

GlassRx - Project Plan

Goals and Scope

GlassRx is focused on making it easy for patients and physicians to manage medications and their corresponding schedules. Managing pill dosages and schedules can become very difficult, as age and number of prescriptions increases, and the main goal of GlassRx is to simplify this process using Google Glass. First and foremost, GlassRx is a medication management application. As such, it will focus on maintaining the user's (potentially complex) medication schedule. This will be accomplished through implementation of medication schedule creation/modification, alerts and alarms which notify the user when he/she must take a medication, connecting a Google user account to our database, and barcode scanning of medication bottles, time permitting. Because Glass applications are voice- and gesture-driven, GlassRx will allow for both voice acknowledgment of medication alerts and voice and gesture navigation through views. A similar application will also be implemented for a doctor to manage his or her patients' pill schedules, adding or modifying a prescription for them, and seeing potential conflicts, all while maintaining face to face contact with the patient.

Since controls and gestures can be cumbersome on the Glass device, and voice recognition of complicated pill names is a potential problem, our focus for the glass app will be on the display of schedules and the notifications, instead of the editing and creation of schedules. The phone app will be capable of editing schedules and adding medications, as well as providing additional notifications for the patients.

Design Goals

The uses of the app are fairly straightforward, and as such there will probably not be a ton of flexibility in terms of separate uses. We are assuming that we are working with medications, and an agreed upon set of information about the pills, i.e. what times to take it, its use, its side effects. Within these categories, the information will be mostly user inputted. We are assuming a somewhat rigid definition of a "schedule" in that it will be the same every week. We hope to have some access to a pill database, so that when a pill is entered, information about side effects, images, etc. will be read in.

The scheduling and notification aspects of our program are core. At minimum, a user should be able to create a schedule through the mobile app, view it from the glass app, and receive notifications on both. We hope to implement some sort of barcode scanning, or text scanning, that would be able to read in most of the information without having to input it manually.

Dependencies

Currently one of our main dependencies is our one Google Glass. Because there is no real emulator for Glass available, we can't really test and write code effectively without more Glass platforms to work on. We also depend on Glass's ability to safely transmit client-patient information. Currently, this is not the case, but we are writing it assuming that one day Glass will be able to securely store and transmit data.

A large dependency in the long run will be the Google Glass itself. Right now, information cannot be transferred safely and securely. So in order for this to actually be useful, we're depending on that to change. But it's mutually agreed that we will just use fake patients for now, and hopefully at a later point we will be able to use this with actual patients and pills.

Concerns

None of us, and really not very many people in the world, is very experienced coding for Glass, so we are not able to make incredibly confident estimations for how easy or hard it will be to implement certain features. However, this has been made known and we have made conservative estimates, and expect we will be able to get a lot of cool stuff done.

Team Organization

Evan Kaplan, Technical Lead - edk14@duke.edu

Vincent Wang, Project Manager - wang.q.vincent@gmail.com

Will Knowles, Business Analyst, Test Lead - willknowles@gmail.com

Task Deliverables for each Sprint

Prototype Demo (10/1):

GUI for patient Glass in place. Glass app should be mostly functional with notifications and navigation, but data based on hard-coded values.

Baseline (10/22):

GUI for both Glass apps in place. Will be able to create and store a simple schedule with the phone app.

Alpha (11/5):

Apps should be able to communicate with each other. Glass app will be able to import a schedule from a phone app, and set data from that.

Beta (11/19):

App will be capable of importing information about a pill from a database. We also hope to be able to add medications through barcode scanning.

Robust Full Functionality (12/11):

All aspects of patient/doctor sides will be functional for both the glass and the phone apps, and they will be communicating. We will have as many stretch goals implemented as we have time for.

Individual tasks

Sprint 1 (Pretotype):

Evan

1. Acquire Google Glass device, install and explore SDK, and configure IDE (4 hours)
2. Install Glass drivers on PC in order to allow for USB debugging (2 hours)
3. Develop pretotype UI mock-ups for the patient-side GlassRx application (2 hours)
4. Implement basic UI with voice and gesture commands and deploy to device (4 hours)

Vince- Communicate with Ryan and Glenn to determine detailed specs (3 hours)

-

Will- Created mockups for patient UI

Sprint 2 (Prototype):

Evan

1. Add physician-side user interface (4 hours)
2. Adjust and add transitions/navigation for Client pretotype, including voice and gesture commands (4 hours)
3. Implement device-side HTTP request functionality in preparation for database connection (4 hours)
4. Create JSON parsing functionality (2 hours)

Vince- Communicate with Glenn to figure out database setup on their servers (2 hours)

-Set up LAMP stack on Linux virtual server

-Set up Linux VM (1 hour)

-Set up Apache and MySQL and PHP (3 hours)

-Set up API for backend model/controller logic (4 hours)

-Setup JIRA outline and overview (1 hour)

Will-

1. Get and add mockup data from Ryan (2 hours)
2. Acquire additional Google Glass devices, from Duke CS or Ryan (1 hour)
3. Add simple time based push notifications (4 hours)

Sprint 3

Evan

1. Complete provider UI for phone application (4 hours)
2. Implement basic account functionality for handling different patients/doctors (2 hours)
3. Begin patient UI for phone application (4 hours)
4. Create core networking classes for handling web requests(4 hours)

Will

1. Install glass drivers for Mac (3 hours)
2. Familiarize with glass development (1 hour)
3. Implement provider UI for glass (4 hours)
4. Figure out how account functionality will work (Connected through Google+? (2 hours)
 1. Implement login/account handling (3 hours)

Vince

1. Design simple SQL database (2 hours)
2. Host sample web server on own computer for sample testing (4 hours)
3. Configure nursing school database stack using CakePHP (4 hours)

Sprint 4

Evan

1. Complete patient UI for phone application (4 hours)
2. Explore possibility of integrating barcode scanning into application (2 hours)
 - a. If feasible, implement barcode scanning (4 hours)
3. Improve phone application navigation based on client feedback (2 hours)
4. Extensively test networking functionality (4 hours)

Will

1. Implement patient UI for glass (4 hours)
2. If feasible, integrate barcode scanning into Glass application (4 hours)
3. Test Glass GUI (with friends, patients) (3 hours)
4. Make according adjustments to UI (4 hours)

Vince

1. Optimize database by talking with Prof Yang (1 hour)
2. Integrate Android back end with database (4 hours)
3. Pipe content from JSON to Glass and Android application (3 hours)
4. Communicate with David and Will from Nursing school to effectively transfer data to them