Project Requirements

1. Project Overview

- **Application Name**: Kinnect
- **Purpose**: Kinnect is designed to centralize and streamline various types of information for children of aging parents, making it significantly easier to manage healthcare details. It will serve as a real-time "hub" accessible by trusted family members, ensuring everyone stays updated on the latest scheduling and medication details for consistency.
- **Core Features**: The application will include a Medication Log, an Appointment Calendar, and a Vitals Tracking Dashboard.

2. Functional Requirements (FR)

- **User Management**
  - **FR01**: The software must enable the user to create an account.
  - **FR02**: The software must enable the user to log in to their account.
- **Medication Log**
  - **FR03**: The software must enable the user to add new medications, including name and dosage.
  - **FR04**: The software must enable the user to view a list of all added medications.
  - **FR05**: The software must enable the user to mark a medication as "taken."
  - **FR06**: The software must enable the user to delete a medication from the list.
- **Appointment Calendar**
  - **FR07**: The software must enable the user to add new appointments with details such as date, time, location, doctor's name, and purpose of the visit.
  - **FR08**: The software must enable the user to view all appointments on a collaborative calendar.
  - **FR09**: The software must enable the user to click on an appointment to view its details.
  - **FR10**: The software must enable the user to add a post-visit summary for each appointment.
- **Vitals Tracking**
  - **FR11**: The software must enable the user to enter and save readings for various vital signs.
  - **FR12**: The software must enable the user to view a chart/graph displaying vital sign trends over time.

3. Non-Functional Requirements (NFR)

- **Usability**: The application must feature a clean and straightforward interface, easily understandable by non-technically inclined users.
- **Performance**: The application should load quickly, with all data appearing as instantaneously as possible.
- **Security**: All user data must be stored securely. User passwords must be hashed before

being stored in the database.
- **Compatibility**: The application must function correctly on the latest versions of all major web browsers.
- **Technology Stack**: The project must be built using React for the frontend, a Node.js/Express server for the backend, and MongoDB/SQL for the database.

4. Domain Requirements

- **DR01**: Marking a medication as taken must create a timestamp and record the name of the family member who administered it.

5. A Use Case Specification

Use-case name	Brief description	Actor	Basic path	Alternative paths
Mark Medication as Taken	Description of the user's process of marking a medication as taken	Family Member	1. The user views Medication Log. 2. The application shows a list of medications 3. The user choses the medication they want and pick the "mark as taken" option 4. The application records the user's name and timestamp (DR01). 5. The application updates to display the new information	**A01: Network Error** If the application can't connect to the server, then an error message will display