individual assessments.

4.13.2 Stimulus/Response Sequences

User has logged in to their account, has navigated to the course and has chosen to input their grade details.

4.13.3 Functional Requirements

RECORDGRADE-1 The program shall prompt the user for their overall grade in the course

RECORDGRADE-2 The program shall prompt the user for a breakdown of their grade in a course

RECORDGRADE-3: The program shall display a user's overall grade for the chosen course

RECORDGRADE-4: The program shall display the breakdown for a user's overall grade for a chosen course

RECORDGRADE-5: If incorrect number formatting is used, the input will be rejected and a prompt to re-enter the grade will be given.

RECORDGRADE-6: All inputted information is saved and stored in the server database

4.14 Calculate minimum required grade

Calculate minimum required marks on outstanding assessment based on the user's expectation of the final grade

4.14.1 Description and Priority

The system shall calculate the grades required on remaining tasks with grading schemes such that the user will achieve their overall target grade for that course.

4.14.2 Stimulus/Response Sequences

User has logged in to their account, has set up the required information about a specific course, and has imputed their desired GPA

4.14.3 Functional Requirements

CALCULATEMIN-1 The program shall display an appropriate error if the desired grade is not possible

CALCULATEMIN-2 The program shall automatically update the required average when new grades are inputted.

CALCULATEMIN-3 The program shall round up required grades to provide room for error or ambiguity in grade rounding.

CALCULATEMIN-4 The program shall assume a mark of "0" on previous assignments should information be unavailable

CALCULATEMIN-5 The program shall only accept grades on a numerical scale between 0-100.

4.15 Course Detail Input

Allow user to input course detail information

4.15.1 Description and Priority

The program shall allow users to input information about a course they would like to have to be saved and preserved for convenient access.

4.15.2 Stimulus/Response Sequences

User has logged in to their account, and clicked on and filled out a form to input various types of course information.

4.15.3 Functional Requirements

COURSEDETAIL-1 The program shall allow users to detail the instructor's e-mail

COURSEDETAIL-2 The program shall allow users to detail the mark breakdown of the course

COURSEDETAIL-3 The program shall allow users to detail the schedule of the course

COURSEDETAIL-4 The program shall attempt to automatically fill out information about the instructor's contact information, and mark breakdown by scraping the course outline if it is provided.

COURSEDETAIL-5 The information for the specific course is saved in the server database

4.16 Contact Faculty

Allow users to contact the faculty of an enrolled course

4.16.1 Description and Priority

Allow users to contact the faculty of an inputted course through a provided means of communication

4.16.2 Stimulus/Response Sequences

User has logged in to their account, and has uploaded information about the course instructor's contact information.

4.16.3 Functional Requirements

CONTACT-1 If a course instructor's email is available within the course information for a course, it will be treated as a hyperlink.

CONTACT-2 If a course instructor's phone number is available within the course information for a course, it will be treated as a callable number.

CONTACT-3 If an email is provided, and the link is clicked, the program shall use the device's default email system to compose and display a blank email to the instructor.

CONTACT-4 If a phone number is provided, and the link is clicked, and the device is a mobile phone, the program shall use the device's default calling system to dial the associated number.

4.17 Allow user to upload course outline file

4.17.1 Description and Priority

The program shall allow the user to take their existing course outline provided to them by the professor and store it in the application under the corresponding course page for easy reference. This is of low priority

4.17.2 Stimulus/Response Sequences

When adding a new course user clicks on the "Upload Documents" button and is prompted to upload a file.

4.17.3 Functional Requirements

OUTLINE-1: When the user clicks on "Upload Documents" they are prompted to upload a local file.

OUTLINE-2: The system will accept files of type PDF, docx, png and jpeg otherwise the user will receive an error message asking them to choose an appropriate file type.

OUTLINE-3: The file is stored in the database along with which course and user it is associated with.

OUTLINE-4: The file shall be displayed on the front end on the individual course page.

4.18 Generate weekly study plan

4.18.1 Description and Priority

The program shall attempt to generate a weekly study plan based off of available time, and an algorithm that weighs the urgency and importance of a task.

4.18.2 Stimulus/Response Sequences

User is logged in and has navigated to their dashboard, and has confirmed that a generated weekly schedule is accepted.

4.18.3 Functional Requirements

GENERATETT-1: If accepted, the generated time table will be displayed on the weekly timetable block in the dashboard, as well as higher level views.

GENERATETT-2: The generated time table should take all registered courses into account.

GENERATETT-3: If new courses are added after a schedule has been generated, the old one will be displayed until a new one is generated.

GENERATETT-4: The algorithm should weigh the hours allocated based on an algorithm that calculates priority, as well as user preference.

GENERATETT-5: If there are no remaining graded tasks, an empty timetable will be generated.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The application should run on device browsers with a stable internet connection. This includes phones, tablets, and personal computers. The application will be able to request and modify information from the database in real time, and requests such as email verification should be sent as soon as possible. This makes the application as usable as possible, as delays in data transmission can cause confusion and frustration to the user.

5.2 Safety Requirements

This application does not need, and thus should not be able to access any personal information from the client.

5.3 Security Requirements

Sensitive user information such as passwords, should not be stored in the database. This information should be encrypted, and only transmitted through a verifiably safe system. This ensures user safety in the case of a database breach. Additionally, two factor authentication may be implemented to ensure only the rightful owner of an account may access the service.

5.4 Software Quality Attributes

The most important quality characteristics of this application are maintainability, usability, and portability.

- Maintainability is important, to allow for more features to be added in the future, as well as modify existing ones.
- Usability is one of the most important characteristics, as an application with poor usability does not entice users to use a product in the long term. For an application such as an agenda, poor usability will deter users from even using the product. The product should be simple to use, while offering a rich diversity of features, and provide a clean attractive user interface.
- This product needs to be portable. The product should work in the same way with no issues on multiple platforms, giving users more freedom to use the application whenever it is needed.

5.5 Business Rules

All users will have the same permissions in using this product. All required functions will be offered to the user, including basic personal administrative functions; and thus there are no situations where special permissions may be required.

6. Other Requirements

Database contents will be backed up in case of failure, data loss, or breach of data integrity.