Learn-By-ToDoing

Aiden Rodriguez
Brandon Powell
Dylan Watanabe
Martin Rodriguez

Created On: 11/21/2024
Last Updated: 3/11/2025
https://github.com/Bpowell5184/ToDoList-308

Product Vision and Scope

a. Overview

Organization is a large problem for a lot of people. Students, parents, teachers, managers, and many others have to spend their time effectively in order to accomplish their tasks and manage their responsibilities. Most people struggle to fit in their own responsibilities with their necessary work, along with their hobbies, and Learn-By-ToDoing aims to help alleviate this problem.

By providing a user with a way to list all of their tasks into one easy application, people can manage their own time effectively. The calendar view functionality can help a user visualize their week and their obligations in an easy to see way, and the filtering and sorting functionality can help a user prioritize tasks that may be important to finish immediately.

The tag functionality can help a user to give their own modifiers to filter by to further add some customization to their list. Users can add, remove, and edit tasks on their to do list to help with their organization.

b. User Personas

ROMAN'S PERSONA TEMPLATE

romanpichler

PICTURE & NAME	DETAILS	GOAL
What does the persona look like? How is it called? Choose a realistic and believable picture and name.	What are the persona's relevant characteristics? For instance, demographics such as age, gender, occupation, and income; psychographics including lifestyle, social class, and personality; behavioural attributes like usage patterns, attitudes, and brand loyalty. Only state details that are relevant.	What problem does the persona want to solve or which benefit does the character want to gain? Why would the persona use or buy the product? If you identify more than one goal, prioritise them and state the main goal at the top.
Johnathan 	Johnathan is a college student who struggles to keep organized. He has many classes, events, and activities that he needs to attend, but struggles to manage them all. He is 22 years old, male, and just wants to get through his college classes. He has a part time job at Starbucks as well, further digging into his time.	He struggles with organization and wants to keep better track of his commitments and responsibilities. His grades seem to slip, not because he isn't capable, but he has trouble keeping track of the due dates and various commitments he has. With the Learn-By-ToDoing application, his organization would be able to increase greatly, and he would be able to keep track of his responsibilities visually through the app, especially through the calendar view function.

ROMAN'S PERSONA TEMPLATE

romanpichler

PICTURE & NAME	DETAILS	GOAL
What does the persona look like? How is it called? Choose a realistic and believable picture and name.	What are the persona's relevant characteristics? For instance, demographics such as age, gender, occupation, and income; psychographics including lifestyle, social class, and personality; behavioural attributes like usage patterns, attitudes, and brand loyalty. Only state details that are relevant.	What problem does the persona want to solve or which benefit does the character want to gain? Why would the persona use or buy the product? If you identify more than one goal, prioritise them and state the main goal at the top.
Karen 	Karen is a mom who has trouble dealing with her 3 kids. She has trouble juggling dealing with them, as well as her job in the local kindergarten. Sometimes Karen needs to hire a babysitter for her children, as there are times when work obligations make it neccisary to hire a babysitter.	Karen wants to stay organized and see what times she needs to do what to see when exactly she would need to hire a babysitter because of her unavailability. Learn-By-ToDoing helps with this as being able to map her various chores, duties, and job she can decide when a good idea to hire a babysitter may be. It helps her to see her availability, or lack of availability throughout the week.

c. Stories/Requirements

- **Functional**:
    - Add Tasks
        - As a student, I want to add tasks to my to do list tracker so I can stay accountable and remember what homework I need to complete.
    - Complete Tasks
        - As a student I want to be able to complete tasks on my list so that I can see my accomplishments.
    - Delete Tasks
        - As a student I want to be able to delete tasks from my list in case I accidentally add a task or if I decide not to do a task.
    - Edit Tasks
        - As a student I want to be able to edit my tasks incase due dates change, or I make a mistake when making the task.
    - Task Point Value
        - As a student, I want to be able to assign a point value to each task so that I can choose to prioritize certain tasks over others.
    - Sort By Functionality
        - As a student, I want to be able to change the priority of the list by date and point value so that I can keep my schedule organized and do things in the order they are due.
    - Login Functionality
        - As a student, I want to be able to login to my account so that I can see my to-do list, so that I can keep my tasks private and secure.
    - Calendar View Functionality
        - As a student, I want a to do list that conveniently organizes tasks in a calendar/modular view.
    - Filter By Functionality

        - As a student, I want to be able to filter my tasks by completed and pendings so that I can look at pending tasks more easily.
    - Tags Functionality
        - As a student, I want to be able to put tags on tasks so that I can separate educational tasks from personal tasks.
- **Non Functional**:
    - Login Performance
        - A user should be able to login to their account within 1.5 seconds of clicking the “Login” button 90% of the time. A user should be able to login within 3 seconds 100% of the time.
    - Monitor Display
        - A user should be able to use the application regardless of the size of their monitor. The application should be usable on large and small monitors alike. Additionally, browsers such as chrome, firefox, and opera will be supported.

d. Non-Goals or Out of Scope

- Gamification of the program
    - Originally intended to be a core aspect of the program, but now is just the point value associated with the task. Gamification would likely be distracting from the work on and share currently. It is meant to make a singular person’s life easier to keep track of, and take care of their obligations. This may be considered in a future quarter, but for now is not in scope, and there are more pressing goals to attend to. We may come back to this once we have finished other features, however this is fairly low on the priority list currently as we work to implement more primary and core features.

e. Future Goals / Unaccomplished goals

- Sharing of tasks between people

    - Originally thought about as a concept, but deemed also out of scope of the program during this quarter. This program is not meant for groups of people to organize together, but rather for individuals to organize themselves
- Canvas API
    - Originally meant to be implemented, but ran out of time during sprint #8. Was meant to be a big feature, but was ultimately decided it was better to have a full working program, than a buggy API implementation.

3. Design (Approaching the Solution Space)

Tech Stack:

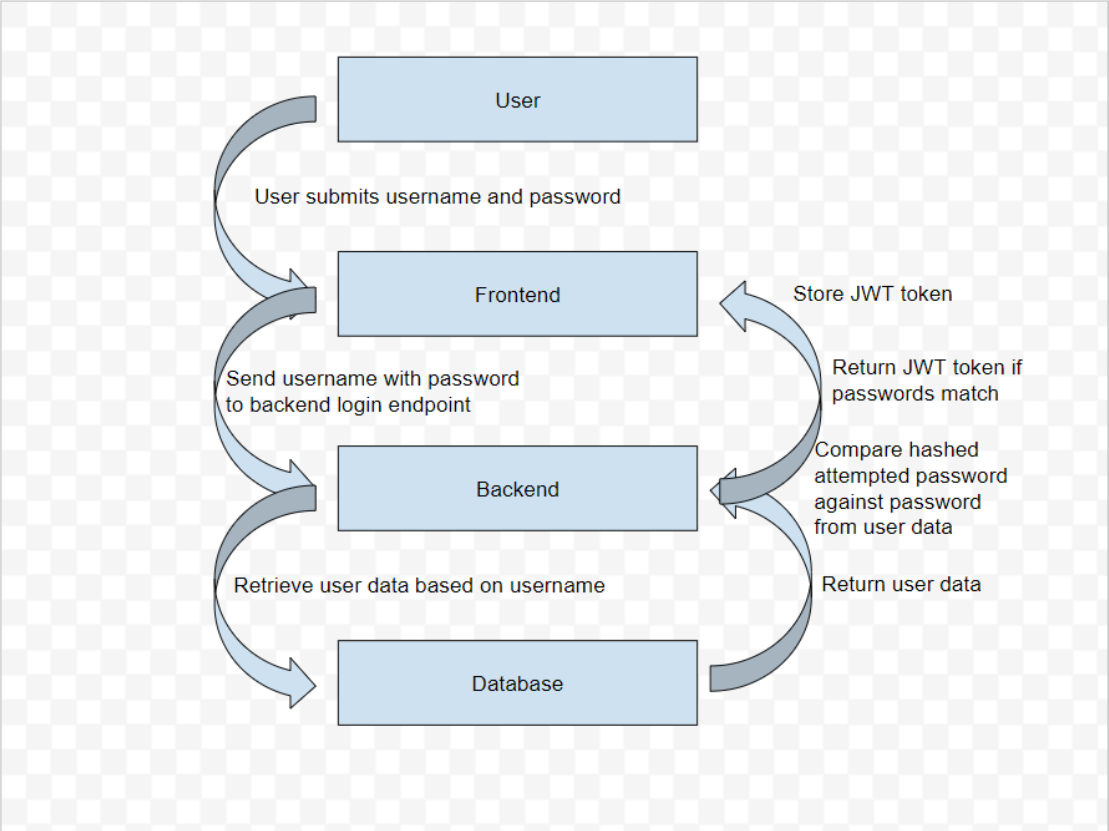
Frontend, Backend, Database Interaction:

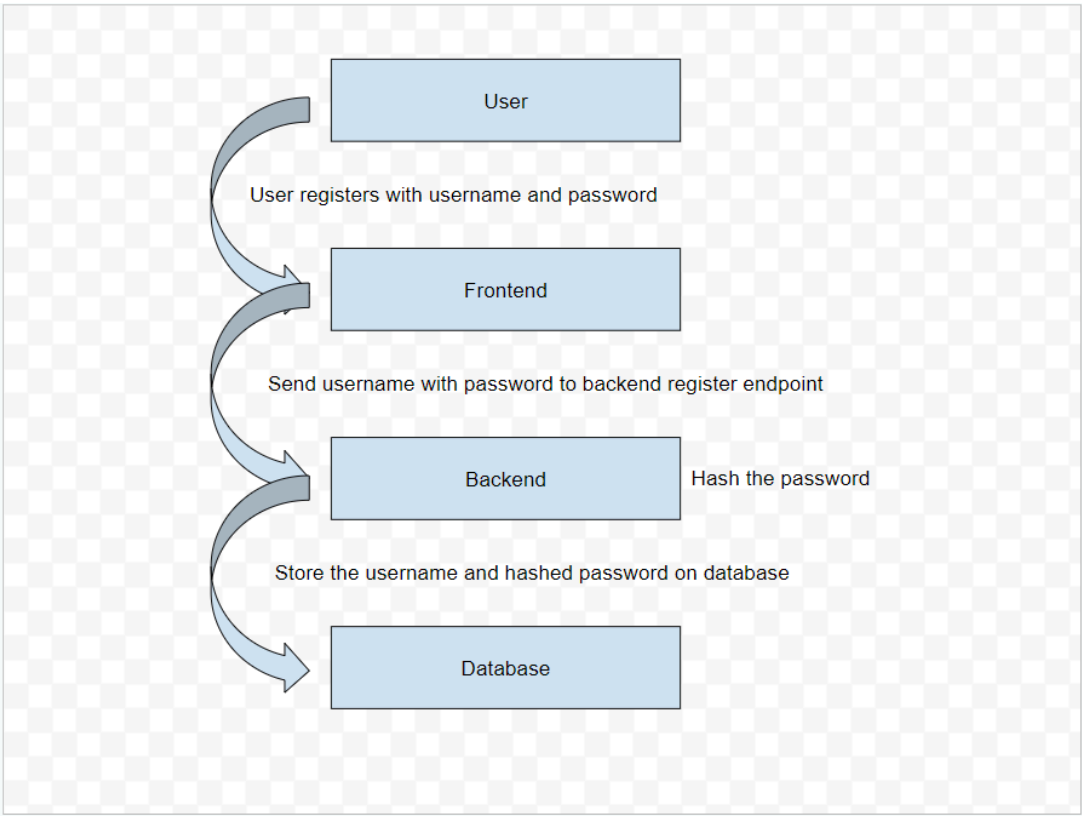
The database stores users, which has the required login information, along with tasks and their userId. Within the backend, there exists the schema for tasks, for which it will read, write, and add tasks to the database. The backend links to the frontend, giving the user's information, confirming or denying login, and most importantly providing the list information of the user from the backend.

The database being used is a cloud mongo database, that way we can all have access to the same remote database while working on the project. Mongoose is installed in the backend to interact with the cloud database, and the backend sends that information. Axios is installed on the frontend to send and receive information to and from the backend easily. Javascript is being used on both the frontend and backend, and css is also used on the frontend.

• UI Prototype:
https://www.figma.com/design/Fo3GVc4iEkTNKzMTFA4MtO/Learn-By-To-Doing?node-id=0-1&node-type=canvas&t=1KxqbofC8QSIlY73-0

• Data Schema/Model.
https://lucid.app/lucidchart/148f21a7-ac0c-46a6-b969-617cd685dbf4/edit?viewport_loc=-518%2C212%2C2960%2C1295%2C0_0&invitationId=inv_fba16893-43c7-413d-965a-624ae90ec543


User
User submits username and password
Frontend
Store JWT token
Send username with password
to backend login endpoint
Return JWT token if
passwords match
Backend
Compare hashed
attempted password
against password
from user data
Retrieve user data based on username
Return user data
Database


User
User registers with username and password
Frontend
Send username with password to backend register endpoint
Backend
Hash the password
Store the username and hashed password on database
Database

The entity sets are as follows:

Data, which contains the list of users, User, which stores data of said user, List, which has information of a list, and contains the Ids of Tasks, which are the main datatype. Tasks have many attributes, such as date, points, title, etc. Finally, the last set is Tag, which can be on a task, and can be another thing that can be filtered by.

• Entity sets:
- Data
  - UserList
- Users
  - UserName
  - Password
  - UserId
  - ListId
- List
  - TaskList
  - ListId
  - UserId
- Task
  - Title
  - Description
  - CreationDate
  - DueDate
  - Points
  - Priority
  - TagsList
  - CompletionStatus
  - TaskId
- Tag
  - TagStr
  - TagId

• Relationship sets:
- Data is made up of users
  - One to many relationship
- Users are included within data
  - One to one relationship
- Users have a task list
  - One to one relationship

- One list is made from one user
    - One to one relationship
- Lists are comprised of many tasks
    - Zero to many relationship
- A task can only be in one list
    - One to one relationship
- Tasks can have tags
    - Zero to many relationship
- A tag can be used to describe multiple tasks
    - Zero to many relationship

4. Deliberation

a. Discussion and Open Questions

Gamification aspect: Initially was a big part of the program but kind of dissolved. Focus is now more on the efficiency of the user using the list. May be implemented in the future, but with limited scope as to keep a user focused on staying organized.

Should a mobile implementation be offered as well?

• Questions about things you do not know the answers to. These may include aspects of the problem you don't know how to resolve yet.

How can we expand upon this program further past what we have stated earlier? What add ons would help a user go through this process more efficiently?