Fall 12-7-2021

Project Fe - Anemia Support

Kevin Grzela

FE

FE

HEALTH CARE INDUSTRY | CAPSTONE PROJECT
HONOURS BACHELOR OF COMPUTER SCIENCE
(MOBILE COMPUTING)

STUDENT TEAM

Jayce Merinchuk, 4th Year Student
E: merinchu@sheridancollege.ca

Kevin Grzela, 4th Year Student
E: Grzelake@sheridancollege.ca

SUPERVISOR

Prof. Cristina Ribeiro
E: cristina.ribeiro@sheridancollege.ca
T: 905-845-9430 ext. 2904
Sheridan College Professor

DOMAIN EXPERTS

Margaret Crawford
E: Mlcrn1989@hotmail.com
T: 905-928-4352
Nurse

ABSTRACT

The problem examined in this capstone project is centered around the medical condition of anemia, specifically the kind causing iron deficiency. The main symptoms of iron deficiency anemia are extreme fatigue, fast heartbeat, shortness of breath, dizziness, lightheadedness, cold hands/feet, and chest pain. This project proposes a solution of using an iOS mobile phone device paired with a custom watch/sensor developed by the team to assist the user in monitoring and managing their symptoms of anemia.

The mobile phone is the platform where users can log in, manage their profile details, view their current and past sensor data, schedule and receive notifications, connect with our chatbot, and experience various other integrated features. The mobile phone also holds the Altimeter sensor used to pull readings for elevation and air pressure, which can also contribute to symptoms but are secondary to heart rate and blood oxygen.

The custom watch device will send heart rate and blood oxygen sensor readings to the mobile phone every 5 seconds. There is no onboard UI needed for this device, so it can be miniaturized to look more like a bracelet and less like a watch.

We will use cloud computing functions to analyze user data every 5 minutes to detect if sensor readings are above or below the user-set thresholds. If an anomaly is detected, a push notification will be sent to the user's phone, and the user also has the option to have a text message sent to an emergency contact of their choosing.

We expect this solution will minimize users visiting their doctor or an emergency room as they will be able to monitor and manage their symptoms before they reach a critical point and, in turn, will put less stress on the healthcare system and their body.

Sheridan | Get Creative

ABOUT CAPSTONE PROJECTS

TIMELINES • PROGRAM • SCHOOL

- **January 2020 – April 2020**: Capstone Project Inception, 4-credit course (18 hours/week)
- **September 2020 – December 2020**: Capstone Project, 4-credit course (18 hours/week)

PROGRAM • SCHOOL

- Hons. Bachelor of Appl. Computer Science (Mobile Computing)
- Applied Computing, Faculty of Applied Science and Technology

Table of Contents

INTRODUCTION

The purpose of this document is to provide a general overview of the project, a domain and industry overview, a description of the problem being solved, the solution being proposed, the impact the solution has on the industry, the feasibility of the solution, the project requirements, the system architecture, a detailed project plan, validation and testing strategies, and a conclusion detailing the success of the project.

01/21/2021 - Version 1.0.0 - Document Inception
02/27/2021 - Version 1.1.0 - Requirements Updates
04/07/2021 - Version 1.2.0 - Architecture Updates
04/14/2021 - Version 1.3.0 - Review and Revision Updates
09/10/2021 - Version 1.4.0 - Revision of Software Architecture and addition of new features
10/08/2021 - Version 1.5.0 - Review and update document after latest changes
11/25/2021 - Version 1.6.0 - Review and update document with newest features
12/08/2021 - Version 1.7.0 - Review and update document for the final release

PROJECT OVERVIEW

The project's name is Fe, which is the symbol for the chemical element iron. The project will require an iPhone running iOS version 13 or newer and a custom watch with sensors developed by the team. The goals of project Fe are to help people suffering from anemia better manage their symptoms. To this end, Fe will actively monitor the user’s heart rate, blood oxygen, and altitude and alert the user or emergency contact of any unusual or concerning sensor readings beyond their set thresholds. We expect this project to be feasible as many devices built today have the necessary sensors to track the user’s heart rate. The custom watch contains all the required sensors to work for this project. A constraint on the project will be ensuring we follow all proper legal procedures for handling user data.

Mr. Merinchuk is taking the role of project owner and risk analyst. He is a 4th-year student in the Sheridan College Mobile Computing Degree Program. He has over seven years of experience in the health care industry in the accounting field and one year of experience developing mobile applications in health care.

Mr. Grzela is taking the role of scrum master and risk analyst. He is a 4th-year student in the Sheridan College Mobile Computing Degree Program. He has eight years of experience in software development and two years of experience in Integrated Circuit development. His current relevant project is an in-car companion that responds to changes in the vehicle’s state.

Ms. Ribeiro is taking the role of project supervisor. She has a Ph.D. in Computer Science from the University of Waterloo, an MSc from the University of Guelph, and a BSc in Computer Science from Ryerson University. She brings a positive attitude and outstanding leadership to the table and will be an excellent supervisor.

Mrs. Crawford is taking the role of the project’s domain expert. As a nurse, she has 30 years of experience in the healthcare industry and will be an excellent resource and tester for our project as we progress towards the finish line.

There will be a detailed domain and industry overview explaining the health issue we are tackling, known as anemia, and the problem it presents in the body. We will then describe the various technologies we are using to solve the issue, such as a mobile application for tracking, an online cloud database, a wearable device with the appropriate sensors, and a chatbot used to assist the user with any questions they may have regarding the app or their symptoms.

DOMAIN AND INDUSTRY OVERVIEW

62 – Health Care and Social Assistance
This industry code is the blanket code for the Health Care and Social Assistance Industry, which covers anemia and support that we are investigating. Due to this code's treatment and residential care section, our project is in this domain. We provide a service to the user by allowing them to self-monitor their symptoms instead of seeking constant medical assistance and care [1].

6216 / 62161 – Home Health Care Services
These two industry codes focus on patients' nursing services in their homes and providing personal care to patients. Medical equipment is also covered under these codes. Our project is covered under these two codes because we use the apple watch and mobile phone as medical devices [1].

624 – Social Assistance
This industry code focuses on aiding the patients directly via social interactions and focuses on residential services related to short stays. This project will diminish the need for short visits by allowing the patient to monitor for symptoms without the need for intervention or assistance [1].

The health care industry is enormous. There are 44,149 establishments, most of which have at least 0-99 employees. The average revenue in the industry is $248.7 thousand, and they are 91.4% profitable. There is a significant chance that a working product in this industry will succeed.

PROBLEM DESCRIPTION

The problem we are facing is the detection of symptoms of anemia. Users with anemia want to monitor their bodies for signs of anemia-related symptoms setting in before it is too late. Often, people with anemia will feel shortness of breath, chest pain, dizziness, or like they will pass out. Using a custom sensor device developed by the Project Fe team, an array of sensors will track the user's vital signs to predict the symptoms of anemia.

We plan to use Firebase and Core Data on the device to store user data, including past and current sensor readings. The mobile device will then act as a hub for users to interact with their current and past information and alert them to unusual or concerning readings from the sensors. We will use a local notification system to ensure the user receives timely reminders to take their medication and more urgent alerts to more pressing concerns.

The cloud computing solution used in this project is Firebase. We will use several systems on Google's Firebase, such as their authentication for logging in, Firestore for storing user profile data, and functions for completing some heavy computations when analyzing data for anomalies. This system is easily scalable as the application grows. This system will be part of the back end of the application. Firebase, by default, has no security turned on. Still, Firebase has a well-developed area where we can implement authentication and rule-based authorization so only users with the proper access level can view or manage the data [2]. Firebase also has documentation for penetration tests to ensure we are securing our data appropriately.

The community that would benefit from this type of application will be anyone with an iron deficiency type of anemia. The application can also be easily transferred into assisting with other medical conditions. The mobile company, Apple, will also benefit from this project as it will be designed for their mobile devices and wearables. Apple's current iPhone, the iPhone 13, for a base price of USD 1099.00 [3].

Our current option for wearable devices is a custom wrist-mounted sensor array. This option is necessary because we cannot gather the data needed through the Apple Watch or Fitbit smartwatches reliably or gather enough data to suit the project's needs. Price is another argument for a custom sensor array over the off-the-shelf smartwatches listed above. As making anemia support accessible is our primary goal, any opportunity for price reduction will be explored.

The inspiration for the project is for personal reasons. All participants in the project are interested in helping people with anemia become more self-sufficient and monitor and manage symptoms easier. Another connection is to one of the developers. They have a close friend diagnosed with anemia, so they feel personally invested in the project.

SOLUTION DESCRIPTION

The proposed solution for this Capstone Project is a system that allows a user to monitor for symptoms and have a custom profile designed for notifications to keep on top of medications. We will use mobile devices to show current sensor readings and maintain a history of readings in a local database. Our advanced areas will cover wearables that hold the sensors we monitor and a chatbot to assist the user with any questions regarding the application or sensors.

Mobile Computing

This solution will be designed for Apple's iOS mobile phone device. The iPhone will require an operating system of iOS version 13 or newer as its minimum requirements. There has been a massive surge of people owning mobile devices in recent years, and we believe that offering this application on the Apple mobile device will reach a large audience. An Apple iPhone 8 or newer can support iOS version 13 or newer.

This capstone project proposed is ideal for the selected problem because it monitors and manages symptoms of anemia much better than current processes do. It allows us to use our knowledge of technology to solve a real-world issue in a domain that has been growing over the last few decades. The mobile computing component will be simple enough for the older generation to use.

Cloud Computing

The cloud computing section is broken into several parts as we utilize many different features within our application. There are many services in Google Firebase offered that we found were very useful when designing the app.

The first section is Google Firebase Authentication. This section contains a list of email addresses used to create accounts and log in to the app.

The second section is Google Firebase Firestore Database. We will store user profile information in Firestore, including names, addresses, phone numbers, emergency contacts, the user's FCM Token, and the user-specified thresholds for each sensor.

The third section, which we have put on hold, is Firebase Storage. We initially used this section to store user documents such as blood tests or doctor's notes. We have decided not to go in this direction as it created more difficulties with PIPEDA, which would seriously limit our application.

The last section is Google Firebase Functions. We have developed several functions here that will analyze the user's sensor data to determine if there are any sensor readings above or below their sensor thresholds. If unusual readings are detected, the Firebase Function will notify the user to alert them of the anomaly so the user can decide if the reading is concerning to them.

By leveraging cloud computing, we can tailor the presentation of information, alerts, warnings, and thresholds to each unique user with their unique set of symptoms. We make the application significantly lighter on the end user's device by offloading profile compilation and the calculations off to Google's powerful servers.

Advanced Areas of Computer Science

Fe uses two advanced areas of Computer Science. The first is IoT/Wearable computing. This will be essential to gathering the necessary bodily metrics to ensure that Fe can accurately track potential symptoms of anemia. Fe will be using a custom watch sensor device to collect metrics such as blood oxygen saturation and heart rate. We will also use the phone's Altimeter to determine altitude and elevation as these can contribute to user symptoms if they are extreme. Combining all these data points will provide an accurate glimpse into the users' current state of well-being.

One important update that occurred in September 2021 was the change in the sensor device for the project. Unfortunately, there does not appear to be a reliable method of accessing the onboard SPO2 (Blood Oxygen) sensor for the Apple Watch Series 6 nor the Fitbit Sense (our backup device). Instead, the team has developed an Arduino-based Bluetooth pulse-oximetry sensor to track the user's blood oxygen saturation and heart rate and relay this information to the iPhone device.

The secondary Advanced Area of Computer Science for Fe is a Cognitive Computing Chatbot. The FE Chatbot will leverage Amazon's AWS technology to provide the user with contextual responses to various questions they may have as they use the application. Primarily, this chatbot will provide a more conversational response to areas that may be daunting or confusing to the user. Examples of such topics include symptom checking, explanation of sensor readings and their significance, and application usage.

- Symptom checking
    - The chatbot will ask the user several questions to determine the urgency for assistance.

  - A detailed explanation of each symptom will be provided so the user understands why the chatbot is asking about it.
- Sensor Readings
  - A detailed explanation of what the sensors track, the healthy/acceptable quantities, the range of concern for each sensor
    - The range of concern refers to the sensor values (i.e., Heart rate at 40) that cause concern.
    - The chatbot will provide additional information regarding what these numbers are for each sensor, why they are chosen, and its impact on the user.
- Usage of the application
  - While the Fe team has done their best to make a user friendly and clear UI, we understand that confusion can arise in all situations
  - The Fe Chatbot will guide the user for accessing and using different features of Fe
    - For example, if a user wants to upload a document but is unsure how, the chatbot will provide instructions on using this feature.
    - Another example would be using the sensor data features. If a user has some confusion over how to scale the charts to view their heart rate over the past week, that will be explained with the chatbot

SOLUTION IMPACT

This solution will have a decent size impact on the industry as it will give the users the power to monitor their symptoms without the need for a nurse or their medical devices to be deployed out in the field. The user will be led to stay on track with their medications. It will reduce the number of expensive medical devices people need to buy to track their symptoms and instead allow them to monitor most issues with the wearable and mobile devices. Depending on the type of anemia the user has, they may require an injection of erythropoietin (EPO) which could cost them USD 120.00 for 10,000 units [4]. Tests to check for anemia may be cheap, but there is comfort in knowing that your symptoms are being monitored passively without you having to check via tests every day. These injections try to trigger the body to produce more red blood cells. This may only last the user a few weeks, at which point, they will require more injections. One of the aims of project Fe is to be fully inclusive to all anemic users. This would include reminders for those needing EPO injections.

The solution hardware is an Apple iPhone 8 or greater, which, at the end of 2021, is at least 4 versions behind the latest iPhone version released on the market. These phones tend to decrease in value as new ones are released. The custom watch device was not costly to create but took time to set up correctly. With further development of the watch device after the project's main components are completed, the watch device could be optimized and made smaller to be worn like a bracelet. The sensor device will not replace smartwatches as it does not require an onboard UI. It can be worn complimenting a watch, not as a replacement.

SOLUTION FEASIBILITY

This section will explain, in detail, the solution’s feasibility concerning the design, deployment, and adoption of the solution. The design section will cover the solution's device choices and the potential risks for an unsuccessful project. The deployment section will explain how the solution appears to be straightforward with the use of current technology in commercially available devices. Lastly, the adoption section will present the potential barriers to adopting the solution and defend our choices.

Design and Construction

The design and construction of this project are relatively feasible, though it will be the most challenging part. Using commercially available products (i.e., Apple Watch, Fitbit) instead of custom-made hardware for this project would have helped decrease the barrier to entry for the development team and software users, but they could not provide enough data. The technology already exists. Our solution will be implementing it in a new way which may pose risks such as lack of accuracy by monitoring hardware, permissions issues with the wearable devices, and consistency of measurements.

Given that most of our risks are associated with the hardware onboard the wearable device of choice, we have a mitigation strategy that will help alleviate most risks. We can create custom hardware using available IC kits such as Arduino and Raspberry Pi. We have taken that route to ensure a functional product is created.

The product will be a wrist-mounted device like a watch if we attempt to create a custom sensor bank. The watch-like device will house the required sensors to mirror as closely as possible the data collection available to us with an Apple Watch Series 6. For the sake of the capstone project, this device is an Arduino Uno soldered to a breadboard with an HM-10 UART Serial Bluetooth Low Energy module and a MAX30102 pulse oximetry development daughterboard. Before production, the Arduino device and HM-10 would need to be replaced with an IC equivalent to an Atmel ATMega 328 and a Bluetooth module equivalent to a Texas Instruments CC2541, and the Maxim Integrated MAX30102 sensor would likely be integrated into the production model.

The secondary risks for the project would be related to time management and the collaboration of the group members. Both members of the team have full-time class schedules along with part-time jobs. These other responsibilities could hold the team back from constructing a successful solution. There is also a possibility that the team members may not work well together. This will be mitigated with constant contact between the team members and their supervisor to ensure everyone is on task and has assistance if they reach any roadblocks.

Deployment

Operating under the assumption that the design and creation of Fe go according to the original plan, the deployment will be the most feasible and straightforward facet of this solution. As the package will be entirely software-based and targeting Apple devices, Fe will be launched on the App Store as a free app. Should we choose to pursue this project as a business idea, we have explored the options available for making money. We have determined that a monthly subscription model and selling collected data model would be the best options. The biggest roadblock for this project reaching its intended audience is the cost of the devices. Over time, the sensors and device costs are expected to come down, making the accessibility to the solution more viable.

Using a flexible storage solution such as Firebase and keeping as much processing off the user's device as possible, we aim to minimize any potential bottlenecks in deployment from the perspective of infrastructure throughput.

Adoption

The team behind Fe has worked hard to ensure there are as few barriers to adoption as possible. While we have not been able to reduce the requirements of Fe to just the user's smartphone, we believe that the condition of a smartphone and the custom wearable device will not be overly burdensome on potential adopters. The current competition in the industry is Apple Inc., but they have not yet used the sensors available on their devices in this way. To make money, this application is considering using a monthly subscription fee model to be sustained. We have determined with primary research methods that a monthly cost of CAD 9.99 would be appropriate for this product. This may lower potential adoption, but we must make money for the business to survive. Advertisements were considered, and they may be the better way to go depending on the UI of the system. The initial money output for the two devices is currently the most significant barrier, but these costs will decrease with time.

The current leading plan for monetizing Fe is with a monthly subscription model. This will allow us to receive continuous income for marketing and further development of the application. We have also considered selling collected user sensor data to research centers or schools for extra income.

PROJECT REQUIREMENTS

This section reviews the system context, use-cases, and user interface. The context will describe any groups of people that will be affected by the existence of this product. The use cases will be a high-level view of the major functional areas of the system. The user interface will describe the major elements of the application and holds any wireframes or screenshots. The detailed software requirement specification document will cover the intricate details of connecting the devices and storing the data in the database. There are links below to the different systems we use to manage the project.

VPository - https://online.visual-paradigm.com/w/ujjwxvpo/drive/
Project Plan - https://project-fe.atlassian.net/jira/software/projects/FE/boards/1
Risk Management Plan - https://project-fe.atlassian.net/jira/software/projects/FR/boards/2
Test Plan - https://project-fe.atlassian.net/jira/software/projects/FT/boards/3

SYSTEM CONTEXT

This system will affect many distinct groups, each of which will be described below. The existence of this product will help society in various ways but mainly aims to improve the user's quality of life.

The user that has anemia is the primary stakeholder. The user will interact directly with the input and output of the device. They are the ones that are interested in viewing the sensor data to track and manage their symptoms of anemia.

The user's family member will be a secondary stakeholder. The family may be interested in the sensor data output to ensure the user is doing well. They may also be subjected to the system's output if the user requires assistance and uses the contact system to help the user call a family member for help.

The user's doctor will also be a secondary stakeholder. The doctor will be interested in the output of the sensor data and want to use it to provide the appropriate medication and guidance to the user.

Apple Inc. is a tertiary stakeholder. Apple Inc. will not interact with the system in any way, but their company may see an increase in revenue due to the application's existence.

Home and community support workers are tertiary stakeholders. They will not interact with the system or output in any way, but the existence of this application may impact their jobs. With users being able to monitor themselves for potential symptoms, there may be less need for support workers.

The government of Ontario / Canada is a tertiary stakeholder. They will not interact with the system or output in any way. They will, however, be interested in ensuring the application follows all laws regarding the safety and security of the user's data.

Lastly, the ministry of health is a tertiary stakeholder. The ministry of health will not interact with the application or output. They will also be concerned about the safety of user's data. The ministry of health will also be interested in ensuring the users are not being given lousy advice through the application.

The main top-level use-cases the system is capable of will be described in the use-cases section below. A link to the software requirements specification document will be provided below.

Software Requirements Specification – can be found here.

USE-CASES

One of the most significant functions of the application will be the user account system. The user must be able to create an account, log in, edit their details, delete their account, and delete their data. The user account system will be set up with Google Firebase using Authentication and Firestore to hold user profile data.

The Sensor system will be the next crucial system in the application. We will be pulling sensor data from the blood oxygen sensor, the altimeter sensor, and the optical heart sensor. The data pulled from these sensors will be displayed on the application's home screen, and each sensor will have a history of all previous sensor readings. This data will be stored in core data on the user's device.

The Firebase backend service will be another significant system in the application. Data relating to the user account and settings will be stored in Firebase Firestore and User Defaults on the phone when necessary. We are also using

Firebase Functions for data analysis. We have conducted extensive research in the Google Firebase documentation to ensure the user data is stored safely and securely according to the governing bodies' laws.

The notification system will use the user-specified thresholds for each sensor along with the sensor data to determine if a notification should be sent to the user to alert them of a detected anomaly. We will also create a notification function to remind the user to take their daily medication should they choose to use this. Depending on the severity of the sensor reading, the user or the user's family member may be notified if they choose to set this up.

A link to the Software Requirements Specification can be found here.
A link to the Visual Paradigm Diagrams can be found here.

USER INTERFACE

The user interface of Fe is designed around a few central tenets:

- Simplicity
    - By keeping the design of Fe as simple as possible, we ensure that there is no needless bloat stopping the user from seeing their valuable information or directing the user away from where they need to be.
    - One of the pet peeves of the development team in terms of user experience is dealing with endless menus and unintuitive app-flows resulting in the user constantly hunting for the feature they wish to use. We aim to avoid those user experience issues through simplicity.
- Accessibility
    - The userbase of Fe is not narrowed down to a single age range or cultural group. Instead, the users of Fe will be from all walks of life with all different levels of technical expertise and physical restrictions. We will implement a full suite of accessibility features to ensure that Fe is available to as wide a range of people as possible.
- Urgency
    - The users of Fe are trusting us with their well-being. When an alert is required, the time to act may be short. As such, Fe's alert system takes top priority over other UI elements in the event of an urgent alert. When an alert is presented, the UI of the alert will be straightforward.

A link to a live wireframe of the current project can be found here.

PROJECT ARCHITECTURE

In the sections below, you will find information about the project architecture. While the project is constantly evolving and specific details are subject to change, the project has progressed to a point where the major architecturally significant components have been identified. The overall project architecture has been determined. Below, you will find a general overview of the system architecture design, the architecturally significant components that can be identified, and an overview of the deployment model.

A link to the visual paradigm diagrams for each model can be found here.

ARCHITECTURE OVERVIEW

The system architecture decided for this project is the Layered Architecture model. This model was chosen as it provides a logical structure for the flow of information from user interaction down to the details in the database. This type of architecture is also well understood by the team as we have experience from previous classes using this architecture layout. The layers selected for the model are as follows:

- Interaction Layer
  - This layer contains the methods of interaction with the user. Such methods include:
    - User Input (gestures, on-screen taps, buttons) - Inbound
    - Notifications – Outbound
- Presentation Layer
  - This layer contains the application's GUI. There will be two separate GUI's. One GUI for the Apple Watch Application and one for the Mobile Application.
- Business Layer
  - This layer contains the logic that fetches or determines the data viewed in the presentation layer.
  - Significant components of the business layer include:
    - Logic to interact with the HealthStore persistence layer package for current sensor data.
    - Logic to interact with the Firebase persistence layer package for storing sensor data and account details.
    - Logic to trigger timers for pulling sensor data or activating the notification logic in Firebase Cloud Functions. Logic to move between screens when buttons are pressed.
- Local Persistence Layer
  - This layer contains the methods to interact with the persistent storage systems found within the mobile device and watch:
    - Core Data: Core data will store user settings and temporary sensor data.
    - Health Store: Access the Apple Watch and iPhone Health Stores to access sensor data.
    - User Defaults: This storage will save the state of any buttons or switches.
- Device Layer
  - This layer contains the methods to interact with any sensors or devices related to the application.
    - Custom Sensor Object: The logic required to access data from the custom sensor device.
    - Phone Sensors: Access the altimeter sensor on the phone.
- Remote Persistence Layer
  - This layer contains the connections to the remote Firebase servers used to power Fe. The components found in this layer are:
    - Firebase Authentication: This contains a list of accounts that have been set up and are allowed to access the application.
    - Firebase Firestore: This is where the user data will be stored.
    - Firebase Functions: These functions will complete some heavy computations on sensor data looking for anomalies and sending out notifications when necessary.

Architecturally significant components include:

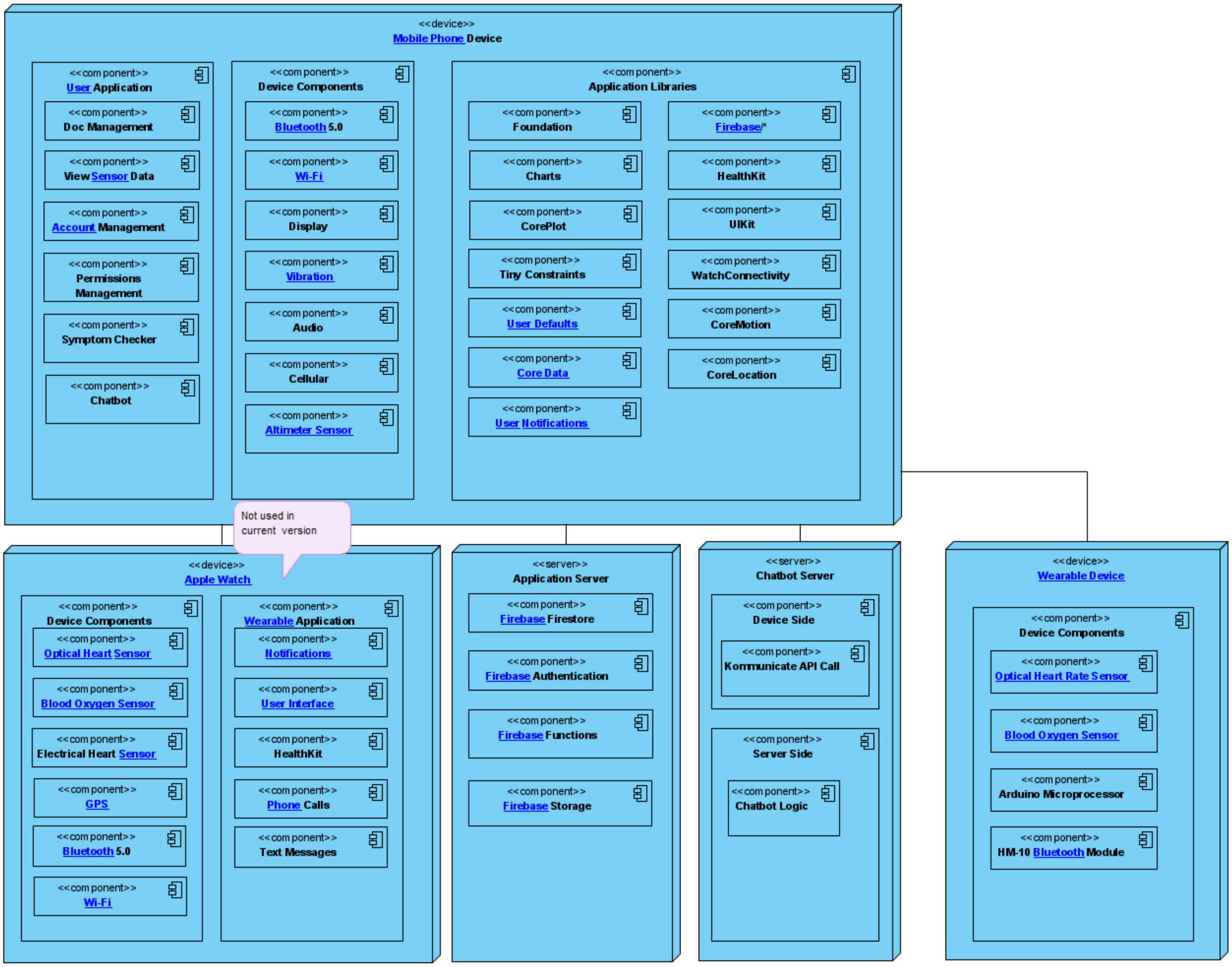
1. Data persistence (Firebase, HealthStore, Core Data, User Defaults)
   a. Firebase
   b. This will be the primary storage system for Fe and stores vital information such as user login data, user account information, and user metrics. Health store
      i. Health store is an internal database used within Apple’s HealthKit to store information regarding the user’s biotelemetry read from the Apple Watch. This information is stored locally on the device for a short duration before being compiled and transferred to the Firebase database for long-term storage and user retrieval.
   c. Core Data
      i. Core Data is another internal database used within the Apple iOS ecosystem. This will be used in tandem with Health Store to provide data persistence and a smooth user experience while minimizing calls to Firebase where possible.
   d. User Defaults
      i. User Defaults is a way to save the state of a switch or button in the application so the app can remember if a switch was turned on or off after the app is closed and re-opened.
2. Interaction Components
   a. Custom Watch
      i. This is the primary data collection device for Fe. The watch will provide sensors to read the user’s current heart rate, blood oxygen.
   b. Apple iPhone
      i. This is the primary interaction device between the user and Fe. The user will have complete control of their account through this app. They can view past sensor data, notifications, and manage their account details.

DEPLOYMENT MODEL

Deployment of Fe will require four primary resources.

1. The mobile device. This is the user’s primary interaction point, where the user can view their data and interact with the GUI of the application. More notably, Core Data tables on the local device store the user’s sensor data.
2. The Custom Watch device. This device is responsible for collecting sensor data and uploading it to the mobile device via its Bluetooth connection to the phone.
3. The Application Server. The server contains Firebase Firestore, Authentication, and Functions.
4. The Chatbot Server. This server receives requests from the user, evaluates the data, and returns a response.

A diagram of the deployment model is shown below:


<<device>>
Mobile Phone Device
<<component>> User Application
<<component>> Doc Management
<<component>> View Sensor Data
<<component>> Account Management
<<component>> Permissions Management
<<component>> Symptom Checker
<<component>> Chatbot
<<component>> Device Components
<<component>> Bluetooth 5.0
<<component>> Wi-Fi
<<component>> Display
<<component>> Vibration
<<component>> Audio
<<component>> Cellular
<<component>> Altimeter Sensor
<<component>> Application Libraries
<<component>> Foundation
<<component>> Firebase/*
<<component>> Charts
<<component>> HealthKit
<<component>> CorePlot
<<component>> UIKit
<<component>> Tiny Constraints
<<component>> WatchConnectivity
<<component>> User Defaults
<<component>> CoreMotion
<<component>> Core Data
<<component>> CoreLocation
<<component>> User Notifications
Not used in current version
<<device>>
Apple Watch
<<component>> Device Components
<<component>> Optical Heart Sensor
<<component>> Blood Oxygen Sensor
<<component>> Electrical Heart Sensor
<<component>> GPS
<<component>> Bluetooth 5.0
<<component>> Wi-Fi
<<component>> Wearable Application
<<component>> Notifications
<<component>> User Interface
<<component>> HealthKit
<<component>> Phone Calls
<<component>> Text Messages
<<server>>
Application Server
<<component>> Firebase Firestore
<<component>> Firebase Authentication
<<component>> Firebase Functions
<<component>> Firebase Storage
<<server>>
Chatbot Server
<<component>> Device Side
<<component>> Kommunicate API Call
<<component>> Server Side
<<component>> Chatbot Logic
<<device>>
Wearable Device
<<component>> Device Components
<<component>> Optical Heart Rate Sensor
<<component>> Blood Oxygen Sensor
<<component>> Arduino Microprocessor
<<component>> HM-10 Bluetooth Module

PROJECT PLAN

Within the Project Plan section, you will find information regarding the detailed responsibilities of each team member, the management process being used by the team, detailed information on each iteration throughout the Project Inception Release, and finally, a detailed explanation of our risk management plan.

The team is operating off a small SCRUM management process. Week-by-week, we have planned out our sprints in advance using Atlassian's JIRA management software. Due to the small nature of our team, the work is highly collaborative, and communication is essential. We have chosen to meet twice weekly, once for code review and the other for a full SCRUM meeting with our Capstone Advisor, Cristina Ribeiro.

A Responsibility Matrix has been shown below to show how the team has divided the work.

Project Responsibility	Member 1 - Jayce Merinchuk	Member 2 – Kevin Grzela
Project Owner	Yes	
SCRUM Master		Yes
Requirements Analyst	Yes	
Stakeholder Champion	Yes	
Functional Area Champion		Yes
User Experience Design Lead		Yes
Software Architect	Yes	
Requirements Model Lead		Yes
Domain Model Lead	Yes	
Design Model Lead		Yes
Deployment Model Lead	Yes	
Interaction Model Lead		Yes
Full-Stack Developer		Yes
Integration / DevOps Lead	Yes	
QA Lead		Yes
Verification & Validation Champion	Yes	
Test Model Lead		Yes
Tool and Devices Support		Yes
Communication Support	Yes	

ITERATION PLAN

Inception Release:

- Sprint 01 (Iteration 1): Initialize project, enable user login.
    - Begin iOS application.
    - Create a Firebase account.
    - Establish initial Firebase tables, rules, and configuration.
    - Login Screen.
    - Login and account creation.
    - Home screen.
    - Settings screen.
- Sprint 02 (Iteration 2): Implement edit and delete account options.
    - UI and logic to update user details.
    - UI and logic to delete user account and/or data.
- Sprint 03 (Iteration 3): Implement data gathering, storage, and retrieval.
    - Users can view online documentation about anemia.
    - Users can take a pre-screening test before calling the doctor for an appointment.

Elaboration Release:

- Sprint 04 (Iteration 4): Enable more user features, implement some more core functions.

  - Implement the upload document system.
  - Implement the document view all list system.
  - Implement the system to view individual documents.
  - Implement the delete document system.
  - Finish Edit Account details system.
  - Implement necessary packages into the iOS application for sensor reading.
  - Extract data from the Apple Watch into the iOS application.
  - Display data on the home screen.
- Sprint 05 (Iteration 5): Review past sensor data and complete current sensor readings
  - Implement warnings based on calculated historical averages.
  - Implement basics warnings based on static high/low values for vital metrics.
  - Implement UI for viewing past heart rate data.
  - Implement UI for viewing past blood oxygen data.
  - Implement UI for viewing past altitude data.
  - Implement logic to pull past heart rate data and display it in UI.
  - Implement logic to pull past blood oxygen data and display it in UI.
  - Implement logic to pull past altitude data and display it in UI.

Summer Release:

- Sprint 06 (Iteration 6): Improve UI and Implement Altimeter Data
  - Upgrade the Heart Rate UI
  - Upgrade the Blood Oxygen UI
  - Implement the Altimeter UI - Minor code improvements

Alpha Release:

- Sprint 07 (Iteration 7): Implement notifications
  - Implement local and Firebase Cloud Function for HR notification
  - Implement local and Firebase Cloud Function for Blood Oxygen notification
  - Implement UI and Firestore for Emergency Contact
  - Research viability of Fitbit, Garmin, and Custom Watch Sensors
  - Implement background functions for application
  - Fix Blood Oxygen to pull from Health Store
  - Finish UI for Detail screens chart data for all Charts
- Sprint 08 (Iteration 8): Implement Settings and Security, Custom sensor
  - Create custom sensor device
    - Develop logic for custom sensor SPO2 tracking
    - Develop logic for sensor HR Tracking
  - Begin to develop logic for sensor device communication to IOS Device
  - Notification for medication reminder
    - Set up User Defaults and new keys
  - Notification for sensor anomaly
    - Link button so notifications are either sent or not sent

Beta Release:

- Sprint 09 (Iteration 9):
    - IBM Watson Chatbot creation
    - IBM Watson Chatbot integration into iOS project
    - IBM Watson training
    - Emergency Contact Notification
    - Code reorganization for layers
- Sprint 10 (iteration 10):
    - Set up application permissions screen and logic
    - Set up notification permissions screen and logic
    - UI adjustments on the Home Screen
    - Update Questionnaire system to provide context on each screen as to why this symptom is important.
    - Security Testing
        - Login testing
        - Firestore testing
        - CoreData Testing
        - Firebase functions testing incl. transmission between iOS and Firebase
- Sprint 11 (Iteration 11):
    - Implementation of Background Readings
    - Test Plan for Functional Areas:
        - Heart Rate
        - Blood Oxygen
        - Login
        - Create Account
        - Delete Data
        - Delete Account
    - Creation of Logo for app
    - Optimization of code and bug fixes
        - Home Screen bugs
        - Past sensor data screen charts loading in Background thread

Final Release:

- Sprint 12 (Iteration 12):
    - Testing
        - Edit Account
        - Notifications
    - Remaining Bug Fixes

Future Work:

- New Features
    - Mood/Feeling updates daily for user engagement

  - Billing system for each user with monthly subscription model
  - Persuasive Tech – Contextual languages
  - Persuasive Tech – Awards system
  - Persuasive Tech – Gamification system
  - Persuasive Tech – GUI positive reinforcement system
  - Persuasive Tech – Points system
  - Persuasive Tech – Social feedback
- Upgrades
  - Medication reminder to allow for scheduling and managing multiple reminders
  - Hide Emergency Contact details if feature disabled
  - UI pop up if many anomalies detected in short period of time asking to contact help or adjust thresholds

RISK MANAGEMENT PLAN

The team sat together and conducted a brainstorming session to determine the different risks that could potentially happen during the development life cycle of this project. The risks identified varied from the potential of poor cooperation of team members to the unlikely scenario of a company going out of business.

Some significant risks that have been identified are related to poor time management. Both team members are balancing a full work and school load with five courses. Both team members have families, and one of them has two children, which can reduce that team member's ability to contribute to the project while adjusting to new home routines. The other significant risk is failure to connect the smartwatch and iOS app to gather sensor readings in a reasonable amount of time. As this is new research, it may take longer than expected to acquire this new knowledge and develop that functional area of the application.

Another set of risks that must be addressed are related to the device requirements. The application will currently require an iPhone running iOS version 13 or greater, limiting the iPhones we are targeting to an iPhone 8 or greater. If the user has an iPhone 8 or greater, the device components such as battery, CPU, and memory should be enough to run the application. Currently, we cannot estimate data usage. However, during the testing phase, we will evaluate the amount of data required to transfer the sensor data and documents to the Firebase backend so our customers will be aware of the requirements. One risk we analyzed regarding the Apple Watch has been realized. We are now developing our project to work with our own custom watch sensors as the Apple Watch and Fitbit could not provide enough data for us to validate the requirements of the project.

We also run the risk of misunderstanding our users' goals, such as the user, doctor, caregiver, or customer. The doctor evaluating the data from this application will want the information presented logically, so it is easy for them to understand and make recommendations. The user will want to see the data and understand everything, so we must include information bubbles that describe the functions on the UI. The customer wants a high-level description of how the application works and what it does to decide if it is suitable for their needs. We can manage these risks by having meetings with a potential customer, a doctor, and our domain expert to ensure we are presenting the data to all 3 of them in a way that they can each understand. If we are not clear about any of the requirements, we will meet with the users and break down the requirements to design the application properly.

The time management risks are managed by ensuring each team member blocks off 18 hours per week to dedicate to working on the capstone project. We also have a log of our communication in Microsoft Teams.

The connection risks are being managed in the research section of the project. We have now completed all major use cases of the project and have successfully connected our custom sensor device via Bluetooth to the iPhone. Our connections to Firebase for our database and AWS for our chatbot all appear to be functioning properly.

Risk Management Plan - https://project-fe.atlassian.net/jira/software/projects/FR/boards/2

VALIDATION AND TESTING

This next section will provide details about the testing strategies and validation results. For an in-depth explanation of the testing and validation, please visit the test plan document.

The Test Plan Document can be viewed here for more information regarding the team's DIT and SIT strategy.

TESTING STRATEGY

Throughout the project development lifecycle, the team has been conducting Development Independent Testing (DIT) in line with the completion of code associated with various functional areas. The DIT testing is a paired-down subset of the complete testing strategy below, which is the System Integration Testing (SIT) plan. As the project is nearing the stage of completion, the team is conducting tests to ensure each functional area is working appropriately. Upon completing SIT, the team will release the project to a small subset of users for User Acceptance Testing (UAT). The project will stay in UAT until all tests are passed before being released to the public.

Below is the SIT plan.

Note: Document Storage and Management has been discontinued on this project due to issues with PIPEDA.

- Upload Document
  - Trying to upload a document with no information filled out and no image selected fails
  - Trying to upload a document with some information filled out and no image selected fails
  - Trying to upload a document with all information filled out and no image selected fails
  - Trying to upload a document with all information filled out and an image selected succeeds
  - Trying to upload a document with some information filled out and an image selected succeeds
  - Trying to upload a document with no document name and an image selected succeeds
- Update a Document

  - Trying to update every field in the document with appropriate values succeeds
  - Trying to update every field in the document with inappropriate values (spaces and special characters) fails
  - Trying to update every field in the document with spaces fails
  - Trying to update every field in the document with no input fails
  - Trying to remove the image uploaded and upload a document with no image selected fails
  - Trying to upload a new image and update the document succeeds
- View all Documents
  - When the documents button is pressed, all documents under that user appear on screen
  - Upload 100 documents and ensure they all populate on screen
- View a Specific Document
  - Click the first document and ensure it opens
  - Click the last document and ensure it opens
  - The document clicked on is the document that opens on screen
- Delete a Document
  - Try to delete the document from view all documents screen
  - Document is removed from view
  - Document is removed from Firestore
- Sort a Document
  - Sort documents by date
  - Sort documents by name
  - The appropriate documents are shown based on the search
- Search for a Document
  - Search for a document by name
  - Search for a document by date

We currently have some ideas for testing strategies for the sensor functional areas.

- Heart rate sensor
  - Gather Heart Rate Data
    - Ensure data is being read from watch at regular intervals
    - Try reading data with accepted permissions
    - Try reading data with permissions denied
  - Store Heart Rate Data
    - Ensure data is correctly populating Firebase tables
    - Try pushing data over cellular
    - Try pushing data over Wi-Fi
    - Try pushing with no network connection
    - Try pushing data with incorrect date
    - Try pushing empty data
  - View Average Heart Rate Data
    - Ensure averages are populated correctly
    - Ensure averages update with data changes

        - Test with data out of range
    - View Current Heart Rate Data
        - Ensure data is updating frequently
        - Ensure handler stops when screen is not in the foreground
        - Ensure value is accurate to what the Watch sensor is reading
        - Ensure value shows up on button
    - View Past Heart Rate Data
        - Ensure date variables are accurately reflected on chart view
        - Try searching for dates too far in the future
        - Try searching for dates too far in the past
        - Try with cellular connection
        - Try with Wi-Fi connection
        - Try without internet connection
- Blood Oxygen Sensor
    - Gather Blood Oxygen Data
        - Ensure data is being read from watch at regular intervals
        - Try reading data with accepted permissions
        - Try reading data with permissions denied
    - Store Blood Oxygen Data
        - Ensure data is correctly populating Firebase tables
        - Try pushing data over cellular
        - Try pushing data over Wi-Fi
        - Try pushing with no network connection
        - Try pushing data with incorrect date
        - Try pushing empty data
    - View Average Blood Oxygen Data
        - Ensure averages are populated correctly
        - Ensure averages update with data changes
        - Test with data out of range
    - View Current Blood Oxygen Data
        - Ensure data is updating frequently
        - Ensure handler stops when screen is not in the foreground
        - Ensure value is accurate to what the Watch sensor is reading
        - Ensure value shows up on button
    - View Past Blood Oxygen Data
        - Ensure date variables are accurately reflected on chart view
        - Try searching for dates too far in the future
        - Try searching for dates too far in the past
        - Try with cellular connection
        - Try with Wi-Fi connection
        - Try without internet connection
- Altimeter

    - Gather Altitude Data
        - Ensure data is being read at regular intervals
        - Try reading data with accepted permissions
        - Try reading data with permissions denied
    - Store Altitude Data
        - Ensure data is correctly populating Firebase tables
        - Try pushing data over cellular
        - Try pushing data over Wi-Fi
        - Try pushing with no network connection
        - Try pushing data with incorrect date
        - Try pushing empty data
    - View Current Altitude Data
        - Ensure averages are populated correctly
        - Ensure averages update with data changes
        - Test with data out of range

Below are the following ideas for the account testing strategies:

- Create an Account
    - Create an Account Using Email in proper format
    - Create account using format other than email@email.com format
    - Create account with blank password
    - Create account with password having just 1 space
    - Create account using proper password format
- Log in to Account
    - Log in to account using proper email format with an existing account brings you to the screen asking for password
    - Log in to account using improper email format produces an error message alerting the user to fix their format
    - Logging into account with correct email and correct password moves the user to the Home Screen
    - Logging in to account with correct email and incorrect password alerts the user about incorrect email/password match
- Update Account Details
    - Updating all possible fields with correct values updates the user account successfully
    - Updating each field with blank fields produces an alert error
    - Update each individual account field with special characters
    - Updating a field with only a space character produces an alert error
    - Updating a field with more than 50 characters produces an alert error
- Delete Account
    - Try to delete your account and then cancel the action on the confirmation prompt
    - Deleting your account removes it from Firebase Authentication
    - Deleting account logs the user out and moves them back to the first login screen
- Delete Account Sensor Data

  - Deleting your sensor data on the app removes it all from Firebase too
  - Try to delete all your sensor data and then cancel the action
  - Trying to delete when there is no data will show an alert that no data exists

Below are the following ideas for the questionnaire testing strategies:

- Answer Yes on Question N moves to the Doctor Screen
- Answer No on Question N moves to the next question
- Answer No on the last Question moves to the Everything OK Screen
- The Return button pops all view controllers back to the home screen
- The Call doctor button opens the slide-up menu to call the doctor

Below are the following ideas for the notification testing strategies:

- Receive Heart Rate Anomaly Notification
  - Ensure switch is on in settings
  - Ensure heart rate readings are being stored in core data
  - After 5 minutes, ensure notification is received on mobile phone
- Receive Blood Oxygen Anomaly Notification
  - Ensure switch is on in settings
  - Ensure blood oxygen readings are being stored in core data
  - After 5 minutes, ensure notification is received on mobile phone
- Receive Medication Notification
  - Ensure switch is on in settings
  - Set time to any time during the day
  - Ensure notification is received on mobile device at set time
- Receive Heart Rate Anomaly Text Message for Emergency Contact
  - Ensure switch is on in settings
  - Ensure proper phone number is added and updated
  - Ensure heart rate readings are being read and stored
  - After 5 minutes, ensure text message is received on different mobile device
- Receive Blood Oxygen Anomaly Notification for Emergency Contact
  - Ensure switch is on in settings
  - Ensure proper phone number is added and updated
  - Ensure blood oxygen readings are being read and stored
  - After 5 minutes, ensure text message is received on different mobile device

VALIDATION RESULTS

The team is now in the process of completing testing of the different components to ensure they pass both the equivalence class tests and the boundary value tests for any input text fields that we have control over.

For the login use case tests, the team has noted that they do not have full control over the text fields at this time due to the decision to use FirebaseAuthUI to login. The team has concluded that the rest of the account use cases have been verified and closed.

The team has concluded that the sensor use cases have been verified and closed.

The team has concluded that the notification use cases have been verified and closed.

The team has concluded that the questionnaire use cases have been verified and closed.

The team has shown the final version of the project to the domain expert and have received feedback on what areas could be improved upon. These areas have been noted in the future work section.

CONCLUSION

Project Fe was founded upon the simple goals of creating an application that would improve the quality of life of Anemia sufferers by taking some of the mental processing involved in maintaining good health off the user's plate. By constantly monitoring the user's health and alerting them to urgent health changes, the user can worry less about continually keeping tabs on their own body's signals and enjoying the moment a little more. By tracking trends in the user's health and storing periodic reminders such as medication, the user can worry less about catching minute changes in their well-being.

Fe is close to a release-ready state, but still requires some minor bug fixes to be completed and handling of data older than 1 day, whether it be used in graphs or dropped to save space. The implementation of the app in the background is not ideal and would currently be rejected by the AppStore if submitted due to our use case not ticking any of the current approved reasons for allowing our app to run in the background.

In its current state, Project Fe demonstrates the ability to access cloud storage systems (Google's Firebase), authenticate users, access biometric data from a wearable sensor (our custom sensor), send notifications to users for abnormal sensor readings, send a scheduled notification for medication reminders, and send text messages to an emergency contact. With the project in its current state, it satisfies all of the requirements for a Capstone Project.

PROJECT SUITABILITY

Project Fe began as an ambitious concept, with some major uncertainties existing at the conception stage of the project. However, as research and development have continued throughout the project, the team has overcome many of these uncertainties. The major difficulties for the feasibility of this project lie in the ability to retrieve sensor information from the sensor device in real-time, an issue which the team has taken great pains to develop robust and detailed solutions to overcome. With the current state of the project, the team is confident that the major feasibility concerns surrounding the successful creation and implementation of Project Fe have been resolved. The team's confidence is based upon the successful implementation of Functional Areas that represent the major technical challenges in the creation of Project Fe.

DOMAIN EXPERT EVALUATION

Throughout the development of Project Fe, the team has consulted their domain expert for guidance and assurance that the design and functionality decisions of the project were in line with the requirements of the future users.

The team has held a recent meeting with the domain expert in November to evaluate the progress and status of the project. The domain expert has provided the following feedback.

- The expert is satisfied with the look and feel of the home screen. The text is large, and the screen is not too busy. The colors chosen don't distract the eye from the vital information.
- The expert noted a bug that the sensors needed to be turned on after initial account creation before they could be used. This has been addressed.
- The expert was satisfied with the features for notification on low or high heart rate or blood oxygen.
- The expert loved the daily medication reminder feature and noted they would like to select a time instead of having it at 8 AM every day. This has also been addressed.
- The expert was happy with the time interval that data was being pulled.
- The expert was happy with the time interval that data was being analyzed.
- The expert provided a suggestion where if a medAlert button were pressed, our emergency contact would be alerted with the current sensor readings and the person's GPS coordinates. This is being considered for future work.

As we are now in the final development stage of the project, we have submitted the project to our domain expert for a final evaluation and validation. The domain expert has made the following comments regarding the status of the project:

- The application has a very clean look and is very easy to navigate
- I'm happy the app works in the background, and I don't have to keep it open
- It would still be really nice if it could connect to a medAlert button
- The chatbot looks excellent but couldn't answer a couple of questions about the app like:
    - What is altitude for?

Notable decisions the team made based on domain expert recommendation include:

- Questionnaire symptom list
- Reviewing past sensor readings
- Sensors for heart rate and blood oxygen needed
- Notifying the user of unusual sensor readings

USER TESTIMONIALS

As Project Fe has now ended the Final Release phase, we can release it to stakeholders for user testimonials. The development team continues to develop Fe with the stakeholder's primary use-cases in mind and anticipates positive user testimonials upon completing SIT and the beginning of UAT.

FUTURE WORK

Currently, the project has completed all major use cases and can run in the background for as long as the user has the application open in the background. The future work for Project Fe contains the following sections, divided by their planned implementation timeframe:

- To be implemented at a future date
    - Notifications for tertiary and quaternary use-cases
    - Persuasive Technologies
    - Background Readings

Please find each of the future work sections detailed below:

Notifications:

There are several different practical uses for notifications within the scope of project Fe. They are listed below:

- The initial and most urgent use of notifications within Fe is the urgent health alert system. This system will be developed to notify the user when any of their sensor readings exceeds or falls below a given threshold, indicating a dangerous state of health. The user will be provided the appropriate recommendation from the application and/or a hotkey to dial emergency services (if applicable, given the scenario).
    - An example of such a notification is as follows: "Your heart rate seems to be dangerously high. You may be in danger. Tap here to call emergency services."
    - Such notifications will also be accompanied by an audible alarm of the user's choosing.
- The secondary use-case for notifications in Fe is to remind the user of non-urgent health-related items. These include reminders to take medication, exercise, maintain a specific diet (such as intermittent fasting). It will also offer more information regarding the notification, should the user feel the need for more context.
    - An example of such a notification is as follows: "Reminder: It is time to take your iron supplement pills. Taking them will earn you 100 Fe-points! Tap here for more information on the importance of iron supplements in anemic patients."
    - These notifications will carry the same level of urgency as a text message or email.
- The tertiary use-case for notification in Fe is as a reminder for appointments and to book appointments. When an appointment is booked into the user's calendar, Fe will remind them of the appointment. If the user has no

future appointment booked or the appointment is deemed too far from the current date, Fe will remind them to book an appointment and suggest a date.

  - An example of a notification for a doctor's appointment: "Fe Reminder: You have a doctor's appointment this upcoming Tuesday at 4 PM."
  - These notifications will carry the same weight as a calendar notification for the user's device.
- The quaternary and currently final use-case for notifications within Fe is a gentle notification when data is transmitted to a doctor or caregiver. This is an automatic process that happens in the background. To ensure transparency for the user's comfort, the application will deliver a notification when this happens.
  - An example of such a notification is as follows: "Health metrics for the past week have been sent to Dr. Merinchuk. Tap here for more information."
  - These notifications will just be a slight "ping" with a top banner.

Persuasive Technologies:

The team will employ Persuasive Technology strategies to encourage user participation and retention within Fe, as well as improve the overall impact that usage of Fe provides on a user's quality of life. Persuasive Technologies – Proposed by B.J. Fogg and further researched by Sajanee Halko and Julie Kientz – refers to technology designed to change behaviors of users through persuasion and social influence. Halko and Kientz's research on the persuasive strategies used to modify health-related behaviors provides the basis for the team's approach in implementing persuasive technology, specifically using the four primary categories and their complementary approaches.

- Instruction Style
  - By customizing the language for the user, we can provide a non-authoritative instruction style that will guide the user in a manner that does not feel strict or oppressive. By tailoring the language used within the application to the user, the user can relate more personally to the information. This reduces the "abstract" feeling that medical information can have on users, making the user more comfortable with the information and suggestions while making the application more user-friendly. This can increase patient compliance regarding their diet, exercise, and regularly taking medication, among other factors.
  - According to the U.S. National Library of Medicine, patient noncompliance results in hospitalization costs as high as $13,350,000,000.00 USD annually. [5]
  - Unfortunately, the more medication a patient is required to take, the lower the chances they will adhere to their medication regimen. The adherence rate of patients taking one pill daily was 84% [5] this percentage dropped drastically to just 59% among patients taking three pills daily. [5]
  - Adherence to recommendations involving lifestyle changes such as increased exercise also poses an issue for doctors. A study in the Netherlands found that just 35% of patients prescribed a physical therapy regiment adhered to it wholly. [6] Scientists have found that regimens requiring the modification of existing habits (such as eating) can have a nonadherence rate of up to 70%. [5]
  - An example of the difference in instruction style from user to user would be:
    - Adult user: "Your heart rate is up 25% from last week. See here for advice."

    - Child user: “Last week’s heart points: ★★★★☆, this week’s heart points: ★★★☆☆. Let’s get your score up! Click here for tips.”

- Social Feedback
  - By providing the user with community metrics, we can offer a collective sense of community. This sense of community allows the user to better understand where they stand among their peers in their health.
  - This social feedback can also behave as a motivator. Those who are in the lower half of their class may see it as a “wake-up call” and make an effort to improve their score (and thereby their health). Those who are acceptably healthy may see it as an opportunity to “compete” against their peers, also improving their health.
  - Example metric: “Your blood pressure is among the top 10% in your weight class.”
  - Example metric: “You have remembered to take your medication ten days in a row. That’s better than 63% of Fe users! Keep it up!”
- Motivation type
  - Gamifying the health-related goals of the user – by offering trophies, badges, and an overall score, we can provide extrinsic motivation. Motivating the user, keeping them engaged, and returning to Fe fulfills the desire of the user to accomplish something meaningful, one of humanity’s eight Core Drivers. [7]
  - Such trophies and rewards must be diverse, public, and be scarce to satisfy more of humanity’s core drivers, which Yu-kai Chou postulates. [7]
    - Social influence – by making these trophies public, the user accruing these points has something to display for their accomplishments publicly. Integration with major social media sites can be implemented to further this goal.
      - Example: Ran every day, 2km for 20 days. A trophy is awarded to the user, which shows up on their Fe profile. This trophy can (with user permission) also be posted to Facebook for their friends/family to see.
    - Unpredictability – by having some level of randomness to the awards given, we further entice the user to collect as many achievements as possible.
      - Much like video games use “Shiny” or other unique characteristics to differentiate random, rare items, a “Shiny” trophy will be awarded at random upon achieving the related trophy goal. This further motivates the user to reach these goals.
  - Some gamification examples:
    - “You have logged in every day for 30 days in a row! You have earned the Bronze participation trophy.”
    - “Your health has been stable with no alerts or alarms for 15 days! Keep up the great work. You have earned the Bronze health trophy.”
    - “Congratulations! You have taken your medication daily for 20 days. You have earned 100 Fe-Coins. Fe-Coins can be redeemed for profile customizations!”
- Reinforcement type
  - By providing gentle, positive reinforcement within the GUI of the application, we can provide the user with a small reward for their repeated positive behavior.

  - For example, by adding a glowing effect to their app home screen, we can show they have met their exercise goals every day for a 7-day streak.
  - Another example: User-unlockable rewards. As the user collects Fe-Coins they can add "charms" to the application. These "charms" are small graphical elements that will appear on the screen only when the user is above specific thresholds/goals. "Charms" can be butterflies that float on the top of the screen or effects when selecting buttons.
  - This effect is subtle, but the slight positive effect of adding warm, positive elements to the UI of the application and, conversely the lack of those elements when the user falls below their health thresholds can make a notable impact on the user's desire to adhere to the recommendations of Fe.

Background Readings:

The team has implemented a solution to have the application constantly running in the background to pull sensor data, but the solution is not ideal. The current solution involves running a song called "silence.mp3" in the background to ensure the timer can continue to run and pull data. We had to develop the solution like this because timers are not allowed to be run in the background for any reason. Functions usually are only allowed to run for a few seconds to up to 3 minutes after an application has closed. Our use case does not sit within these parameters, which is why we have had such a difficult time finding the right solution.

TECHNICAL RESOURCES:

This section is a collection of technical resources used in creating the project and a brief description of their benefit to this project.

- StackOverflow (Website) [8]
  - Used multiple community pages within StackOverflow to solve development hurdles
- GitHub (Website) [9]
  - Used multiple GitHub repositories as examples and guidance for specific problems with Project Fe
  - Some specific examples include:
    - Heart Rate Realtime [10]
    - ZSwift [11]
    - NYXO Blogposts [12]
    - Charts Master [13]
- Apple Developer Documentation [14]
- Firebase Developer Documentation [15]
- Previous PBACS Projects
  - Parking Project from iOS Semester 5

BIBLIOGRAPHY

[1] "Health Care and Social Assistance - 62 - Summary - Canadian Industry Statistics - Innovation, Science and Economic Development Canada," Government of Canada / Gouvernement du Canada, 01 March 2019. [Online]. Available: https://www.ic.gc.ca/app/scr/app/cis/search-recherche?lang=eng. [Accessed 21 January 2021].

[2] Google, "Privacy and Security in Firebase," Google, [Online]. Available: https://firebase.google.com/support/privacy. [Accessed 29 01 2021].

[3] "Buy iPhone 12 and iPhone 12 mini," Apple (CA), 2021. [Online]. Available: https://www.apple.com/ca/shop/buy-iphone/iphone-12. [Accessed 22 January 2021].

[4] "MultiBrief: Everything you always wanted to know about blood doping and EPO," MultiBriefs, [Online]. Available: http://exclusive.multibriefs.com/content/everything-you-always-wanted-to-know-about-blood-doping-and-epo/sports-fitness#:~:text=It%20has%20been%20a%20useful,4%20weeks%20could%20cost%20%24500.. [Accessed 29 01 2021].

[5] "Therapeutics and clinical risk management - The challenge of patient adherence," NLM-NIH, September 2005. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1661624/. [Accessed 14 April 2021].

[6] E. M. Slujis, "Correlates of Exercise Compliance in Physical Therapy," American Physical Therapy Association Inc., 01 November 1993. [Online]. Available: https://academic.oup.com/ptj/article-abstract/73/11/771/2729054. [Accessed 14 April 2021].

[7] Y.-k. Chou, "Octalysis: Complete Gamification Framework," Yu-kai Chou, 08 March 2021. [Online]. Available: https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/#.Vv0TRBJ95E4. [Accessed 14 April 2021].

[8] "Stackoverflow," [Online]. Available: http://www.stackoverflow.com.

[9] "Github," [Online]. Available: http://www.github.com.

[10] "Github : Swift Heart Rate Real Time," [Online]. Available: https://github.com/nhathm/swift_heart_rate_real_time.

[11] H. Truong, "Github : ZSwift," [Online]. Available: https://github.com/hungtruong/Zswift.

[12] "Github : Swift and Data Blogposts," [Online]. Available: https://github.com/hello-nyxo/swift-and-data-blogposts.

[13] D. Gindi, "Github : Charts," [Online]. Available: https://github.com/danielgindi/Charts.

[14] "Apple Developer Documentation," [Online]. Available: https://developer.apple.com/documentation/.

[15] "Firebase Developer Documentation," [Online]. Available: https://firebase.google.com/docs.

[16] "Apple Watch Series 6," Apple (CA), 2021. [Online]. Available: https://www.apple.com/ca/apple-watch-series-6/. [Accessed 22 January 2021].

[17] "Fitbit product protection," Advanced Health Smartwatch | Fitbit Sense, 2021. [Online]. Available: https://www.fitbit.com/global/en-ca/products/smartwatches/sense?sku=512BKBK. [Accessed 22 January 2021].

[18] healthline, "Understanding Barometric Pressure Headaches: How Does Weather Affect Your Headaches?," 7 March 2019. [Online]. Available: https://www.healthline.com/health/headache/barometric-pressure-headache#symptoms. [Accessed 2021].

[19] Hypoxico, "Altitude to Oxygen Chart," 2021. [Online]. Available: https://hypoxico.com/altitude-to-oxygen-chart/. [Accessed 2021].

[20] Devfright, "How to Access the iPhone Barometer with CMAltimeter," 2 August 2018. [Online]. Available: https://www.devfright.com/how-to-access-the-iphone-barometer-with-cmaltimeter/. [Accessed 2021].

[21] I. Virnik, "Tutorial: get user's altitude in Swift," 6 March 2019. [Online]. Available: https://medium.com/@ilya.virnik/tutorial-get-users-altitude-in-swift-33d23b299bbe. [Accessed 2021].

[22] Hacking With Swift, "Scheduling notifications: UNUserNotificationCenter and UNNotificationRequest," 2021. [Online]. Available: https://www.hackingwithswift.com/read/21/2/scheduling-notifications-unusernotificationcenter-and-unnotificationrequest. [Accessed 20 September 2021].