Level Up Meds Vision Document

Version 2.0

Team Members:

Chance Biddle

Cayla Hale

Mary Johnson

Ibigoni Inye-Tam

Revision History

Date	Version	Description	Name
10/3/2025	1.0	First draft	All
10/10/2025	2.0	Revision	All

Table of Contents

1. Introduction

1.1. Overview

LevelUp Meds was founded by our sponsor to assist in managing the care of her aunt living with dementia. Coordinating medications, home healthcare, and caregiver schedules required constant communication between companions, family members, and healthcare providers. This real-world challenge inspired the creation of LevelUp Meds, a web application designed to simplify medication tracking, appointment scheduling, and communication among multiple caregivers.

The purpose of this project is to enhance the existing LevelUp Meds application by:

- Improving overall usability and user experience for caregivers.
- Increasing scalability through AWS cloud integration.
- Expanding functionality with new tools to support caregivers managing multiple recipients.
- Modernizing the interface for better accessibility across desktop and mobile devices.

1.2. Definitions, acronyms and abbreviations

- **Node.js -** A JavaScript runtime environment used on the server side to handle API requests, manage business logic, and communicate with the database.
- **Next.js -** React-based web application framework used to build the front-end interface of LevelUp Meds with server-sider rendering for faster performance.
- **Amazon Web Service (AWS) -** A cloud computing platform used for hosting, deployment, authentication, and database management for the LevelUp Meds web application.
- **Dynamo DB -** a fully managed NoSQL database service provided by AWS that stores caregiver, recipient, and medication data in a scalable and reliable way.
- **API -** Application Programming Interface, allows different software systems to communicate.
- **Caregiver -** refers to a user who manages medication schedules, health records, and appointments for one or more care recipients.

- **Care Recipient -** refers to the individual receiving care from one or more caregivers. Their health data, medications, and appointments are tracked through the system.
- **CRUD Operations -** this represents Create, Read, Update, and Delete which are the core data management actions performed on user, medication, and schedule information.
- **The Cloud -** refers to remote servers accessed over the Internet that host and process the LevelUp Meds application instead of relying on local storage or infrastructure.
- **UI -** User Interface, refers to the layout and visual elements of the application that users interact with, such as forms, buttons, and menus.
- **UX -** User Experience, focuses on how users interact with and perceive the application, ensuring that it is intuitive, efficient, and accessible.
- **Amplify** - AWS framework that simplifies configuration and deployment by providing tools for authentication, backend services, and hosting.
- **HIPAA -** Health Insurance Portability and Accountability Act, a significant law in the United States that governs medical privacy and grants individuals rights over their health information.
- **2FA/MFA -** Two-Factor or Multi-Factor Authentication, are security methods that require users to verify their identity through multiple forms of authentication, such as a password and email code.
- **Version Control(GitHub) -** this is a collaborative development platform that tracks code changes and enables multiple developers to work on the same project efficiently.

1.3. References

- Amazon Web Services (AWS): https://aws.amazon.com/
- Node.js Documentation: https://nodejs.org/en/docs
- Amazon Amplify: https://docs.amplify.aws/
- AWS DynamoDB: https://aws.amazon.com/dynamodb/
- Next.js Documentation: https://nextjs.org/docs
- HIPAA: https://www.hhs.gov/hipaa/index.html

2. Problem Statement

2.1. Current situation:

Currently, the website for LevelUp Meds already has many of its main core functionalities implemented. Adding clients, appointments and medications, are all possible to log within the app.

2.2. Problem:

Many tiny yet essential functions do not work. There are no notifications, some functions on the calendar are not implemented, limited medication tracking, etc. Because of this, it is limited in how much detail caregivers can put to keep track of their patients to provide the best care possible.

2.3. Proposal:

The goal of this project is to add functions to the web application that will allow better client tracking, involving increased detail and communication between all parties. The site will have more robust ways to track client information and keep an adequate log of all user actions. Extra care will be used to make an intuitive, easy to use interface, keeping accessibility in mind.

3. Stakeholder and User Descriptions

3.1. Stakeholder summary

Name	Description	Responsibilities
DeLury Enterprise	Project Sponsor	Provide project requirements, coordinate capstone team with development team, supply access to development tools
Dr. David Liu	Project Advisor	Guide the progression of the project, aid students with team work, monitor progress

3.2. User summary

Name	Description	Responsibilities
Caregiver	The user that is taking care of another individual.	Creating clients, scheduling appointments, adding new medications, marking medications as taken, communicating with other caregivers

3.3. User environment

A user begins with creating and logging into an account with their credentials. Then they can start creating clients with basic, medical, and insurance information, emergency contacts, and any other information that might be important as well. Once they have a client, they can schedule appointments and medications to be shown in the calendar. Caregivers can also invite other caregivers to share a client which will share information, medications, and appointments for that client as well as allow the caregivers to message each other through the website.

3.4. Operating environment

The website employs the following technologies:

- **Cloud Service Provider:** Amazon Web Services (AWS)
- **Database:** DynamoDB
- **Server Environment:** Node.js
- **Front-End Web Application Framework**: Next.js
- **Back-End Web Application Framework:** AWS Amplify

To use the website, a user only needs a device with a web browser and an internet connection. They can connect with a computer, phone, tablet, and even a kindle with any browser.

3.5. Key stakeholder or user needs

Need	Priority	Concerns	Current Solution	Proposed Solution
Knowing who did a task and	Low	Not all tasks have a way to say that they	N/A	Audit trail on the dashboard.

when did they do it		are completed.		
Knowing upcoming medications	High	Risks the dashboard becoming cluttered.	Checking the calendar or checking client pages	Adding a notification on the dashboard.
Know which medication is for which client	Med		Checking the client page	Filtering medications based on the client.
Knowing how often a client takes medications compared to how often they are forgotten	High	Has the possibility to increase loading times.	N/A	Marking a medication as taken or missed and calculating how often it is missed.

4. Product Overview

4.1. Overview

Our sponsor started LevelUp Meds to care for her aunt with dementia. Companions, home health care, and appointments needed to be coordinated while our sponsor was working. LevelUp Meds is intended to allow for cross communication amongst multiple caregivers as well as make communication easier.

4.2. Summary of capabilities (Major features)

Users are currently able to create an account as a caretaker. They are then able to create clients as well as add prescriptions and appointments for their clients. They can also add allergies and other important information about their client. There is a calendar view that makes it easy to see upcoming prescriptions and appointments.

4.3. Requirement analysis (Scope and limitations)

Our team has been tasked with the addition of several new features, one of which includes several UI/UX enhancements. Increased accessibility such as multi-language support and an improved dyslexia friendly mode. We will also work on several new features such as:

Project Requirements:

- Tracking Missed vs. taken meds
- Upcoming Meds Display
- Audit Trail Implementation
- More Language Support

Optional Requirements:

- Notifications
- File sharing
- Pinning Direct Messages

4.4. Assumptions and dependencies

Assumptions

- Users have reliable access to the internet and a modern browser
- AWS and third-party APIs will maintain uptime and stable service during project development and deployment.
- The sponsor (LevelUp Meds/Delury Enterprise) will provide access to test data and design guidelines.

Dependencies

- AWS Amplify for backend hosting and deployment.
- Next.js and Node.js for front-end and backend development.
- GitHub for version control and collaborative deployment.
- VSCode for development and testing environment.

4.5. Other product requirements

- **Performance:** The system must handle multiple caregivers updating data simultaneously without lag.
- **Security:** All users data should be encrypted both in transit and at rest.
- **Scalability:** Architecture must allow the addition of new modules (e.g., vital logging, health report exports) in future releases.

- **Usability:** Interfaces must comply with accessibility standards (WCAG 2.1) for inclusive design.
- **Testing:** Functional, usability, and API integration testing will be conducted before deployment to ensure reliability.

4.6. SWOT Analysis

Strengths

- Good team communication
- Weekly input from project sponsor
- AWS system already in progress
- Project advisor with 4 years of experience working on the project

Weaknesses

- Limited experience with implemented google APIs
- Need for continuous data synchronization between caregivers may increase server load and introduce sync delays.
- Limited experience with HIPAA-style data compliance and advanced AWS configurations.

Opportunities

- Expansion into mobile apps for broader accessibility.
- Working in a team environment
- Gives the team hands-on experience with full-stack development using AWS and Next.js.
- Offers learning opportunities in API integration, cloud deployment, and security best practices.

Threats

- Data security risks
- Stretch goal expansion delaying delivery.
- API service outages could disrupt critical functionality.
- Data breaches or security incidents could harm user trust and legal compliance