Antidose

Seng 499

June 12 2017

University of Victoria

Group Information

Name	V Number
Graeme Bates	V00810000
Heather Cape	V00795197
Jake Cooper	V00794804
Geoff Lorne	V00802043
Jordan Vlieg	V00779758
Tanner Zinck	V00801442

Contents

I Objectives

The core objectives for this project include, but are not limited to:

Overall Objectives:

- Use technology to help curb BC's overdose crisis.
- Connect naloxone kits to people having an overdose.
- Reduce the time for a first responder to arrive on scene.
- Reduce the number of overdose deaths in BC per year
- Create a mobile application to achieve above objectives

Member Roles:

Graeme Bates - Android Developer
Heather Cape - User Interface Designer
Jake Cooper - Server Developer
Geoff Lorne - Database Designer
Jordan Vlieg - Android Developer
Tanner Zinck - Database Designer

Application Deliverables:

- Include a map and navigation to help Naloxone carriers find overdose victims.
- Audio signal/alarm to notify people in vicinity of ongoing overdose.
- Automatically contact emergency services for aid.
- Provide lifesaving information to first responders on how to deal with an OD.
- Provide a streamlined one touch solution for those in need of assistance.

Class Deliverables:

- Class presentation pitching the idea to our peers.
- Class presentation updating our peers on our progress, and providing more details on the project.
- Written progress report submitted to both the supervisor and the coordinator.
- Final report detailing design and implementation of the project.
- Final presentation showcasing the features and abilities of the product.

II Introduction

British Columbia is currently in the midst of an Opioid overdose crisis. 2007 saw approximately 200 overdose deaths, and in 2017 the number is already at 900. Over 60

percent of overdose fatalities in 2017 have involved fentanyl[1]. Fentanyl is a highly potent synthetic opioid pain medication, being between 50 and 100 times stronger than morphine. The frequency of overdose calls is putting a very heavy burden on our Emergency Medical Services. Wait times for an ambulance can be more than 30 minutes on a weekend in the lower mainland, due to the number of overdose calls being attended to. Fortunately, Naloxone is a readily available drug that can block the effects of opioids, and keep someone alive during an overdose situation. British Columbia needs an effective and efficient way to get Naloxone kits to people that are experiencing an overdose.

This project was chosen by our team because of the severity of the need for a solution. The overdose crisis is affecting everyone, not just drug addicts, and is continuing to get worse. The social impact of this project could be immense. It has the potential to save numerous lives. No one should die of an overdose, especially when there are pro-active citizens, carrying Naloxone and willing to help. The way we see it, all that is needed is the piece to link the Naloxone carriers to the drug users.

Our goal with Antidose is to create an application which readily pairs volunteers who carry Naloxone with those who need it. We would like to be able to minimize the number of overdose fatalities around the world through the use of networking and location based services. With the advent modern telecommunications, almost everyone has access to a networking device which can alert them of overdoses in their area. We feel that this project has the capacity to create lasting social change and save the lives of thousands of people.

The potential user base for this project includes two groups of people. First would be people who are currently, or would like to become, Naloxone kit carriers. These people would install the app, and then receive alarms when they are in proximity to a potential overdose situation, as reported by the second group of users. The second group would consist of drug users, party goers, or anyone else who may be around people taking drugs. This group of users would install the application, and then report any potential overdose situation they encounter.

III Literature Survey & Specifications

The team came up with three different solutions to the primary mapping and navigation system. To have a cohesive and contained application experience, the mapping system should ideally keep the user inside the Antidose application without the need of loading external applications.

Navigations Technology

1) **Mapbox Navigation API**

   Mapbox uses open-source technologies and hosts many APIs on GitHub which provides control to custom tailor the look and feel of the map. Their Navigation API lets the user have total freedom on the front-end, while also being embeddable. This means the user-experience can be made to look and feel exactly as we would like. This added degree of customization means that we can build the application the way we envision it. The downside to using Mapbox is that the available documentation is limited and less widespread than Google's documentation.

2) **Google Directions API**

   Google's Directions API is very similar to any map application they provide. This option also allows the map to be embedded into Antidose, similar to Mapbox. The user-interface would be a top-down map view with lines denoting the path to the person in need. The caveat here is that turn-by-turn navigation cannot be provided through this API, which is a service we would very much like to offer users. This option constitutes the "middle of the road" option, where the difficulty is minimal without negatively affecting the user experience.

3) **Google Navigation Services**

   This solution unfortunately means using the included Google Maps application that already exists on the user's phone which pulls the user out of Antidose. So when the helper responds to a request, it will open Google Maps and navigate from inside the other application. The application will then externally poll the user's location and when they have reached their destination they will be notified through the notification drawer. We feel this option is incredibly fragmented and does not provide the simplicity and immersion that we're looking for. While it is incredibly easy to implement, it is the least effective of the three options discussed in this report.

For navigation technology, we are opting to utilize the Mapbox Navigation API. We are doing this for a number of reasons, but primarily we feel that it adds a lot of features to the application without increasing the cost. The only downside of this API is that it is a little more foreign to the members of our team than the Google API. However, it provides features such as turn by turn navigation, map customization, location updating, and a couple more that we find incredibly valuable for our project. The ease of use of Google Maps may be useful if Mapbox increases the difficulty too much.

Wearable Device Integration

Originally, the Antidose team had a keen interest on integrating with a wearable device to keep track of your vitals. The wearable device options are discussed below

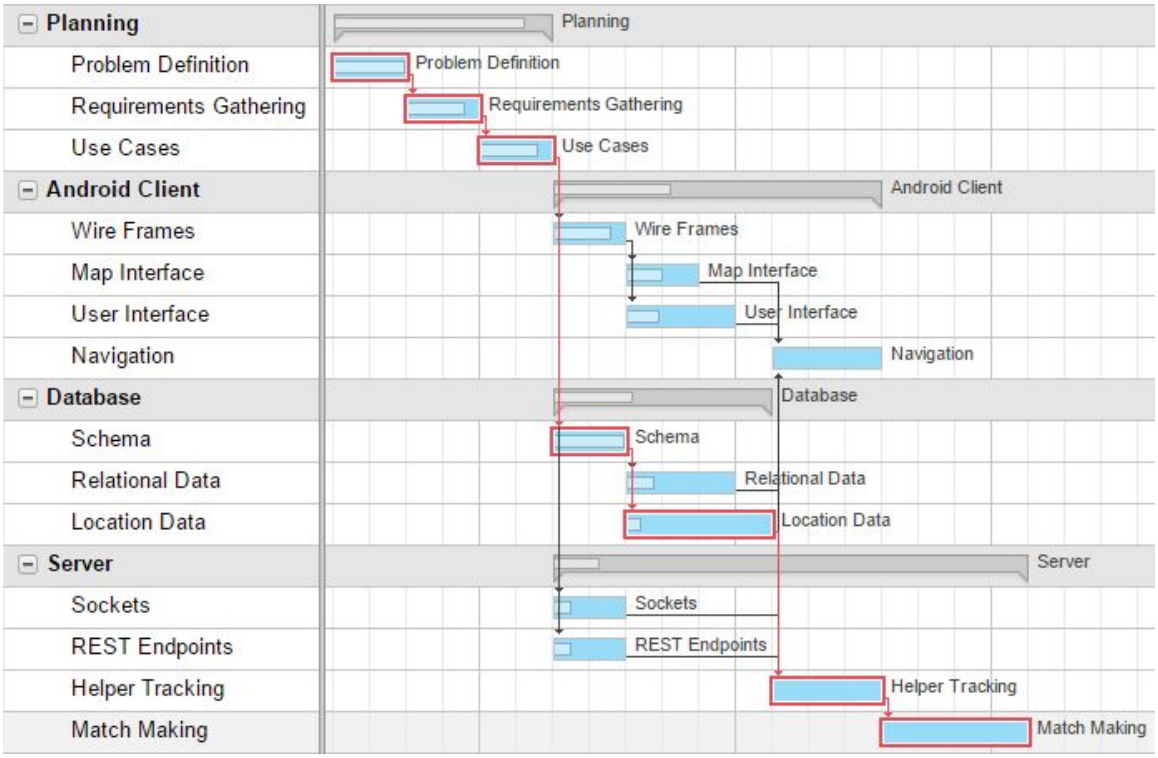
1) **Android Wear/Apple Watch Support**
   AndroidWear and Apple Watch are the current contenders for largest wearable market share in the world. Both Google and Apple provide verbose API/SDK access for these devices[4][5], but as we've found out, there isn't an accurate metric for determining overdoses. Ideally we would like to be able to measure the O2 content within the bloodstream, but this functionality is not supported by either platform. On top of this, often heart rate monitoring was not effective/consistent.

2) **FitBit**
   We also wanted to explore the Fitbit API for ways to measure and track overdoses. However, we soon ran into the same issue as the Android Wear/Apple Watch software. While we could reliably get heart rate information[3], according to Dr.Price, this information isn't usually useful as you're usually dead by the time the information is relevant.

In the end, we've opted to leave the wearable device integration out of the project planning. It simply does not offer enough value to constitute a good investment of our time.

IV Team Duties & Project Planning

The tasks for the project can be grouped into three major categories: Android client, server, and database. Many tasks are dependant on one another so it is important to identify tasks that must be completed before others to prevent bottlenecks.



The project plan is organized in the above Gantt chart. Note the critical path highlighted in red. The most likely bottleneck is the PostGIS database that stores location data about the helpers. With this in mind, we assign tasks as follows:

Wire Frames

Wire Frames will be made to plan out the user interface of the application, as well as assist with prototype creation.

- To be completed by: Heather Cape.

User Interface

The User Interface will be written in XML and will involve careful planning to ensure that the app is easy to use and provides a smooth user experience during high stress situations.

- To be completed by: Heather Cape and Graeme Bates.

Map Interface and Navigation

The integrated map will allow helpers to navigate to the location of an overdose. The navigation system will use sockets to continuously update the location of the helper and the overdose.

- To be completed by: Graeme Bates.

Database Schema

An entity relationship diagram of the relational database schema will be created before the database is implemented.

- To be completed by: Geoff Lorne.

Relational Database

The relational database will store information about helpers, help requests, and overdose incidents.

- To be completed by: Geoff Lorne and Tanner Zinck.

Location Database

Location data for helpers will be stored in a geospatial database to allow for faster search times.

- To be completed by: Tanner Zinck.

Sockets

The helpers device will be fed live location data about an overdose using sockets for more accurate navigation.

- To be completed by: Jake Cooper.

REST Endpoints

REST Endpoints will allow for communication between the Android client and the server.

- To be completed by: Jake Cooper.

Helper Tracking

Current location of active helpers will be periodically sent to the server so that the system will have a general idea of which helpers are nearby in the event of an overdose.

- To be completed by: Tanner Zinck and Jordan Vlieg.

Match Making

When a help is requested, the server will search for nearby helpers and inform the helper of the overdose. The search radius will increase until a sufficient number of helpers have agreed to respond to the overdose.

- To be completed by: Jake Cooper and Jordan Vlieg.

V Milestone & Progress Made

Progress Made

Progress structured to mimic GitHub repo organization

Documentation and Planning

- Trello and GitHub organization made
- Flow diagrams of high level system

- Discuss map framework alternatives and choose best option
- Plan DB schema in entity relationship diagram
- Write Gherkin use cases for system (50%)

Server
- Server basis running in Go, with Sockets and basic Roll-Your-Own Authentication
- DB schema implemented in Postgres with PostGIS integration

Application
- Test app set up with Android studio
- Navigation view being developed with lots of learning and experimentation of MapBox framework
-

Website (Deliverable)
- Domain purchased: antidose.ca
- Server written in Flask and running on Heroku with GoDaddy hosting
- Free one page template chosen (rather than starting from scratch, a template will save us time)
- Branding decisions such as color palette, logo, iconography to be used

Milestones

Milestone June 30th

At this point we would like to have the registration use cases working. This would be a simple flow through the whole system. It allows us to implement a simpler use case in order to integrate the backend, frontend, and to do text authorization. At this point the system should also be polling and storing 'helper' locations of registered users.

At this point we should also have a very clear idea of the features we would like visible on the navigation page (eg. should it show distance, time, options for different paths, 'first person navigation', list of direction) and have researched that they are possible with the Map Framework

Milestone July 10th

Progress report 2 due

At this point we should have a complete prototype of the minimum viable product. This would include:
- Requester: Can click the help button, app will text a number (simulating contacting police), app display how many people are nearby, app should display when someone responds to the request
- Helper: Should receive a notification when someone nearby clicks help, should be

brought to a yes/no screen and shown the distance to the requester, should be shown a map of where the requester is (if they respond with 'yes')

There should be placeholder pages for overdose response information. The information does not need to be filled in at this point, but the pages should be accessible.

The project may have bugs at this point that will be smoothed out over the next milestone.

Milestone July 20th
Web presentation due
We should have the website finished at this point. This includes information about the features and team, and in app pictures. It can be visited at www.antidose.ca, but should also be on one of our ~engr pages.

Milestone July 27th
Final presentation.
App should be able to demonstrate several use scenarios. Notable use case to be completed include:

- Firing an overdose event
- Responding to an overdose event
- Navigating to an overdose event

The website should also be live for more information. We will also have marketing information such as a banner to present.

VIII Summary & Future Work

British Columbia is currently in the midst of an Opioid overdose crisis, and it is getting worse with each year. Our group would like to use technology to help the problem. With a mobile application that can connect people with naloxone kits or medical training to those in an overdose emergency, we hope to help in reducing the number of overdose fatalities. By the end of the semester we hope to have followed our milestones and produce a demoable application and informative website. We hope to be in a good place with the application to possibly continue working on it into the future as well.

References

[1] "Illicit Drug Overdose Deaths in BC January 1, 2012 to February 28, 2017," BC Coroners Service, rep., Apr. 2017.
[2] https://dev.fitbit.com/docs/heart-rate/
[3] https://developer.android.com/training/building-wearables.html
[4] https://developer.apple.com/documentation/healthkit/hkworkout

Appendix

Work log since June 6th.

Meeting Log

Members Present	Date	Length	Covered
All	June 6	4hrs	Make Trello, flow diagrams, DB schema, discuss map framework
All	June 7	4hrs	Start Gherkin use cases, make progress presentation, turn diagrams into wireframes
All	June 8	1hr	Practicing presentation
All	June 18	2hr	Finalize progress report, talk about server implementation

Heather

Date	Duration	Work
June 6th	2hr	Wireframing and diagram transcription
June 9th	3hr	Finding website template, setting up website
June 11th	2hr	Researching and writing

		Gherkin use cases
June 12th	1hr	Various website work
June 16th	2 hr	Website work and progress report
June 19th	2 hr	Various website work

Tanner

Date	Duration	Work
June 7th	1 Hr	Setup schema creation script
June 9th	1 Hr	Schema creation script spatial support
June 12th	2 Hr	Setting up Android environment, working on App
June 18th	40 min	Report writing
June 19th	40 min	Report writing
June 19th	1 Hr	Working on Golang server

Jake

Date	Duration	Work
June 18th	4 hr	Golang Server + Postgres work + Docker work
June 19th	2 hr	Report writing + Timelog

Jordan

Date	Duration	Work

May 24th	2 Hr	Background research and investigating existing solutions
June 1	1 Hr	Investigating tools and technologies. Set up and ramp up for the project
June 4th	2 Hr	Android studio set up, experimenting with application
June 19th	1.5 Hr	Report writing

Geoff

Date	Duration	Work
May 20th	1 hr	Setup android environment
June 6	2 hr	Database Schema
June 8th	2 hr	Progress presentation slide deck and practice
June 18th	2.5hr	Progress report writing

Graeme

Date	Duration	Work
May 20th	3 hr	Create initial Android app
June 2nd	2hr	Researched map and navigation APIs
June 3rd	3 hr	Android application requirements research
June 6th	2 hr	Database Schema
June 8th	2 hr	Presentation creation

June 10th	2 hr	Domain creation and website setup
June 12th	4 hr	Android application requirements research
June 18th	1 hr	Progress report writing