**SW Engineering CSC 648-848**

**OnTask**

**Section 4 Team 5**

Rhoanna Perez (Team Lead), Christopher Su (Backend Lead), Jia Li (Frontend Lead)

Nelson Pang (Scrum Master), Seongjung Kim (Github Master), Michael Abolencia

**Milestone 1**

**February 28, 2022**

**History Table**

|  |  |
| --- | --- |
| **First Submission** | **February 28, 2022** |
| **Revisions** | **March 17, 2022** |
| **Last Revision for M5** | **May 13, 2022** |

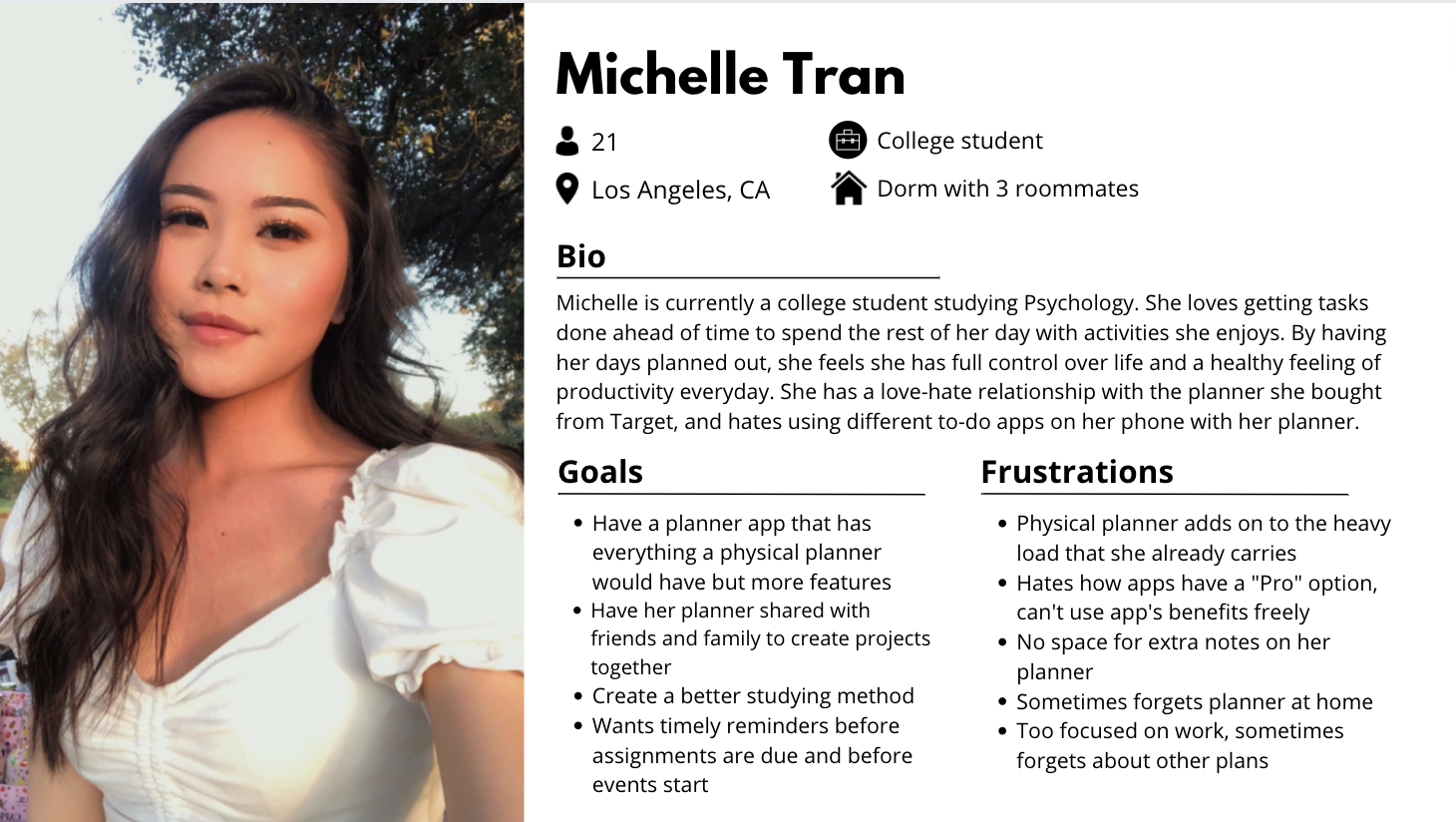
**1. Executive Summary**

Many of us juggle a lot of things in life— work, school, daily chores, appointments, routines, and the list goes on. It’s hard to stick with our schedules, it’s easy to procrastinate, and more often than not, it’s hard to remember and visualize tasks that we have yet to complete in a day. It can be overwhelming to have to keep a mental note of the things that need to be done, especially if many tasks have deadlines attached to them. It’s important to have a list of things to do to visualize. About 76% of US citizens rely on to-do lists to help them throughout the day, and to-do lists have been psychologically proven to reduce anxiety, even just by making a plan to get them done. In the digital world, keeping a list of tasks on a piece of paper might be hard to keep track of, and OnTask is one of the easier and user-friendly alternatives to help you. OnTask is an application that can make sure you stay on task and organize your tasks to keep your day manageable.

OnTask is an easy-to-use task manager app that allows everyone to keep their tasks on one app and organize the things they need to get done in a timely manner. Users can prioritize tasks, categorize tasks, add deadlines, all to make sure that they’re organized for users to view every day. OnTask allows users to share to-do lists, copy to-do lists, and view to-do lists with other users, making it easier for teachers, students, employers, employees, and teams to visualize what work needs to get done. OnTask will make sure you get your work done without procrastinating. OnTask will send reminders before the task deadline creeps up, making sure you mark your task as done without waiting for you to cram your work right before it’s due. The best unique part about OnTask is the ability of users to use the Pomodoro Technique, a technique to help users work and/or study efficiently through working in intervals and prioritizing a break in between. This function allows users to prevent themselves from cramming all their work in one day and ensures you take a well-deserved break. The goal of OnTask is to make sure you’ve successfully completed your productive routine, including breaks for yourself, leaving you satisfied at the end of the day.

**2. Personas and User Stories**

**P.1: Michelle Tran, Student**



**U.1: Michelle Tran, Student**

As a student, Michelle wants to have alarms for set interval studies so that she won’t forget to submit and complete her class assignments on time.

Constraints:

* Users only need to open one application for all their productivity.
* App should coordinate with user’s timezone when sending out reminders and notification
* Easy to use and easy to plan
* Connecting over network: 5G, 4G, 3G, 2.5G, Wifi, etc
* App must have a user ID for each user to create and share one’s weekly activities to another, either by adding them as a friend using the user ID, or just the user ID by itself.
* A clock icon that acts as an alarm button which engages one mouse click. Once the alarm has been engaged, the best study-to-break interval alarm will start running. This study-to-break interval will be easily adjustable to user as well as the recommended settings

**P.2: Crystal Hyun, Nurse**



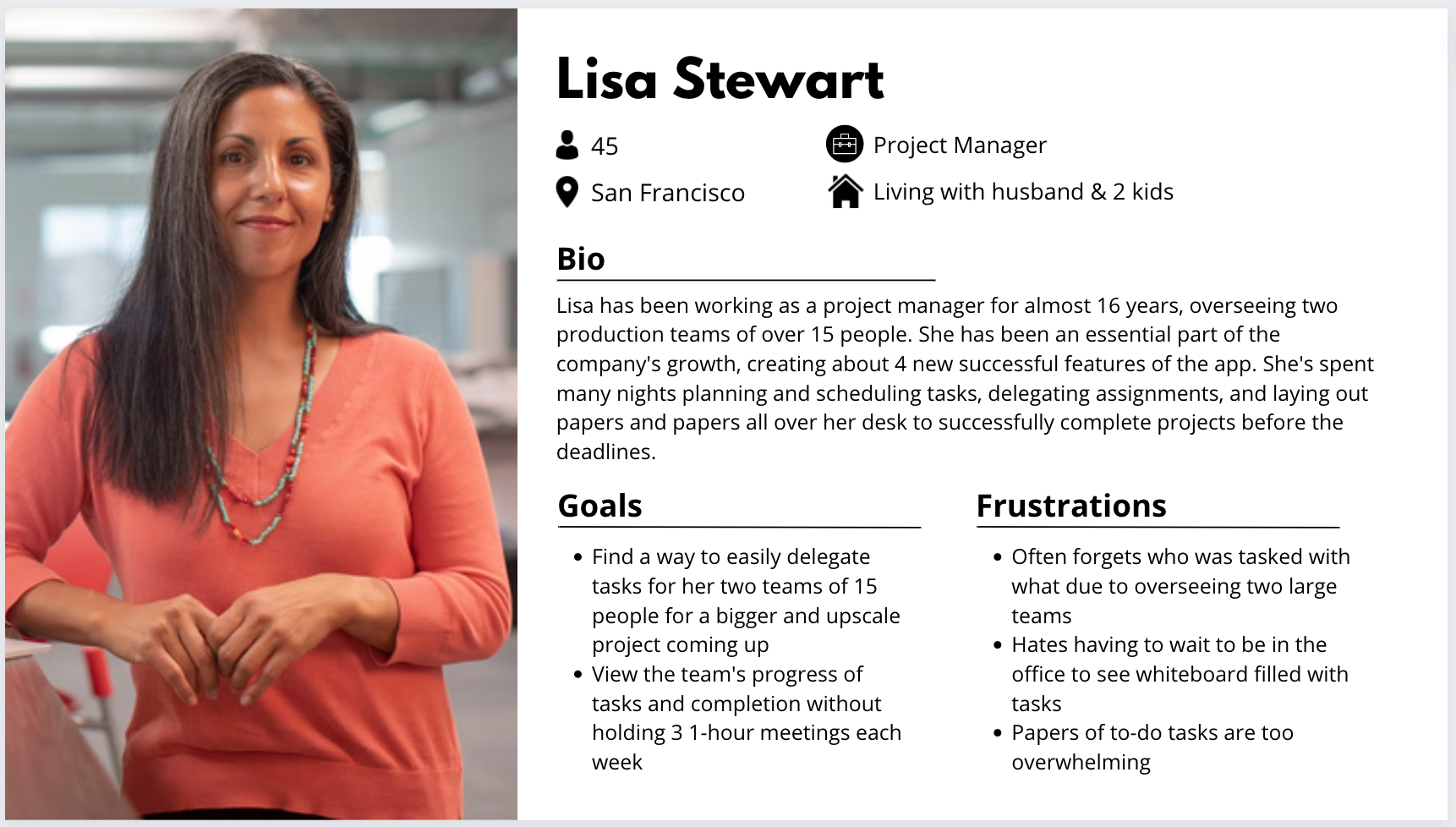
**U.3: Crystal Hyun, Nurse**

As a mom and a nurse, Crystal would like to collaborate on todolists so that she will be able to better organize what she needs around the house with her family.

Constraints:

* Share option to allow for collaboration
* Simple User Flow to help those unfamiliar with technology navigate through the app
* User should be able to add to other users task lists

**P.3: Lisa Stewart, Project Manager**



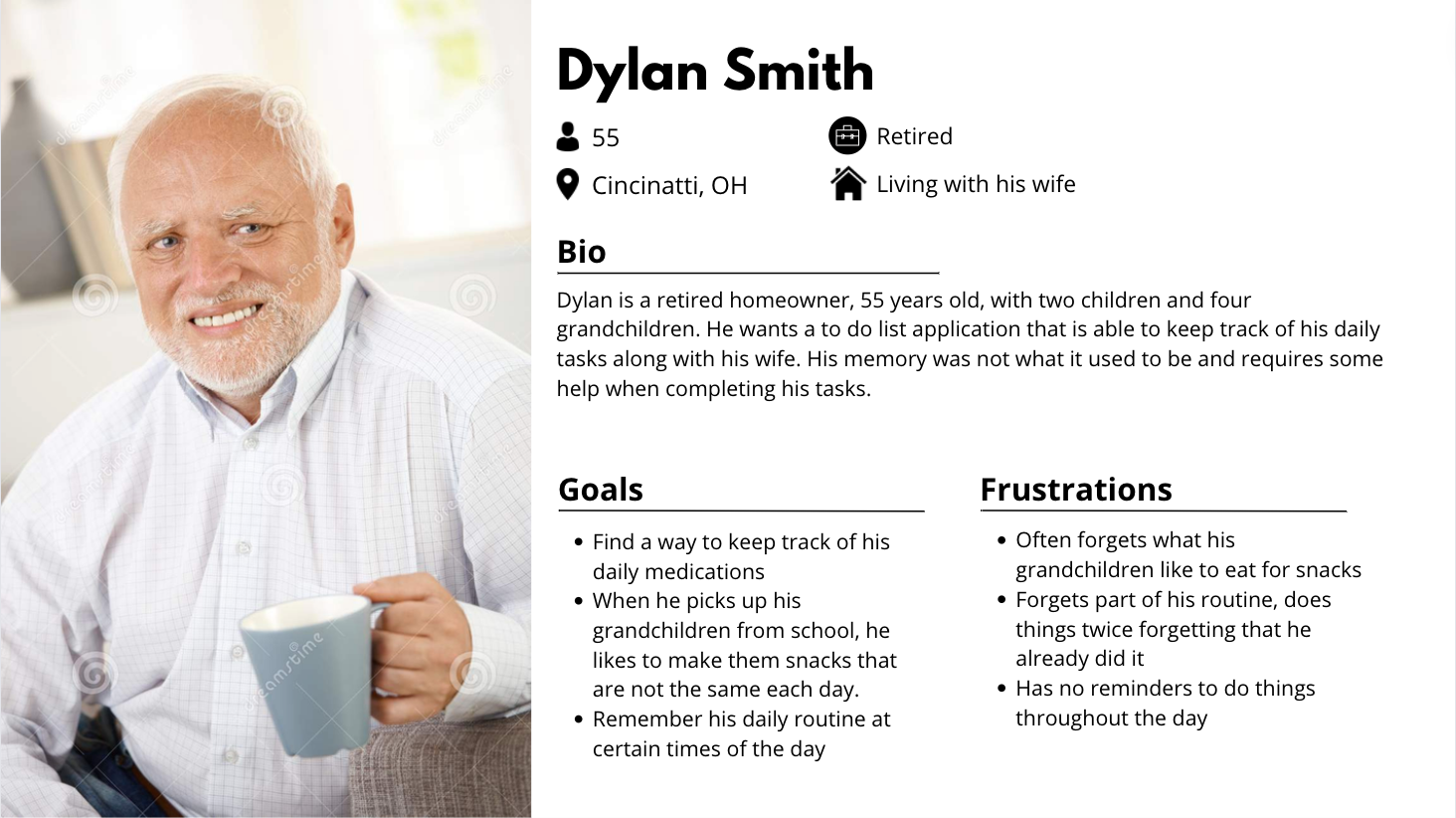
**U.3: User Story, Project Manager**

As a project manager, Lisa would like to share a digital to-do list with her two teams to better delegate tasks for each of her teammates and view their progress so that she won’t have to conduct 1-on-1 meetings with each of her teammates to understand their path to completion.

Constraints:

* Share button that allows her to invite members using their emails to edit/view a to-do list
* Option to assign tasks to each member involved in to-do list
* User should be able to access to-do list and view it in list format or board format
* User should be able to mark tasks as done
* Owner of to-do list should be able to be notified when the task is marked complete by assigned team member

**P.4: Dylan Smith, Retired**



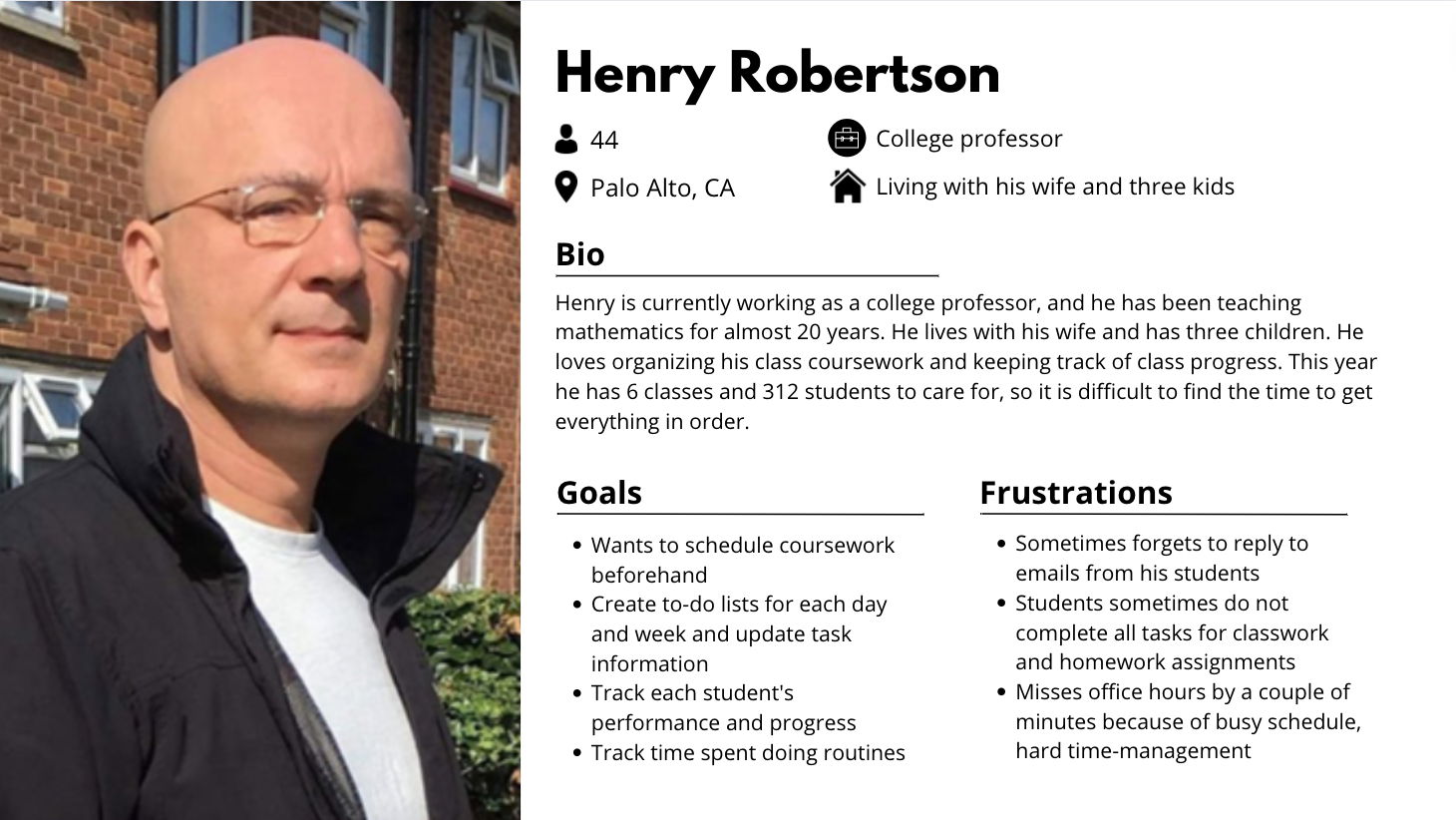
**U.4: User Story, Retired**

As an elderly, Dylan would like reminders to keep track of his medicine so that he won’t forget what his routine is each day.

Constraints:

* Notifications to remind to do tasks
* User should be able to invite other family members to view and edit
* User wants to be able to customize the view on the to-do list app.
* User should be able to mark tasks as completed or not
* User wants accessibility such as IOS talk over and magnifying.

**P.5: Henry Robertson, Teacher**



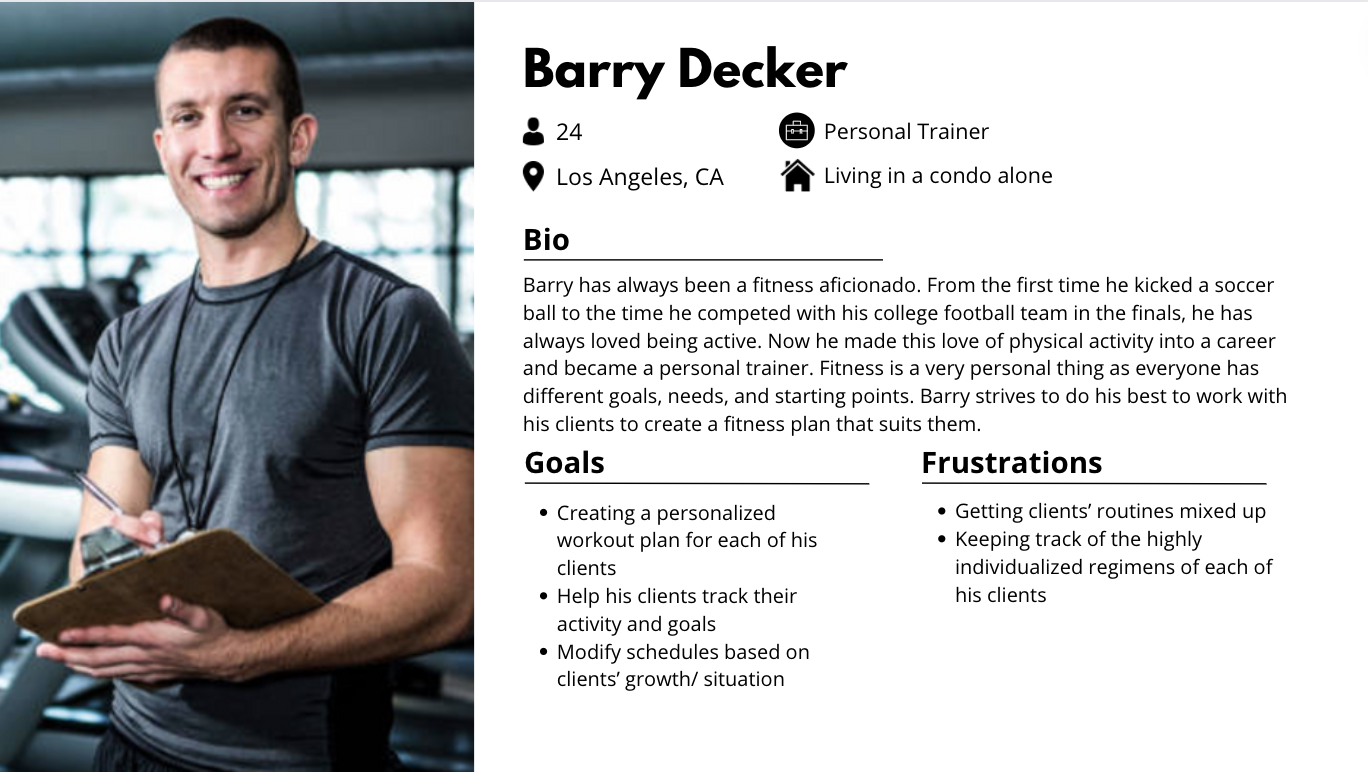
**U.5: User Story, Professor**

Henry would like to share a to-do list with his students so that he can easily distribute weekly tasks and assignments.

Constraints:

* When a user receives an email or comment, a notification pop up is shown.
* When a user sets a due time for a task, the app automatically sets a reminder and sends the user a reminder 1 hour beforehand.
* User should be able to invite other users to the user’s project and it could be shared
* Users should be able to check on approaching deadlines, easily create or reschedule tasks.

**P.4 Barry Decker, Personal Trainer**



**U.6 User story, Personal Trainer:**

As a personal trainer, Barry would like to have a list of exercises to share with his clients to help him coordinate with others to assign a client a set of exercises to do and check their progress on a weekly basis.

Constraints

* User should be able to assign tasks to others
* User should be able to mark tasks as complete
* In the case of a shared list, others should see which tasks have been marked as complete
* User should be able to add on to list based on permissions (Read only or Read and Write)
* Users should be able to leave notes/feedback on tasks that can be seen by others.

**3. Data Definitions**

|  |  |  |
| --- | --- | --- |
| **Name** | **Definition** | **Usage** |
| Task | Individual tasks created by the user in the to do list, containing the date, title, category, and priority | Users create tasks that they can complete and see in a list format |
| To do list | A list of tasks that a user creates | Student users have the ability to use collaborative to-do lists set upon their **teacher**, given a user ID |
| Date | The day that a user assigned a task | Users can see when they created a task |
| Title | The name of a user’s task (ex: Finish assignment 1 for CSC 413) | Users can view a list of tasks that they’ve titled themselves |
| Category | A group that a user creates for each task (ex: categorizing a task under **exams**, **assignments, reminders**) | Allows user to view tasks by category |
| Priority | The importance of a task established by the user (ex: Priority 1 for assignment due tomorrow) | Allows users to view tasks by priority (1st priority first to 3rd priority last) |
| Deadline | The due date that a user defines for each task (ex: Finish assignment due 3/19) | Allows users to view tasks in chronological order of deadline |
| Collaborative to-do list | A to-do list that is shared among other users | Keeps track of tasks between users |
| Creator | The person that creates a to-do list, term commonly used for collaborative to-do lists | Specifies who created the to-do list |
| Viewer | A person who can view a to-do list, often shared from a collaborative to-do list | Shows who can view a creator’s list. Creator can specify who can edit. |

**4. Initial list of functional requirements**

**F.1: Sign-in/Login/Logout, Priority 1:**

* Users can sign up for accounts and set it up based on what they need it for.

**F.2: Login/Logout, Priority 1:**

* Users are able to log in and log out through their accounts.
* All users within our list of personas will be able to create an account as a way to register with the to-do list application, and will be able to login and logout.

**F.3: Categorize Tasks, Priority 3:**

* All users within our list of personas will be able to create a to-do list and categorize each task into different categories. The user can create their own categories and filter them out into each. In the different view layouts, users can sort their tasks by the categories.
* Students like Michelle are able to determine what categories tasks can be sorted into such as school work or feeding their pets which can be categorized into home necessities.

**F.4: Assign Priority, Priority 2:**

* A user is able to assign different importances to their to-do list items.
* Users can see tasks based on importance (ex: Priority 1 at the top of the list, Priority 3 at the end of the list)

**F.5: Add a Friend, Priority 3:**

* Users will be able to add a friend through a user’s registered email. A user will invite and request another user to become a friend, and if they would like to be friends, they can accept the invite.
* A friends list will allow users to easily share a to-do list with a friend and invite a friend to either view/edit a to-do list.
* Clicking onto a friend’s icon will bring the user to their friend’s tasklist/calendar. Tasks that are created private will not be visible unless changed by the owner. **F.5.1-Priority 3**

**F.6: Creating a Collaborative To-do List, Priority 1:**

* Users will be able to create a collaborative to-do list. As opposed to having your own to-do list (which is private and only viewed by the owner), a collaborative to-do list allows a user to invite other users to a list that they can edit and view.
  + In order for a user to be added to the collaborative to-do list, user A needs to invite user B to the to-do list. User B needs to accept an invitation to edit/view. If they do not want to be added to the list, they can decline the invitation.
* A collaborative to-do list will allow users to assign tasks to other users and view what collaborative tasks need to be completed by all users who can view the list.
  + When adding a task, a user will have the option to name the task and assign the task to another user within the collaborative to-do list.
* Once a user is assigned to a particular task, the user will be able to see the task on their personal dashboard as a list things to complete. **F.6.1-Priority 2**

**F.7: Notification Pop-Up, Priority 3:**

* When the project task is created or deleted. **F.7.1-Priority 3**
* When someone has sent a comment or notes to a user. **F.6.2-Priority 3**
* When someone has assigned a task or project. **F.7.3-Priority 3**
* When a user has received email or comments. **F.7.4-Priority 3**
* When someone accepts or declines others invitation to join a project **F.7.5-Priority 3**
* When a task that a user has either created or assigned is completed. **F.7.6-Priority 3**
* When a task that a user has either created or assigned is uncompleted. **F.7.7-Priority 3**
* If the user sets a due date and time, notification pop up will be shown before the due date. **F.7.8-Priority 3**

**F.8: Add Notes to Tasks, Priority 1:**

* Users can save all the relevant task information right inside user’s task **F.7-Priority 1**
* Users can add notes for collaborative to-do lists for themselves or for others to see **F.7-Priority 1**

**F.9: Mark Task As Complete, Priority 1:**

* Users can mark a task as complete by clicking the checklist **F.8-Priority 1**
* After marking it as complete, the task will be crossed off their to-do list and will be removed, showing the rest of the tasks that are yet to be completed **F.8-Priority 1**

**F.10: Work-Study Intervals, Priority 1:**

* User sets the amount of time they want to study and decide how long they’d like to study, and how long they’d like to have a break.
* With the information given by the user, the app will set a timer for the user to study.
* Once the timer goes off, it’ll allow the user to take a break for the designated amount they initially requested.
* The intervals will continue until the set amount of time the user wants to study will end. This will allow students like Michelle, teachers like Henry, project managers like Lisa, to work and study efficiently and be able to take adequate breaks throughout the day.

**5. List of non-functional requirements**

**Compatibility:**

* The application should be compatible with Chrome, Safari, Mozilla Firefox, Microsoft Edge
* The application should be built with full mobile responsiveness in mind, employing a mobile-first mindset since most users are expected to be using the todo-lists “on the go”

**Storage Space:**

* The application will be hosted on Amazon AWS
* The database will be hosted on Atlas using a cloud MongoDB database

**Usability/Product Requirements:**

* Application should be straightforward and intuitive
* Usability studies should be used to make sure the application is easy to use to the general public

**External Requirements:**

* Security: The application will run under https at all times in order to keep connections secure
  + Login information will be hashed and taken care of to maintain user confidentiality.
* Ethicality: User information and confidentiality will be protected and users will be notified on how their data is being used
  + The application will follow accessibility guidelines and be developed with inclusivity in mind
  + Application will leverage off-line first capabilities to help those without access to stable connection

**Development Requirements:**

* Github: Code will be well maintained and documented
  + Code should be developed in such a way that allows for future expansion
* Figma: Designs should be well documented and easy for other developers to understand designs
  + A design system should be implemented to keep both developers and designers on the same page

**6. Competitive Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Features** | **On Task** | **Asana** | **Todoist** | **Notion** | **Google Calendar** |
| **Checklist** | yes | yes | yes | yes | no |
| **Calendar** | yes | yes | yes | yes | yes |
| **Planner** | yes | yes | yes | yes | yes |
| **Assign deadline** | yes | yes | yes | yes | no |
| **Collaborative** | yes | yes | yes | yes (Premium plan) | no |
| **Reminders** | yes | yes | yes (Reminders are only available on Pro plans) | yes | yes |
| **Notes** | yes | yes | yes | no | no |
| **List** | yes | yes | yes | yes | no |
| **Accessibility** | yes | yes | yes | yes |  |
| **Study/work interval** | yes | no | no | no | no |

After taking a look at the table of features that other apps have like Asana, Todoist, Notion, and Google Calendar, there are varying features that aren’t provided for every single application like OnTask does. Apps, like Todoist and Notion, have certain features that aren’t available for free users, and are only for those who pay for Pro/Premium plans. Because of that, it prevents other users from using the app to it’s fullest potential, sometimes having to use other apps to use the feature for free on another app. What makes OnTask different is having every feature available for the user to have access to in one place. All of these to-do list apps will allow users to understand and visualize what tasks they have to complete, but OnTask will make sure that users complete the task on time, by giving reminders, notifications, and having a study-work interval that allows users to plan the times they can work or study for. Our app will allow others to work and study in intervals, then allow the user to take a break after a long period of studying.

**7. High-level system requirements**

Server Host: Amazon Web Services EC2 8 GIB

Operating System: Ubuntu 20.04

Database: MongoDB Atlas

Web Server: Apache Server 2.4.52

Server-Side Language: NodeJS

Web Application Framework: Express

Other Tools: Figma

**8. Team**

* Rhoanna Perez, Team Lead
* Christopher Su, Backend Lead
* Jia Li, Frontend Lead
* Nelson Pang, Scrum Master
* Seongjung Kim, Github Master

**9. Checklist**

* Team found a time slot to meet outside of the class. **DONE**
* Scrum Master shares meeting minutes with everyone after each meeting. **DONE**
* Github master chosen. **DONE**
* Everyone sets up their local development environment from the team’s git repo. **DONE**
* Team decided and agreed together on using the listed SW tools and deployment server **DONE**
* Team ready and able to use the chosen back/front-end frameworks. **DONE**
* Team lead ensured that all team members read the final M1 and agree/understand it before submission **DONE**