Technical Specification and Requirements

ONCURO

Costas Christodoulou

costas1379@gmail.com

Table of Contents

Introduction

The objective of this project is to create a mobile app for *Android* and *iOS* and an admin web panel acting as a content management system and analytics tool.

The main objective of the app is to act as a guide through a daily program for cancer patients. Each day will consist of media, text, and tasks to be completed by the patient. The app will also record medication taken by the user, and daily symptoms. Initial wire frames can be found below.

The admin panel will allow doctors to add programs and modify daily content and tasks as well as view statistics regarding usage.

The backend will be setup on AWS on EC2 for the API server and RDS for the database. Since sensitive medical data is involved the database and API servers need to be compliant with the Level 1 benchmark of the Center for Internet Security (CIS).

Programming Language and Technology requirements:

- Frontend (Mobile app):
    - React Native
        - iOS
        - Android
- Backend:
    - NodeJS / Express.js
    - MySQL
    - AWS
        - EC2
        - RDS
        - Route 53
- Frontend (Admin panel):
    - ReactJS
    - HTML
    - CSS
- Versioning:
    - GitHub

Initial App wire frames
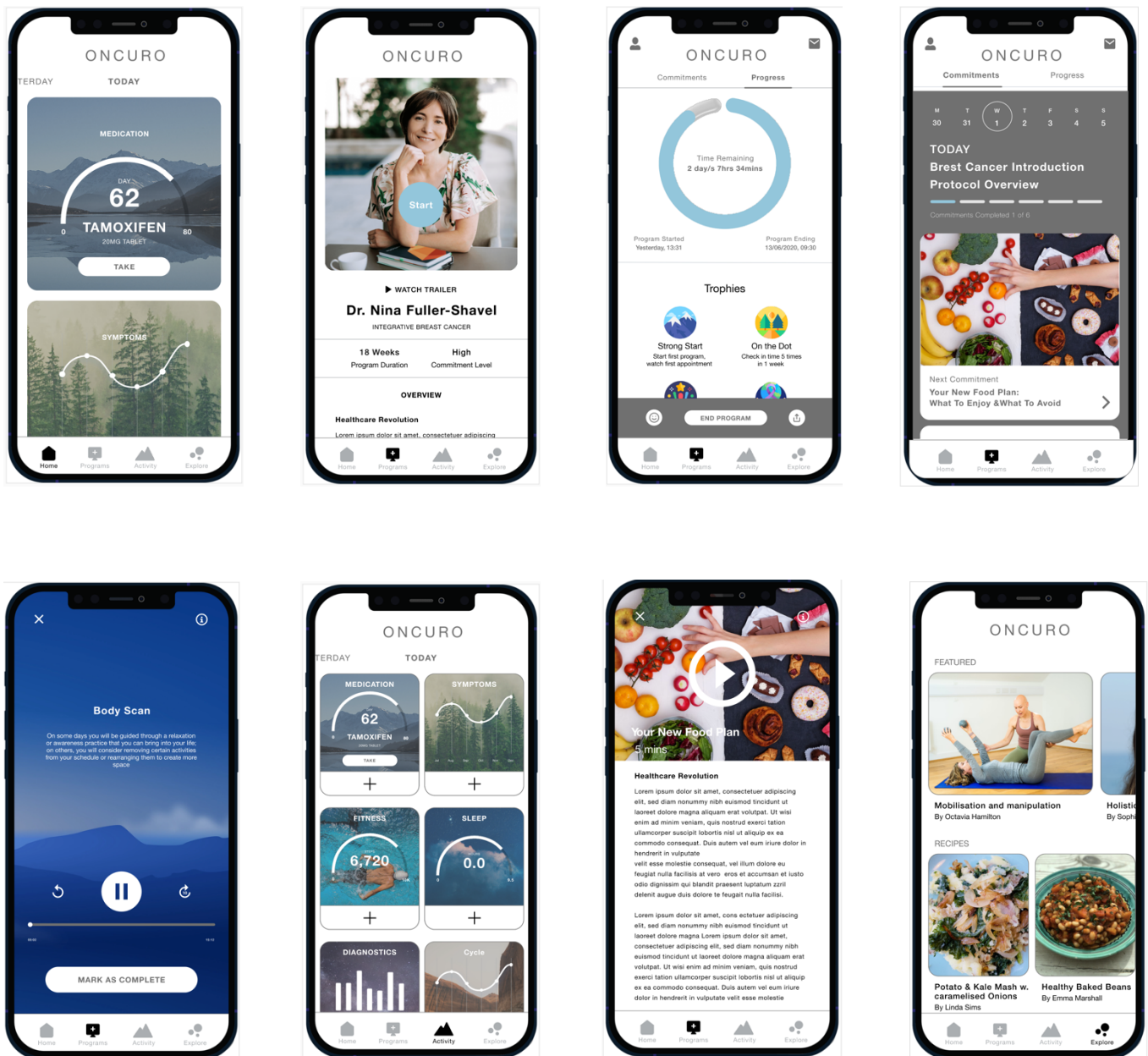
The wire frames can also be found online using this link.

Backend Requirements

Authentication / Authorization:

- Three types of users:
    - Admin / Full Access
    - Doctor (Web User)
        - Can use admin panel to create and modify program
        - Can see data / statistics for own program
    - Patient (App user)
- Login Types:
    - Email Login
    - Social Media Login
        - Google
        - Apple
        - Facebook
- API Authorization:
    - JWT tokens should be used for specific routes with a low time constraint and renewed on each api call
- Database authentication should be done through AWS IAM credentials with limited access rights

Deployment Requirements:

- Deployment on an AWS EC2 instance
- Deployment through CodeBuild and CodeDeploy
- Instance AMI Specifications for auto scale configuration
- Environment Variables in AWS Parameter Store

API Routes

- User
    - POST: Create user
    - PUT: Update user information
    - DELETE: Delete user
    - GET: Get user in formation

- Program
    - POST: Create program
    - PUT: Update Program information
    - DELETE: Delete program
    - GET: Get program information
- Program day
    - POST: Create program day
    - PUT: Update Program day information

- o DELETE: Delete program day
  - o GET: Get program day information
- Task
  - o POST: Create task
  - o PUT: Update task information
  - o DELETE: Delete task
  - o GET: Get task information
- User Task
  - o POST: Add completed task
  - o GET: Get completed tasks
- Medication
  - o GET: Get list of user medications
  - o POST: Add new medication for user
  - o DELETE: Delete medication for user
- Medication History
  - o POST: Add medication and time taken
  - o Get: Get user medication history

Admin Web Page Requirements

Dashboard

- Number of Patients
- Number of Programs
- Number of Doctors

Users

- Table of all users (clickable)
  - Email
  - Program used
  - Date added
  - Id
- Clicking on a row will take you to a user profile page with all their information

Doctor

- Table of all doctors (clickable)
  - Email
  - Program used
  - Date added
  - Id
- Clicking on a row will take you to a doctor profile page with all their information

Program

- Table of Programs
  - Title
  - Date added
  - Number of Users
  - Id
- Clicking on a row will take you to the program page with all relevant information and allow modification of program
  - Days
  - Tasks
  - Content
  - Users

- Create new program button that will take you to a program creation form.

Initial Database Design