CareCoord
Design Document

A mobile application for coordinating elderly care

The Caretechers

Members: Aaron Heo, Brynnli Borrowman, Benjamin Hatch, Seng Horn Rith

Course: CS4000 Fall 2022 - Senior Capstone Design

Date: October 18th 2022

Executive Summary

Many of us will become a primary caretaker of an elderly relative at some point in our lifetime. However, we will likely have busy lives of our own, and many people simply don't have time to check up on elderly relatives consistently. Leaving elderly relatives alone at home without consistent caretaking visits can be incredibly dangerous physically, mentally, and financially (food can run out, heating/cooling issues can arise, falls can become lethal, bills can go unpaid, etc.).

Most families attempt to ensure elderly loved ones are consistently taken care of by sharing the burden of caretaking between multiple family members. However, coordinating care between family members can be messy. Family members could forget which days they are responsible for caretaking visits, or neglect important caretaking tasks (such as refilling medication). Thus, by attempting to split up caretaking visits, the quality of caretaking for the elderly loved one might actually be *worsened*, which paradoxically exposes the elderly loved one to the same dangerous situations previously described!

Our project, CareCoord, is a mobile application for family caretakers (the users) that will mitigate the problems of shared caretaking by accomplishing four main goals, which are:

1. Ensure that the elderly relative does not go more than "X" days (defined by the user) without a wellness-check. This will be accomplished through an in-house scheduling solution that allows caretakers to coordinate different tasks and schedule time off, as well alert caretakers (SMS, email) that a visit is needed.
2. Ensure that crucial caretaking information is centralized and easily accessible. This will be achieved by an in-app group messaging system for conversations about caretaking, as well as a notes section. This messaging system and notes will have search functionality so that caretakers can search for any information needed.
3. Make sure that caretakers complete all of the essential household chores and perform a proper wellness check. In order to accomplish this goal, we will implement a feature to create a checklist of items that need to be completed around the house or for the elderly person.
4. Facilitate better tracking of the elderly person's health. There will be a feature that allows the caretaker to enter health vitals (heart rate, temperature, etc.) which will then be displayed using auto-generated graphs so that a caretaker can easily recognize positive or negative patterns in the elderly person's health.

After extensive discussion with people that have shared caretaking responsibilities with other family members, we are very confident that our project will ease the burdens of shared caretaking and bolster the safety of elderly loved ones living alone if the goals above are accomplished.

Background and Technical Requirements

Why CareCoord?

Elderly loved ones are subjected to a variety of serious dangers as a result of poorly coordinated/low quality shared caretaking. Bills can go unpaid, heating/cooling issues can arise, food can run out, fall risks can become more lethal, and more. Unfortunately, shared caretaking between family members is often not executed well. Family members often forget which days they are responsible for visiting elderly relatives, or neglect important caretaking tasks, such as checking if there is food in the pantry. Additionally, important caretaking information is often "lost in the weeds" of various text messages between caretakers. All of these scenarios expose elderly loved ones in need of caretaking to the dangers previously described.

Our project, CareCoord, will ensure that family-centered shared caretaking is properly coordinated and executed correctly, therefore mitigating the aforementioned dangers. It will accomplish this by providing family caretakers with a mobile application that will facilitate easier coordinated scheduling and reminders for caretaking visits, sharing of relevant caretaking information between family members, caretaking visit checklists, and health tracking mechanisms, all in *one place*.

Currently, family members responsible for shared caretaking tend to use existing tools such as iMessage, Google Calendar, Notes, the Health App, or a combination thereof to achieve the proposed functionality of CareCoord. Having to use many different tools can be overwhelming. There is no current platform that keeps track of the different aspects of caretaking in one place and in an easy to use manner. Additionally, these tools are not tailored for caretaking of a loved one. CareCoord is unique because it provides a centralized platform for caretakers, and it is designed specifically for caretaking.

Required Technologies

Frontend

React Native

React Native is a frontend framework that is very similar to React for the web. It has the same great features of the web version such as component based html elements and hooks to keeping track of and manipulating various state variables. React Native was the framework of choice because it allows us to seamlessly create user interfaces for mobile devices for both Android and iOS using one codebase. In

addition, our team possesses a fair amount of prior experience with either the web version of React or React Native.

Backend

Express.js
Express.js is a JavaScript framework that allows us to easily implement a web server that will contain all of our REST API endpoints.

MySQL
For our database, we will be using a MySQL database. We chose this database because it was what we had the most experience with as a team.

Other

AWS
AWS is a cloud platform that provides many services for building and maintaining software. We will mainly be using AWS to host our servers (frontend, backend, database).

SendGrid
Iif an elderly person has not been checked on for a certain amount of time, our mobile app will send a notification to notify users that they need to be checked on. We will send these notifications using a push notification on mobile devices as well as email. SendGrid provides an API that allows us to easily send these email notifications.

Google OAuth 2.0
Google OAuth 2.0 will allow our team to easily implement login functionality in a simple and secure manner. In addition, Google OAuth 2.0 allows us to pull important information such as an email when a user logs in using Google.

Docker
Docker is a technology that allows us to containerize our project so that we can start up a clean version of our mobile application every time. If we run into a situation where we need to restart our app because of an error or unexpected behavior, we will be able to consistently restart it to a working state.

Git
Version control is a crucial part of software development. It allows us to keep track of changes that have been made and combine different branches. Git is widely used and our team has previous experience with it which was why we decided to utilize this technology.

Assets & Engines

Although most of the mobile application components will be built from scratch, we can leverage certain resources in order to speed up development time. There is a rich library of open-source React Native mobile UI components available for reuse online (that can be installed through Node JS's npm package manager). This will allow us to develop the frontend (the mobile interface) much faster.

Rather than implementing our own secure login functionality from scratch, we will be using Google OAuth (see "Required Technologies" section above). This will save us plenty of development time, as well as ensure proper security protocol is being adhered to.

Most of the application will be designed and developed from scratch, from the structure of the database to the individual API endpoints on the backend. This will provide us with flexibility to make the app function exactly as desired (see "UI Sketches").

Software/Hardware Requirement

The CareCoord application will be developed and tested as a mobile app for iOS on an Apple iPhone. Our clients will download an iOS application to their iPhone, which should be updated to the most recent operating system. At the moment, we do not foresee any abnormal memory usage from our app. It will be cloud-operated, without the need for excessively large files on the client's IPhone.

We are planning on using Google OAuth 2.0 as our account management system, and this will require our clients to have a Gmail account.

Since we are using React Native to develop the CareCoord application, it will likely work on Android phones as well. However, we are focusing on iPhone development, and are considering Android compatibility as a stretch goal. Thus, Android compatibility will not be guaranteed.