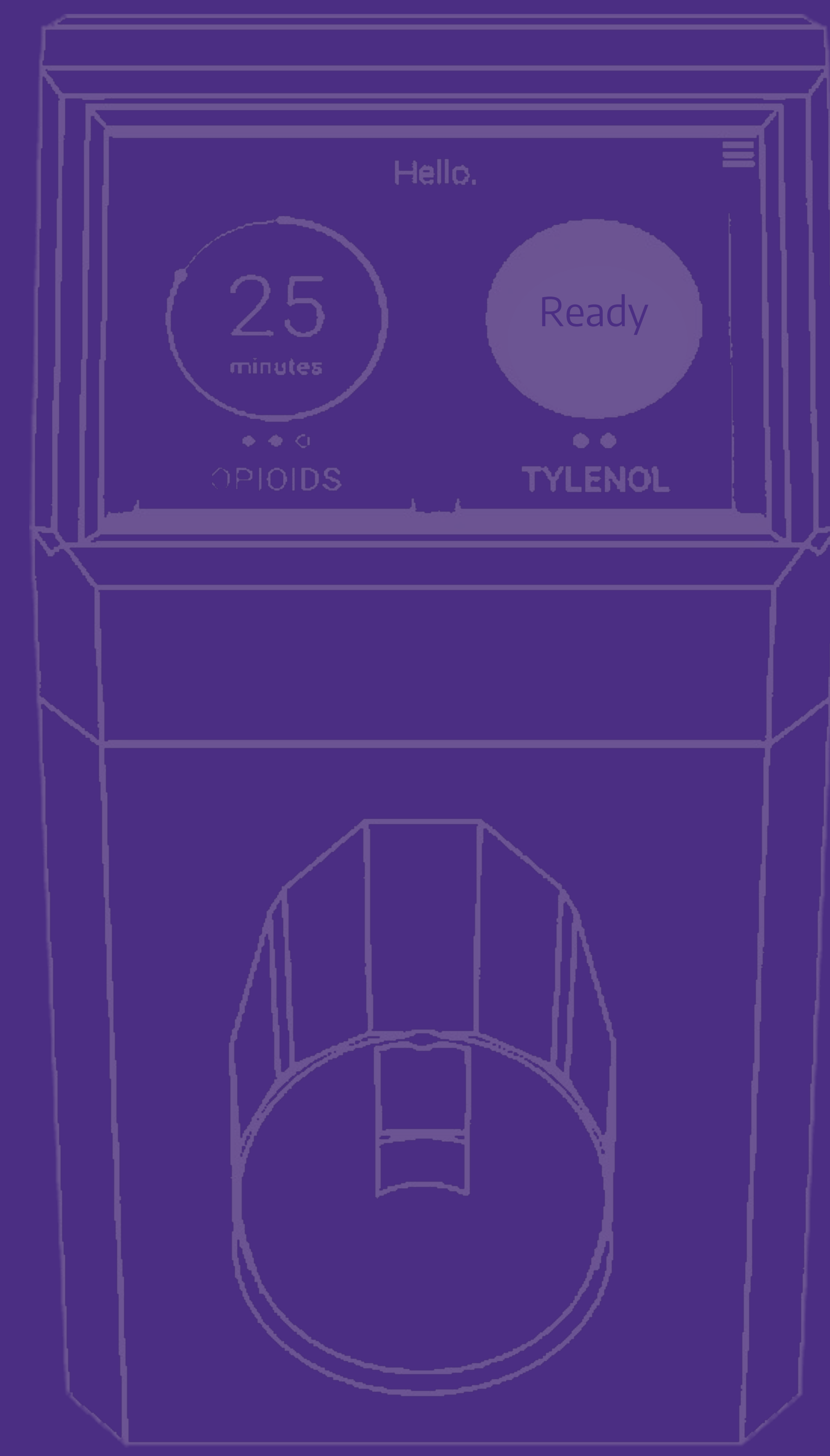
Home Opioid Patient-Controlled Analgesia (PCA) Box



Team 11 Recovery Improvement Interactive Technologies (RIIT)
Ian Russell, Finn Thompson, Ali Morgan, Michael Beach

Sponsor Seattle Children's Hospital & The University of Washington
Dr. Lance Patak, Dr. Stuart Solomon, Dr. Jacob Gross

Finn Thompson - HCDE B.S. Capstone

Process Book (June 6, 2018)

HUMAN CENTERED DESIGN & ENGINEERING
UNIVERSITY of WASHINGTON

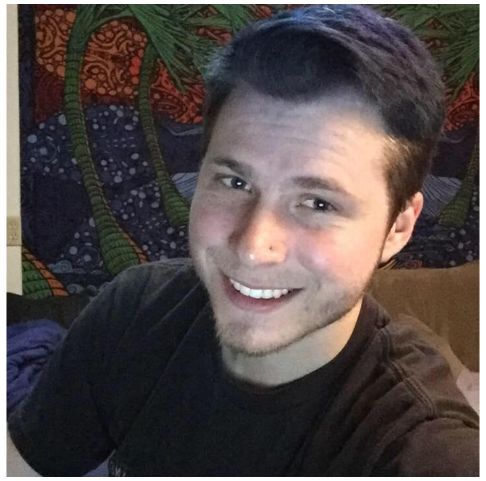
Table of Contents

Meet the Team

Ian Russell

Glad to be here, I feel lucky to be part of a project that is a combination of all what I like and what I care about. For me academia has been a journey from 3D animation, psychology, biology, economics, teaching and now to HCDE. I've found HCDE to be a place where my diversity of skills, interests and experiences can shine. Currently I am the webmaster for the Universities 3D printing club called WOOF3D. I have a passion for creativity, design of all kinds, prototyping, finding and creating solutions.

Finn Thompson



Hi, I'm Finn. I'm an HCDE major with a knack for the nuts and bolts of how the world works, and I'm pursuing a math minor and have a background in computer programming. I am interested in programming, prototyping, and usability research. When I'm not doing schoolwork, I am a supervisor at a Starbucks and I am passionate about coffee and team-focused communication.

Ali Morgan

Hi! I'm Ali. I am a senior working towards a double degree in Human Centered Design Engineering and in Medical Anthropology and Global Health. On the weekends, I work in the Home Department at JCPenney. I am excited to work on a project in the medical field creating a device that could hopefully make a difference in people's lives.

Michael Beach

Hello! I am a senior undergraduate at the University of Washington working toward a double degree in Human Centered Design & Engineering (HCDE) with a focus on HCI and Comparative History of Ideas (CHID) with a focus on media studies. I have been working in the CSC Lab (Computer Supported Collaboration) in HCDE for three years on MoCA (Model of Coordinated Action), a theoretical framework that describes collaboration. We are currently working on a new papers to be submitted to CSCW this April.

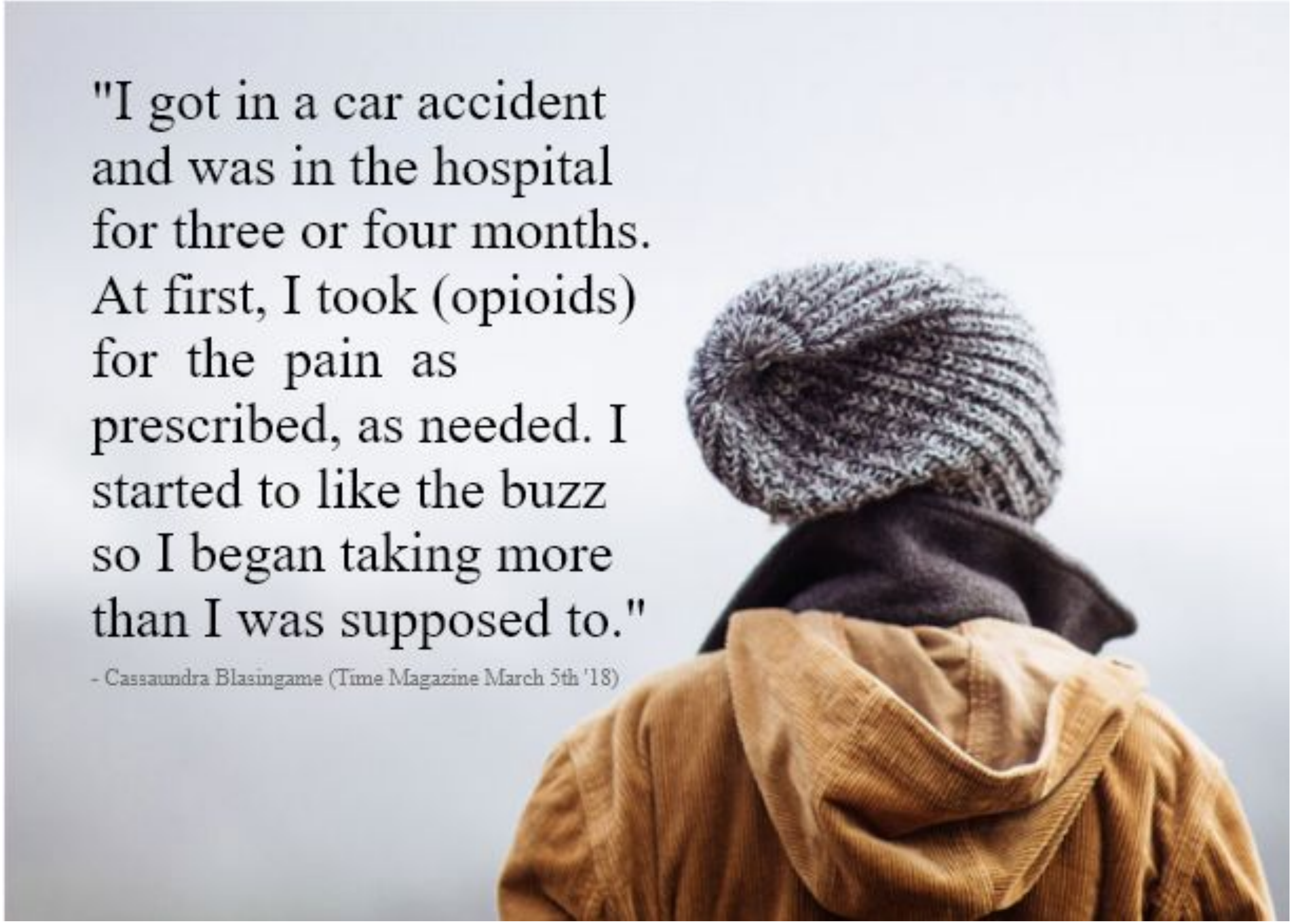
For my CHID thesis, I am designing a creative agency with a funky org structure that brings the power of CHID and Google's Design Sprint methodologies to nonprofits and other socially and ecologically conscious orgs.

For this capstone project, I am hoping to incorporate some design sprint methodologies to help with efficiency and contribute my design, prototyping, and research skills.

Project Overview

Opioids provide necessary pain relief to postoperative patients. However, opioids can be addictive, dangerous, and are often subject to misuse. Our project, an in-home oral Patient-Controlled Analgesia (PCA) box, will address these issues by providing patients with the guidance they need to manage their pain effectively during their postoperative recovery.

Our team researched, designed, prototyped, evaluated, and iterated a Patient-Controlled Analgesia (PCA) device and companion app that monitors and manages opioid prescriptions while connecting patients to doctors throughout the postoperative recovery phase.



The Opioid Epidemic

The United States has witnessed a growing opioid epidemic in recent years. In 2016, 11.7 billion opioid pills were prescribed, equating to 36 pills per American, and 3.3 billion of those pills went unused [1]. In 2015, 12.5 million people misused prescription opioids while there were nearly 90 deaths every day from opioid overdoses [2][3]. Deaths from opioid overdoses continued to increase, highlighting an extreme uptick in opioid-related deaths in recent years, as shown in Figure 1.

With this project, our sponsor, Seattle Children's Hospital seeks to provide an oral PCA device and companion app that make opioid prescription information less ambiguous and allows pain medication control to be guided by a device and companion application. Additional goals include being able to better regulate safe return of unused opioids as well as the ability to remotely adjust prescriptions as needed. One goal is to generate individualized patient usage data that can be observed in real time by medical staff which can lead to identifying high risk behaviors before addiction occurs, thus allowing improvements to be made to treat future pain needs, reducing opioid waste, incidence of addiction and overall costs.

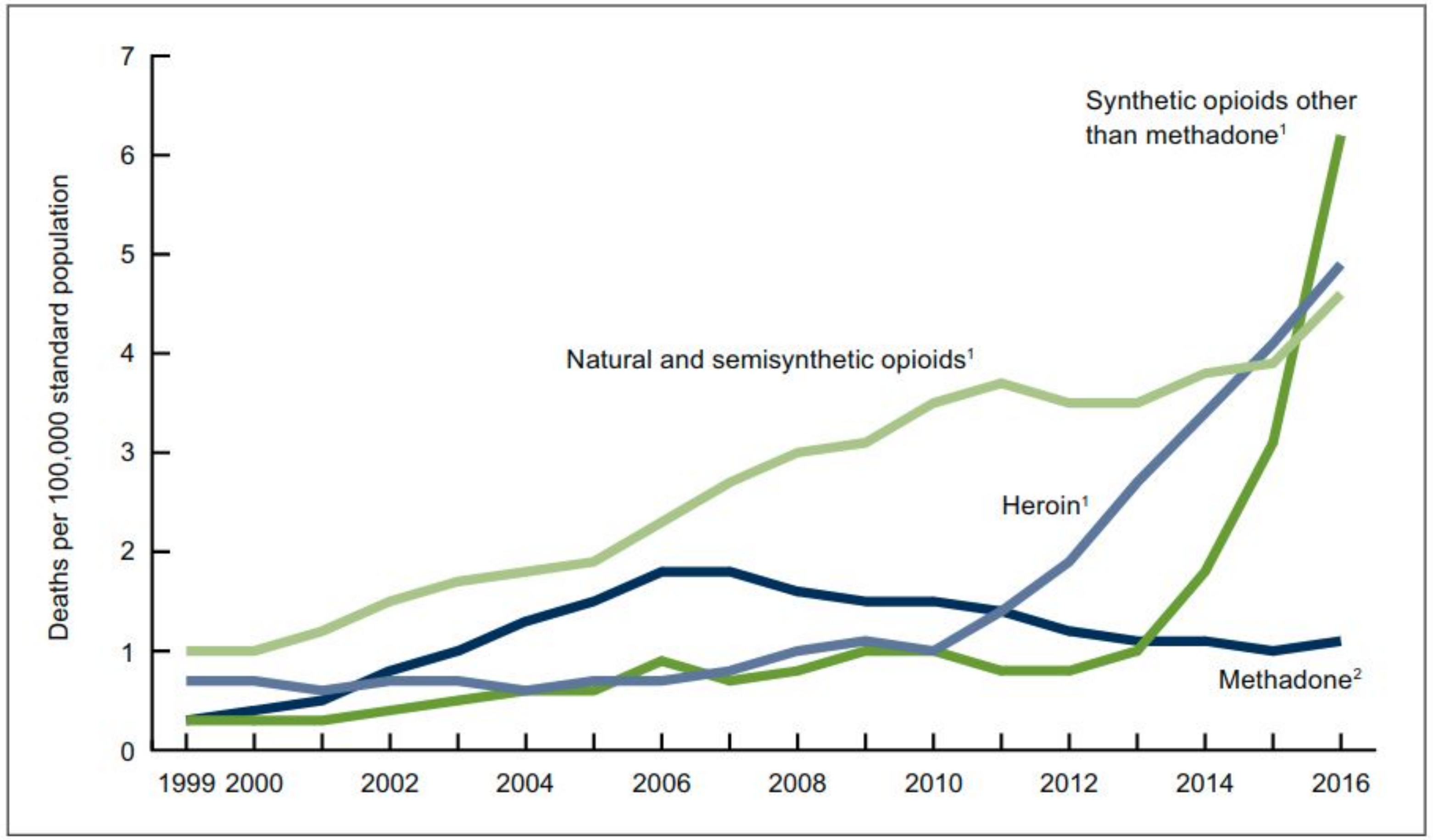


Figure 1: Age-adjusted drug overdose death rates by opioid category: USA, 1999-2016. [4]

Our Design Process

Phase 1
Design

Sketches and Flow
Companion App 1.0
PM: Michael Beach

Phase 2
Prototype

Pill Box 1.0
Interactive Demo
PM: Ian Russell

Phase 3
Evaluation

Usability Testing
Data Analysis
PM: Finn Thompson

Phase 4
Iteration

Pill Box 2.0
Companion App 2.0
PM: Ali Morgan

Pre-Design Research

Literature Review

As part of our Capstone Planning process, we conducted a brief literature review to identify scholarly research related to medical adherences and smart devices used to manage pain. Our sponsors supplied us with a drive full of research articles and their summaries, so we had much of our research work cut out for us from the start, but the articles they provided covered primarily opioid crisis statistics and research related to pain management. Our research, then, focused on filling in the gaps and provide us background knowledge that would help us design our smart pill box. Each teammate read and summarized one of the articles we identified, which I summarize here.

Michael

"Of pill boxes and piano benches: home-made methods for managing medication" (2006) [5]

This article describes how elderly people in Denmark manage their medication. The researchers concluded that new systems should focus on creating benefits for the users rather than for clinicians and work on the principle of *Technology by Invitation* where users have the option to "turn on" the new system.

While we, as HCDE practitioners, understand to prioritize the user as the primary stakeholder, this article served to remind us to stay focused on the users (patients and caregivers) first before the many secondary stakeholders present (doctors, pharmacists, insurance companies).

Ian

"Simulation of Mobile Treatment Monitoring System" (2011) [6]

This article describes how medication non-adherence substantially increases medical costs, recovery time, and required medical visits. The researchers designed a Mobile Treatment Monitoring System that sends users reminders to take their pills and tracks their usage data for doctors to stay in the loop. Their study concludes that such a system would make medication treatment over 200 times more cost-effective.

While the researchers in this study did not appear to actually implement their concept, we took this article as solid supporting evidence for our sponsor's goal to provide data to doctors. A goal of ours, then, was to implement this data collection, though we did not quite meet this goal throughout the project.

Ali

"Opportunities to Support Parents in Managing Their Children's Health" (2008) [7]

This article describes a study where researchers found that many parents are dissatisfied with their ability to provide adequate healthcare for their children, as they are unsure of what information they should be recording for doctors and they often do not share the same information about children's healthcare between family members, school nurses, and pediatricians. The researchers then propose a smart medical bottle that tracks a child's medications and alerts parents about it.

We found this article relevant as we initially planned to develop a similar, pill-bottle-like smart device. While we did later design a pill box instead, this article remained useful to emphasize how the primary user might be a dependent, such as in the case of children or elderly patients, so we need to design for caretakers.

Finn

"Mobile Health: Medication Abuse and Addiction" (2014) [8]

This article describes a design concept for monitoring and analyzing medication use patterns in opioid prescription patients. The author used existing medical adherence data, self-reported by patients, to define algorithms that analyze probability of multi-dosing, probability of accidentally dosing, probability of current abuse, and likelihood of addiction.

Early in the stages of our process, we envisioned our product would dynamically update prescriptions in order to best supply the patient with the opioid medication they need, so the algorithms in this article seemed helpful. Later, we decided that the device should not have this authority, but we still believe these algorithms could prove useful on the doctor's portal to identify patterns in patient's adherence data.

Competitive Analysis

Our literature review provided quality scholarly insight into our design problem, but much of it was theoretical, so we conducted a competitive analysis to identify how existing products address problems in this design space. With this exercise, we sought to identify strengths and weaknesses of various products that seek to help patients with medication or opioid management. We identified five currently available products, ranging from small smart bottles to coffee-machine-sized multiple pill dispensers. Ali reviewed the Livi and I reviewed the remaining four products, based off of information available on YouTube or their respective websites.

Product	Notification	Connectivity	Lockable	Form	Portable	Adherence data collection	Adherence data access	Doctors have data access	PRN Support	Target audience	Power	Cost
SMRxT / Nomi	Text messages	LTE	Unknown	Pill bottle	Yes	Yes	Web portal	Yes	Unknown	Anyone, families, caregivers	Unknown	Unknown
tricella	Bluetooth LE	No	Pill box / drawer	Yes	Yes	Yes	App with pill history	No	Yes	Anyone, families, caregivers	Coin battery	$55 - 75
ProsperSafe	Alarm, lights	None	Yes	Wheel / hopper	Yes, but needs power	No	No	No	No	Addiction recovery / clinics	Wall + backup battery	$400 - 850
Livi	Light, sound, text alerts	LTE	Yes	Hopper	Yes, portable travel packs	Yes	Web portal	No	Yes	Seniors / senior caregivers	Wall	Buy - $1999 Rent - $79/month
AdhereTech	Text, phone calls, lights, chimes	LTE	No	Pill bottle	Yes	Yes	Web portal	Yes	Yes	Anyone	Rechargeable battery	Unknown

Figure 2: The feature comparison chart resulting from our competitive analysis.

While we found several products that are designed to help patients keep track of medication, and one designed to manage dosing for people suffering from opioid addiction, none of these products are designed to guide users through their opioid prescription as they wean through the use of over-the-counter medication like Tylenol. As such, we planned to design our prototype to support two kinds of pills.

Personas

Anyone can face a painful operation requiring postoperative pain medication. Still, we found ourselves asking “Who are we designing for?” To answer this question, we designed three personas:

- A patient who wants to wean off as quick as possible (Liz)
- A patient who wants to manage his pain successfully (Bill)
- A caregiver of a patient (Eleanor)

While we did not base these personas off of user research, we felt they would help guide our decisions through the design phase.

Eleanor

Age: 64
User Status: Caregiver
Identifier: Grandmother
Occupation: Librarian
Location: Bremerton, WA

Eleanor just had her 64th birthday, which was a guided her to thoughts of mortality. She lost her son and daughter in law in a car accident six years ago and has been raising her grandson Tom ever since, Tom has started to make rebelious decisions in his teen years. Eleanor wants the best for her grandson, but worries about him and the friends he has started to keep. Recently Tom was out past curfew with his friends and got into a bad bicycle accident, shattering his lower arm. He just returned home from surgery and Eleanor has to help him manage his pain during post operative recovery. Eleanor is concerned about introducing Tom to such powerful drugs at this impressionable time in his life.

Goals / Desires
- Wants to ensure Tom is taking enough medication to manage his pain successfully
- Wants to ensure Tom is not taking more opioids than he should be
- Wants to help Tom wean off of his opioid prescription

Pain points
- Worries about Tom since his accident
- Doesn’t know if Tom is following his opioid prescription correctly as she leaves for work before he wakes up
- Feels like she is becoming disconnected from Tom as he becomes more independent

Triggers
- Learning that Tom has been in pain
- Finding that Tom has fewer opioid pills remaining than he should

“My grandson has lived with hardships all his life, I don’t want his pain management to be another one.”

Figure 3: Eleanor, a caregiver persona.

Liz

Age: 31
User Status: Patient
Identifier: mother
Occupation: Illustrator
Location: Sammamish, WA

Liz is a full time mother of 2 working part time at home. She has two boys 4 and 7. She is an illustrator and does commision work for children's books and magazines. Liz prides herself on being a great mother, homeschool teacher, homemaker and enjoying her weekends with the family. On the weekend her and her husband of 9 years teach the boys to ski. Liz is a faithful church goer which has helped her stay away from drugs and alcohol all her life. Her father and extended family have had severe addiction issues in the past. When she isn't skiing with the family she is spending time at her mother's helping take care of her elderly, struggling father. Liz recently had some major pains emanating from her stomach, which she ignored for days before finally going to the doctor. She was then immediately taken to the hospital to have her gallbladder removed. She was given high doses of pain relief before, during and after the surgery to cope with the immense pain. This is her first experience with this caliber of pain relief, and she still has a lot of responsibilities to keep up with at home.

Goals / Desires
- Do at least the minimum to stay on top of work and family responsibilities
- Recover quickly
- Get away from opioids as quickly as possible

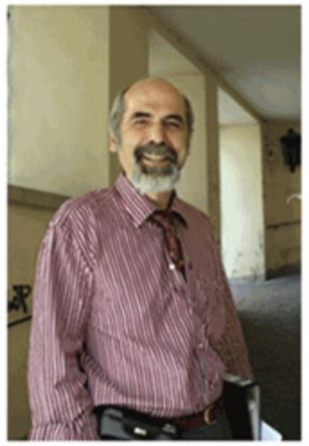
Pain points
- Causing family issues while she is recovering
- Losing her work relationship with her top clients
- Craving pills beyond pain relief

Triggers
- Feeling she is putting her pain relief ahead of work and family

“I don’t have any substitutes, I can’t pause family and work while I recover, I need help.”

Figure 4: Liz, a persona who wants to wean off opioids as quick as possible.

Bill



Age: 52
User Status: Patient
Identifier: Father
Occupation: University Professor
Location: Lynnwood, WA

Bill and his wife both work full time. They each make five-figure incomes that allow them to travel during the holidays with his wife and their two kids. Bill uses the web from work and home. He checks his email and administers online classes. He also looks for events and places that the whole family could visit. He is impatient with the internet because his back gets sore if he sits at the computer too long. Bill manages his high blood pressure with diet. He eats healthy and tries to exercise at least two or three times a week. He uses glasses when he reads and surfs the web. He hates sites with small print because they make him feel old. Bill recently had surgery after a car accident where he suffered a back injury. His wife is often at work, one son is in college, and other son is either at home or with a caregiver, Bill is responsible for his own pain management and opioid prescription

Goals / Desires
- Wants to ensure he is taking enough medication to manage his pain successfully
- Wants to ensure he is not taking more opioids than he should be
- Wants to wean off of his opioid prescription successfully

Pain points
- Worries about getting back to work and helping his family since his accident
- Worries about becoming addicted to pain meds due to his family history (father suffered from alcoholism).
- Worries that his kids may have inherited the family’s “addictive gene”

Triggers
- Being reminded that he is not spending quality time with his family
- Being reminded that he is not working to help support the family

“I’m a big baby when it comes to pain tolerance, I want relief without enslavement.”

Figure 5: Bill, a persona who wants to manage his pain successfully.

Phase I - Design

Project Manager: Michael Beach

Phase 1	Phase 2	Phase 3	Phase 4
Design	**Prototype**	**Evaluation**	**Iteration**
Sketches and Flow Companion App 1.0	Pill Box 1.0 Interactive Demo	Usability Testing Data Analysis	Pill Box 2.0 Companion App 2.0

Mapping Exercise

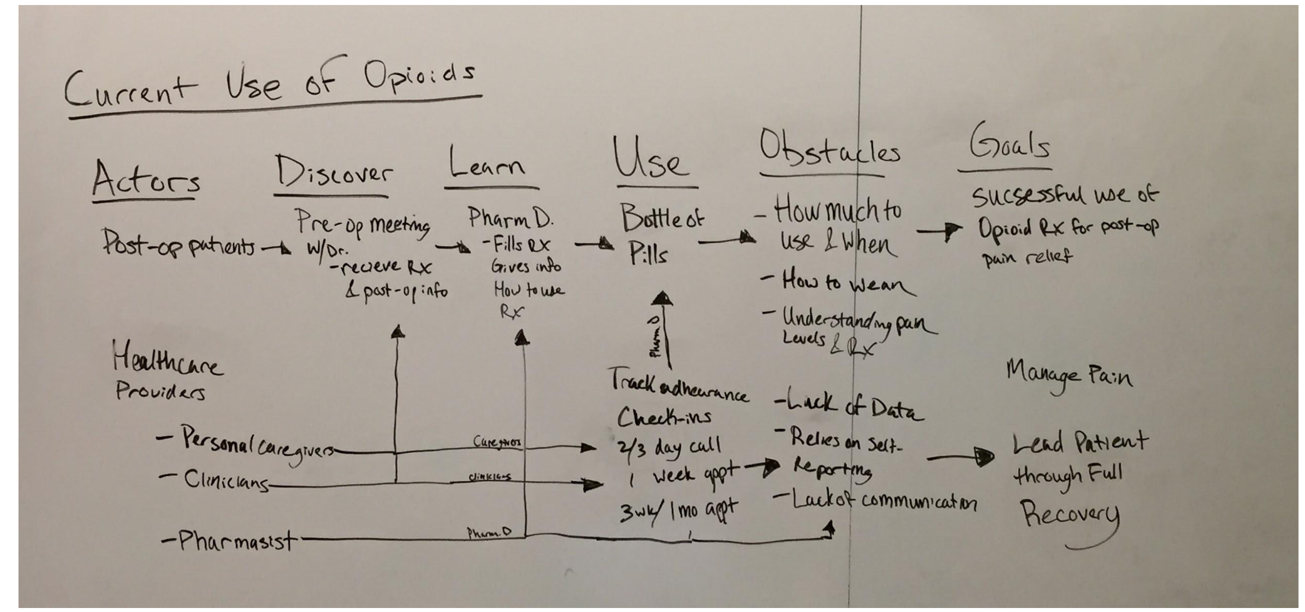


Figure 6: Our mapping exercise identifying the current opioid paradigm.

Purpose

We began the design phase with a mapping exercise in which we sought to map the current state of postoperative pain use. This exercise provided us with scope and foundation to our design sessions and our understanding of our problem space. Specifically, we looked at how stakeholders (actors) currently discover, learn, and use opioid prescriptions, what obstacles they face during this process, and what their goals are with opioid prescriptions.

Exercise

In this exercise, we began by listing all actors who have a stake in opioid prescription acquisition and use: postoperative patients and healthcare providers, including personal caregivers, clinicians, and pharmacists. We then listed how each stakeholder discover, or are introduced to, opioid prescriptions: the patients and caregivers would learn about them in a pre-operative meeting with their doctor. Then we addressed how the stakeholders learn how to properly use their opioid prescription: the patient and caregiver learn from the pharmacist at the time they pick up the prescription. After “Learn” we listed how each stakeholder uses opioid prescriptions currently: the patient has the orange pill bottle, the caregiver checks in a few times a day, clinicians organize an appointment after one week or as necessary, and pharmacists refill the prescription when directed to do so by a clinician. We then listed obstacles or concerns each stakeholder faces: patients are unsure of how many opioids to take and when, how to wean, and how to define their pain levels; healthcare providers face a lack of adherence data and communication, instead relying on self-reporting. As the last entry in this exercise, we listed each stakeholder’s goals: patients want to minimize their pain while successfully weaning off of opioids, and healthcare providers want to manage pain and lead the patient through full recovery.

With this exercise complete, we had identified what obstacles we need to design for in order to support all relevant actors.

Lightning Decision Jam

Purpose
After the mapping exercise, Michael led a lightning decision jam in which we sought to define the scope of our project and its features.

Identifying Problems and Concerns
We started this exercise by brainstorming all problems and concerns we had about the project and product space. We each spent a few minutes writing down every relevant thought we had on sticky notes, which we then sorted into major categories: Use of Device, Device Features, Scope, Housekeeping, and Miscellaneous. This exercise allowed our team to recognize individual team members concerns such that we could address them and keep such considerations in mind moving forward. This step also produced a list of device features to be used in the next step of this exercise.

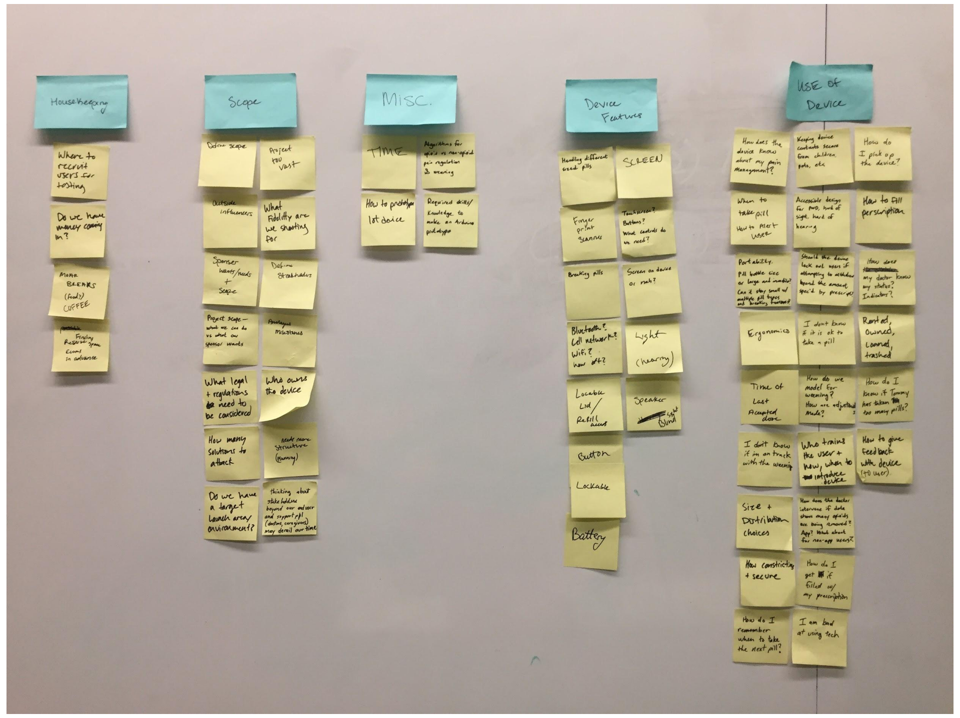


Figure 7: Concerns and problems we identified.

Effort-Impact Scale
We then placed each feature sticky note from our brainstorm onto an impact-effort scale. To do so, Michael held a feature note at the center of the chart while the rest of the team would point right or left for higher or lower effort and up or down for higher or lower impact. This approach allowed us to come to common agreement on how much impact each feature would have and how much effort each feature required.

Figure 8: Effort-Impact Scale, where the bottom left is low impact and low effort.

This exercise resulted in a useful scale, where the top-left quadrant contains the most rewarding features and the bottom-right contains the least rewarding. We deemed the features on the far-right of the graph to be too difficult to implement in the time we had, as indicated by the line separating those features from the rest of the scale.

Concepts and Sponsor Feedback

Purpose

With our priorities established and features defined, we began sketching concepts for how our product should look and function.

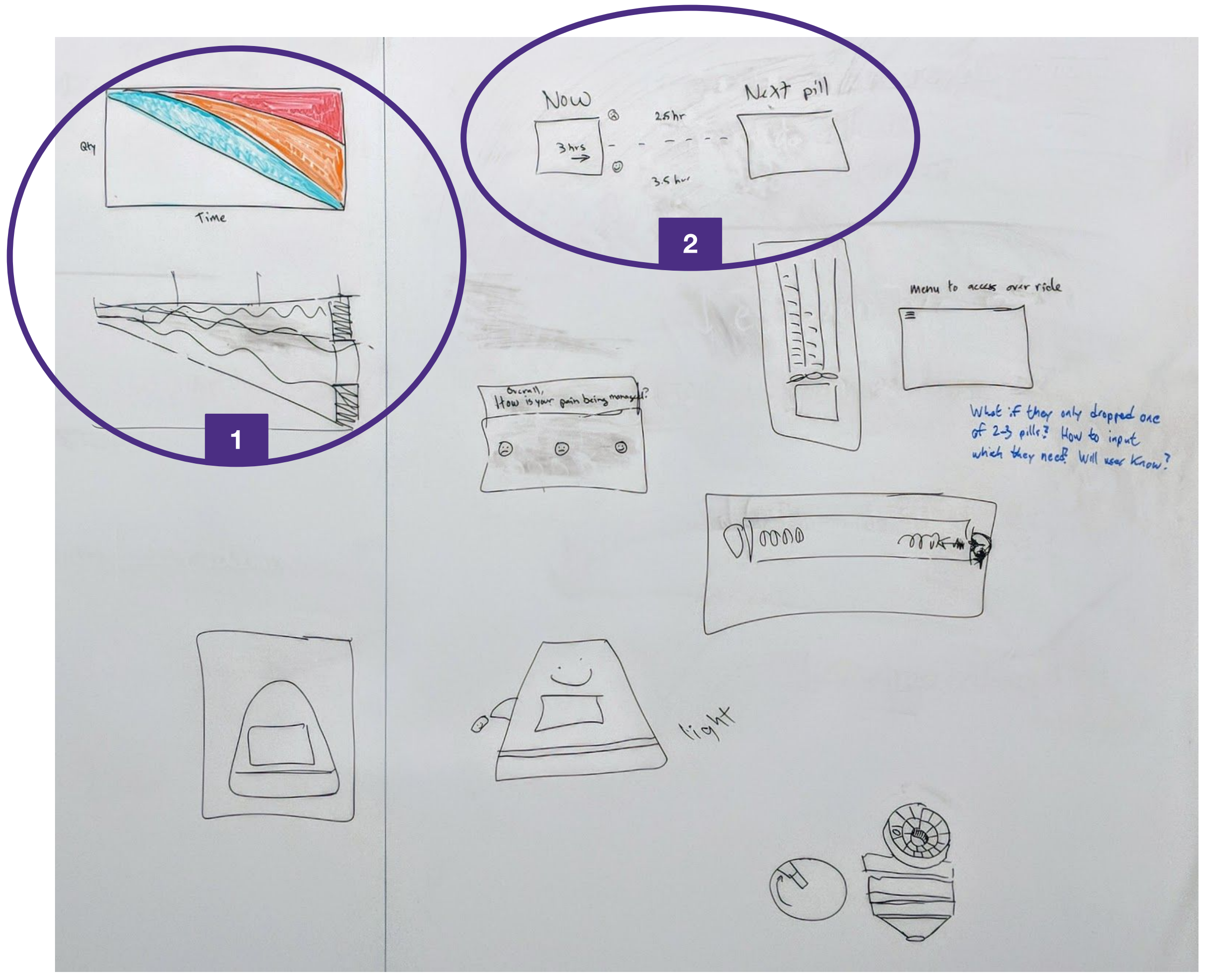


Figure 9: Sketches exploring pill dispensing mechanisms and how the device will use the user's response to pain questions for its dose timing algorithm.

1. This graph, which evolved later into the weaning graph seen in the companion app, shows use of opioids over time. Patients with data points in the green are on track to wean off successfully, while patients in the red may hear from their doctor if they are unable to correct their trajectory.
2. This sketch shows how the device will modify the time until next dose depending on whether the user responds with a happy face (longer duration) or unhappy face (shorter duration).

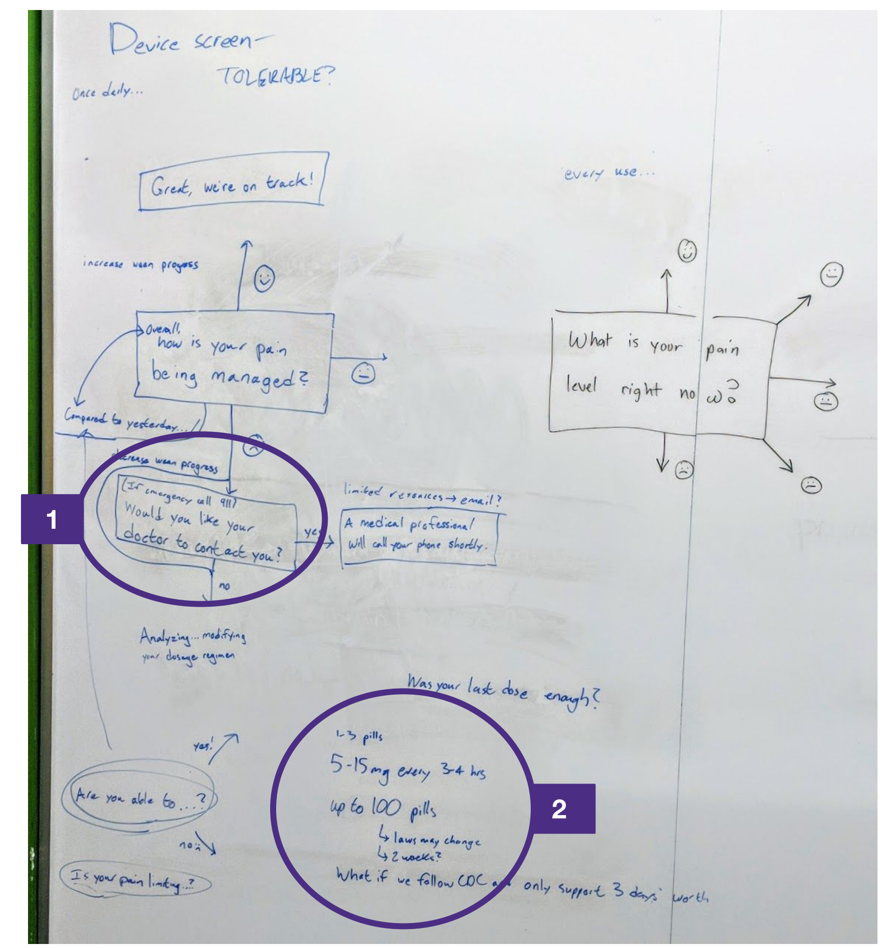


Figure 10: Sketches exploring device interaction flow and how users could use the device to contact their medical care team when in pain. We met with a member of our sponsor team during these sketches.

1. We initially imagined that the device would ask the user if they want their doctor to contact them if they responded with the unhappy face. Our sponsor noted that doctors do not have time to constantly respond, so we later removed this flow in favor of simply recording user responses.
2. We asked our sponsor about typical prescription parameters in order to allow us to design for actual use cases.

Sketches and User Flow

We continued sketching concepts and refining our design through inclusive discussion, considering our priorities and scope.

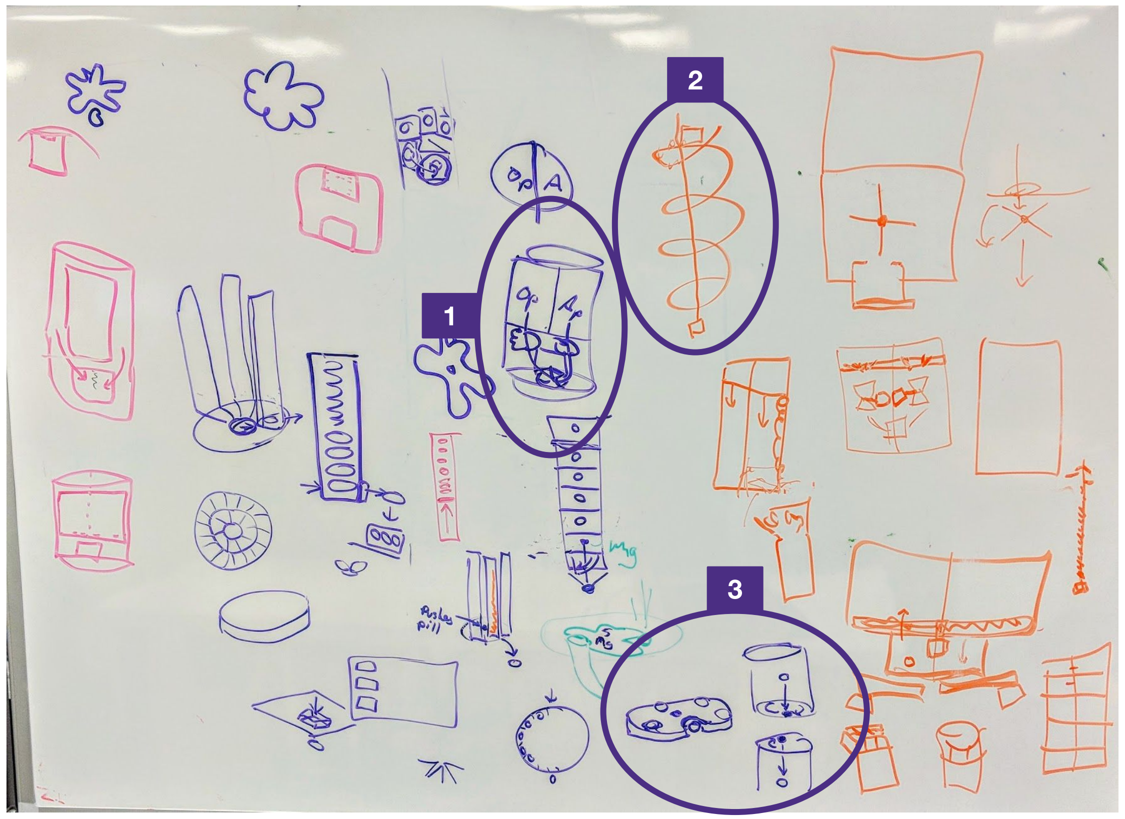


Figure 11: Sketches further exploring and considering the feasibility of various pill dispensing mechanisms and how to support two kinds of pills in one device.

1. The overall form factor of our device that we had decided on, with two compartments dispensing into the same pill collection area.
2. A spiral pill mechanism where pills are lined up one after another throughout the threads of a spiral, with a block on one end that moves downward as the spiral rotates, allowing the device to continue functioning correctly even if shaken or tipped. While my teammates were optimistic about this approach, I pointed out that it would be difficult to model, difficult to print, and difficult to load—I strongly believed this approach would be impossible to produce even though it would dispense pills well. We decided not to take this approach.
3. A rotating gear mechanism where pills fall into a pill-shaped slot which then rotates around to drop the pill into the collection area. We had concerns about this approach consistently dispensing pills when requested.

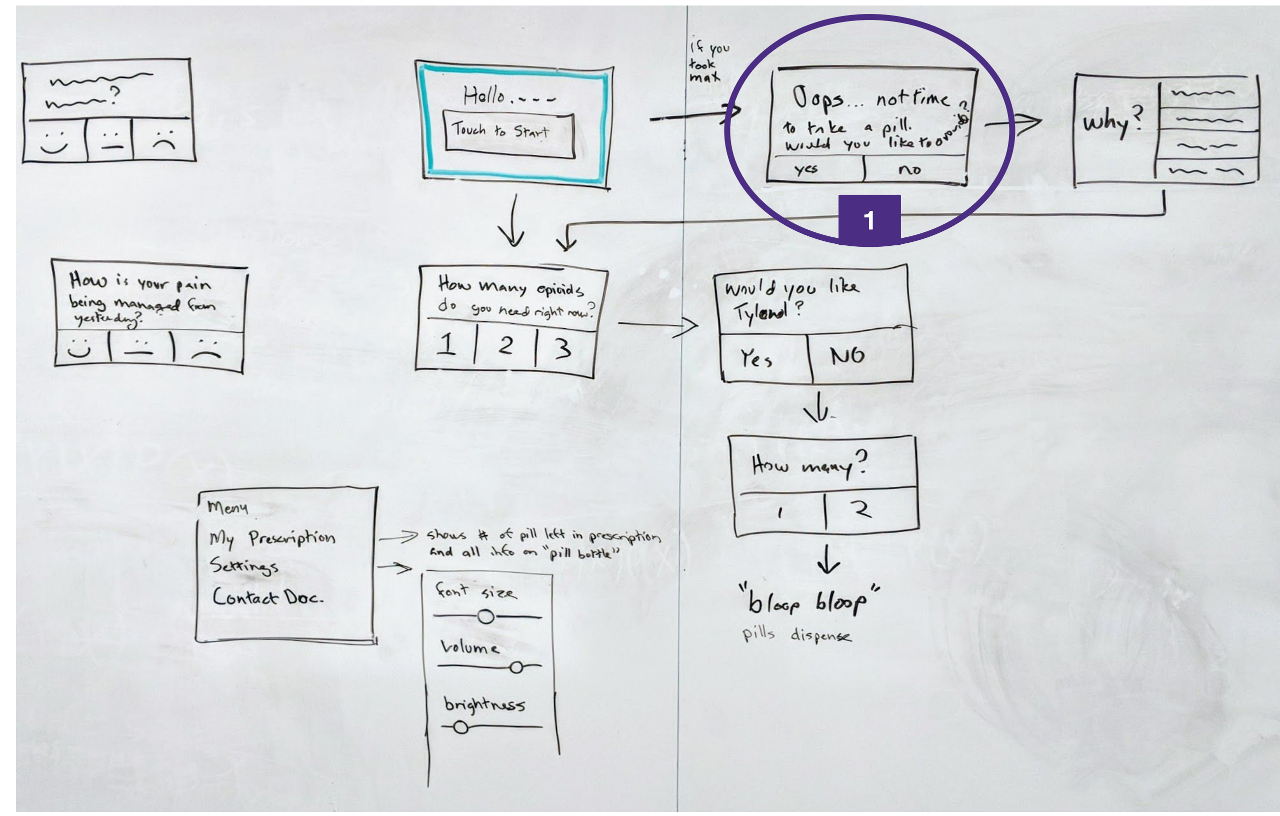


Figure 12: Sketches defining our device screens and interaction flow. In this iteration, we decided that the interaction flow would be the same regardless of the user's response to the pain questions. We considered how to handle users who request more opioids than their prescription allows for. Our team repeatedly came back to the discussion point of whether or not we should lock out users from their opioid prescription, but in the end, we decided to always allow access, as we did not want to stray too far from the existing easy-access pill bottle paradigm. We also discussed how to support users who may need to travel with our device (though travel immediately following an operation is uncommon), but we did not define any solutions to this problem.

1. The "override" screen that is shown to users when they request more opioids than their prescription allows. With this message, we hope to remind users who may have forgotten they recently took a pill, or otherwise at least prompt users to consider that they are exceeding their prescription.

Sketches and User Flow

After a few days off, we returned to our sketches with fresh eyes to finalize our initial designs.

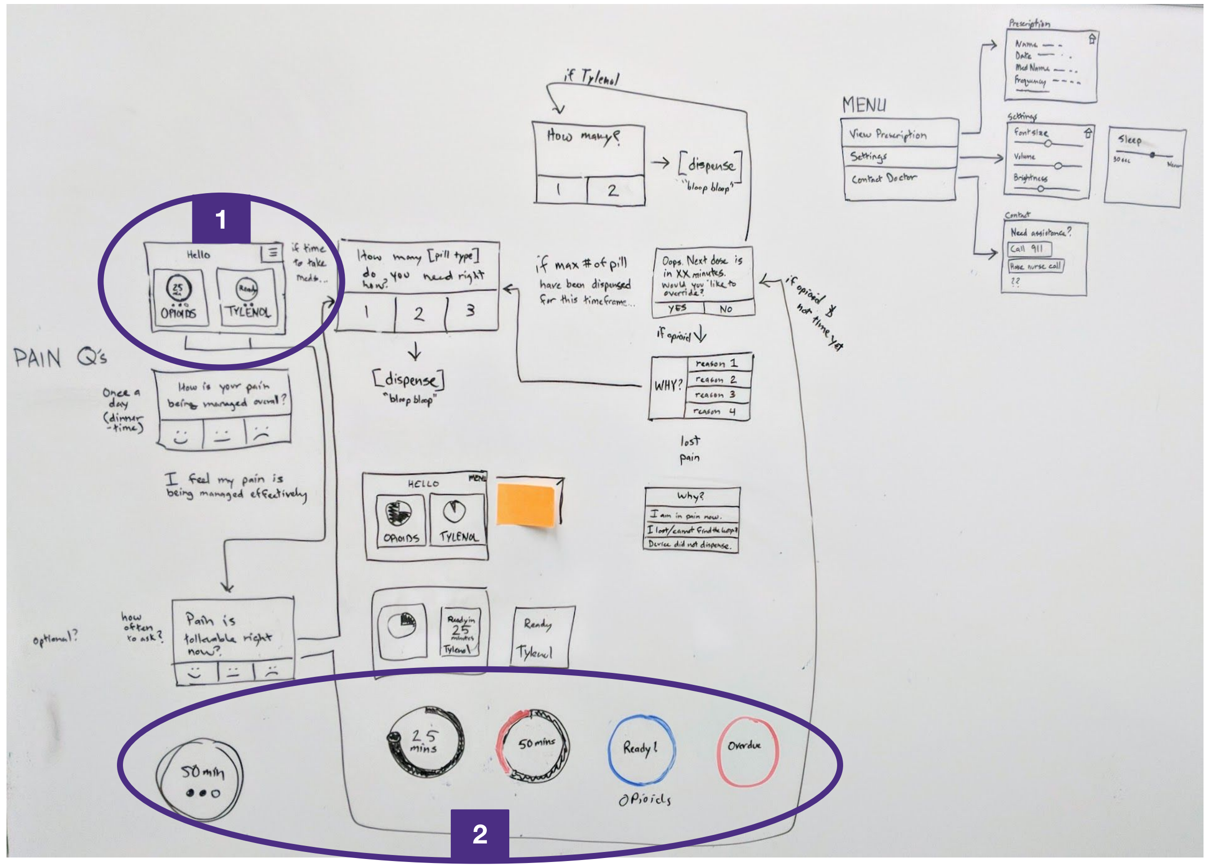


Figure 13: Sketches demonstrating the device screens and interaction flow that we would design in Figma and implement in our first prototype. We also considered here how the home screen of the device could display useful information.

1. The home screen, where users can choose either opioids or their over-the-counter medication like Tylenol. We also added a menu button in the top right from which users could access the menu revealing options to view their prescription, change the device settings, or contact their doctor.
2. Various concepts for the dose readiness information on the home screen. I suggested the idea of a filled arc around the edge of the circle to indicate time progressing until the next dose, a countdown feature I have seen work effectively in video games. With these sketches, we explored how to indicate how many pills are ready and how long until the next dose.

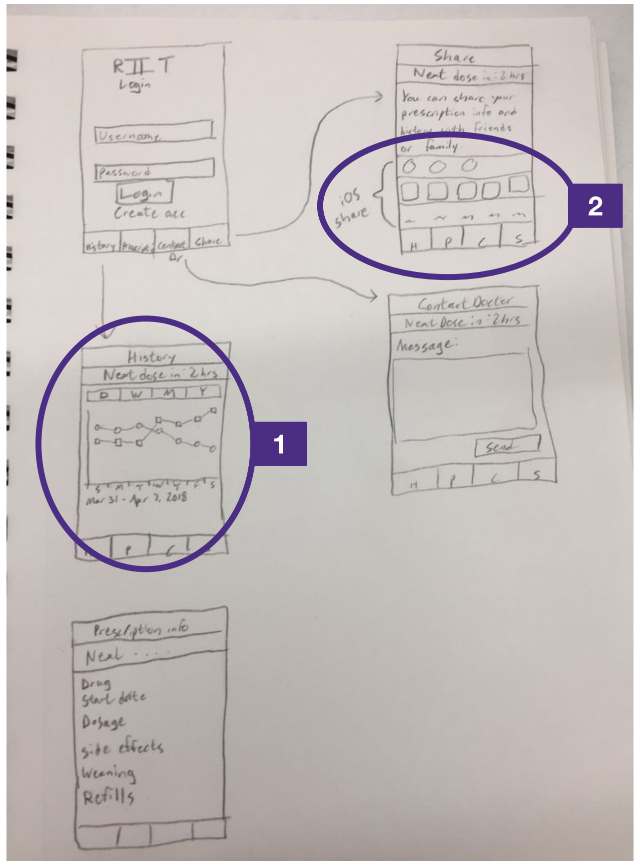


Figure 14: Sketches demonstrating our companion app screens and interaction flow. We considered how the app would fit in with iOS design guidelines and we planned out a login screen that would additionally support iOS' fingerprint login feature. Users would receive or create login info at the hospital and the app would display the user's dosing history, as reported by the device to a cloud server. Users could then share their history and prescription information with friends or family to allow for accountability if desired. We imagined caregivers would get their own login information linked to their patients' account. I produced these sketches.

1. The dosing history screen, inspired by the iOS health app. Michael later merged this concept with his weaning gradient idea, culminating in the weaning graph seen later in our higher fidelity designs.
2. The iOS share interface, allowing the user to pick from their contacts or other apps on their phone to share their information through.

Phase II - Prototyping

Project Manager: Ian Russell

Phase 1	Phase 2	Phase 3	Phase 4
Design	**Prototype**	**Evaluation**	**Iteration**
Sketches and Flow Companion App 1.0	Pill Box 1.0 Interactive Demo	Usability Testing Data Analysis	Pill Box 2.0 Companion App 2.0

Additional Sketching

In the design phase, we primarily focused on the device screen and companion app's design and interaction flow. To finalize our device's physical design and dispensing mechanism, we began this phase with additional brainstorming and sketching as a team.

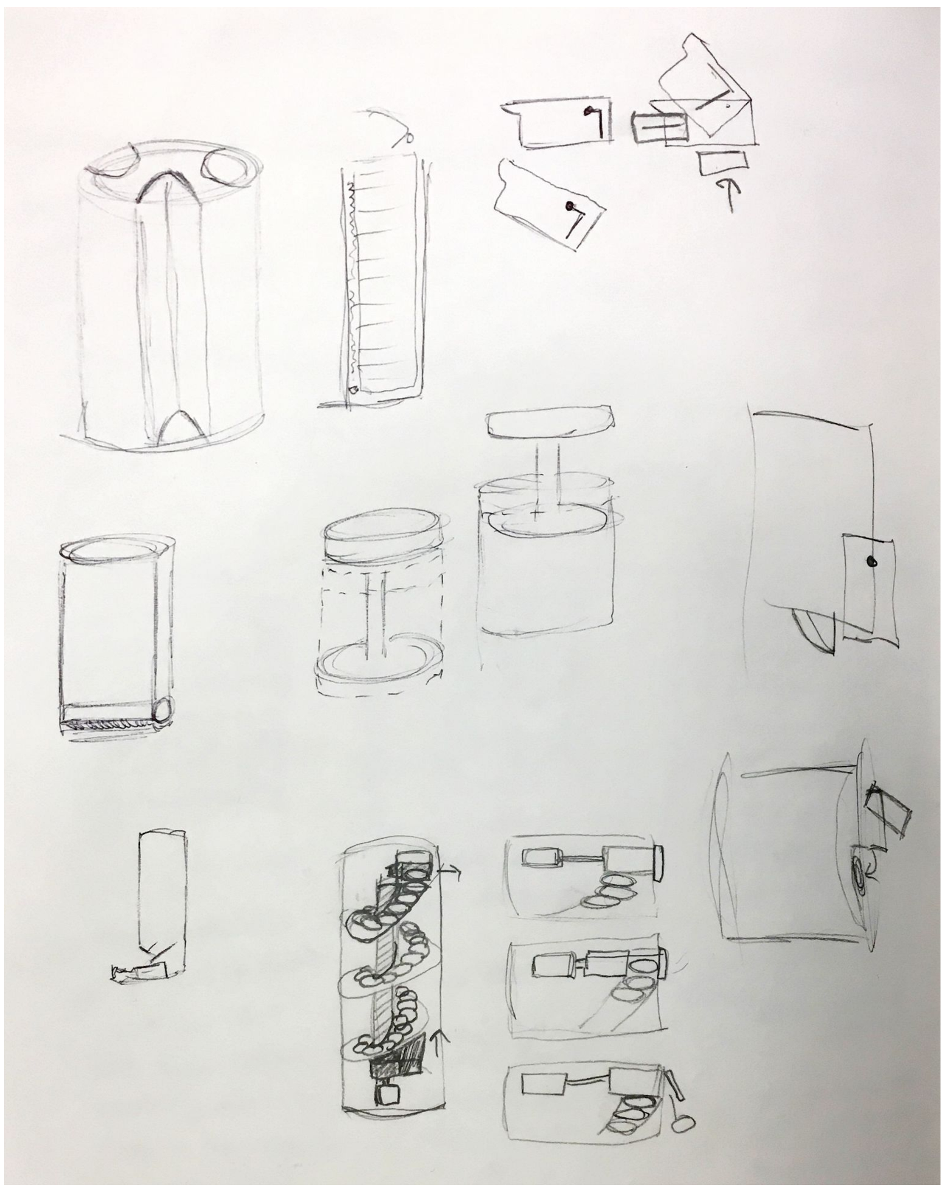


Figure 15: We considered the benefits and drawbacks of various dispensing mechanisms, drawing inspiration from existing applications such as pez dispensers and gumball machines.

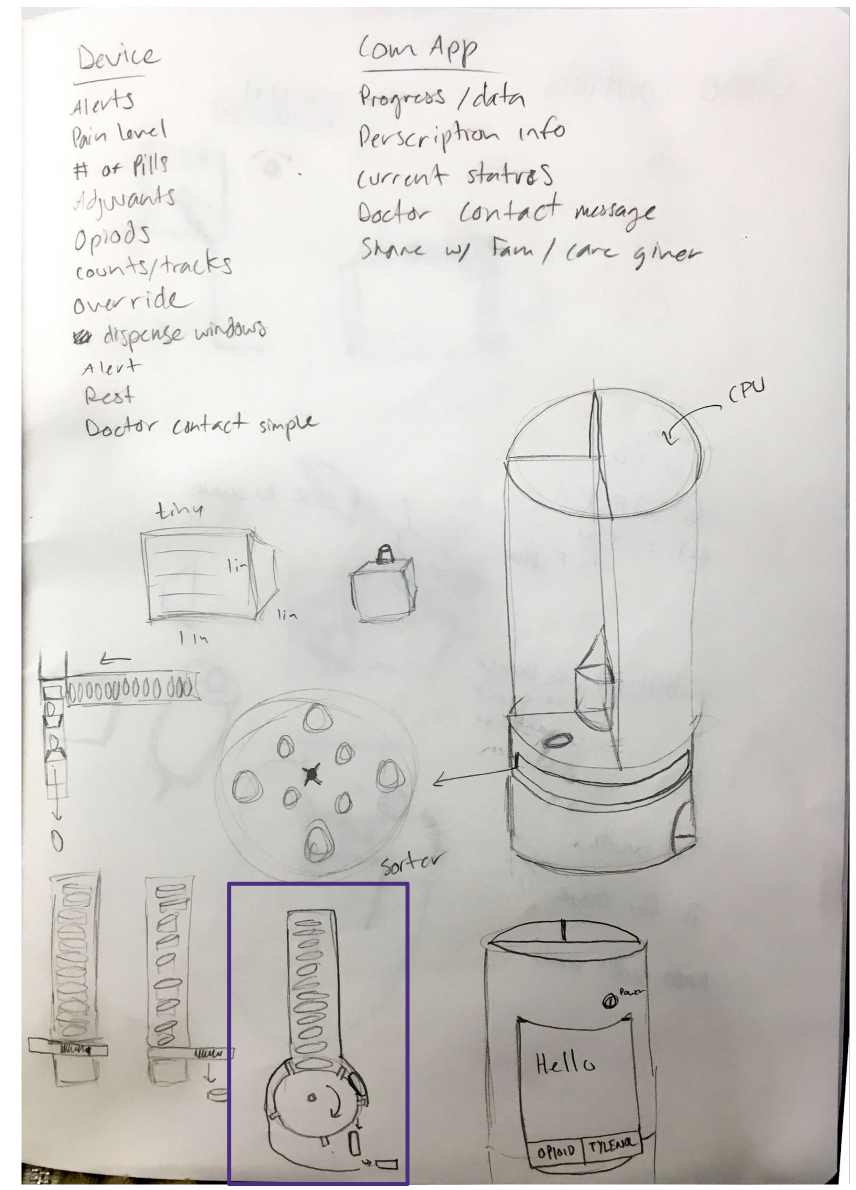


Figure 16: We laid out device and companion app requirements (top) and finalized device design concepts. Highlighted in purple is the gumball-machine-like mechanism we later decided on after recommendation from our TA Wendy, where pills funnel down into a vertically-oriented wheel that can fit one pill, allowing a motor to rotate it to dispense the pill.

3D Modelling

Laser Cut Prototype
Ian, as manager of the prototyping phase, took charge of 3D modelling our device. He began by creating a laser cut version (left) to visualize our device's footprint before he moved on to create a version to be 3D printed.

First 3D Printed Prototype
From the laser cut version, we realized that the entire device needed to be somewhat larger to hold all of the internal physical computing components. The first 3D printed build (middle) achieved this increased space, though at the cost of the size of the pill collection area—we could not fit our fingers into the small gap. Ian explored a horizontal door-based dispensing mechanism here, but we decided that approach would not be viable.

Second 3D Printed Prototype
Ian then redesigned the 3D model to create what would become our first functional prototype (right). We returned to the vertical wheel-like dispensing mechanism that we had identified as most viable in our additional sketching exercise.

Complications
Throughout this phase, several of our 3D prints failed. Thankfully, Ian had to split the model into many pieces in order to print it in a way we could assemble it, so no failed print set us back too far, but we had our eyes opened how finicky and unreliable 3D printers could be. Additionally, the pills would not consistently dispense as they would get stuck sideways, so Michael came up with the idea of building a tube out of index cards to ensure the pills land in the dispensing wheel in the correct orientation. However, this fix limited our capacity to 6 pills per compartment.

Figure 17: The three iterations of our 3D model produced in the prototyping phase. While even the final iteration pictured here was not without its problems, we were able to make it functional for use in our later usability evaluations.

Both 3D printed versions share features such as two large pill-containing areas and a slot in the center for wires to connect the Arduino (housed below) to the screen (placed in the front of the top).

Device Interface

Michael produced medium fidelity digital versions of our device screen sketches in Figma. These screens served as my reference when implementing the first prototype on an Arduino later in the prototyping phase. Michael also created an interactive demo using InvisionApp. [The interactive demo can be found here.]

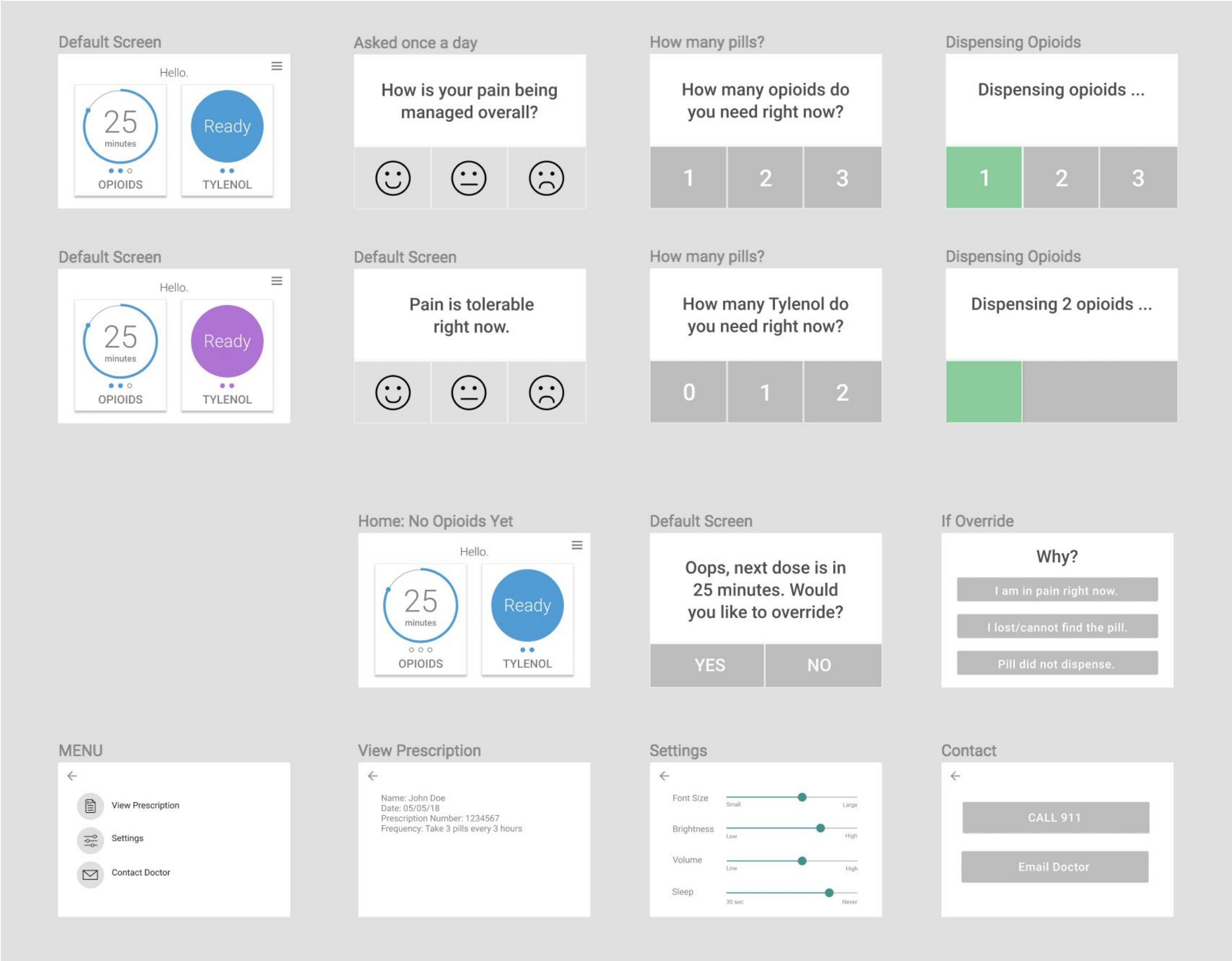


Figure 18: Our full set of device screen prototype screens. In the top left is the home screen with dosing display, where the filled arc represents time until next dose and the dots below the countdown display indicate how many pills are available as defined by the prescription.

Companion App Interface

Michael also produced high fidelity versions of our companion app sketches in Figma and again created an interactive demo using InvisionApp. [The interactive demo can be found here.]

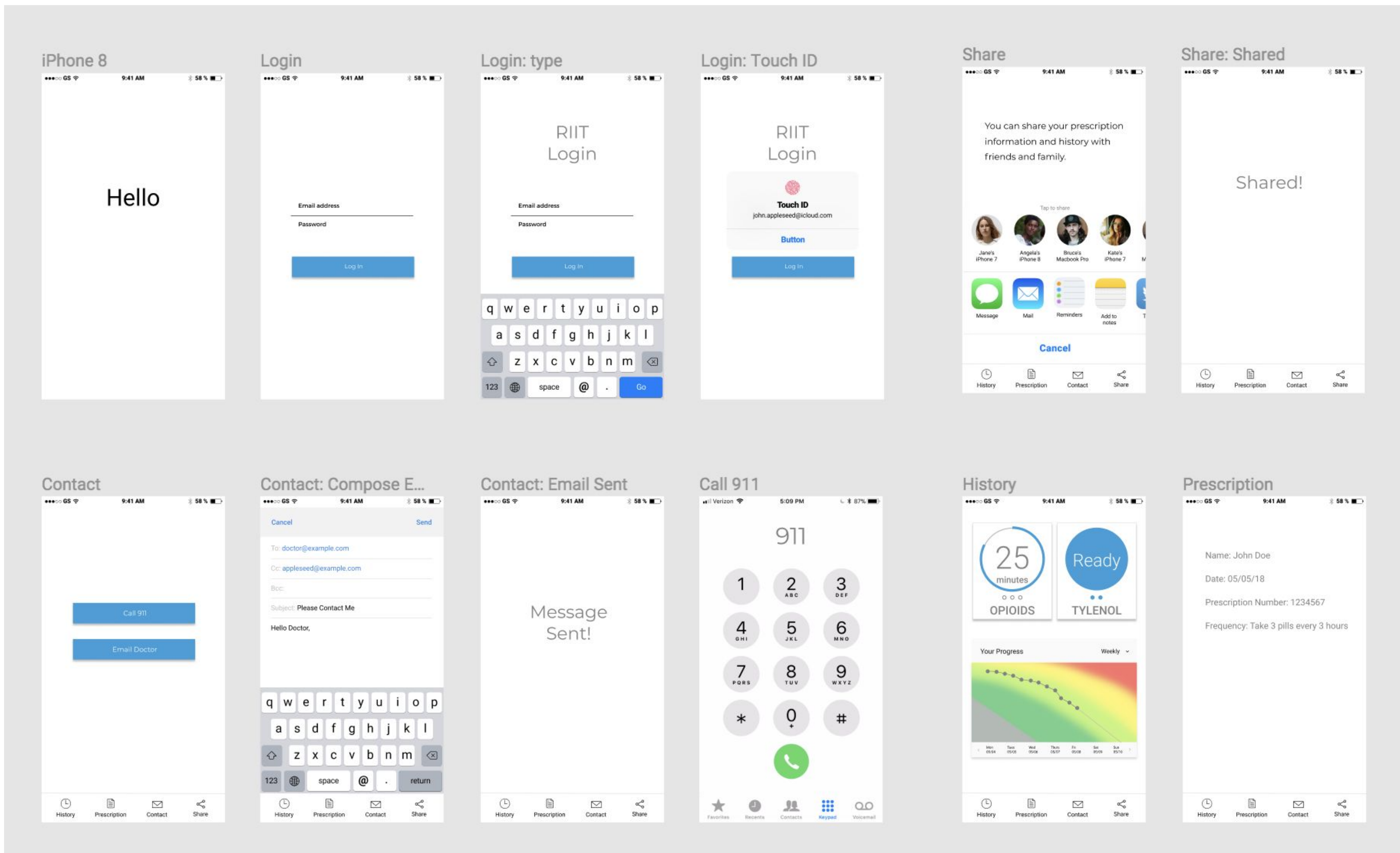


Figure 19: Our full set of companion app screens. Near the bottom left, labeled "History" is effectively the homepage, showing the same dosing information as the device (but cannot activate the device to dispense) as well as the weaning graph demonstrating a user's progress weaning off of opioids versus expectations (green is good, red is bad).

Building with Arduino Physical Computing

Why Arduino?

For the first version of our prototype, we used an Arduino called the TinyDuino, which has a footprint around the size of a nickel. We chose Arduino as Ian and I were simultaneously taking the Physical Computing course, so we had some exposure to its use. Furthermore, Arduino is essentially plug and play, allowing us to get working on the code immediately.

Physical Components

While Ian focused his efforts on the 3D model, I took charge of the physical computing and programming of the Arduino. Initially, I ordered a pair of DC motors (Figure 20), but these were unable to set specific angles, so we instead used a pair of servo motors. I began by assembling the necessary components:

- TinyDuino with battery
- 3.5” touchscreen
- Two servo motors
- Two indicator LEDs

I also connected a speaker and amplifier but removed it before assembling the first prototype due to limited space inside the model.

Challenges

At this point in time, I had very limited experience with physical computing. I learned to solder ahead of my physical computing class in order to use that skill to solder the motor and speaker wires back on after they repeatedly broke off. I also soldered the screen’s header pins on and the power and ground wires to a protoboard (center of Figure 21).

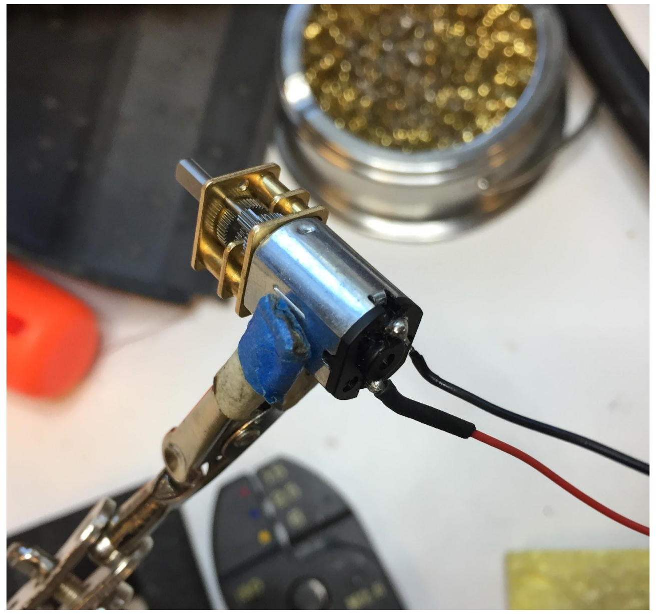


Figure 20: Soldering motor wires.

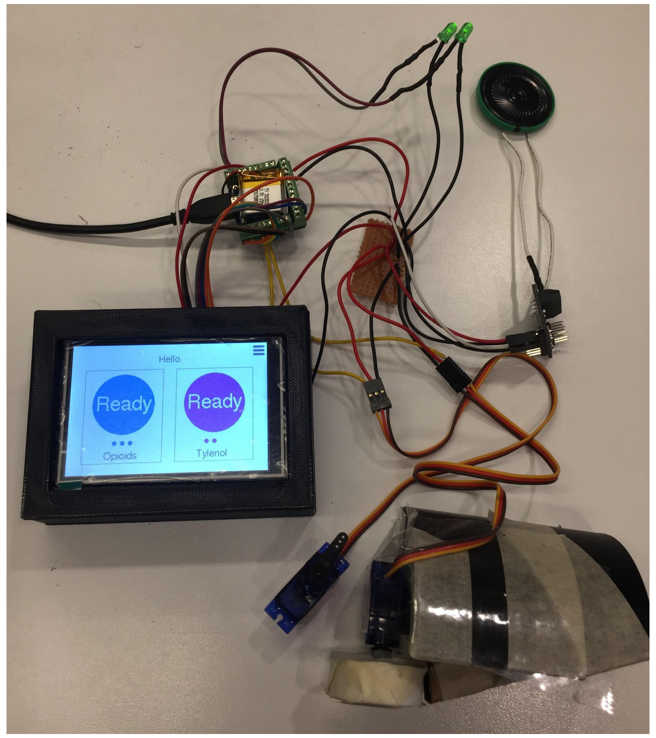


Figure 21: All of the components.

Building with Arduino Microcontroller Programming

Approach

In my team, I was also in charge of all microcontroller programming. As I explored Adafruit's graphics libraries for the 3.5" touchscreen we used, I found that they offered no way to mesh graphics and interaction together in any clean, object-oriented way that would allow us to easily realize our designs from Phase 1.

To begin, I wrote a simple Component object (header file pictured in Figure N) that allowed me to create custom components based off of a common base functionality by just defining when a component is valid and what that component should do when painted, pressed, or clicked. I also wrote custom touch handling code to discern whether the current touch is either a press or a click on any visible component, as the built in library only supplied the current touch simply as a point on the screen any time a touch is present. Beyond that, I broke down all functionality into objects in separate files for easy code management, and I created specific objects to handle a Prescription with *x* doses every *n* hours and a Pill Door mechanism, to name a few (see Figure 22).

Challenges

While I have substantial prior programming experience, this code was my first time working with C and Arduino in any serious capacity. To write this code, I learned on the fly how to properly set up header files and work with C-specific constructs such as pointers, among other quirks of the language. I also found myself breaching the Arduino's maximum memory capacity, so I learned how to do necessary optimizations in order to fit the code onto the Arduino. The limited memory also prevented me from using more than one font in two sizes.

- Colors.h
- CommonGFX.cpp
- CommonGFX.h
- Component.cpp
- Component.h
- Controls.h
- DoseComponents.cpp
- DoseComponents.h
- FadeableLED.cpp
- FadeableLED.h
- LED.cpp
- LED.h
- MenuComponents.cpp
- MenuComponents.h
- PillDoor.cpp
- PillDoor.h
- Prescription.cpp
- Prescription.h
- README.md
- RIIT_TinyDuino.ino
- States.h

```
#ifndef Component_h
#define Component_h

#include <Adafruit_GFX.h>
#include "States.h"

class Component {
  bool needsClear = false;
  bool needsRepaint = true;

protected:
  int x;
  int y;
  int w;
  int h;

public:
  Component(int x, int y, int w, int h);
  int getX();
  int getY();
  int getWidth();
  int getHeight();
  virtual bool isValid(State state);
  virtual void onRepaint(Adafruit_GFX &g);
  virtual void onPress(int x, int y);
  virtual void onClick(int x, int y);
  bool contains(int x, int y);
  int dx(int x); // translate x
  int dy(int y); // translate y
  void paint(Adafruit_GFX &g);
  void repaint();
  void clear(Adafruit_GFX &g);
  void requestClear();
};

#endif
```

Figure 22: Package directory (left), custom Component (right).

Phase III - Evaluation

Project Manager: Finn Thompson

Phase 1	Phase 2	Phase 3	Phase 4
Design	**Prototype**	**Evaluation**	**Iteration**
Sketches and Flow Companion App 1.0	Pill Box 1.0 Interactive Demo	Usability Testing Data Analysis	Pill Box 2.0 Companion App 2.0

Research Questions

Purpose

To provide our usability evaluations with direction, I (as project manager) defined a series of research questions split into device and companion app, as we sought to test the functionality of the two prototypes independently in each evaluation.

Device

1. Do users encounter errors, either as a result of hardware or software design, that impact their ability to successfully acquire medication from the device?
2. Do users understand the dose information on the heads up display?
3. Do users desire any features (additional prescription info, settings, etc.) that we do not have implemented?
4. Are users satisfied with their ability to acquire medication from the device?
5. Do users understand the wording of the pain questions?

Companion App

1. Do users understand the concept of the historical data chart display and prescription progress gradient (the weaning graph)?
2. Do users understand the information hierarchy in the app?
3. Do users desire any features (additional prescription info, settings, etc.) that we do not have implemented?

Methods and Test Kit

Purpose

By defining a test kit, we ensured that our evaluation sessions could be conducted as consistently to one another as possible. Our test kit can be found here.

Screening Survey

We used a “screening survey” to quantify and identify potential participants, asking them questions about if they had previously used prescription medications and if they are a student. As our product could be used by any age group and any demographic, we did not actually screen for any specific qualities here; all respondents were qualified. Their responses, then, could be used to choose a diverse participant pool in order to ideally draw participants who will share a spectrum of experiences.

Test Script

We created a test script, largely modeled after the version Ian created for the HCDE 417 Usability Evaluation team we were on together. This script ensured that each participant would receive the same information so we could focus on answering our study’s research questions. The script mapped every instruction, task, and important feature of the study. Ian, as moderator, read the script to each participant, and we supplied the participant with a copy to follow along, as suggested by our sponsor.

Consent Form

We created a consent form that allowed us to make an agreement with participants concerning how we would use their data in our study.

Pre-Test Questionnaire

We defined a set of questions to ask at the beginning of each study session to collect additional background information.

Task Sheets

We placed each task onto a separate sheet to be printed and brought to each session. This approach allowed participants to clearly see and reference the task being done as well as prevent participants from looking ahead to the next task. Furthermore, we placed post-task questions (asking about difficulty, satisfaction, and perceived helpfulness) on each task sheet such that the participant could fill them out after completing each task.

Post-Test Questionnaire

At the end of each evaluation, we planned to ask questions according to interesting participant actions or comments observed throughout the session.

Recruitment and Study Sessions

Approach
We sought to evaluation our prototypes with a diverse set of 6-8 participants with varied experience using prescriptions and medication dispensers. To collect participants, we posted our survey, noting a $5 incentive, on Craigslist and social media. These approaches proved fruitless, so Ian collected two participants whom he knew personally, and Michael and I conducted “guerrilla” recruiting where we asked individuals on campus if they would be willing to participate, netting us four participants.

Participant Demographics
- 6 participants total
- 4 students (1 Med student, 1 pre-Med, 1 HCDE, 1 Computer Science), 2 non-students
- 2 inexperienced with prescription medication
- 2 married with children
- All male—a possible nonideal result of our guerrilla recruitment approach

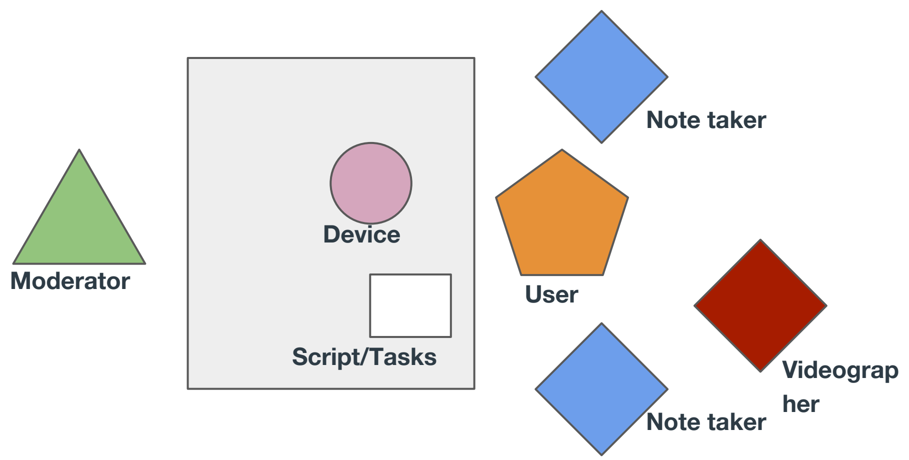


Figure 23: Layout of our usability test sessions.

Study Sessions
We conducted our studies in private rooms in UW libraries. Ali and I took notes, Michael recorded video, and Ian moderated each session. We placed the device in front of the participant seated at a table (see Figure 23).

If any participant had yet to fill out the screening survey, we first had them do so before proceeding with the evaluation. The moderator began by guiding each participant through a think-out-loud warm-up exercise, the consent form, the pre-test questionnaire, and each task, before opening the floor to the team to ask post-test questions.

After the task-based device session, I gave each participant a guided tour of our companion app followed by additional questions. We chose to take this approach rather than tasks for the app due to its simplicity. At the end of each study, we thanked the participant and gave them a $5 Amazon gift card as compensation for their time.

Analysis

Purpose

With data collected from several usability study sessions, we then worked to translate the data into key findings.

Approach

To begin, each team member uploaded their notes to folders in our team drive organized by participants. I then coded our data by defining notable events and checking off each participant who encountered each event (see Figure 24). By noting events common to multiple participants, we could identify root problems that could then be used to define our findings.

Severity Scale

To classify our findings, I defined a three-point severity scale. These ratings would allow us to prioritize design recommendations by impact in the event that we were unable to apply every recommendation.

Post-task forms (1=positive, 5=negative)		P1	P2	P3	P4	P5	P6	Average
Task 1 difficulty		1	1	1	1	2	1	1.17
Task 1 satisfaction		2	1	2	1	2	1	1.5
Task 1 helpfulness		1	1	4	2	1	2	1.83
Task 2 difficulty		1	1	1	1	1	1	1
Task 2 satisfaction		1	1	2	2	2	1	1.5
Task 2 helpfulness		1	1	1	4	1	1	1.5
Task 3 difficulty		1	1	1	1	1	1	1
Task 3 satisfaction		1	1	1	1	2	1	1.17
Task 3 helpfulness		1	1	2	1	1	1	1.17
Event	Spon.	P1	P2	P3	P4	P5	P6	Count X
Trouble on removing pills from device	X	X	X	X	X	X	X	7
Wasn't sure what the device did or was meant to do		X	X					2
Thought pain responses might impact pill availability	X						X	2
Noted or confused by being asked pain question multiple times	X	X			X			3
Thinks pain question could be worded differently	X				X		X	3
Wanted a different way to respond to pain questions	X	X			X	X	X	5
Misinterpreted dosage display information		X		X		X		3
Misinterpreted dosage display interactivity			X			X		2
Screen not detecting taps successfully sometimes			X	X	X	X		4
Wanted more information on/about app graph		X		X	X			3
Commented on device's slow dispensing time				X		X		2

Figure 24: Screenshot of our coding spreadsheet, with quantitative and qualitative data.

Low Severity

A low severity issue can be an annoyance to a user but does not impact their ability to use the system. Issues classified as low severity should be addressed after those classified as medium or high severity.

Medium Severity

A medium severity issue moderately annoys some users and may impact their ability to easily use or understand the system. Medium severity issues should be addressed after high severity issues.

High Severity

A high severity issue significant annoys users and impacts their ability to use or understand the system. Issues with this classification will prevent the system from consideration for mass adoption.

Findings and Design Recommendations

Approach

To convert our coded data into findings and design recommendations, we identified common root causes between events, comments, and feedback noted throughout evaluation sessions. Once we grouped common root causes into findings and classified their severities, we discussed possible solutions and defined design recommendations such that we could work effectively in the following iteration phase.

Device Design Recommendations

Finding	Severity	Design Recommendation
Many aspects difficult to understand for first time users	High	Make a printed Quick Start Guide to help clarify the device's affordances and use for first-time users
Difficult to collect pills from collection cup	High	Redesign pill cup to be larger and smoother for easier pill retrieval
Pain question asked too frequently	High	Adjust software to only display current pain question at most once per 15 minutes
Device tilts when screen is pressed hard	Medium	Add weight to the device to prevent tilting and tipping
Unsatisfactory pain response options	Medium	Add an additional emotive face to the pain question response options
Slow pill dispensing time	Low	Decrease the time it takes for the device to dispense pills
Unclear pain question	Low	Change the wording of the pain question to ask how bad pain is rather than how tolerable it is

Figure 25: Summary of device usability findings and design recommendations.

Companion App Design Recommendations

Finding	Severity	Design Recommendation
Context needed for weaning graph	Medium	Add y-axis label and context to the weaning progress graph
Visual clarify wanted for app hierarchy	Medium	Update the app aesthetic to better convey hierarchy

Figure 26: Summary of companion app usability findings and design recommendations.

Phase IV - Iteration

Project Manager: Ali Morgan

Phase 1	Phase 2	Phase 3	Phase 4
Design	**Prototype**	**Evaluation**	**Iteration**
Sketches and Flow Companion App 1.0	Pill Box 1.0 Interactive Demo	Usability Testing Data Analysis	Pill Box 2.0 Companion App 2.0

Quick Start Guide

Purpose
Based on the findings of our usability evaluation, we identified the need for a Quick Start Guide that users could refer to as they become familiar with the device, similar to what typically comes with consumer appliances/electronics.

Approach
Michael and Ali collected example user manuals and conducted a lightning decision jam, similar in format to the one Michael conducted in the design phase, to identify priorities for the guide before they began sketching ideas. Michael then produced a high-fidelity version of the guide that the entire team provided suggestions and feedback on. While I was absent for the lightning decision jam, I offered several suggestions on making the guide seem more professional (specific wording) and easier to read (increased contrast and font size), with hard-of-sight users in mind.

Features
With the Quick Start Guide, we mainly sought to answer several questions common to multiple participants, such as the meaning of the home screen display, the meaning of the weaning chart, and whether or not their response to the pain question would impact their dose. The guide contains the following:

- Device overview
- Getting started instructions
- Weaning chart explanation
- Companion app introduction
- Frequently asked questions

Figure 27: The Quick Start Guide, which folds in the middle.

Companion App Iteration

Michael updated the companion app to include our design recommendations.

Changes

- Clarified app hierarchy with an updated aesthetic
- Added y-axis, context, and clarify to the weaning graph to make its function more useful and understandable
- Added option to customize the contact page with new contacts, such as friends or family members
- Added option to customize the data points shared via the share page

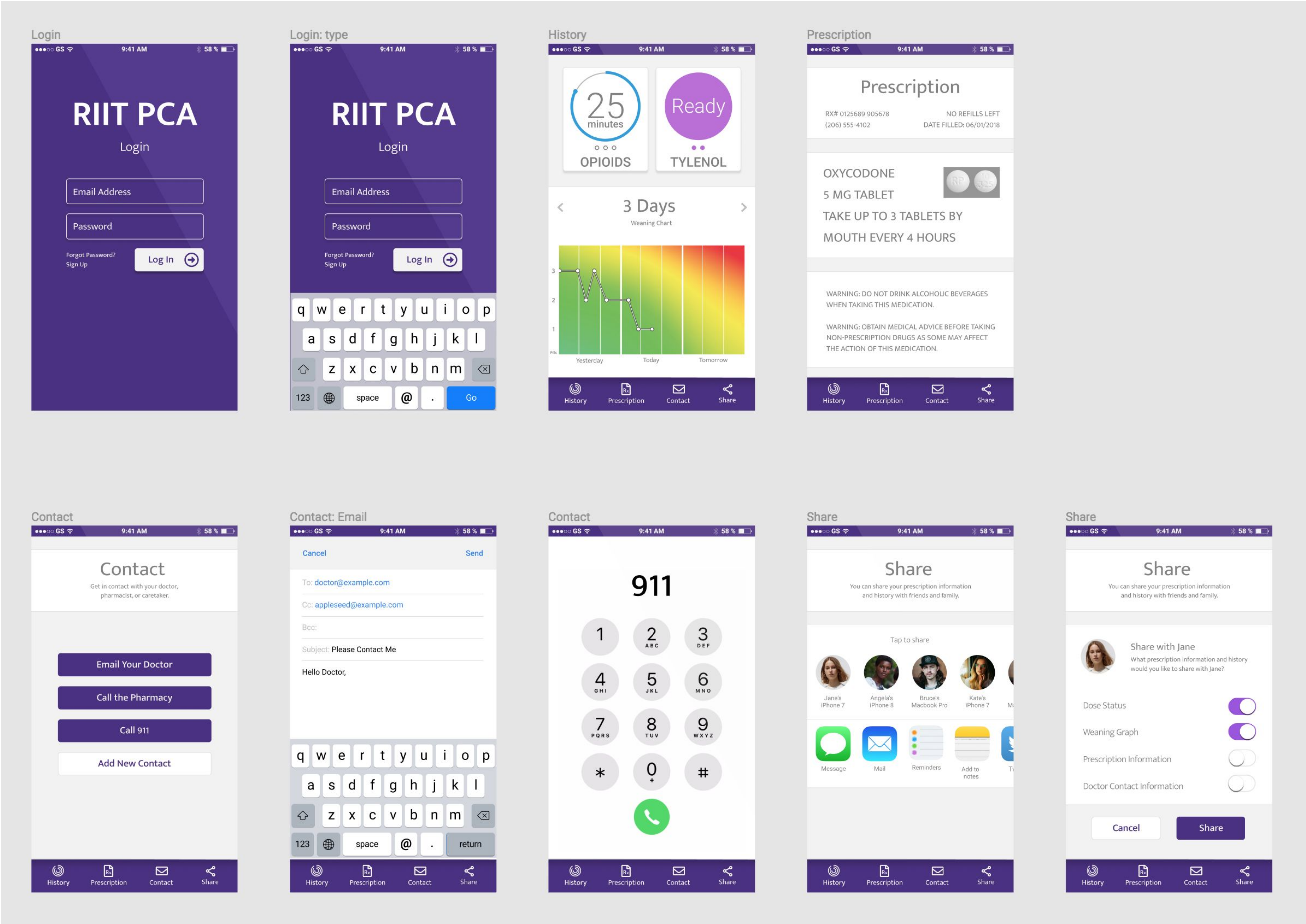


Figure 28: Companion App 2.0 screens.

Device Iteration

Ian redesigned the device 3D model from scratch incorporating our design recommendations as well as working in a new pill dispensing mechanism that the team came to after discussing how to improve our functional pill capacity from 12 pills. We designed a 15-slot horizontally rotating cartridge where each slot is a tube that holds 6 pills, allowing the device to hold 90 pills in total. Ian additionally explored solutions to this new cartridge holding two kinds of pills, though we were unable to successfully get that feature working. When we tested the new design, we found that pills would roll and jump out of the cup due to the angle they are dispensed at. To fix this issue, I added a dot of hot glue to the inside center to break each pill's roll.

Changes

- Increased pill capacity from 12 to 90
- Increased pill dispensing speed
- Redesigned pill cup for easier pill retrieval
- Increased device stability by widening base
- Added a fourth face to the pain question response options
- Reworded and reduced the pain question frequency

Figure 29: Device 2.0 model render.

Building with Raspberry Pi Physical Computing

Why Raspberry Pi?

While the Arduino worked well for a first prototype, it is a fairly weak microcontroller, and this was evident in both its limited memory and the screen taking 1-2 seconds to refresh on every click. For the second prototype, I made the decision to use a Raspberry Pi Zero W:

- Much more powerful, allowing for a quick, responsive screen
- Python, which is easier to work with than C (for me) and offers better graphical libraries
- WiFi connectivity for uploading prescription use data
- The Pi mounts directly onto the screen, allowing for a smaller footprint

Changes

- Rebuilt entire codebase in Python with more maintainable code
- Added audio feedback via speaker and amplifier
- Ability to add settings controls like volume, brightness, font size sliders
- Communicating ready dose via screen and audio instead of indicator LEDs
- One stepper motor (powered by a ULN2003 integrated circuit) and one servo motor.

Challenges

Unlike the Arduino, the Raspberry Pi requires a full Linux operating system. Few things are plug and play, instead requiring drivers installed via the terminal, and I had to set up the Raspberry Pi connect to my home WiFi or phone hotspot so I could connect to it from my laptop on the same network in order to install and run drivers and my code. During this phase, I spent an entire week reinstalling the operating system and drivers and trying countless approaches before I finally had the touchscreen's graphics and interaction both working such that I could start programming. Still, to run the prototype, I must connect to the Pi over WiFi (via SSH specifically) and manually tell it to run the code, though it is possible to set it up to run it on boot.

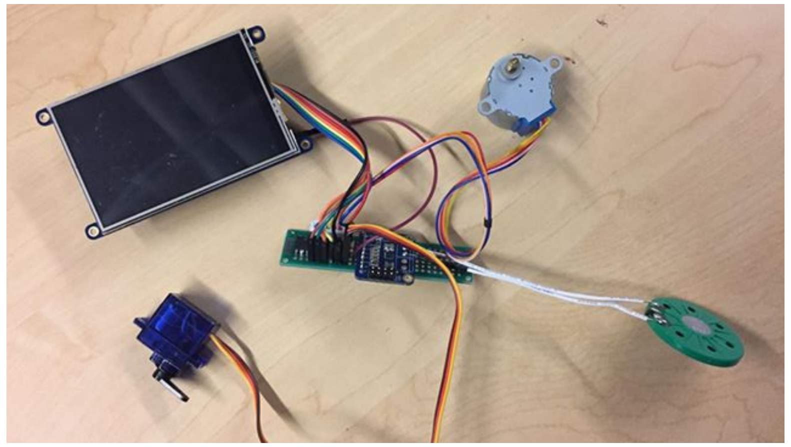


Figure 30: The screen, servo motor, stepper motor, speaker, and soldered protoboard containing integrated circuit and amplifier.

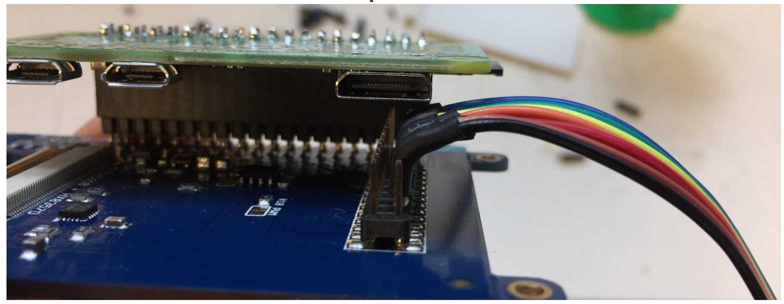


Figure 31: I bent jumper wires to fit in the gap.

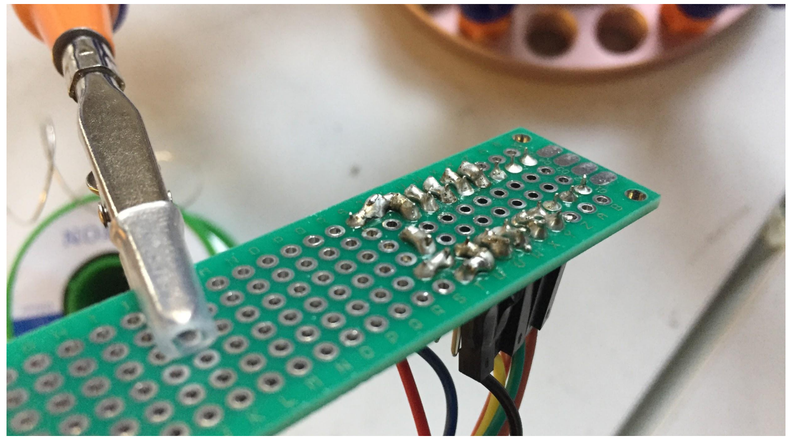


Figure 32: Soldering the integrated circuit and motor headers to protoboard.

Building with Raspberry Pi Microcontroller Programming

Approach

While Python offers far more comprehensive libraries than Arduino, I found myself limited to one library called pygame due to the way the 3.5” Pi screen operates. This library offers everything necessary for drawing graphics, playing audio, and more, though it does not offer an object-oriented way to handle scenes and components. I converted my Arduino Component class over but also created a container to hold multiple components, called a Scene. By mapping device states to scenes, I created a clean UI codebase that is easy to maintain and easy to add to (see Figure 33). I used a library called pigpio to handle writing to the Pi’s physical pins to control the servo motor, stepper motor, and backlight.

As I was no longer limited by the Arduino IDE’s inability to use files in more than one folder, I organized the project code cleanly between a handful of packages, splitting code into physical components (like motors and backlight), prescription logic components, common UI components, scenes (collections of specific UI components), and utilities (see Figure 34).

Challenges

Although I had some prior experience with Python (in HCDE 310 writing server code that draws from multiple APIs), I had never written object-oriented code in Python. Like with coding C and Arduino, I constantly referred to StackOverflow and other online resources to develop my understanding of proper Python syntax to define classes and work with Pythonic constructs such as tuples and dicts. Overall, I found working with Python to be much easier than working with C and Arduino, and the resulting code came out far cleaner and more future proof.

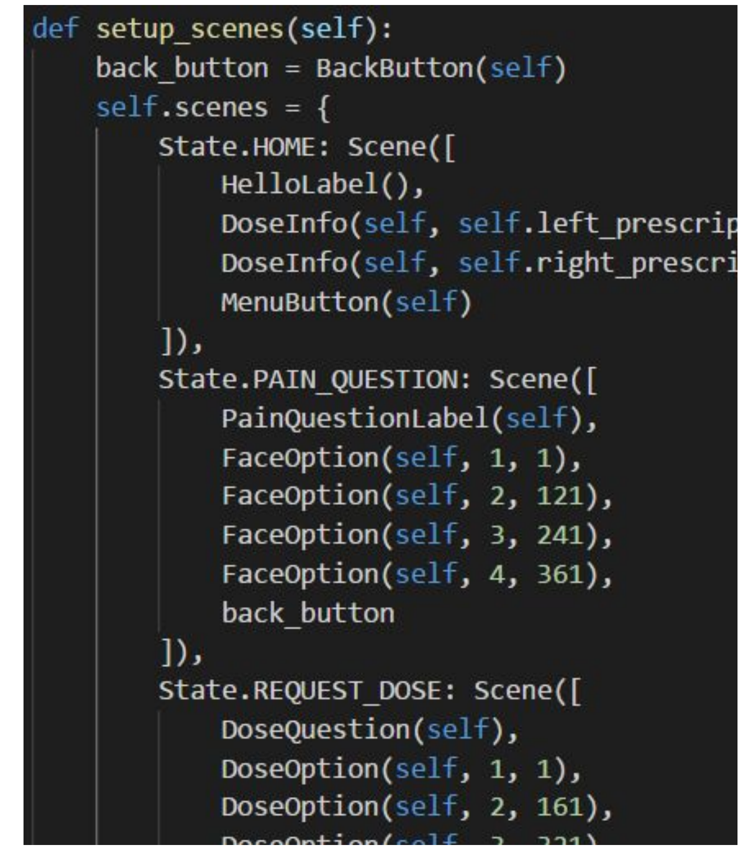


Figure 33: Scene setup code (top), where each device state is mapped to a scene created from multiple components.

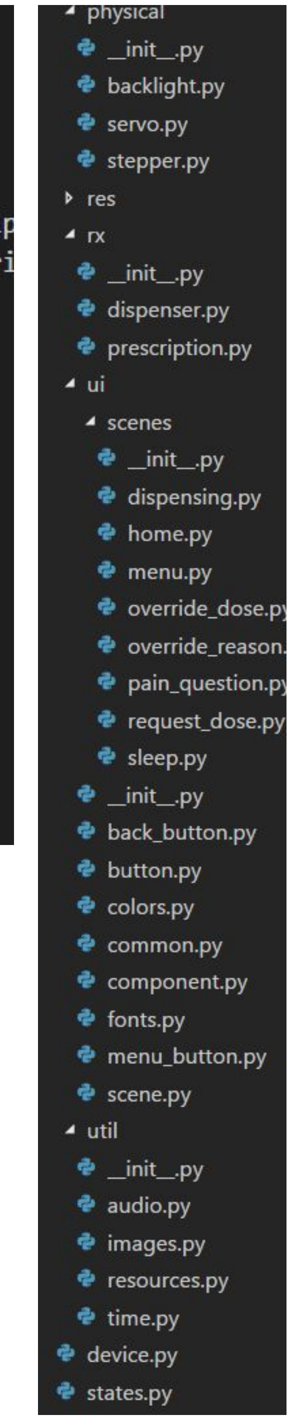


Figure 34: Project package layout (right) with directories for physical components, prescription components, UI components, scenes, and utilities.

Reflection

Reflection

Our Capstone project was defined by learning on the fly and hoping our methods would work. As a team, we decided to mark this project as our first choice both for the product's impact in our world and for the challenge it offered—none of us had the experience with physical computing and 3D modeling necessary to easily tackle this project. While Ian had taken HCDE's Internet of Things course and otherwise had more experience 3D modeling, as a member of the 3D printing club, than the rest of our team, who had only been exposed to it in HCDE's User Experience Prototyping course. This project required that we seamlessly blend the intersection of 3D modeling, physical computing, and UI/UX design such that we could create a functional mechanical project with a user interface that actuates and reflects the state of the physical components encapsulated within.

When I volunteered, as the member of our team with the most experience programming, to take charge of the physical computing and software development, I worried that I was getting myself in over my head. I made this decision in our Capstone Planning phase, knowing that the first two weeks of Spring quarter in HCDE's Physical Computing course would be my only experience Arduino before our prototyping phase where I would need to wire a screen and other components, solder protoboard, and create a comprehensive codebase written in C, which I had never used before this quarter. I soon discovered that we would not learn soldering until Week 4, after we sought to have our first functional prototype, but thankfully Dr. Brock Craft generously taught me how to solder when I asked in Week 2 and he continued to provide helpful tips—and the protoboard I soldered components to—when I asked him questions. Without his help, I would have not been able to achieve what I did in the prototyping phase.

One of the biggest surprises for me in the prototyping phase was that the Arduino was far slower than I had realized. I originally envisioned creating an interactive prototype directly from our Figma screens by defining clickable areas on each screen image that direct to another screen, but I discovered early on that it took a whole 10 seconds for the Arduino to draw an image covering the entire screen! I realized then that my use of images would be limited to icons, but once I rotated the screen orientation from portrait to landscape, I found that Adafruit's image drawing code is broken at any non-default orientations. Consequentially, I had to create every line and shape on first prototype's screen from scratch, defining code to draw an arc of a circle or a smiley face as a combination of an arc and circles, for example.

During the iteration phase, I began exploring if a more powerful Arduino existed that we could easily use in our second prototype to create a more responsive screen. At this point, I remembered the Raspberry Pi (as I have a model from several years ago), and I learned about the Raspberry Pi Zero W, which seemed like a perfect solution for its much more powerful processor, WiFi connectivity, and ability to use other programming languages. I spent a week researching the feasibility of our project on the Pi, looking up the required screen and code/software examples for setting up the necessary motors and audio that we would need. Over this week, I wavered back and forth on whether I thought it was possible or not, but finally I got to a point where I believed it was feasible, though I still gave it a 10-20% chance of failure, either due to unforeseen complications or simply requiring too much time to completely rebuild the physical computing components and rewrite the code from scratch in another language I had little previous experience with (Python). I shared with my team what I believed the pros to be and the chance of failure, and they told me that if I thought it would be worthwhile, go for it—so I took the risk and ordered the parts, committing myself to what would inevitably be another extremely time consuming prototype on my end.

Once the Raspberry Pi parts arrived, I began setting up the necessary operating system and drivers. As I could only connect and control the Pi via a shared internet connection, I ended up "bricking" the Pi several times, where it would no longer boot to the point where it would connect to WiFi and allow me to control it. I ended up reinstalling the operating system, and different ones each time at that, around five times. After a week of trial and error approaches to setting up the Pi as I needed it to even *begin* writing the code from scratch in Python, I finally got it working—though I now had little over a week to build everything on my end, software and hardware.

In the end, I was able to finish my end of the project in time for our second prototype and the final deliverable. While rewriting my code in Python, I took the opportunity to take a far better approach to the code and create a much cleaner, easier codebase to work with. Now, I can say I am proud of how both versions of the code (first prototype in C, second prototype in Python) came out, and I am glad to have these as portfolio pieces once we determine with our sponsors whether I can publish the code publicly or not—it is for this reason that the codebases are not included or linked in this process book.

Looking back, I know that my end of this project would have been far more manageable had I more previous experience with physical computing, C/Python, or Raspberry Pi. Despite that, I am glad that my team and I aimed high and achieved our primary goals. I got more out of this capstone project than I ever could have imagined coming into it, and I am thankful that my team had the opportunity to work on a challenging 3D project rather than a research and app design project.

Finn Thompson

References

[1] “The United States for Non-Dependence: An Analysis of the Impact of Opioid Overprescribing in America,” QuintilesIMS Institute, Parsippany, NJ, USA, 2017.

[2] Center for Behavioral Health Statistics and Quality, “2015 National Survey on Drug Use and Health,” Substance Abuse and Mental Health Services Administration, Rockville, MD, USA, 2016.

[3] R. A. Rudd, P. Seth, F. David, L. Scholl, “Increases in Drug and Opioid-Involved Overdose Deaths - United States, 2010-2015,” Morbidity and Mortality Weekly Report, no. 65, pp. 1445-52, U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, Atlanta, GA, USA, 2016.

[4] H. Hedegaard, M. Warner, A. M. Miniño, “Drug overdose deaths in the United States, 1999-2016,” NCHS Data Brief, no. 294, U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, Hyattsville, MD, USA, 2017.

[5] L. Palen, S. Aaløkke, "Of pill boxes and piano benches: home-made methods for managing medication,” in Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, Banff, Alberta, Canada, November 4-8, 2006, pp. 79-88. Available: ACM.

[6] E. Ermisoglu, “Simulation of Mobile Treatment Monitoring System,” Thesis, Department of Computer Science at the Donagley College of Engineering and Information Technology, Little Rock, AR, USA, 2011.

[7] H. Y. Jeong, S. Y. Park, J. Zimmerman, “Opportunities to support parents in managing their children’s health,” in CHI'08 extended abstracts on Human factors in computing systems, Florence, Italy, April 5-10, 2008, pp. 3225-3230. Available: ACM.

[8] U. Varshney, "Mobile Health: Medication Abuse and Addiction,” in Proceedings of the 4th ACM MobiHoc Workshop on Pervasive Wireless Healthcare, Philadelphia, PA, USA, August 11, 2014, pp. 37-42. Available: ACM.