Milestone 2

PREPARED FOR
SW Engineering CSC648 Section-01,

PREPARED BY
Team 04 - Doseedo
Yuto Mori - Team Leader,
Aleia Natividad - Back-End Lead,
Paige Hodgkinson - Scrum Master,
Carlos Posadas - Front-End Lead,
Yakoub Alkabsh - Front-End Lead,
Wing Lee - Github Master

March 18, 2024

1. Data Definitions V2

- **Table User** - Holds all the User's personal information.
    - **id** - to identify our User across all tables
    - **first_name** - first name of the user
    - **last_name** - last name of the user
    - **email** - email for the user
    - **password** -password for account verification
    - **address_1** - address of the user
    - **address_2** - additional address information (ex/ apt, suite, etc..)
    - **state** - state of the user's address
    - **city** - city of the user's address
    - **zip_code** - zip code of the user's address
    - **phone** - phone number of the user. For caregivers to keep track of patient info, and emergency Alerts will be sent directly to their phone via text

- **Table Account** - This holds the type of the account that the user holds, each with different permissions.
    - **user_id** - is associated with id from the User table to link them
    - **permission_number** - to determine which functions are accessible to
    - **account_type**
        - Caregiver - Holds information on patient data, medication, dosage, and reminders. They hold administrative permissions over patient medication cycles and schedules.
        - Patient - Holds information on medical records and medical reminder completions

- **Table AccountLink** - This links the the caregiver users to their patient users via their User ids
    - **caregiver_id** - to establish which User is a caregiver for the associated patient
    - **patient_id** - to establish which User is the patient to the associated caregiver

- **Table Session** - Keeps track of when a user has logged on and logged off, which device they're on , and when the last time they accessed their account via the same session.
    - **id** - to uniquely identify each session
    - **user_id** - associated with id from the User table
    - **device** - to ensure singleton for device login
    - **recently_accessed** - when was the last time the session was active
    - **login_time** - what time the user logged in
    - **logout_time** - what time the user logged out

- **Table Alert** - Similar to a reminder or notification that will eventually be sent to a User
    - **id** - to uniquely identify each alert across active and inactive (backlogged) data
    - **alert_name** - a label for an Alert
    - **sender** - If this alert is being sent from a user via User id
    - **receiver** -Indicates who will be receiving the alert via User id
    - **priority** - An alert can be low, medium, or high priority. This is mainly for the front end, so we know what color and alert sound to make
    - **alert_type** - to indicate the type of alert (picture gram, medication, etc) via the Alert_Types table
    - **prescription_id** - Accesses detailed prescription information from Prescriptions table
    - **send_time** - indicates the time for the alert to be sent
    - **reminder_interval** - set time interval for reminders in minutes
    - **max_send** - the maximum amount of times an alert can be sent
    - **is_active** - Indicates whether the Alert is still active
    - **attachment** - to link if contains picture or other large files in the alert

- **Table Alert_Types** - There can be different alert types such as a medication reminder, an emergency message, or a photogram notification
    - **id** - to uniquely identify alert types and will be referenced in the alert able
    - **type** - to describe the alert

- **Table Prescription** - Holds all information on a User's medication
    - **id** - to uniquely identify each medication
    - **user_id** - The associated User with this medication
    - **med_name** - for the user friendly text name of medication ie "warfarin"
    - **description** - A place for the User to put important notes or other info like with water, interval, side effects, etc
    - **dosage_amt** - the number value such as 5 in "5 mg"
    - **med_type** - to indicate the form such as powder pill etc
    - **dose_unit** - for the units of measure like mg, g, ml, etc
    - **start_date** - date to indicate when to start taking medication
    - **end_date** - to indicate stop date (ex/ for short term mediations)
    - **refill_date** - the refill by date indicated on their prescription bottle
    - **doctor_first_name** - Prescribed doctor's first name
    - **doctor_last_name** - Prescribed doctor's last name
    - **doctor_phone** - Prescribed doctor's phone number

- **Table AlertHistory** - Keeps track of any alert/notification that has been sent
    - **id** - to uniquely identify each alert in storage
    - **alert_id -** Indicates which alert we are referencing
    - **sender** - If the patient or caregiver is the one sending the alert, their User id will be here
    - **receiver** - If the patient or caregiver is the one receiving the alert, their User id will be here
    - **create_date** - Date and time the Alert was sent

- **Table Picture -** Keeps track of all the photos that are sent
    - **id** - to uniquely identify each photo in the db
    - **sender** - to display picture sender info to recipient
    - **receiver** - to allow access to photo history for patient
    - **time_sent** - the time the photo was sent
    - **full_path** - We will store photos in AWS S3, and this will be pathing information to retrieve the photo

- **Table Permissions** - A list of different permissions that are allowed for a User/Account. Being able to edit, add, or delete medications or reminders are types of permissions
    - **id** - to uniquely identify each permission type
    - **permission_name** - to describe the permission

- **Table User_Permissions -** Keeps track of which users has which permissions
    - **id** - to uniquely identify the user permission
    - **permission_id** - to store their permissions
    - **user_id** - to associate their user id with their permissions

2. Functional Requirements V2

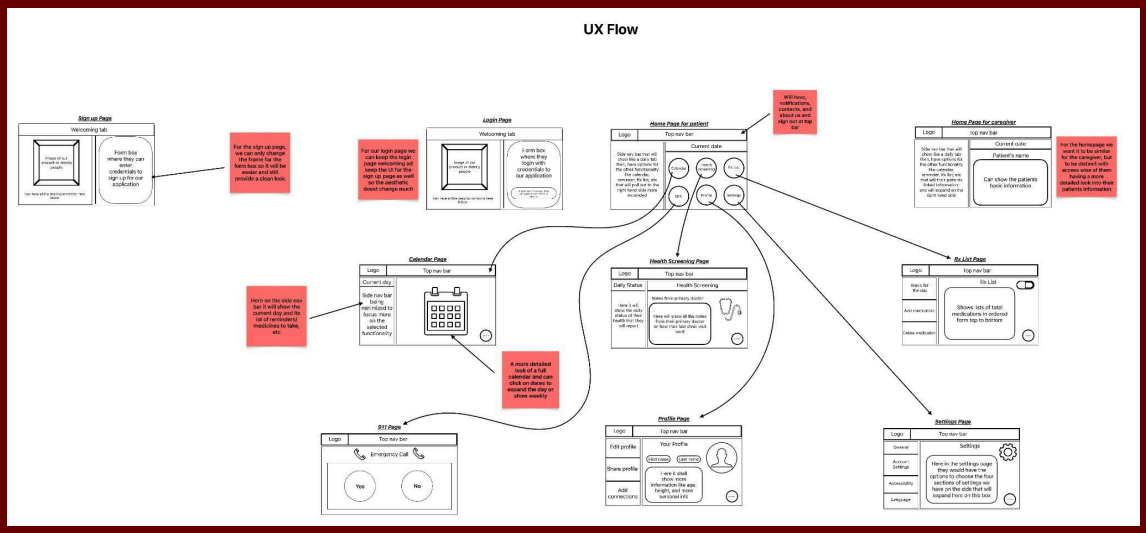
Functions	Associated Persona	Priority Number
User-General (User log in and out)	For all three personas, security is a requirement for medications	1
User-General (Users shall not be timed out of log in.)	Michael's grandfather leaves the web app on the computer 24/7, and would not like to struggle with login and logout often.	1
Account (The initial account shall have one primary user associated with caregiver permissions)	Ellie's daughter wants to be added to her medication routine in case she begins to forget- this can indicate potential memory loss if it notifies them frequently.	1
Account (A secondary account may be added to many primary accounts.)	Caregiver Andrea has more than one patient.	2
Account (A primary account may be associated with only one secondary account)	Andrea patients do not need more caregiver accounts than the one for Andrea, since he is setting it up and managing it for them.	2
Account (Once a secondary account becomes associated with the primary account, permissions shall change)	As Ellie has added another account, it implies that she may need assistance remembering her pills with full functions from the solo account. She may accidentally remove a reminder alert or a medication from her list.	1

	Michael has more cognitive faculty or electronic savvy than his grandfather, so it is easier for the both of them if he manages the app for his grandfather.	
Task notification bar (For patient)	Ellie is reminded to take her medication as she keeps the web app open.	1
Task notification bar (For caregiver)	Andrea can see that his patient is up and around before he has gone to their apartment. Ellie's daughter can see that she has used the app for the day, even though she isn't answering the phone	1
Reminder (Dosage)	All Personas require dosage.	1
Reminder (Unit of measurement i.e. mg, g etc)	All personas require measurement units per dose.	1
Reminder (Medication "form"- i.e. capsule, pill, powder)	All personas require knowing what the pill looks like.	1
Reminder (Portion of form optional i.e. full, half)	All personas require an amount because they take half a pill.	2
Reminder (Option click boxes for common requirements such as 'with food' and 'with water')	All personas require dosage.	1
Reminder (Time)	All personas require a time for their medication.	1
Reminder (Date)	All personas require a date because some medications are only once a month.	1
Reminder (For Patient)	Ellie has forgotten to take her meds for the day, and so her daughter will remind her.	1

Reminder (display reminder alert pop ups at the time set.)	Ellie has forgotten to take her meds for the day while cooking, but will remember at the next popup.	1
Reminder (include a reminder alert interval 10 min, 30 min, 1 hr option.)	Ellie has forgotten to take her insulin which is a very time dependent medication.	1
Reminder (Alert Pop-ups) : Shall require a yes or no input, block the entire app until input is given, include a time out of 5 minutes, send time outs as a “no”	Michael’s grandfather is going to press the button so that he can go back to his tv without his grandson ‘bothering him’ about taking his medication	1
Calendar (Displays a schedule of medication from reminders list)	Andrea can view his patient’s medication list instantly while talking to a doctor.	1
Medication List (Medications will be shown as a list and can be added/deleted/edited from here)	One of Andrea’s patient’s medication changes every few days based on blood levels.	1
User (User can upload photos and follow associated user [like social media])	Adrea can see on the scroll tab that her patient is having a lovely time with her granddaughter at the park	3
User (can send messages to other secondary linked accounts using messaging API)	Ellie had forgotten to take her meds and now her caregiver had sent her a message to make sure she's reminded Ellie wanted to send a message to her granddaughter about how lovely the day was and catch up	3
User (can send message asking about medication information Chat-gpt API)	Adrea wants to remind and check the prescription since she manages many medication for her patient	3

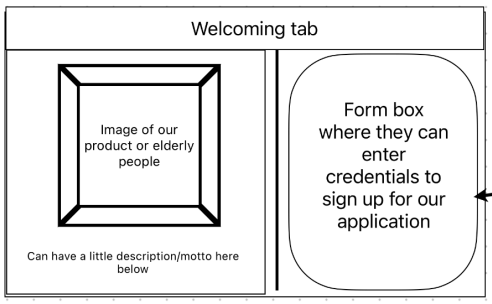
3. UI Mockups and Flows

Overall View



And below will be enlarged view:

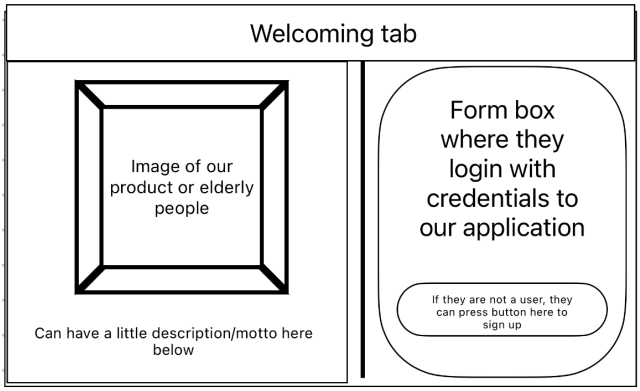
Sign up Page

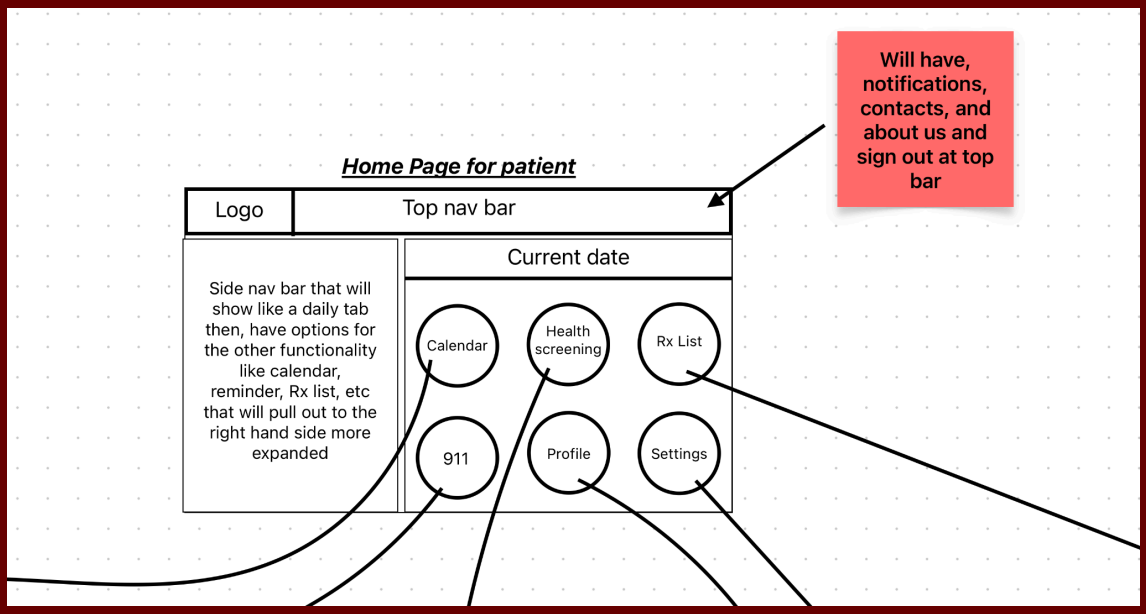


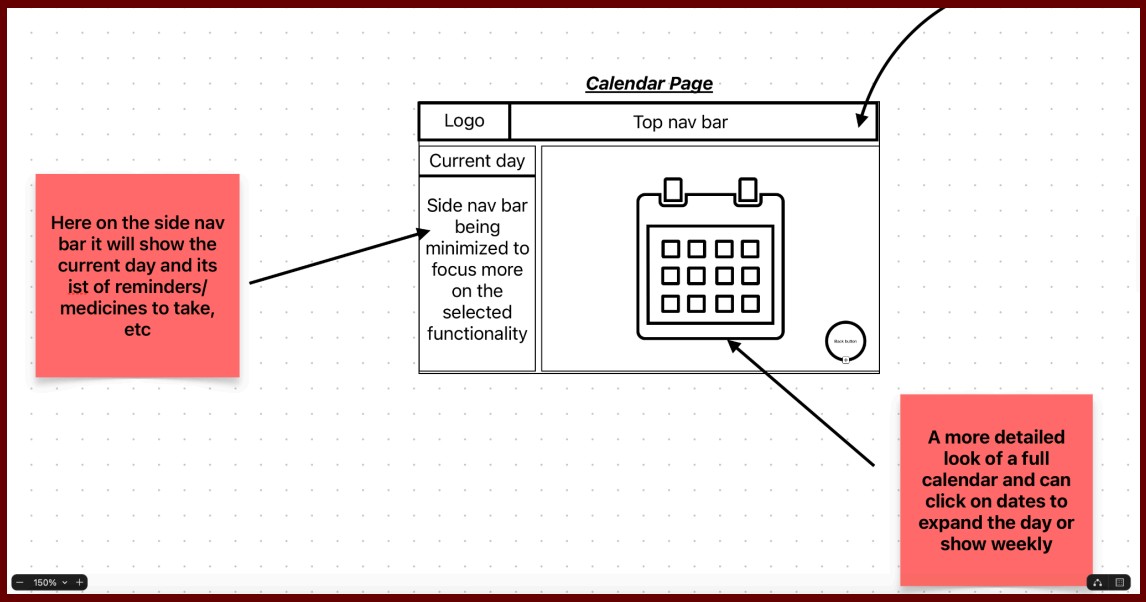
For the sign up page, we can only change the frame for the form box so it will be easier and still provide a clean look

Login Page

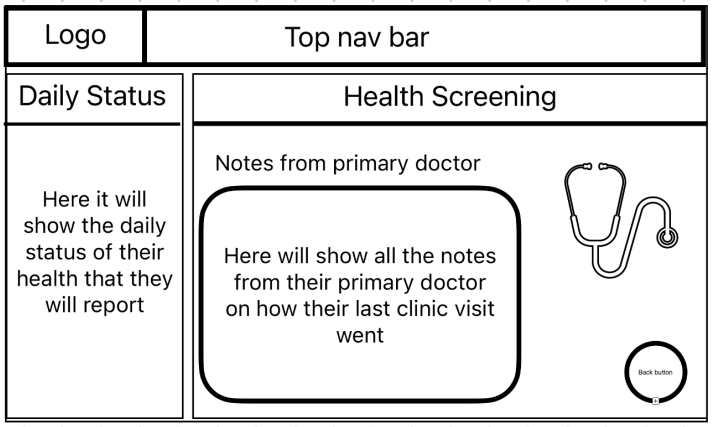
For our login page we can keep the login page welcoming ad keep the UI for the sign up page as well so the aesthetic doest change much



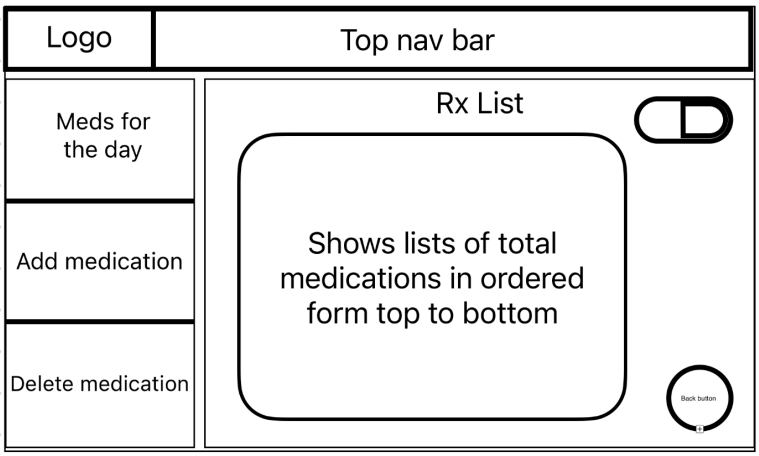

Will have, notifications, contacts, and about us and sign out at top bar
Home Page for patient
Logo
Top nav bar
Current date
Side nav bar that will show like a daily tab then, have options for the other functionality like calendar, reminder, Rx list, etc that will pull out to the right hand side more expanded
Calendar
Health screening
Rx List
911
Profile
Settings


Calendar Page
Logo
Top nav bar
Current day
Side nav bar being minimized to focus more on the selected functionality
Here on the side nav bar it will show the current day and its ist of reminders/ medicines to take, etc
A more detailed look of a full calendar and can click on dates to expand the day or show weekly
150%

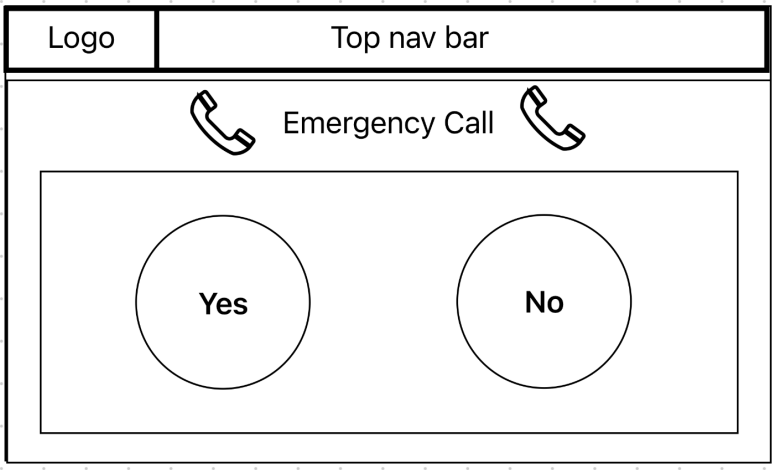
Health Screening Page

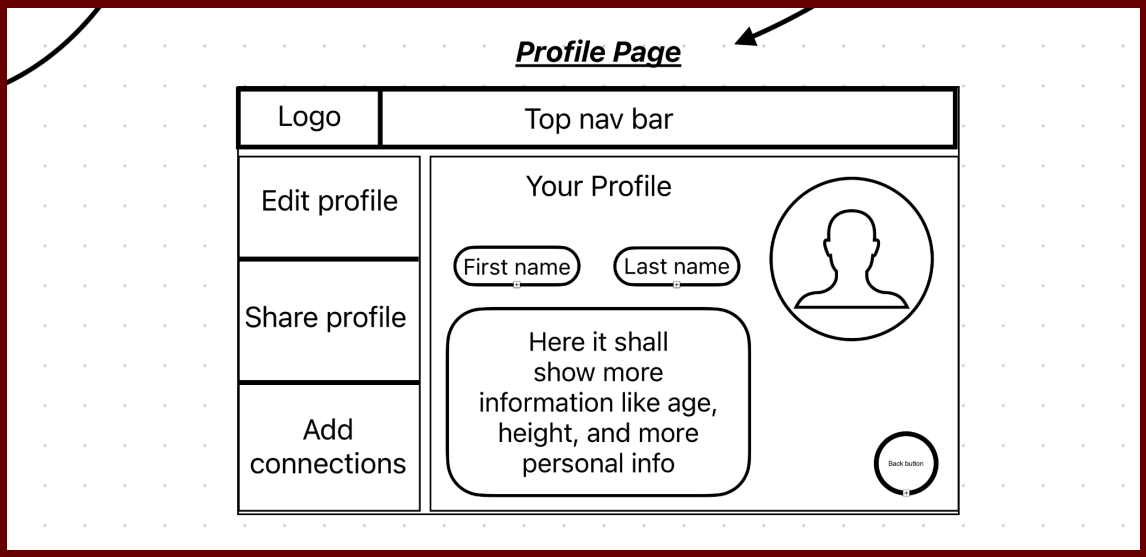


Rx List Page

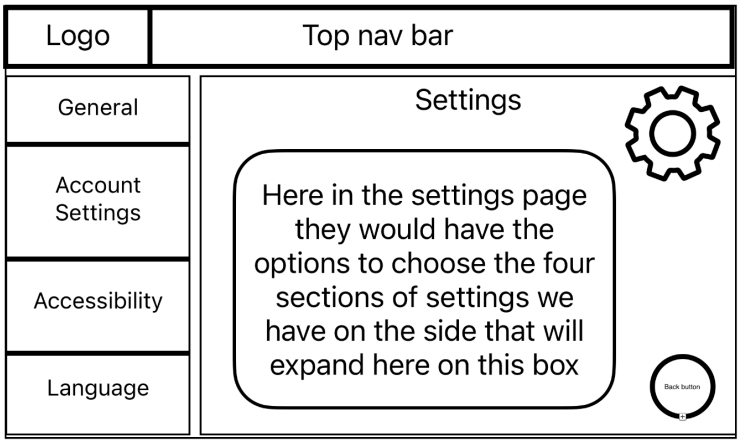


911 Page

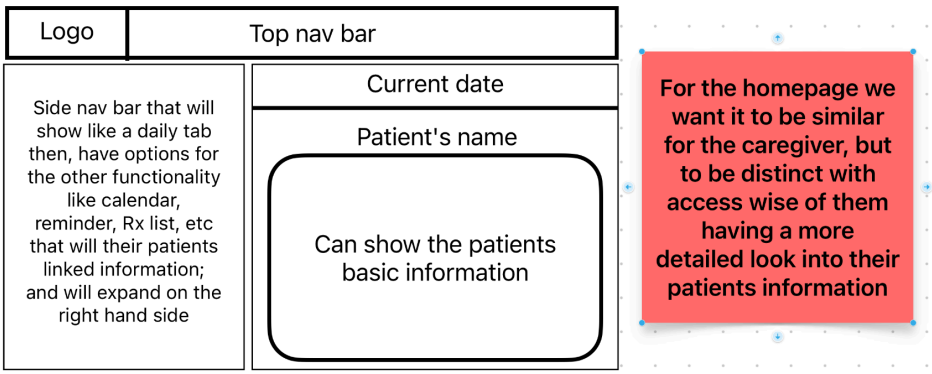



Profile Page
Logo
Top nav bar
Edit profile
Share profile
Add connections
Your Profile
First name
Last name
Here it shall show more information like age, height, and more personal info
Back button

Settings Page



Home Page for caregiver



4. High-level Architecture, Database Organization

DB organization:

- Please refer Data Definition for the database table
- To see an in depth structure, you can see our diagram here: https://dbdiagram.io/d/Doseedo_DB-65d273d0ac844320ae6b0f47

Add/Delete/Search architecture:

- Search architecture: Reminder, User, Medicine, AccountLink
- Add architecture: Reminder, User, Prescription, Picture, Account, AccountLink
- Delete architecture: Reminder, User, Prescription, Picture, Account, AccountLink

APIs:

- Axios API

  connects to the backend via from the frontend then uses express to create a server that listens for HTTP requests. When a GET request is made to these routes, the server executes the corresponding asynchronous function. This function attempts to retrieve data from a MySQL database through the selecttest() function.

- **Chat-Gpt API**
  Responding explanation about medicine to users
- **RxNorm**
  DataBase for Medicine

- Twilio API
  A texting API that we can use to send alert text to the caregiver in case of an emergency

5. Identify actual key risks for your project at this time

Skills risks and mitigation plan

1. **Fixed Time Constraints for Study Plans:**

Risk: Study plans have fixed timeframes set by the team leader, which may not be sufficient or efficient for study groups.

Mitigation: Empower study plan leaders to set detailed goals for each study plan, allowing for more flexibility in adjusting timeframes. Encourage communication with the team leader to modify schedules if necessary, either shortening or expanding study plan completion times based on the actual progress and needs of the study groups.

2. **Lack of Detail in Study Plans:**

Risk: The study plans lack detailed information, making it unclear what specific topics are covered.

Mitigation: Instruct study plan leaders to provide comprehensive details and goals for each study plan, specifying the content to be covered, such as the principles of HTML and CSS or React. This ensures clarity for all team members and helps in aligning expectations regarding the study content.

3. **Schedule Alignment with Academic Tasks:**

Risk: All study plans are designed to be completed by the end of April, potentially conflicting with academic tasks.
Mitigation: Keep the study plan completion date flexible. If necessary, adjust the completion date to accommodate the academic responsibilities of team members. Encourage open communication among team members to ensure

that adjustments can be made as needed while maintaining the overall goal of completing the study plans.

Teamwork risks (any issues related to teamwork):

1. **Teamwork Risks and Mitigation:**

Risk: The team leader has missed some meetings due to emergencies.
Mitigation: Create an emergency meeting schedule for cases where the team leader or other key members cannot attend. This ensures a steady and organized meeting even in their absence.

2. **Imbalance in Workload for Back-end Team:**

Risk: The Front-end team seems lightweight, while the Back-end team, particularly dealing with the Database, faces challenges and struggles to keep pace.
Mitigation: Have the Team Leader and Scrum Master actively participate in Back-end tasks, especially in database configuration, to provide additional support and help maintain the pace of progress.

3. **Legal Risk with 911 Functionality:**

Risk: Implementing a function that calls 911 requires obtaining a proper license, posing a legal risk.
Mitigation: Instead of directly integrating a 911 function, implement an emergency contact list within the web-app. While a user-friendly button for 911 is desirable, the practical inconvenience makes an emergency contact list a more feasible and legal alternative.

6. Project Management

The project has been progressing well overall despite encountering unexpected issues such as team members falling ill, unforeseen errors on the server, and communication challenges. We have consistently adapted to new technologies and acquired additional knowledge while advancing the project. Each assigned task has been actively addressed, with effective communication observed between the Back-end and Front-end teams. During meetings, team members openly discuss challenges they face, fostering a transparent sharing of difficulties.

Our progress is not only reported in each scrum meeting but also shared through the Discord group chat. This dual communication approach enhances transparency, providing a comprehensive overview of our advancements. To further improve transparency, as mentioned in point 6, we plan to implement a dev logs channel. This addition aims to facilitate better communication among team members.

Starting from Milestone 2, we have decided to transition from using Google Drive, Google Excel sheets, and Google Calendar to utilizing Notion as our primary project management tool. The decision to choose Notion is based on its ability to seamlessly integrate with Google Calendar and provide a more user-friendly UI for project management compared to Excel sheets. This shift aligns with our goal to enhance communication and transparency within the team.