

ThinkItThru: Requirements Specification

By: Jensen Schmidt, Levi Wright, and Zachary Travis

Table of Contents

	Page #
Project Name	3
Team Member Names.....	3
Abstract.....	3
Tools & Technologies.....	3
Requirements List	3
1. (Onboarding) Sign-in Form	3
2. (Onboarding) Registration Form:	5
3. (General) Modal.....	7
4. (General) Dialog Box	7
5. (General) New Task Modal.....	7
6. (Dashboard) Navbar.....	8
7. (Dashboard) Tasks	10
8. (Dashboard) Game.....	13
Updated Timeline	15
Appendix	16
Figure 1: Team Member Names.....	16
Figure 2: Tools and Technology*	16
Figure 3: Tentative Schedule*	16

Project Name

Being a web application focused on aiding students by promoting consistent daily work, we decided to call the application **ThinkItThru**. We felt this name accurately conveyed what we wanted students to be able to do when they create any goals or receive any assignments: break them down into manageable daily workloads that could alleviate stress in a healthy way.

Team Member Names

See Figure 1 in the appendix for a detailed description.

Abstract

Students struggle with workloads at most points in their academic career. It is common to find those students talking about “all-nighters” and various unhealthy habits—generally related to studying in large quantities just before a test or project, also known as “cramming”—that contribute to an overall unhealthy lifestyle of chronic stress. ThinkItThru’s main goal is to help students visually see the hours of work they need to accomplish daily, help students create positive habits of daily work, as well as help students fit in any additional goals while they are completing their schooling. The web application aims to incentivize these habits with both visual feedback and “gamifying” their progress, as well as socially connecting users to common groups within their majors. This gives students an additional tool to help create positive changes in their daily lives, helping to lead them onto a path of personal growth for their future career.

Tools & Technologies

See Figure 2 in the appendix for a detailed description.

Requirements List

1. (Onboarding) Sign-in Form

- 1.1. The sign-in form will be a single message, with the following:
 - 1.1.1. A header text label, “Sign-in Using Email and Password”.
 - 1.1.2. A text label to indicate the input of an email address.
 - 1.1.3. An input field for the email address of the user.
 - 1.1.3.1. The input field will have a placeholder value of “[example@gmail.com](#)”.
 - 1.1.3.2. The input field will only allow valid email addresses.
 - 1.1.3.2.1. Emails must contain an @ symbol.
 - 1.1.3.2.2. On an invalid email, user is notified on the field.
 - 1.1.4. A text label to indicate the input of a password.
 - 1.1.5. An input field for the password of the user.

- 1.1.5.1. The input field will be blank.
- 1.1.5.2. The input field will hide user's input behind * characters for each character entered.
- 1.1.5.3. The input field will only accept valid passwords.
 - 1.1.5.3.1. The password must be at least 8 characters.
 - 1.1.5.3.2. The password must be under 24 characters.
 - 1.1.5.3.3. The password must contain at least one special character (!, _, #, @, >, <, ., (,), &, or ?).
 - 1.1.5.3.4. The password must contain at least one capital and one lower case letter.
- 1.1.6. A clickable button labeled "Sign-in".
 - 1.1.6.1. On-click, a database will be queried for the user's credentials.
 - 1.1.6.1.1. If a match is found in the database, the user will be taken to the **dashboard**.
 - 1.1.6.1.1.1. If a match is found in the database, and the user has not gained any **Login Streak** since the previous day, the application will add a point to the **User's Login Streak**.
 - 1.1.6.1.1.1.1. The user may only gain 1 point in the **Login Streak** per day.
 - 1.1.6.1.1.1.2. If the user has not logged into the system in two days, the **Login Streak** points will reset.
 - 1.1.6.1.1.1.3. For every point in the **Login Streak**, the user will gain an additional overall 1% more **Experience** across all **Experience** gained.
 - 1.1.6.1.1.2. If a match is found in the database, and the user has not gained **First Login of the Day Experience**, the user will be awarded that **Experience** as they are taken to the **dashboard**.
 - 1.1.6.1.1.2.1. The user may only gain **Experience** for their **First Login of the Day** once per day.
 - 1.1.6.1.2. If no match is found in the database, a message will display to the user stating, "Invalid login credentials, please try again."
- 1.1.7. A subheader text, "Sign-in Using a Different Method:"
- 1.1.8. A clickable button to sign in using a Google account, labeled, "Sign-in with Google".
 - 1.1.8.1. On-click, the Google account authentication will occur.
 - 1.1.8.1.1. If the Google account has any prior information in the database, the user will be taken to the **dashboard**.
 - 1.1.8.1.2. If the Google account does not have prior information in the database, the user will be taken to the **registration form**, modified to not require an email address or password.
- 1.1.9. A clickable button labeled, "Sign-up".
 - 1.1.9.1. On-click, the user will be taken to the **registration form**.
- 1.2. The sign-in form should adhere to best practices for ease-of-access from the user.
 - 1.2.1. Every label should be navigable by keyboard.
 - 1.2.2. Every label should select their input fields when clicked by the user.
 - 1.2.3. Every label should use aria text.
 - 1.2.4. Every input field should be navigable by keyboard.

- 1.2.5. Every input field—except the password field—should have default placeholder text.
- 1.2.6. Every input field should use aria text.

2. (Onboarding) Registration Form:

- 2.1. The sign-up form will contain one component including the following:
 - 2.1.1. A text label, “First Name”.
 - 2.1.2. An input text field connected to the “First Name” label.
 - 2.1.2.1. The input text field will have a placeholder value of “Jane”.
 - 2.1.2.2. The input box will not accept numbers and special characters, only letters.
 - 2.1.2.2.1. With invalid input, a message under the field will state that this field can only accept letters.
 - 2.1.2.3. A “*” symbol will be marked next to the field to indicate a required component.
 - 2.1.3. A text label, “Last Name”.
 - 2.1.4. An input text field connected to the “Last Name” label.
 - 2.1.4.1. The input text field will have a placeholder value of “Doe”.
 - 2.1.4.2. The input box will not accept numbers and special characters, only letters.
 - 2.1.4.2.1. With invalid input, a message under the field will state that this field can only accept letters.
 - 2.1.4.3. A “*” symbol will be marked next to the field to indicate a required component.
 - 2.1.5. A text label, “Major”.
 - 2.1.5.1. A drop-down menu field with the available majors with the application.
 - 2.1.5.2. A “*” symbol will be marked next to the field to indicate a required component.
 - 2.1.6. A text label, “Email”.
 - 2.1.7. An input field connected to the “Email” label.
 - 2.1.7.1. The input field will contain a placeholder text of example@gmail.com
 - 2.1.7.2. The input field will accept only valid emails.
 - 2.1.7.2.1. Emails must contain an @ symbol.
 - 2.1.7.2.2. With an invalid email, the user will be notified under the field.
 - 2.1.7.3. A “*” symbol will be marked next to the field to indicate a required component.
 - 2.1.8. A text label, “Password”.
 - 2.1.9. An input field connected to the “Password” label.
 - 2.1.9.1. The input field will not have any placeholder values.
 - 2.1.9.2. The input field will hide user’s input behind * characters for each character entered.
 - 2.1.9.3. The input field will only accept valid passwords.
 - 2.1.9.3.1. The password must be at least 8 characters.
 - 2.1.9.3.2. The password must be under 24 characters.
 - 2.1.9.3.3. The password must contain at least one special character (!, _, #, @, >, <, ., (,), &, or ?).
 - 2.1.9.3.4. The password must contain at least one capital and one lower case letter.
 - 2.1.9.4. A “*” symbol will be marked next to the field to indicate a required component.
 - 2.1.9.5. With invalid input, a message under the field will state “Invalid Password”.
 - 2.1.10. A text label describing the valid password structure requirements.
 - 2.1.11. A text label, “Verify Password”.

- 2.1.12. An input field connected to both the “Password” input field and the “Verify Password” label.
 - 2.1.12.1. The input field will not have any placeholder values.
 - 2.1.12.2. The input field will hide user’s input behind * characters for each character entered.
 - 2.1.12.3. The input field will only accept valid passwords.
 - 2.1.12.3.1. The password must be at least 8 characters.
 - 2.1.12.3.2. The password must be under 24 characters.
 - 2.1.12.3.3. The password must contain at least one special character (!, _ , #, @, >, <, ., (,), &, or ?).
 - 2.1.12.3.4. The password must contain at least one capital and one lower case letter.
 - 2.1.12.4. The input field will only accept the same password as the user-inputted data in the “Password” input field.
 - 2.1.12.5. A “*” symbol will be marked next to the field to indicate a required component.
- 2.1.13. There will be password status text indicating the validity of the password input fields.
 - 2.1.13.1. The default status message is left blank.
 - 2.1.13.2. If both password fields do not match, the status display message will show, “Passwords do not match.”
 - 2.1.13.3. If both passwords match, the status display message will show, “Passwords are matching.”
 - 2.1.13.4. If the input field for “Password” contains an invalid password, the status display message will show “Invalid Password”.
- 2.1.14. A clickable button labeled “Create Profile”.
 - 2.1.14.1. When clicked, the email is checked to make sure that an account is not already created using it.
 - 2.1.14.1.1. If an existing email matches the user-inputted email, a dialog box message will appear, stating “Email already taken”.
 - 2.1.14.1.2. If the email is open, no dialog box will be displayed.
 - 2.1.14.2. When clicked, the two password input fields will be checked to see if both contain the same appropriate input.
 - 2.1.14.2.1. If both passwords are the same and adhere to password requirements, no message will be displayed.
 - 2.1.14.2.2. If both passwords are the same, but do not adhere to password requirements, a dialog box will be displayed, stating, “Invalid Password”.
 - 2.1.14.2.3. If both passwords are different, yet adhere to password requirements, a dialog box will be displayed, stating, “Passwords do not match.”
 - 2.1.14.2.4. If both passwords are different and have invalid input, a dialog box will be displayed, stating, “Invalid Password”.
 - 2.1.14.3. When clicked, if any required fields are empty, a dialog box will display, stating, “Please enter all required material.”
 - 2.1.14.4. When clicked, if there are no status messages and all the fields are filled out, the information is submitted to the back-end.

- 2.1.14.4.1. The user is then logged into their account and the application navigates to the dashboard.

3. (General) Modal

- 3.1. All **modals** will have an “X” button that will close the **modal**.

4. (General) Dialog Box

- 4.1. The dialog box will have a clickable button labeled, “Ok”, that closes the dialog box when clicked.
- 4.2. The dialog box will display a message depending on what has triggered it.

5. (General) New Task Modal

- 5.1. The **modal** should contain a clickable close button indicated with an “X”.
- 5.2. The **modal** should contain a title label, “New Task”.
- 5.3. This **modal** will have text labels for the following:
 - 5.3.1. “Name of the Task:”
 - 5.3.1.1. By this label, a text input field should be set with a placeholder value of “Task Name”.
 - 5.3.1.2. The input field should only accept a minimum of 3 characters and a maximum of 80 characters.
 - 5.3.1.3. The input field should only accept letters, numbers, and specific special characters (:, ", ', (,), -, &, /, !, and .)
 - 5.3.1.4. If the input field contains invalid characters, a message should be displayed below the field stating so, along with the only valid characters allowed.
 - 5.3.2. “What is the date the task should be done by? Or is it recurring on a daily/weekly basis? ”
 - 5.3.2.1. By this label, a text input field should be set with a placeholder value containing the current date in the correct format (Month/Day/Year).
 - 5.3.2.1.1. The input field should only accept dates with the correct format.
 - 5.3.2.1.2. The input field should only accept dates past the current date.
 - 5.3.2.1.3. The input field should only accept letters, numbers, and one special character (/).
 - 5.3.2.1.4. If the input field contains invalid characters or an invalid date format, a message should be displayed below the field stating so, along with the input restrictions and format.
 - 5.3.2.1.5. If the input field contains an invalid date, a message should be displayed below the field stating, “Invalid Date”.
 - 5.3.2.2. Also by this label and field, a clickable option will be labelled as “Daily Recurring”.
 - 5.3.2.3. Also by this label and field, a clickable option will be labelled as “Weekly Recurring”.
 - 5.3.3. “The Task’s Level of Difficulty, On a Scale from 1 (Easy) to 10 (Difficult):”
 - 5.3.3.1. By this label, there should be an input field that should only accept integers between 1-10.
 - 5.3.3.2. If the input field contains invalid input, a message should display beneath the field stating so.

5.3.4. “What is the estimated number of hours it will take to complete this task?”

5.3.4.1. By this label, there should be an input field that should only accept positive integers and should have a maximum of

5.3.5. “What is the minimum amount of time, in minutes, you can work on this task in a day?”

5.3.5.1. By this label, there should be an input field that only accepts positive integers and a number less than the biggest number of minutes the user has allotted for any day.

5.3.5.2. If the input field contains invalid input, a message should display beneath the field stating so.

5.3.6. “What is the maximum amount of time, in minutes, you can work on this task in a day?”

5.3.6.1. By this label, there should be an input field that only accepts positive integers and a number less than the biggest number of minutes the user has allotted for any day.

5.3.7. “Is this a low-priority task, mid-priority task, or high-priority task?”

5.3.7.1. By this label, there should be three clickable and labeled options of High-Priority, Mid-Priority, and Low-priority.

5.3.8. “Are there any subtasks you would like to add?”

5.3.8.1. There should be a clickable button labeled “+” next to this label.

5.3.8.1.1. When clicked, this button should take the user to a copy of the “New Task” screen, where the title is now “**New Subtask**”, and the subtasks input and field are now gone.

5.4. The **modal** will have a clickable button at the bottom of the **New Task** screen labeled “Submit”.

5.4.1. When clicked, the button will validate the data in each field.

5.4.1.1. If there is any invalid data, a dialog box will display, stating “Invalid: Please check your information for the correct formats.”

5.4.1.2. When validated, if there is no dialog box, the new task will be submitted to a database, and the user will be redirected back to the **Tasks** section of the dashboard.

6. (Dashboard) Navbar

6.1. There will be a place to indicate the user’s **Login Streak** points (see sign-in form for additional information on **Login Streaks**).

6.2. There will be a place to include a user avatar/profile picture.

6.2.1. A user’s active profile picture is displayed, depending on the user’s settings.

6.2.2. A default profile picture will be displayed if the user has no active profile picture set within their settings.

6.3. There will be a link labeled “Work Hour History”.

6.3.1. When clicked, this will take the user to the **Work Hour History** section of the dashboard.

6.3.1.1. This section will display a statistical bar graph with “Number of Hours Worked” on one axis, and “Days” on the other.

6.3.1.1.1. The bar graph will be pulled from the user’s data within the database to gather the total number of hours worked on every task within each day of the current week.

6.3.1.1.1.1. Each bar will also be divided into which tasks took up that portion of the bar’s work hours.

6.3.1.1.2. Next to the bar graph will be a clickable button indicating “Year History”

- 6.3.1.1.2.1. When clicked, the graph will switch to a bar graph that shows the hours you've worked over the past year across 12 bars, representing each month.
- 6.3.1.1.2.2. When clicked, the button will switch to a button indicating "This Week".
- 6.4. There will be a link labeled "Mind Your Garden!".
 - 6.4.1. When clicked, this will take the user to the **Game** section of the dashboard (**See item 8**)
- 6.5. There will be a link labeled "Connect with Others".
 - 6.5.1. When clicked, this will take the user to the **Connect with Others** section of the dashboard.
 - 6.5.1.1. This section will display a message that incorporates the user's major, indicating that the following information will lead the user to relevant communities within the user's major.
 - 6.5.1.2. This section will display several names to **Discord** servers that are relevant to the student's major.
 - 6.5.1.2.1. Each name will have a clickable button that will redirect the user to a new tab in their browser to receive the **Discord** invitation.
- 6.6. There will be a link labeled "Profile Settings".
 - 6.6.1. When clicked, this will take the user to the **Profile Settings** section of the dashboard.
 - 6.6.1.1. This section will show the user's current profile settings, including the following:
 - 6.6.1.1.1. The user's first name.
 - 6.6.1.1.1.1. This field will not be editable.
 - 6.6.1.1.2. The user's last name.
 - 6.6.1.1.2.1. This field will not be editable.
 - 6.6.1.1.3. The user's email address.
 - 6.6.1.1.3.1. This field will not be editable.
 - 6.6.1.1.4. The user's current avatar/profile picture resource.
 - 6.6.1.2. A clickable button will be visible—the button will be labeled, "Adjust Info".
 - 6.6.1.2.1. When clicked, a **modal** will appear on screen.
 - 6.6.1.2.1.1. The **modal** should contain a clickable close button indicated with an "X".
 - 6.6.1.2.1.2. This **modal** will have text labels and fields for the following:
 - 6.6.1.2.1.2.1. First Name
 - 6.6.1.2.1.2.1.1. The placeholder should be the user's current first name.
 - 6.6.1.2.1.2.1.2. The input field should only accept letters.
 - 6.6.1.2.1.2.2. Last Name
 - 6.6.1.2.1.2.2.1. The placeholder should be the user's current last name.
 - 6.6.1.2.1.2.2.2. The input field should only accept letters.
 - 6.6.1.2.1.2.3. Email
 - 6.6.1.2.1.2.3.1. The default value should be the user's current email.
 - 6.6.1.2.1.2.3.2. The input field should only accept valid emails.
 - 6.6.1.2.1.2.3.2.1. Emails must contain an @ symbol.
 - 6.6.1.2.1.3. This **modal** will have the current avatar/profile picture resource.
 - 6.6.1.2.1.4. A clickable button should be connected to the resource, labeled "Edit".
 - 6.6.1.2.1.4.1. When clicked, an avatar view should be displayed, showing:
 - 6.6.1.2.1.4.1.1. A label, "Old Profile Picture".
 - 6.6.1.2.1.4.1.2. The current source of the profile picture.

- 6.6.1.2.1.4.1.3. A label, "New Profile Picture".
- 6.6.1.2.1.4.1.4. An image uploader to receive a new source image.
- 6.7. The Navbar will be shown within every section it reroutes the user to, within the dashboard.
 - 6.7.1. When navigating off of the default "Tasks" section, the Navbar will contain a link at the top of the link list labeled, "Tasks".
 - 6.7.1.1. When clicked, this will take the user to the **Tasks** section of the dashboard.

7. (Dashboard) Tasks

- 7.1. The Tasks section of the dashboard will be the default section displayed to the user when coming from the **Sign-In** or **Registration Forms**.
 - 7.1.1. Displayed will be **Daily Tasks** taken from each individual task the user has input from using the **New Task Modal**.
 - 7.1.1.1. There will be a clickable button with a "+" character.
 - 7.1.1.1.1. When clicked, a **New Task modal** will appear on screen.
 - 7.1.2. The tasks should be shown from the stored data of "New Tasks" that the user has input prior.
 - 7.1.2.1. Each task for that day will be determined by the web application.
 - 7.1.2.1.1. The web application will determine the specific tasks available for that day, based on the prior factors entered by the user when creating a "New Task."
 - 7.1.2.1.1.1. If the user has subtasks attributed to the main Task, the program will try and use those instead of the main Task.
 - 7.1.2.1.1.2. The program will also add to the text label of the daily task within the **Tasks** section of the dashboard the amount of time it recommends the user to work on the task within that day.
 - 7.1.2.1.1.3. The program will also determine the amount of **Experience** that the task will give.
 - 7.1.2.1.1.3.1. **Experience** will be determined by the work's level of difficulty and will be awarded on a per-minute basis of work.
 - 7.1.2.1.1.3.1.1. **Experience** totals will round up to the nearest minute.
 - 7.1.2.1.1.3.2. **Experience** is awarded to the user's profile when the user stops the **Timer** and the work time is logged into the database.
 - 7.1.2.1.1.3.3. If the user completes the suggested work time in the **Task**, the program will award additional **Experience** equal to 20% of the **Experience** that the **Task** would give for the recommended time.
 - 7.1.2.2. Each task will have a clickable timer button next to it.
 - 7.1.2.2.1. When clicked, the timer button will begin a visible **Timer**.
 - 7.1.2.2.2. When clicked, the new visible **Timer** will have a clickable button labeled "Stop".
 - 7.1.2.2.2.1. When the "Stop" button is clicked, the **Timer** will pause, displaying the time that the timer stopped at
 - 7.1.2.2.2.1.1. The time that the **Timer** stopped at will be communicated to the database for that specific task, under that specific user.
 - 7.1.2.2.2.1.2. The program will use the time that the **Timer** stopped at to record the amount of **Experience** that the user gets, storing that in the database for the user's account.

- 7.1.2.2.2.1.3. The new time label that is displayed will now have a clickable button labeled “Resume” next to the paused time label.
 - 7.1.2.2.2.1.3.1. When the “Resume” button is clicked, the paused time label will continue, with the “Stop” button reappearing next to the running timer.
 - 7.1.2.2.2.1.3.2. The updated time will be communicated to the database for that specific task when the **Stop** button is clicked.
 - 7.1.2.2.2.1.3.2.1. The program will send additional **Experience** using the original time that the **Resume Timer** started at and the time that the **Resume Timer** will stop at.
- 7.1.2.2.3. The web application will pre-design “breaks” into the total time, with a **dialog box** that indicates the user take a 5-minute break at specific points in the timer.
 - 7.1.2.2.3.1. The **Timer** will continue during this time, with the five minutes already factored into the overall time
 - 7.1.2.2.3.2. The dialog box will run a countdown starting at 5 minutes.
 - 7.1.2.2.3.2.1. Once the 5 minutes finishes, the dialog box will automatically disappear.
- 7.1.2.3. Each task will have a button next to it to indicate an “options” menu.
 - 7.1.2.3.1. The “options” menu will expand into three clickable text labels.
 - 7.1.2.3.1.1. A clickable “Edit” label will be amongst the list within the expanded “options” menu.
 - 7.1.2.3.1.1.1. When clicked, the user will be taken to an **Edit Daily Task** section.
 - 7.1.2.3.1.1.1.1. In this section, there will be a title text label, “Edit Daily Task”
 - 7.1.2.3.1.1.1.2. In this section, there will be a text label, “Hours Worked:”
 - 7.1.2.3.1.1.1.2.1. Next to the “Hours Worked:” label, there will be a field that will only accept positive integers equal to or between 1 and 24.
 - 7.1.2.3.1.1.1.2.2. Should invalid input be entered, a message will appear below the field indicating invalid input.
 - 7.1.2.3.1.1.1.3. In this section, there will be a text label, “Completed?”.
 - 7.1.2.3.1.1.1.3.1. There will be a check box near the “Completed?” label, indicating “yes” if filled.
 - 7.1.2.3.1.1.1.4. In this section, there will be a clickable button labeled, “Submit”.
 - 7.1.2.3.1.1.1.4.1. When clicked, the program will check to ensure the “Hours Worked:” field has valid input in its field. If invalid input is found, a dialog box will display, with the message, “Invalid Hours.”
 - 7.1.2.3.1.1.1.4.2. When clicked, if the “Completed?” is checked, then the program will mark the item as completed for

the day and award the user additional **Experience** equal to 20% of the **Experience** gained from the total work time.

7.1.2.3.1.1.1.4.3. When clicked, the program updates the database and awards (or removes) Experience for the user's account based on the disparity between the old amount of work time and the new amount of work time on the **Task**.

7.1.2.3.1.2. A clickable "Task Completely Finished" label will be amongst the list within the expanded "options" menu.

7.1.2.3.1.2.1. When clicked, the task will be removed from the user's **Tasks** section and updated in the database to reflect its "finished" status.

7.1.2.3.1.2.1.1. This task will no longer show up in the user's daily tasks and **Tasks** section.

7.1.2.3.1.2.1.2. The **Daily Tasks** list will be redone for days after the current day, removing this task from the equation.

7.1.2.3.1.2.2. When clicked, the user will receive an additional amount of **Experience** based on 20% of the **Experience** from the total work time on the task across the task's history.

7.1.2.3.1.3. A clickable "Remove from Today's List" label will be amongst the list within the expanded "options" menu.

7.1.2.3.1.3.1. When clicked, the task will be removed from the user's **Task** section for this day only.

7.1.2.3.1.3.1.1. The web application will re-make the **Daily Tasks** going forward, depending on the amount of time that was left on the current day's task

7.1.3. There should be a clickable button labeled "See All Tasks".

7.1.3.1. When clicked, the user will be taken to the **Total Tasks Layout** section.

7.1.3.1.1. In this section, all the user's input **Tasks** will be shown in a list.

7.1.3.1.1.1. Each task will have a clickable button indicating to "edit" the task

7.1.3.1.1.1.1. When clicked, a copy of the **New Task Modal** will display to the user, replacing the title text header with "Edit Task".

7.1.3.1.1.1.1.1. All the fields and options will be selected/pre-filled with the data already in the database for that task.

7.1.3.1.1.1.1.2. When the user "submits" the data, and if the data is valid, the information in the database will be overwritten with the new data in the **Edit Task Modal**.

7.1.3.1.1.1.1.2.1. The same restrictions on input are explained in the **New Task Modal**.

7.1.3.1.1.2. Each task will have a clickable button indicating to "remove" the task

7.1.3.1.1.2.1. When clicked, the task will be removed from the **Total Tasks Layout** section, the **Tasks** section, and the **Daily Tasks** calculated for days beyond the current day will be re-made.

7.1.3.1.2. In this section, there will be a clickable button labeled "New Task".

- 7.1.3.1.2.1. When clicked, the **New Task Modal** will be brought up for the user to input a new task
- 7.1.3.1.3. In this section, there will be a clickable button labeled “Back to Daily Tasks”.
- 7.1.3.1.3.1. When clicked, the user will be taken back to the **Tasks** section.
- 7.2. The **Tasks** section of the dashboard will have a header label for “Objectives of the Week”
 - 7.2.1. By this label will be a list of text labels
 - 7.2.1.1. Each of these text labels corresponds to each task the application has set for the user to work on within this week.
 - 7.2.1.2. Each of these text labels will have labels indicating the total hours for this week.
 - 7.2.1.3. Should the user complete the total hours for the week on any specific task, the application marks this on the label, and will award the user with additional **Experience** equal to 20% of the **Experience** of that task’s total amount of logged work time.

8. (Dashboard) Game

- 8.1. The **Game Section** will include a header label, “Mind Your Garden!”
 - 8.1.1. The **Game Section** will include a sub header label, “Drag items from the store into your garden to upgrade your experience gain!”
- 8.2. The **Game Section** will include an image divided up into a 64 x 96 grid of squares, with the image oriented in landscape and hereby called the **Game Board**.
 - 8.2.1. The **Game Board** will consist of mainly grass, to simulate a top-down view of a grassy field
- 8.3. The **Game Section** will include a “Store”
 - 8.3.1. The **Store Component** will have a header label of “Store”
 - 8.3.2. The **Store Component** will have a sub header text label of “Single Use Items”
 - 8.3.2.1. Underneath **Single Use Items** will be a select variety of clickable images
 - 8.3.2.1.1. Each of the images will also have a text label denoting the required experience to “Purchase”
 - 8.3.2.1.2. When any of the **Single Use Items** are clicked, the program will first verify that the user has enough total **Experience** in their account
 - 8.3.2.1.2.1. The program will verify that the user has more **Experience** than what is noted by the **Single Use Item**.
 - 8.3.2.1.2.1.1. If the user does not have enough experience, a dialog box will display with a message, “You do not have enough experience!”
 - 8.3.2.1.2.2. If no dialog box occurs, the program will remove the value noted with the **Single Use Item** from the user’s total **Experience**.
 - 8.3.2.1.2.2.1. The program will apply the listed modifier to all gained **Experience**, noting the current day and the last day the benefit will be active.
 - 8.3.2.1.2.2.1.1. For as long as the modifier is active, the program will add the listed single item to the **Active Single Use Items**
 - 8.3.2.2. Underneath each **Single Use Item** will be a text label containing the item’s name.

- 8.3.2.3. Underneath each **Single Use Item** will be a text label containing the item's described benefits (a.k.a: the item's modifier on the user's **Experience** gain), as well as how long the item will last
 - 8.3.2.3.1. Items can range from lasting a day, a week, and a month
- 8.3.3. The **Store Component** will have a sub header text label of "Permanent Items and Decorations!"
 - 8.3.3.1. Underneath the **Permanent Items and Decorations** will be a select variety of draggable images.
 - 8.3.3.1.1. Each of the images will also have a text label denoting the required experience to "Purchase"
 - 8.3.3.1.2. When any of the **Permanent Items and Decorations** are clicked, the program will first verify that the user has enough total Experience in their profile
 - 8.3.3.1.2.1. The program will verify that the user has more **Experience** than what is noted by the **Permanent Item and Decoration**.
 - 8.3.3.1.2.1.1. If the user does not have enough experience, a dialog box will display with a message, "You do not have enough experience!"
 - 8.3.3.1.2.2. If no dialog box occurs, the program will remove the value noted with the **Permanent Item and Decoration** from the user's total **Experience**.
 - 8.3.3.1.2.2.1. The user will be able to click and drag the item from the store and onto the field image with the grid.
 - 8.3.3.1.2.2.2. The program will apply the listed modifier to all gained **Experience**, noting in the database what item is active on the user's account.
 - 8.3.3.2. Underneath each **Permanent Item and Decoration** will be a text label containing the item's name.
 - 8.3.3.3. Underneath each **Permanent Item and Decoration** will be a text label containing the item's described benefits (a.k.a: the item's modifier on the user's **Experience** gain), as well as the number of squares it takes up on the **Game Board** (1 x 1, 2 x 2, 3 x 3, 4 x 4, 5 x 5, etc.)
- 8.3.4. The **Store** will include a sub header label, "Experience:"
 - 8.3.4.1. Near this label, the user's current **Experience** will be noted with a positive integer value
 - 8.3.4.2. Anytime the user buys an item, the **Experience** will be updated, as well as the new total sent to the database
- 8.3.5. The **Store** will include a clickable button labelled "Sell"
 - 8.3.5.1. When clicked, the **Game Board** will go into **Sell Mode**
 - 8.3.5.1.1. When the **Game Board** is in **Sell Mode**, each of the current **Permanent Items and Decorations** on the Game Board will have an "X" symbol over them.
 - 8.3.5.1.1.1. When the symbol is clicked, the item will be removed from the Game Board
 - 8.3.5.1.1.1.1. When removed, the **Experience** required to buy the item will be returned to the user's total **Experience**.

- 8.3.5.1.1.1.2. When removed, the database will be updated to remove the item and its modifier from the user's account and **Experience** gain.
- 8.3.5.1.1.2. When the **Game Board** is in **Sell Mode**, the clickable "Sell" button will change to a button labeled "Exit".
- 8.3.5.1.1.2.1. When clicked, the **Game Board** and **Store** will revert to the state it was in, prior to clicking the "Sell" button.

Updated Timeline

See Figure 3 in the appendix for a detailed description.

Appendix

Figure 1: Team Member Names

Name	Role
Jensen Schmidt	Database Engineer, Architect
Levi Wright	Coding Engineer, Architect
Zach Travis	Web Development, Architect

Figure 2: Tools and Technology*

Place in System	What is the Tech	Why
Frontend	Netbeans IDE	A versatile IDE that can be used to develop JavaScript, Html, and work with CSS
Backend	Firebase Platform	To be able to host the web application site
Backend	Supabase Platform	To store and manipulate data for our users within a database

*Subject to change or addition

Figure 3: Tentative Schedule*

Week 1: January 22 nd —January 28 th	All members: <ul style="list-style-type: none">- Initial proposal
Week 2: January 29 th —February 4 th	All members: <ul style="list-style-type: none">- Draft Requirements and Presentation
Week 3: February 5 th —February 11 th	All members: <ul style="list-style-type: none">- Finish Requirements and Presentation
Week 4: February 12 th —February 18 th	All members: <ul style="list-style-type: none">- Assign research and initial steps- Gather what is needed for the project
Week 5: February 19 th —February 25 th	Levi: Construct “Sign-in Form” Jensen: Construct system database Travis: Develop a standard format for html/site design
Week 6: February 26 th —March 3 rd	Levi: Construct “Registration Form” Jensen: Establish Connection for “Registration Form” and “Sign-In Form” with the system database

	Travis: Continue implementation and design of responsive site design
Week 7 and 8: March 4 th —March 17 th	Levi: Begin creation of Task Logic and its Implementation Jensen: Continue integration of database with system, review code logic for possible security issues Travis: Begin construction of “Dashboard” (Navbar and Tasks) HTML/CSS
Week 9: March 18 th —March 24 th	Levi: work with Jensen to design and implement “Task Statistics” logic Jensen: Continue integration with database and work with Levi on implementation of “Task Statistics” Travis: Create “Statistics” graph section of the site.
Weeks 10, 11, and 12: March 25 th —April 14 th	All Members: <ul style="list-style-type: none"> - Evaluate the project so far; this will be a good point for us to establish a stable version 1.0
Week 13: April 15 th —Finals	All Members: <ul style="list-style-type: none"> - Polish product, final presentation

*Subject to change or addition