# ScheduSmart

Sprint 2 Planning Document

Team 30

**Reece Ausmus, Cassie Chang, Stanley Huang, Bradley Norris, Himanshu Sinha, Gloria Xu**

## Sprint Overview

During this sprint, our goals include cleaning up our codebase, enhancing the visual appeal of our user interfaces, and implementing numerous new features to finalize the functionality of the calendar and task manager.

### Scrum Master:

Reece Ausmus

### Meeting Plan:

We will meet weekly on Sunday at 5:00pm with our TA. We also will meet weekly on Wednesday at 7:00pm and Saturday at 11:00am with just our team. Each meeting will be on Zoom.

### Risks and Challenges

For this sprint, we are confronted with several significant risks and challenges that we must address. Firstly, it is imperative that we enhance the aesthetics and user-friendliness of our UI to ensure it meets the expectations and needs of our users. Secondly, we must focus on improving communication and collaboration among team members to effectively address any issues or obstacles that may arise, building on the lessons learned from Sprint 1. Lastly, this sprint marks

the beginning of our implementation of Jenkins to establish a robust pipeline that will not only enhance our code quality but also facilitate Continuous Integration and Continuous Deployment (CI/CD), ultimately leading to more efficient and effective development processes.

# Current Sprint Detail

## User Story #1

As a user, I would like to have the calendar automatically schedule time to complete homework.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Develop functionality that connects the task manager to the calendar system, allowing information to be passed between both systems. | 3 hrs | Himanshu |
| 2 | Create a user interface to mark certain tasks for auto-scheduling | 2 hrs | Himanshu |
| 3 | Develop an algorithm that looks at the user's overall schedule, and optimizes the auto-scheduler to automatically assign time for tasks | 4 hrs | Himanshu |
| 4 | Develop unit tests that checks that auto-assigned tasks do not overlap with other scheduled events as well as auto-assigns time before the due date | 1 hr | Himanshu |

### Acceptance Criteria:

1. Given that the cross-system functionality is properly links the task manager to the event system, tasks assigned for auto-scheduling should automatically be placed as events on the calendar, without any extra effort from the user
2. Given that the auto-scheduler algorithm checks for present events, tasks that are scheduled automatically should not conflict with any other events already placed. Instead, they should fill in the gaps on the calendar.
3. Given that the unit-tests are correctly implemented, tasks should not be placed during "odd-ball" hours, such as those after the due date of the task, nor during time that is set for sleeping or other necessary activities.

## User Story #2

As a user, I would like to attach documents/files to my tasks.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Develop functionality that allows users to upload files and store file data in the database. | 3 hrs | Himanshu |
| 2 | Develop a system that scans files for malicious code or otherwise malicious intent and rejects files if need be | 4 hrs | Himanshu |
| 3 | Create functionality that allows users to download files from the tasks they were uploaded to | 2 hrs | Himanshu |
| 4 | Create unit tests that checks that “bad” files are rejected as well as ensuring the “good” files are correctly uploaded to tasks. | 1 hr | Himanshu |

### Acceptance Criteria:

1. Given that the file-checker checks for properly uploaded file extensions, any file that contains malicious code or otherwise malicious data should be rejected. This should not come at the cost of also rejecting files that are not malicious.
2. Given that the file-upload correctly sends data to firebase and receives data in a similar fashion, files that pass the check should be uploaded and stored in the data-base, under the specific task that they are assigned to. Files uploaded should also be able to be downloaded from the task manager menu.
3. Given that the unit tests are correctly implemented, a known malicious piece of data should be rejected, and a known good file should be accepted by the program.
4. Given that the file-upload is correctly implemented, files uploaded should retain all of their information, and data should not get lost or corrupted.

## User Story #3

As a user, I would like to track progress on my assignments and see how much time I have planned to work on them.

| # | Description | Estimated Time | Owner |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Develop front-end functionality that creates a progress meter to illustrate task completion. | 1 hr | Himanshu |
| 2 | Develop functionality that allows users to create sub-tasks that break down a single task into more measurable quantities | 2 hrs | Himanshu |
| 3 | Create a system that once sub-tasks are marked completed, the progress meter is incremented towards completion | 1 hr | Himanshu |
| 4 | Create a system that pulls information from the calendar, with a “task_id” system that aggregates all planned study time for a task. | 4 hrs | Himanshu |
| 5 | Illustrate the amount of planned study time on the task page | 1 hr | Himanshu |
| 6 | Develop unit tests that check to make sure the progress meter accurately matches the aggregate time of all completed sub-task . | 1 hr | Himanshu |

## Acceptance Criteria:

1. Given that the front-end functionality is displayed correctly, the progress meter should be visually appealing, streamlined with the rest of the UI, and accurately depict the amount of time left to complete a task.
2. Given that the cross-system functionality between the task manager and the event system is seemles , the progress meter should work with the auto-scheduler and correctly display how much time is currently set aside to work on a task, so that users can see if they have enough time scheduled to work on that given task.
3. Given that the unit tests are correctly implemented, the tests should fail on any of three accounts: The progress meter does not accurately represent the sum total time of the tasks marked completed, the progress meter does not accurately display the amount of time the calendar has set aside for a given task, and the progress meter does not re-adjust if new subtasks are added to the list.

## User Story #4

As a user, I would like to be able to navigate the schedule with keyboard shortcuts without using the mouse.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Allow users to navigate between different calendar format through keyboard shortcuts | 3 hrs | Cassie |
| 2 | Allow users to navigate between different day/week/month/year through keyboard shortcuts | 2 hrs | Cassie |
| 3 | Allow users to add events through keyboard shortcuts | 2 hrs | Cassie |
| 4 | Create unit tests to check if users can actually use shortcuts to manipulate the schedule | 2 hrs | Cassie |

### Acceptance Criteria:

1. Given that the keyboard shortcut ‘Shift + d’ is setted up, when users press them, then they switch  calendar format to be daily format
2. Given that the keyboard shortcut ‘Shift + w’ is setted up, when users press them, then they switch  calendar format to be weekly format
3. Given that the keyboard shortcut ‘Shift + m’ is setted up, when users press them, then they switch  calendar format to be monthly format
4. Given that the keyboard shortcut ‘Shift + y’ is setted up, when users press them, then they switch  calendar format to be yearly format
5. Given that the keyboard shortcut ‘Shift + a’ is setted up, when users press them, then they can turn on and off the adding event popup form to add event
6. Given that the keyboard shortcut ‘Shift + n’ and “Shift + p” are setted up, when users press them, then they can change to different day in calendar

## User Story #5

As a student, I would like to be able to take notes.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Create a button to redirect users to the notes page | 1 hr | Cassie |
| 2 | Create a button allowing users to add notes | 1 hrs | Cassie |
| 3 | Display notes with contents taken by users | 4 hrs | Cassie |
| 4 | Create a button allowing users to delete the notes they don't need anymore | 2 hrs | Cassie |
| 5 | Allow users to edit the contents of notes and the edited content can be saved and displayed properly | 2 hrs | Cassie |
| 6 | Create unit tests to check if users can add notes successfully | 2 hrs | Cassie |

### Acceptance Criteria:

1. Given that the button is setted up correctly, when users press a button on the dashboard, they will be redirected to the page where users can take notes.
2. Given the taking notes function is implemented correctly, when users want to take notes, they can press a button to generate notes with desired contents.
3. Given the editing notes function is functioned correctly, when users want to make some changes to the content of notes, they can edit the notes by pressing a button and the modified contents will be kept on the notes.
4. Given the deleting notes function is implemented correctly, when users want to discard a note, then they can make it through a button

## User Story #6

As for maintainability, developers have to use Jenkins and Git to do version control and CI/CD allowing us to efficiently maintain our web.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Integrate GitHub with Jenkins | 4 hrs | Cassie |

| | | | |
|---|---|---|---|
| 2 | Write Jenkinsfile to build pipelines | 3 hrs | Cassie |
| 3 | Test if pipelines are constructing successfully | 2 hrs | Cassie |

### Acceptance Criteria:

1. Given that the integration of GitHub and Jenkins is implemented correctly, when developers push some changes on the repo, they will receive the results of testing
2. Given that the Jenkinsfile is implemented correctly, when developers push some changes to the repo, then the test cases listed on the file should be run.
3. Given that the pipeline is implemented correctly, when developers want to test this pipeline, they can intentionally make some errors to test if they will be reported, then correct the error to see if no error will be reported.

## User Story #7

As a user, I would like to have a consistent style throughout the web application.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Style the main page so that it has a consistent style with others | 3hrs | Gloria |
| 2 | Style the task manager page | 2hrs | Gloria |
| 3 | Create a dashboard so that users can get access to every page easily | 2hrs | Gloria |
| 4 | Create tests to ensure pages are styled correctly. | 1hr | Gloria |

### Acceptance Criteria:

1. Given that the main page is styled properly, when users browse, they will be presented with an attractive interface where all components function smoothly.

2. Given that the task manager page is well styled, when users manage their tasks, they will experience a visually appealing interface where all components function seamlessly.
3. Given that the dashboard is created perfectly, when users click a button, they will get access to their wanted page smoothly.

## User Story #8

As a user, I would like to add locations to my events using text or a map feature (i.e., Google Maps).

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Create a button for users to choose adding locations using text or a map feature | 1hrs | Gloria |
| 2 | Create a button in setting page that asks if users want to use the map feature | 1hrs | Gloria |
| 3 | Display the selected location on the event details page | 2hrs | Gloria |
| 4 | Store the location details in the firebase | 4hrs | Gloria |
| 5 | Create unit tests to check if the users can add locations to my events using text or a map feature successfully | 1hrs | Gloria |

## Acceptance Criteria:

1. Given that the button of inputting locations using text or a map feature is correctly implemented, when users add locations while creating events, the information should be stored in the firebase and be updated on the page accordingly.
2. Given that the button which asks if users want to use the map feature runs well, when users set this feature in the setting page, the feature should appear in the adding-event page and runs smoothly

3. Given that the display of  the selected location on the event details page and storage of the location details in the firebase are implemented, when users add the locations successfully, the details should be clearly shown on the events details page.
4. Given that the unit tests are implemented correctly, all buttons should be verified upon action and log appropriate information in the console.

## User Story #9

As a user, I would like to have access to a complete settings page and a create-account page with a consistent style.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Integrate all the setting functionalities in to one page | 5hrs | Gloria |
| 2 | Create a button which links to the setting page in the dashboard page | 1hrs | Gloria |
| 3 | Make the styles of create-account page and setting page consistent with others | 5hrs | Gloria |
| 4 | Create unit tests to check if everything in setting page and create-account page works well | 2hrs | Gloria |

### Acceptance Criteria:

1. Given that  the settings page is implemented user-friendly, when users want to change their settings, they can easily find a button which allows them to do so.
2. Given that the button of the setting page is properly created in the dashboard page, when users click the button, they will change to the setting page smoothly,
3. Given that the styles of pages are properly changed, when users use our system, all of the pages should have a consistent style.

4. Given that the unit-tests are implemented correctly, when users use our system correctly, they will not receive any error message.

## User Story #10

As a user, I would like to change my time zone.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | All the assignments and works specified with time-related information should be changed to display the selected time-zone. | 2hrs | Bradley |
| 2 | When users select a timezone from the dropdown, update the new field in firebase accordingly. | 3hrs | Bradley |
| 3 | Design a unit test to confirm if the time zone change is reflected accordingly across the assignments list. | 1hrs | Bradley |

### Acceptance Criteria:

1. Given that the time-zone functionality has a working frontend to backend communication setup, when a user selects a timezone from the dropdown, the selection will update a new “timezone” field which is stored in Firebase for each user.
2. Given that the time-zone is correctly stored where it should be, when a user has a chosen timezone, all assignments and works will display the selected timezone (calendar & assignment list).
3. Given that the unit-tests adequately measure time-zone adjustments, the timezone should be correctly reflected across the assignments list with persistence throughout separate logins. Also, any invalid selections by users will be dealt with.

## User Story #11

As a user, I would like to be able to track daily habits.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Create a new page that is accessible once logged in called “Habits” | 2hrs | Bradley |
| 2 | Add button to reach page via the main calendar page, Add buttons to return to calendar + settings from habits page. | 1hrs | Bradley |
| 3 | Add a component that allows users to input their daily eating habits (calories, food eaten, weight, etc) | 3hrs | Bradley |
| 4 | Design a unit test to ensure that all buttons act as expected | 1hrs | Bradley |

### Acceptance Criteria:

1. Given that the “Habits” page is created and displayed, it should only be accessible via the calendar view once logged in.
2. Given that the “Habits” page has proper navigational options, users should be able to navigate back to calendar and their account settings pages
3. Given that the eating habits component correctly parses in user input, users should be able to input information on their eating habits
4. Given that the unit test appropriately measures all navigation tools, all buttons should be verified upon action and log appropriate information in the console.

## User Story #12

As a user, I would like to be able to track my eating habits via the Habits page

| # | Description | Estimated Time | Owner |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Store eating habits as new field in firebase for each user | 3hrs | Bradley |
| 2 | Allow users to add, edit, and delete items to the list component | 4hrs | Bradley |
| 3 | Add capability for the eating habits component to be exported as CSV | 2hrs | Bradley |
| 4 | Add capability for the eating habits component to create formatted graphs using selected variables (weight, calories) | 4hrs | Bradley |
| 5 | Link exporting and graph functionality to buttons with dropdowns that update with context on Habits page | 2hrs | Bradley |
| 6 | Design a unit test to confirm whether or not exporting succeeded | 1hrs | Bradley |
| 7 | Design a unit test to confirm whether or not eating habits are updated correctly in firebase | 1hrs | Bradley |

## Acceptance Criteria:

1. Given that the eating tracker correctly communicates from frontend to backend database, all information input should be sent to firebase once a “submit” button is pressed to add a new item.
2. Given that the eating tracker has the correct functionality with options, users should be able to add, edit, and delete items to the list component with no issues.
3. Given that the export feature is working as expected, users should be able to export their habits list as a CSV.
4. Given that the graphing functionality correctly stores users data, users should be able to download graphs containing information on their trends for variables (weight, calories).
5. Given that the buttons are working as expected, exporting should be context dependent upon the selected options from a drop-down.

6. Given that the unit tests properly measure file functionalities, export failure should be checked and dealt with, and all fields should be updated in firebase accordingly.

## User Story #13

As a student, I would like to manage time schedules before the start of a semester to smooth the course enrollment procedure, such as scheduling class times and breaks.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Create a page for students to schedule courses. | 3hrs | Reece |
| 2 | Create a section on this page to schedule breaks from class such as spring break and winter break. | 2hrs | Reece |
| 3 | Create a system in the backend to store courses in the database. | 2hrs | Reece |
| 4 | Implement a system to show courses on the main calendar. | 2hrs | Reece |
| 5 | Create unit tests to confirm valid inputs, proper storage, and proper display of courses. | 2hrs | Reece |

### Acceptance Criteria:

1. Given the add course page is designed properly, users should be able to add as many courses as necessary with individual schedules for each.
2. Given the add course page is designed properly, users should be able to add breaks in the course schedule. Adding breaks will delete course meetings during the break.
3. Given the courses are stored properly, users should be able to access their and only their courses and breaks.
4. Given the courses feature is implemented properly, users should be able to see their courses on the main calendar in a simple visual format.

## User Story #14

As a user, I would like to invite other users to my events and see events I am invited to.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Create a button to share an event with a user interface. | 1 hr | Reece |
| 2 | Create a system to notify a user of an invitation and allow a response. (this will take a lot of research) | 4 hrs | Reece |
| 3 | Create a connection to the backend to share the event and update the database. | 2 hrs | Reece |
| 4 | Create unit tests to validate input, test notification system, and confirm database updates. | 2 hrs | Reece |

### Acceptance Criteria:

1. Given the share event feature is implemented correctly, users should be able to see invites they have sent and received in a concise way.
2. Given the share event feature is implemented correctly, users should be able to see shared events on each of their own respective calendars.
3. Given the share event feature is implemented correctly, users should be able to share an event when creating it or after it is created.
4. Given the share event feature is secure, invitations to an event should only be visible to the users necessary.

## User Story #15

As a user, I would like to be able to create, read, update, and delete (CRUD) events and tasks.

| # | Description | Estimated Time | Owner |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Create a system of storing events and tasks server-side to reduce calls to the database. | 4 hr | Reece |
| 2 | Create any buttons and UIs that are not already implemented for full CRUD capability. | 2 hr | Reece |
| 3 | Create backend functionality to perform and CRUD operations. | 2 hr | Reece |
| 4 | Create unit tests to ensure events and tasks are correctly manipulated in the backend and in the database. | 2 hr | Reece |

### Acceptance Criteria:

1. Given this system is efficient, tasks and events should be loaded from the database at login, and database calls should be kept to a minimum.
2. Given this system is implemented correctly, users should be able to perform any CRUD operation on a task or event.
3. Given this system safely stores data, the server should periodically save to the database as well as save at logout.
4. Given this system is implemented correctly, the application should inform users when there are unsaved changes.

## User Story #16

As a user, I would like the calendar home page to be simple and not overwhelming while including pertinent information, such as the date and time.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Reformate the main page to make it look cleaner | 3hr | Stanley |
| 2 | In backend, adding parser that can parse string time into meaningful integer that can be used to evaluate return value | 2hr | Stanley |

| | | | |
|---|---|---|---|
| 3 | Putting events to the calendar, which includes day, week | 5hr | Stanley |
| 4 | Connect the event with checkbox and show only the checkbox that is picked | 2hr | Stanley |
| 5 | Test if the page is shown correctly and make sure that other pages format is correct | 1hr | Stanley |

Acceptance Criteria:

1. Given the backend is implemented correctly, all events should be retrieved depending on which calendars are selected. (All events that has been selected will be send as data)
2. Given the frontend logic is implemented correctly, it will request events data every time when a new calendar is  added in the display list or removed from it.
3. Given the frontend logic is implemented correctly, it should parse the event to the calendar and repetitively put it on the calendar depending on which type of events it is.
4. Given the UI is implemented correctly, users will be able to drag and drop on the event for better scheduling only on the calendar, but will not change the data in the backend.

## User Story #17

As a user, I would like to see statistics on my productivity at certain times.

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Create a button in mainpage that pop up a new pages related to tasks' detail info | 1hr | Stanley |
| 2 | Accessing backend that retrieve all data (including assignment and events) among the preferred domain of time including 7 days, 30 days, year and show it in the page as list | 2hr | Stanley |
| 3 | Putting event on the assignment list in | 2hr | Stanley |

| | | | |
|---|---|---|---|
| | mainFrame | | |
| 4 | Add function in backend that stored when the user finish one task | 1 hr | Stanley |
| 5 | Test the UI to shows correct information | 1hr | Stanley |

Acceptance Criteria:

1. Given the assignment list UI is implemented correctly, users can see the unfinished task in the assignment list with the name of the assignment and deadline only.
2. Given the assignment list UI is implemented correctly, user can mark the task as finish with a checkbox, which will automatically remove the task from assignment list
3. Given the assignment list UI is implemented correctly, when user have multiple tasks, user will have a scroll bar to view all the task
4. Given the detail UI is implemented correctly, user will be able to see what is done in the past as a conclusion
5. Given the backend is implement correctly, when a user mark the task as finish, the backend will received the time when the task is finish and update that task in firebase

## User Story #18

As a user, I would like to add other account as my friends and able to manage my friend list

| # | Description | Estimated Time | Owner |
|---|---|---|---|
| 1 | Fix the issue of user account crash the server | 1hr | Stanley |
| 2 | Implement user function in backend to search and add friends | 2hr | Stanley |
| 3 | Create a small box in the right bottom to access friend information, which will pop up a windows that show all friend and a input bar to search and add friends | 3hr | Stanley |

| | | | |
|---|---|---|---|
| 4 | Adding send message function that account can sent message to other friends account | 2hr | Stanley |
| 5 | Adding friend function will sent a message to the friend and confirm whether he knew the person who is adding him as friends | 1hr | Stanley |
| 6 | Adding function to delete friends and in other account, they should not be able to sent message anymore | 1hr | Stanley |
| 7 | Test if the chat box shows all users friends and it's function | 30min | Stanley |
| 8 | Create unit test to see whether friend is correctly modified in backend | 30 min | Stanley |

## Acceptance Criteria:

1. Given the expand friend UI button is implement correctly, user can open and close the chat box (which contains friend list and chat room) by that button
2. Given friend UI search system is implemented correctly, user can view other account username only and able to request adding them as friend
3. Given the friend list system is implemented correctly, when user click on their friend, the UI will be direct to their chatroom where contains all messages from both user and friend in past
4. Given the chat room UI is implemented correctly, the user will have an input bar and sent message button. When the button is click, text in the input bar will appear on the chat room
5. Given the chat room UI is implement correctly, user can send message to the chatroom, but if that person is not user's friends, he or she will not see other message but a notification from system asking to add user as a friend

6. Given the chat room UI is implemented correctly, users can direct themselves back to the friend list with a button.

## Hours Summary

Reece Ausmus: 30 hours

Cassie Chang: 30 hours

Stanley Huang: 31 hours

Bradley Norris: 30 hours

Himanshu Sinha: 30 hours

Gloria Xu: 30 hours

# Sprint 1 - Incomplete User Stories

**Requirements (Backlog)**

**Bold: Sprint 1 User Stories**

*Italics: Sprint 2 User Stories*

Functional:

1. Introduction:

~~**2. Login system:**~~

~~2.1. **As a new user, I would like to create a new account**~~

~~2.2. **As a user, I would like to login and logout into my account**~~

3. Calendar system:

~~3.1. **As a user, I would like to add events to my calendar.**~~

~~3.2. **As a student, I would like to add my homework deadline to the schedule.**~~

~~3.3. As a student, I would like to schedule time to study for tests with recurring events.~~

~~**3.4. As a student, I would like to find the closest available time (15-minute meeting, etc).**~~

***3.5. As a student, I would like to see when I have no class (festival or break).***

*3.6. As a student, I would like to manage time schedules before the start of a semester to smooth the course enrollment procedure, such as scheduling class times and breaks.*

*3.7. As a student, I would like to have the calendar automatically schedule time to complete homework.*

3.8. As a student, I would like to type my schedule or tasks into a chat box and have AI schedule my day for me.

***3.9. As a student, I would like to be able to list times for exams.***

3.10. As a student, I would like to receive suggestions for optimal study plans and breaks.

***3.11. As a student, I would like to receive notifications about deadlines and exams.***

*3.12. As a user, I would like to add locations to my events using text or a map feature (i.e., Google Maps).*

*3.13. As a student, I would like to attach documents/files to my tasks.*

3.14. As a student, I would like to see the weather and have it impact my AI-generated schedule.

3.15. As a user, I would like to add other account as my friends and able to manage my friend list

3.16. As a student, I would like to share my schedule with teammates to find an available time slot for meetings.

~~**3.17. As a user, I would like to be able to add a repeatable event (sprints, meetings, etc. etc.)**~~

~~**3.18. As a worker, I would like to mark a period as "working time" for every week, including commutes, and "out-of-office" time.**~~

~~**3.19. As a user, I would like to visualize my calendar in different formats (day, week, month, etc.)**~~

3.20. As a user, I would like the calendar home page to be simple and not overwhelming while including pertinent information, such as the date and time.

~~**3.21. As a user, I would like to create different calendars that I can turn on and off individually.**~~

**3.22. As a user, I would like to drag and drop events in my schedule.**

~~3.23. **As a user, I would like to change my time zone.**~~

*3.24. As a user, I would like to be able to navigate the schedule with keyboard shortcuts without using the mouse.*

3.25. As a user, I would like to adjust what time my schedule starts and ends, essentially scheduling when I am sleeping.

3.26. As a user, I would like to see what time of day it is in relation to my events.

3.27. As a user, I would like to see a quick view of my calendar without being overwhelmed with all features.

3.28. As a team manager, I would like to have a meeting with all the team members when all of them are free.

3.29. As a nurse, I would like to have all patients' dosing time (with constant intervals)

4. Assignment tracker:

~~4.1. **As a student, I would like to have an assignment tracker.**~~

~~4.2. **As a user, I would like to view my calendar and assignment tracker.**~~

*4.3.* ***As a student, I would like to be able to prioritize my tasks.***

~~4.4. **As a student, I would like to add assignments to the assignment tracker.**~~

*4.5. As a student, I would like to track progress on my assignments and see how much time I have planned to work on them.*

~~**4.6. As a student, I would like to sort my assignments based on due date, creation date, class, type, etc.**~~

4.7. As a user, I would like to set and track long term goals.

4.8. As a user, I would like my to-do list tasks to synchronize with calendar events if toggled.

~~4.9. **As a user, I would like to mark work done as completed.**~~

4.10. As a user, I would like to differentiate events and tasks.

4.11. As a team leader, I would like to assign tasks to employees.

5. Settings:

**5.1. As a bodybuilder, I would like to add the daily calories I have.**

5.2. As a student, I would like to see statistics on my productivity at certain times.

**5.3. As a bodybuilder, I would like a summary of how I work on it (total and average per month).**

5.4. As a user, I would like to have a modular, customizable home screen.

**5.5. As a user, I would like to set up reminders for my events and tasks.**

*5.6. As a user, I would like to invite other users to my events and see events I am invited to.*

*5.7. As a user, I would like to have access to a complete settings page and a create-account page with the same style.*

5.8. As a user, I would like to color coordinate my schedule.

***5.9. As a user, I would like to change the whole displayed text as my native language.***

6. Misc:

*6.1. As a student, I would like to be able to take notes.*

6.2. As a student, I would like to be able to put a to do list aside for convenience.

6.3. As a student, I would like to have a small notebook to record things I have done in the past as a reminder (finished assignments, meetings, etc.)

6.4. As a student, I would like to be able to import assignments from Brightspace, Canvas, or Blackboard.

6.5. As a user, I would like to be able to connect Google Calendar to this calendar.

6.6. As a user, I would like to export my schedule to a printable format.

**6.7. As a user, I would like to track daily habits that I set up, e.g. 5 mins of reading, 10 mins stretching, and cook 3 meals.**

~~**6.8. As a user, I would like to see the weather.**~~

6.9. As a boss, I would like to have all workers see the schedule but unable to change it.

6.10. As a team manager, I would like to have my team members drop down what they have done for the project in the past.

6.11. As a team manager, I would like to have a chart (summary) on how each team member processes their job.

6.12. As a team leader, I would like to receive automated progress reports from the team.

6.13. As an employer, I would like to see when my employees will be out of the office.

6.14. As an employer, I would like to have the ability to accept or deny time-off requests that employees submit.

6.15. As an employee, I would like to schedule when I will be out of the office.

6.16. As an employee, I would like to quickly send meeting invites to coworkers.

6.17. As an employee, I would like to integrate my calendar with Zoom meetings, such as creating a meeting from the calendar, joining a meeting from the calendar, and getting notifications from the calendar about my meetings.

6.18. As a dietician, I would like to track my daily calories and water intake.

6.19. As a software engineer, I would like to add a scrum framework to my schedule

6.20. As a software engineer, I would like to upload design pdfs to the notebook section

6.21. As a software engineer, I would like to integrate my notebook with Github, including adding milestones to my calendar and receiving notifications from the calendar when someone pushes to a repository.

6.22. As a writer, I would like to keep my drafts organized with a simple file management system and events on the calendar when I update a draft.

6.23. As a writer, I would like to keep different notebook entries organized with folders/tags

Non-Functional:

**Performance**

To ensure our users have a positive experience using our website, our server response time must be within 600 ms.

**Scalability**

Because we have great ambition to make our product available to many people, we employ Nginx, known for its ability to handle large numbers of concurrent connections, and expect that it can handle up to 500 simultaneous requests. What's better, we will also try to enable short-term caching by setting up the Nginx configuration to allow more concurrent requests per second. If time allows, we may try to apply Proxmox to realize horizontal scaling by creating more than one server to support our service.

**Security**

For ScheduSmart, we place emphasis on customer privacy, thus designing several mechanisms to protect precious information. First of all, we will prevent SQL injection by using parameterized statements, escaping special characters, etc. Also, users' passwords will be encrypted through password encryption functions provided by the database, such as scram with sha256 so that we can shield our consumers from man-in-the-middle attacks for our users. Moreover, even on our server side, we cannot know the passwords since they are stored as ciphertext. Besides, we also need to deal with cross-site scripting when working on our front-end and back-end pages. To

achieve this, we would be careful with HTML tags and configure CSP properly for our Nginx web server. What's more, we will also let the Nginx server manage a reverse proxy to protect our server. If time allows, to thoroughly ensure our web security, we will take advantage of OWASP ZAP to do penetration testing.

**Usability**

To make it easier for our users to get started with ScheduSmart, we have figured out several requirements for the development team to help our consumers. On the one hand, we should design interactive web tours so that users can be more familiar with our functions and know how to use them by the step-by-step instructions. On the other hand, we will design a user-friendly interface and find more than 6 participants to use our applications and share their user experience with us to help us improve our products. If time allows, we can write a step-by-step user guide on Medium to make sure that our users know how to use this application.

**Maintainability**

*As for maintainability, developers have to use Jenkins and Git to do version control and CI/CD allowing us to efficiently maintain our web.* Besides, we should pay attention to error handling so that the developer who is responsible for maintaining this application can easily know which part causes errors from the clear and concise error messages at once.