# Product Backlog
## Team 21

**Team Members**: *Joe Marshall, Gus Vroustouris, Delun Shi, Pano Kostouros, Abhi Gupta*
**Coordinator**: *Adam Johnston*

## Problem Statement

Students have a lot on their plate and need help managing all their classes, schoolwork, events, and personal life. In order to help organize and plan their schedules, many use calendar apps such as Google Calendar. While the programs these students are currently using are very useful for a variety of applications, they tend to be built for general purpose use. Because of this, these programs can be tedious to use and have somewhat of a learning curve, leading to some students choosing not to use software to keep track of their schedules and miss out on the benefits of doing so. A streamlined program would help students create and keep track of schedules faster and easier, and would encourage more students to use it.

## Background Information

There are many different calendar applications out there for people. Everyone has one and they are given increasing importance in how to live a balanced and productive life. Human memory of course is not perfect and storing reminders someplace else is extremely helpful. The problem with the current calendar applications is that they are not actively attempting to help people schedule their lives. They act merely as an empty canvas for you to put down your tasks. Our calendar will have an algorithm to help optimize your time and will schedule events for you if you forget.

## Environment

Our environment will be rather simple. We will be programming in Java using whichever IDE each developer wants to use. The version control we will be using is GitHub to record our changes. MongoDB will be the database we will use to store the information about the user. Our frontend user interface will be done with JavaFx.

## Functional Requirements

| ID | Functional Requirement | Hours | Status |
|----|------------------------|-------|--------|
| 1 | As a user, I want to quickly set the date, time and duration of an event. | 8 | Sprint 1 |
| 2 | As a user, I want to be able to see set events in a given week | 4 | Sprint 1 |
| 3 | As a user, I want to be able to see set events during a month | 4 | Sprint 1 |
| 4 | As a user I want to store classes for a semester. | 4 | Sprint 1 |
| 5 | As a user, I want to able to click on classes on the calendar and view additional information about them. | 4 | Sprint 1 |
| 6 | As a user, I want to see a summary of a week's | 4 | Sprint 2 |

| | | | |
|---|---|---|---|
| | assignments, commitments and free time and be able to compare this to other weeks. | | |
| 7 | As a user, I want to use a time sorting feature to automatically plan out my free time for completing assignments efficiently. | 6 | Sprint 2 |
| 8 | As a user, I want to be able to choose the type of event I'm adding: choices such as Homework, Class, Midterm, Final. | 4 | Sprint 1 |
| 9 | As a user, I want to be able to edit information for events I created previously. | 4 | Sprint 1 |
| 10 | As a user, I want to be able to delete events I created previously. | 4 | Sprint 1 |
| 11 | As a user, I want to specify different priority levels for different events. | 5 | Sprint 1 |
| 12. | As a user, I want to specify whether an event repeats, and how often it should repeat. | 5 | Sprint 1 |
| 13. | As a developer, I would like to edit the events details in the database. | 4 | Sprint 1 |
| 14. | As a user, I would like to assign a color to an event upon creation. | 4 | Sprint 2 |
| 15 | As a user, I want to add in due dates for homework | 4 | Sprint 1 |
| 16 | As a user, I want to be given a recommendation on which homework to work on based on Priority Algorithm (due date, priority, and time availability) | 4 | Sprint 2 |
| 17 | As a user, I want to mark a homework as completed | 3 | Sprint 1 |
| 18 | As a developer, I want to create a priority algorithm to help users schedule their time. | 7 | Sprint 2 |
| 20 | As a user, I would like to be notified when an event is about to happen. | 6 | Sprint 1 |
| 21 | As a user, I would like to change the time I get notifications for events. | 3 | Sprint 1 |
| 22 | As a user, I would like to know when my allotted time for an event is up. | 3 | Sprint 1 |
| 23 | As a user, I would like to disable notifications for an event | 4 | Sprint 1 |

| 24 | As a user, I would like to disable notifications for all events in a series. | 5 | Sprint 1 |
|----|----|----|----|
| 25 | As a user, I would like to opt-in to constant reminders to remind me what I should be doing. | 5 | Sprint 2 |
| 26 | As a developer, I would like to store the users notification settings in a database. | 4 | Sprint 2 |
| 27 | As a developer, I would like the application to have different preset event creations that have different preset defaults. I.e. class, homework, studying, exams | 8 | Sprint 2 |
| 28 | As a user, I would like to schedule my class for the semester with a recurring event by only going through one creation cycle. | 5 | Sprint 2 |
| 29 | As a user, I would like to have a preset event for homework. | 4 | Sprint 2 |
| 30 | As a user, I would like to have a preset event for meetings. | 4 | Sprint 2 |
| 31 | As a user, I would like to have a preset event for finals/exams. | 4 | Sprint 2 |
| 32 | As a user, I would like to store a directory of contacts/important people in the app. | 4 | Sprint 2 |
| 33 | As a user, I would like to have random reminders scheduled every few weeks. Reminding me to call important people from a directory of important people. | 4 | Sprint 2 |
| 34 | As a developer, I would like to store the user's classes and events in a database. | 4 | Sprint 2 |
| 35 | As a user, I would like to choose the color scheme for my calendar. | 3 | Sprint 2 |
| 36 | As a user, I would like to see all my events laid out in a per month calendar grid format | 6 | Sprint 2 |
| 37 | As a user, I should be able to look ahead at the rest of the semester's events. | 4 | Sprint 1 |
| 38 | As a user, I should be able to see all my events displayed in a weekly calendar grid format. | 3 | Sprint 2 |
| 39 | As a user, I would like to click on a calendar day and have that day's events displayed on screen. | 4 | Sprint 1 |

| 40 | As a user, I would like to have a deleted event cease to appear on the any form of calendar display. | 4 | Sprint 1 |
|----|---|---|---|
| 41 | As a user, I would like to have events on the calendar display the assigned event color. | 4 | Sprint 2 |
| 42 | As a developer, I would like to have the calendar system efficiently query from the database and build said displays. | 5 | Sprint 1 |
| 43 | As a developer, I would like to have the database hosted on Heroku. | 4 | Sprint 1 |
| 44 | As a user, I want to define what a semester is. Or how long events should be repeated in a period of time. | 4 | Sprint 1 |

## Non-Functional Requirements

1. User interface should scale well on desktop.
2. User interface should be as user friendly as possible.
3. User interface should be very responsive.
4. Should provide valuable feedback to user

## Use Cases

**Case: Adding Events to Calendar**

| Action | System Response |
|---|---|
| 1. Click 'Add Event' Button | 2. 'Add Event' form appears |
| 2. Fill out fields for event info: type of event, due date, class, notifications, priority, etc. | |
| 3. Click 'Confirm Event' | 4. Save event to database and add it to the calendar view |

**Case: Editing Events on Calendar**

| Action | System Response |
|---|---|
| 1. Click on event to display more information | 2. Information box pops up with all information about the event |
| 3. Click 'Edit Event' button | 4. Display form to edit the event's information |

| Action | System Response |
|---|---|
| 5. Click 'Submit' | 6. Save modified event in the database, go to calendar |

## Case: Deleting Events on Calendar

| Action | System Response |
|---|---|
| 1. Click on event to display more information | 2. Information box pops up with all information about the event |
| 3. Click 'Delete Event' button | 4. Dialog asking user to confirm event's deletion |
| 5. Confirm deletion of event | 6. Delete event in the database, return to the calendar |

## Case: Editing Calendar View Preferences

| Action | System Response |
|---|---|
| 1. Click "Edit Preferences" on Calendar Page | 2. Display preferences page |
| 3. Make desired changes and click 'submit', ex. Change color scheme of calendar | 4. Save user preferences in the database, display calendar page |

## Case: Viewing Event / Homework / etc. details in Calendar

| Action | System Response |
|---|---|
|  | 1. Calendar view is always visible with event title in the correct date |
| 2. Click on event in calendar view | 3. Display popup with all event details (title, time, priority, class, notes, reminder, custom fields, etc.)<br>As well as options to edit or delete event |

## Case: Adding Homework to Calendar

| Action | System Response |
|---|---|

| | |
|---|---|
| 1. Click 'Add Homework' Button | 2. 'Add Homework' form appears |
| 2. Fill out fields for homework info: name of homework, due date, priority, with optional fields for class, notifications, etc. | |
| 3. Click 'Confirm Homework | 4. Save homework and add it to the calendar view |
| | 5. Recalculate homework priority |

**Case: Receiving notifications for events**

| Action | System Response |
|---|---|
| | 1. Display notification popup with information about the associated event |
| 2. Acknowledge the notification and click 'Dismiss' | 3. Close popup |

**Case: Populating Contacts**

| Action | System Response |
|---|---|
| 1. Click on Contacts tab | 2. Display current list of contacts associated with the user |
| 3. Click on 'Add contact' button | 4. Display form with necessary information about contact |
| 5. Fill out information and click 'Submit' | 6. Add contact to the database and display their information on the contacts page |
| 7. System will schedule random events when available to contact important connections. | |

**Case: Receiving Homework recommendation based on Priority Algorithm**

| Action | System Response |
|---|---|

| 1. Click 'Which Homework should I work on?' Button | 2. Show priority view popup with top 3 homework recommendations in order. |
|---|---|

**Case: Planning out time to work on Homework**

| Action | System Response |
|---|---|
| 1. Already have added homework along with estimates of time to complete each assignment | |
| 2. Requests homework planner from menu | 3. Fits homework into currently available free time, making sure not to interfere with scheduled commitments.<br>Attempts to allocate time to higher priority assignments first.<br>Adds times to do homework as events to the calendar |
| | 5. Displays calendar |

**Case: Marking Homework completed**

| Action | System Response |
|---|---|
| 1. Click on homework in Calendar view | 2. Show details popup |
| 3. Click 'Completed' button | 4. Mark homework as completed, in calendar view.<br>Remove from homework recommendation algorithm<br>Remove events created as times to do that assignment from calendar |
| | 5. Close popup |

**Case: Automatic Time Sorting**

| Action | System Response |
|---|---|
| 1. Click Add Event button | |

| | |
|---|---|
| 2. Enter the information for the appropriate event | 3. An appropriate time is recommended for the users event based on the time sorting algorithm. |
| 4. The user can accept or reject the recommended time | 5. If accepted the program schedules the event for that time. Else it will use the time given by the user. |

**Case: Choosing predetermined event**

| Action | System Response |
|---|---|
| 1. Click Add Event button | 2. System displays different preset options to the user. |
| 3. The user chooses to use one of the preset options to fill out their event | 4. Prefills information for the event for the user. |
| 5.Adds more information to the event and submits | 6. Added to the calendar |

**Case: Editing notifications**

| Action | System Response |
|---|---|
| 1. Click the event | 2. System displays events details |
| 3. Select the notification | 4. Allows you to edit the time of the notification and any other details |
| 5.Changes the time period for the notification | 6. Reflects the change in the event |

**Case: Disabling notifications**

| Action | System Response |
|---|---|
| 1. Click the event | 2. System displays events details |
| 3. Select the notification | 4. Allows you to edit the notification |
| 5.Delete the notification | 6. Notification no longer visible |

**Case: Disabling notifications for series of events**

| Action | System Response |
|---|---|
| 1. Click the event that is in a series | 2. System displays events details |
| 3. Select the notification | 4. Allows you to edit the notification |
| 5.Delete the notification | 6. Asks if you want to disable the notification for all events in the series. |
| 7. Selects yes to disable all notifications | 8. Disables notifications for all the events |

**Case: Constant reminders**

| Action | System Response |
|---|---|
| 1. Click the event | 2. System displays events details |
| 3. Click the constant reminder button | 4. After a period of time the system will remind you about the event during the time period |

**Case: Random reminders**

| Action | System Response |
|---|---|
| 1. Go to settings page | 2. Display current user settings |
| 3. Activate random reminders | 4. Log setting in database, remind user at random times during the week to use their directory of important contacts. |

**Case: Preset events**

| Action | System Response |
|---|---|
| 1. Click Add Event button | 2. Display options to create an event. In this display are multiple options to create either, class, meeting, finals/exam, etc. Classes and some other events will create multiple events that repeat indefinitely. |
| 3. Click on one of the preset elements | 4. Fills in event with presets values |

| | |
|---|---|
| 5. Adds rest of values | 5. Event added to calendar |

**Case: Define Semester**

| Action | System Response |
|---|---|
| 1. Click settings menu | 2. Displays settings |
| 3. Click semester button | 4. Display details on semester |
| 5. Update semester start and end date | 5. Changes the values of the start and end date of the semester, and should be reflected in classes that repeat for only that duration. |

**Case: View events**

| Action | System Response |
|---|---|
| 1. Click application executable | 2. Launches application and weekly calendar viewer containing all the users events |

**Case: View events for day**

| Action | System Response |
|---|---|
| 1. Click day view button | 2. 'Zooms in' on the events for the day |

**Case: View events for week**

| Action | System Response |
|---|---|
| 1. Click week view button | 2. Changes from the current view to 'zoom in' on the events for the week |