Sugar Coded

Requirements Specification (Version 1.2)

December 8, 2017

Project: Prediabetes Intervention Mobile Application

Sponsor: Dr. Natalia Dmitrieva

Mentor: Dr. Eck Doerry

Team Members:

Chantz Spears

Julian Shak

John Bassler

Alfonso Martinez

Accepted as baseline requirements for the project:

Sponsor: ______________________________ Team: ______________________________

Table of Contents

1.0 Introduction

Diabetes mellitus is a chronic condition characterized by elevated blood glucose levels. The reason for these elevated blood glucose levels is due to one's insulin, a hormone produced by the pancreas to filter sugar from your blood, not functioning properly. It is a disorder that currently affects over 30 million Americans. The vast majority of people with diabetes (~90%) have Type II diabetes. Those with Type II have grown resistant to their body's insulin over a long period of time (usually decades) and have a reduced capacity for blood sugar regulation. Causes of type 2 diabetes are often attributed to having a family history of diabetes, as well as a poor diet and exercise routine. Because of their high blood sugar levels, people with type 2 diabetes suffer from frequent hunger, increased thirst, fatigue, and blurred vision. Diabetes is considered to be an irreversible condition, so people with type 2 diabetes are left only to manage their condition for the remainder of their lives through various recommended strategies, including monitoring of blood sugar, carbohydrate intake, and medication use. However, type 2 diabetes prevention programs can successfully lower diabetes incidence.

One such program is the lifestyle arm of the Diabetes Prevention Program (DPP). This was designed to prevent future cases of diabetes by finding high-risk patients and offer moderate lifestyle and diet changes. These high-risk patients are usually overweight and have intermediate elevated blood sugar levels that do not yet meet the criteria for type 2 diabetes. This state describes the condition of prediabetes, and the US Department of Health and Human Services estimated a third of US adults meet the criteria for prediabetes as of 2015. The DPP's goal is to reduce the patient's body weight by 7% and the DPP is successful in delaying and even preventing future diabetes diagnoses. On average, a patient's risk of diabetes onset is reduced by 58%. The program includes patients of all ethnicities with about 45% of program participants self-identified as an ethnic or racial minority. Those who stay on the program all have a similar reduction.

2.0 Problem Statement

However, retention among the Native American community in the DPP is low. Native Americans have a high withdrawal rate from the DPP with approximately one-third prematurely dropping out of the 16-session program. In the United States, compared to individuals of other racial and ethnic backgrounds, those who self-identify as Native

American have the highest prevalence of diabetes and, as such, may require more targeted support from the DPP.

Our client Dr. Natalia Dmitrieva is a developmental health psychologist at NAU who received her PhD at Pennsylvania State University. Dr. Dmitrieva's research focuses on identifying vulnerable subgroups in middle adulthood that can benefit from targeted intervention. To discover the reasons why Native Americans have such a hard time staying on the program, she uses a research technique used is called Ecological Momentary Assessment (EMA). In EMA, the methodology is to frequently ask the participants to record what they are feeling in the moment. Tracking how participants are feeling throughout their days and weeks may give an indication for why they exhibit high attrition in the program.

The current implementations of EMA are done via phone surveys, smartphone applications, and paper-and-pencil questionnaires. Regarding phone surveys, EMA participants are required to verbally respond to phone surveys throughout their day administered by pre-recorded automated systems. Researchers then have to spend time listening to audio of study participants where they answer questions such as their current mood, state of mind, and how they are doing with their diet. These methods lead to data collection being time-consuming and expensive, while also making it difficult to have a large number of survey participants. The current smart phone application methods are very limited, and for these reasons, Dr. Dmitrieva is interested in implementing EMA within a new mobile application to aid in her research. The goal of this mobile application is to allow for more efficient data collection for research purposes as well as creating a less intrusive way to collect data from the participants.

3.0 Solution Vision

Our Solution

Team Sugar Coded will be working closely with Dr. Dmitrieva in order to design and implement a highly configurable mobile application and web portal as a solution to the aforementioned problem above. Figure 1, seen below, depicts a simple diagram of our solution, SCHEMA (Sugar Coded's Handheld Ecological Momentary Assessment), where all users will have an application that receives protocols the administrator sends out, these applications send responses back to the server which then provides the administrator with useful data.

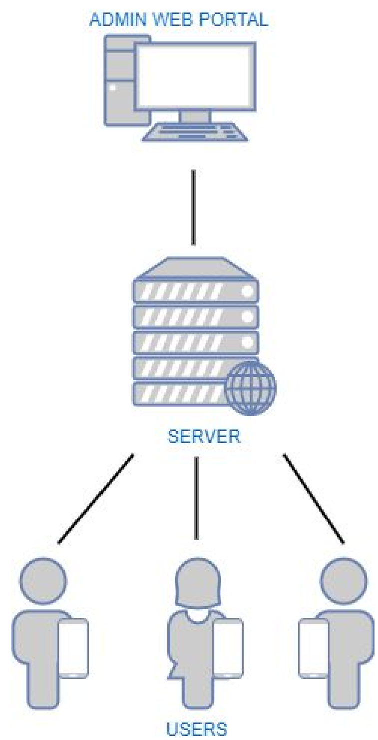

Figure 1: Solution Diagram

The long term objective is to build a smartphone-based application that will support American Indian patients with prediabetes in eating healthier meals and engaging in more physical activity, with the ultimate goal of reducing conversion from prediabetes to type 2 diabetes in this population. The mobile application that we are developing is designed to be a quicker, and cheaper alternative to gathering data compared to what is currently implemented, phone and paper surveys. The web portal will be an easy way for the researcher to view the collected data in real time as the study progresses. What follows is a list dictating the high level features we plan on providing.

High-Level Features

- Obtain Baseline User Data
  - Purpose: Identify stable patient-level characteristics that are associated with poorer health behaviors (e.g., socioeconomic characteristics, individual and family factors, etc).
- Time-Based Data Collection
  - Momentary Fitbit data
    - Purpose: Identify timing and duration of physical activity episodes (e.g., sedentary behavior, vigorous activity, etc)

    - Hourly Questionnaire
        - Purpose: Identify momentary precursors and consequences of poor health behaviors (e.g., negative affect, momentary stressors, social support, physical context, food availability, etc)
- Participant-Initiated Logs
    - Purpose: Identify self-reported eating behavior duration and timing (e.g., normal eating, overeating, etc)
- Administrative Web Portal
    - Purpose: a portal for researchers/clinicians to access data as close to real-time as possible; a potential resource for administering a specific set of questions or interventions to a subset of participants.

The next points provide some further clarification and give a rough outline of what we look to provide.

What data does SCHEMA generate?

The system will generate data logs from each user, everything from their initial demographic responses to their daily meal logs. The individuals data can be categorized on the web portal by the research administrator and conclusions can then be drawn from these larger subgroups. This ease of large quantity data generation and collection is one of the main necessities for this application's development, and will create less work for research staff.

How will the system change the way our client works?

Dr. Dmitrieva and her colleagues will be able to assess momentary characteristics of patients' environment and their health behaviors. Data will be added into a database in close to real time, so data can easily be looked at throughout the study. Through Fitbit data collection, the researchers will gain time-synced information on physical activity.

Other avenues considered for solving this problem

The team looked at over fifteen data collection applications that already exist, the trouble being none of them met all of our clients needs. For example, one of them called Device Magic did not have the ability to send out new protocols, was mainly used for pre-filled data forms, and cost $17 a month per device. Many others lacked the individualized experience our application is looking to deliver as well, they could not send notifications, they were expensive, outdated, lacked Fitbit integration and dynamic data viewing web portal.

How could our project change the world?

If we can make the product modular enough, other researchers and clinicians would be able to take our application and apply it to their field of study, including drug addiction, physical training or rehabilitation, or numerous other behavioral research topics. A free and open source product like this could save researchers thousands of hours of work and lead to more research being completed, and produced effective patient-centered interventions for various chronic conditions.

Workflow

Figure 1 pictured below details the components and their workflow for our solution. The mobile application, web portal, and Fitbit data will all be utilizing a single database to store and retrieve data. The administrator web portal can then view data from the other two components.

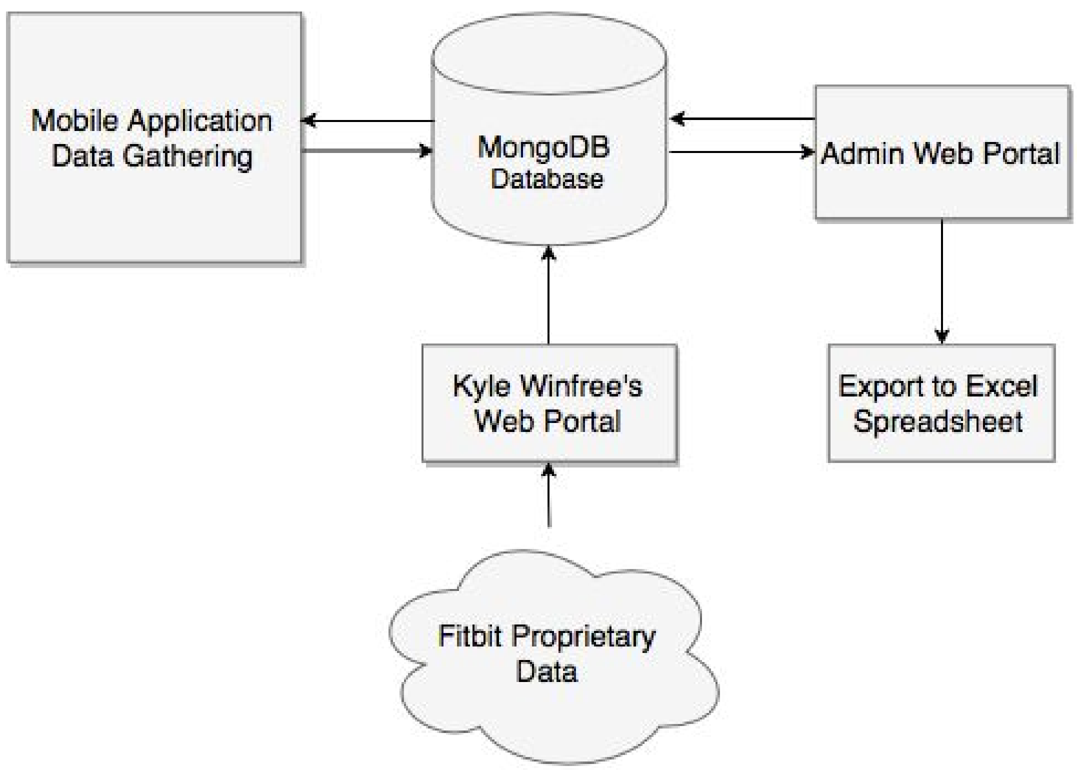

Figure 2: Component Overview

The following section presents the functional, performance, and environmental specifications of SCHEMA in full detail.

4.0 Project Requirements

To develop the following requirements the team has been through multiple client meetings since the start of the project, bouncing ideas off of Dr. Dmitrieva until we found feasible requirements that satisfy her needs. After our technical feasibility was laid out we discussed among the team requirements that can be implemented using the technologies we chose to work with. To start of the section below we go into the domain requirements which lay out the general specifications the client expects out of SCHEMA.

Domain Level Requirements

The following are the features from the domain perspective:

DR1 Keeping Track of Participant's Data

Each research participant will need to have their information easily identifiable, and stored and accessible separate from everyone else.

DR2 Participant Self Reporting

Users need to be able to report on key behaviors such as when they have eaten something, and how they were feeling at the time.

DR3 Send reminders / direct users to questionnaires

The administrator needs to be able to set push notifications to be delivered to users at certain intervals and direct users to additional questionnaires.

DR4 Question Delivery Protocols

Must support creating, editing, and delivering complex question protocols including the ability to specify timing, events, or user initiated protocols.

DR5 Administrative Abilities

The researcher requires the ability to setup and manage study participants, and download datasets from the research for analysis.

DR6 Fitbit data collection

Collect and store individual users Fitbit information which contains data such as steps taken and heart rate at any given time.

DR7 Hybrid application

The mobile application should work for both Android and IOS as to not alienate the majority of research participants.

The following requirements sections will take a deeper look into the finer points of our project and relate them back the domain level requirements. Specifically, each requirement is attributed to a single domain level requirement, where '**DR#**' stands for domain requirement, '**FR#**' stands for functional requirement, and so forth.

4.1 Functional Requirements

The functional requirements entail every function that our product will need to perform or have. The functions will be presented initially at high level and gradually increase into more detailed lower level functionality.

DR1.FR1 **Secure login**

- Users will each be set up with a single account that will have a secure login to protect their information, this information will be kept confidential and is for research staff eyes only.
- Create a username and password interface for users to create an account they can securely log in and out of
- Allow users to stay logged in without the need to log in to the application each time they open it.
- Allow users the ability to log out of the application

DR1.FR1.1 **Store usernames and passwords in a secure database**

- Our database will need columns for both usernames and passwords
- The database needs to be secured and only accessible to the research team

DR1.FR1.2 **HIPAA compliance**

- Login process and data management must be HIPAA compliant
- Use a HIPAA compliant cloud data storage solution such as AWS
- Safeguard PHI data

DR2.FR1 **Track fluctuations in health states**

- User consistently logs events such as eating through in app interfaces
- Alongside each event log the user will be asked questions to determine their current momentary stressors, social support, physical context, food availability, etc.

DR3.FR1 **Send users reminders**

- Push notifications will be used to:
    - remind users to log events
    - notify them of any urgents messages
    - send specific subgroup alerts

DR3.FR2 **Direct users to questionnaires**

- The ability to direct users to a new or recurring poll via notifications
    - A push notification will be sent out whenever the user has a poll they need to take

DR4.FR1 **Question Delivery Protocols**

- Some participants should be able to skip certain questions and actions depending on their subgroup, previous questions answered, and factors chosen by the researcher.
- Users placed into specific subgroups depending on their answers to questionnaires and baseline questions.
- Individuals can specify interruption times to a certain degree.
    - If they go to bed at 8pm they will not want to be interrupted at 9pm
- Special alerts can be sent to users that are in specific subgroups

DR4.FR1.1 **Branching logic**

- Conditional statements will determine what a user is asked, what they are allowed to skip, etc. based on information stored about this user in the database.
- Conditionals determine interruption windows

DR5.FR1 **Create an administrator web portal**

- Utilize web2.0 ideology to create a web application for the research administrator.

DR5.FR2 **Set up and manage research participants**

- Through the web portal the researcher will set up each research group with a ‘study-specific’ to be admitted into the study
    - On the participants end they will need to enter this code to finish their registration
    - The code will be secured in our database and checked against the participants input to insure only verified users are participating in the study.

DR5.F3 **Data viewing**

- Researcher can view data gathered from the participants in near real time
    - This data will be pulled from our cloud hosted database
    - The data will be displayed using a chart based javascript tool such as D3 or chart.js

DR5.F3.1 **Downloading data sets**

- The ability to download .csv files generated from our cloud data storage such as Excel files

DR5.F4 **Manually group participants**

- Researcher can select users based on how they have answered certain questions or individually to assign them to certain sub pools
    - We will query the database to find who the researcher is looking for, then update the individual's 'group' field

DR5.F5 **Send out notifications**

- Researcher can send out notifications to all or specific individuals to alert them of anything they should need
    - Push notifications will be sent to the mobile applications detailing the message the researcher selects
    - Have the researcher select which participants they would like the notification to reach via various selection boxes
        - Query the database to determine which users to send the notifications to.

DR6.FR1 **Gather users Fitbit data**

- Collect users steps, heart rate, and more to deliver to the researcher web portal.
    - Fitbit data collected from Dr. Winfree's web portal and placed into our cloud database. From the database we relay the Fitbit info back to the researcher.

DR7.FR1 **Mobile application runs on Android and IOS**

- The mobile portion of our project needs to be able to seamlessly run on either operating system, Android or IOS.
    - Developing the application in a hybrid framework such as Meteor or Ionic allows for easy integration into either OS.
- The application will be backwards compatible with recent OS versions

4.2 Performance Requirements

The performance requirements detail the conditions in which out product is supposed to function. These functions are measurable and testable, unlike the functional requirements and include some quantitative qualities.

DR1.PR1 **Quick account creation**

- Users will receive a Unique ID from researcher
- Estimated time to fill out single use form
  - Unique ID - ≤20 seconds
  - Button submission - ≤2 seconds
- Process should take ≤ 30 seconds

DR1.PR2 **Fast log on and quick turnaround for users to log events**

- Application will keep users logged into their accounts for quick use
  - Pressing re-login - ≤ 2 seconds

DR2.PR1 **Entering Participant-initiated logs and time based logs**

- User logs emotions, positive experiences, stressors, and behaviors after an activity or responds to a time based query
- Time to enter form respectively
  - Type of log - ≤ 3 seconds
  - Log Data - ≤ 3 minutes
  - Open ended voice to text box ≤ 20 seconds
  - Button submission - ≤ 2 seconds
- Process should take ≤ 3.5 minutes

DR3.PR1 **Mitigate Intrusive Interruptions**

- Application will take in user's initial sleep hours, and any other busy hours in order to create an interruption time for that specific user, which determines the frequency of the application's push notifications
- User's will be choosing a time to take "End-of-Day" questions
- Time to enter form respectively
  - Work hours entry - ≤7 seconds
  - Sleep hours entry - ≤7 seconds
  - Other hours entry - ≤7 seconds
  - "End-of-Day" entry - ≤7 seconds
  - Button submission - ≤2 seconds

- Process should take ≤ 23 seconds
- Interruption time determines frequency of the application's push notifications

DR4.PR1 **Minimal need for application usage training**

- Simple and elegant design so user is able to quickly and efficiently understand the sections of the application
- Should take user no more than 3-8 minutes to fully understand UI if they are comfortable with modern mobile applications, otherwise 12 - 20 minutes

DR5.PR1 **Ability to add participants**

- Any researcher will be able to add any questions to all participants
- Time to enter form respectively
  - Enter Unique ID - ≤5 seconds
  - Enter name - ≤5 seconds
  - Enter dob - ≤5 seconds
  - Enter gender - ≤3 seconds
  - Submit - ≤ 2 seconds
- Process should take ≤ 20 seconds

DR5.PR2 **User data operations**

- GET, PULL, PUSH Requests: Should take no more than 5-10 seconds to retrieve and send User data

DR6.PR1 **Fitbit data operations**

- GET, PULL, PUSH Requests: Should take no more than 5-10 seconds to retrieve and send Fitbit data

4.3 Environmental Requirements

ER.1 **Fitbit integration**

- Allow further data collection through our app by integrating data collected by from a Fitbit worn by the app user.
- Require participants to connect to the internet at least once a week to allow data from Fitbit to be offloaded

ER.2 **The app will be developed for both Android and iOS**

- App functionality must be consistent across both platforms to ensure app users have similar experiences
- Development of the app will be done in a hybrid environment

ER.3 **Prevent researchers from seeing personal info of participants**

- Allow for app users to be anonymous to the researchers tracking a participant's progress

5.0 Potential Risks

Now that we have gone over the requirements, we will look into the potential risks that come with the overall project. Here is a list of possible threats and potential hazards if certain requirements are not met. These are very detrimental because they can cause data uncertainties for the researcher.

Overview Table

Risks	**Severity**	**Likelihood**
User purposely inputs false information	Severe	Low
User ignores prompts	Severe	Moderate-High
Fitbit Integration	Moderate	Moderate-High
Application not synced with server	Low	Low-Moderate

User purposely inputs false information

Our goal is to make sure that our application is able to display the data through the admin portal. One thing that we are not able to know is if the user is purposely inputting false answers for the questions. This would be very severe because it will result in false conclusions, hurting the researchers study.

The reason why a user might purposely input false information is to make it look like they are on track with their dieting and exercising. This is common when job interviews require someone to complete a survey. They want to look good for the employer so they have a higher chance to be picked.

The good news is that according to studies, the chance for this to happen is relatively low. During these types of studies, users tend to answer the questions truthfully.

User ignores prompts

There are times when users are already on their phone, looking through social media. During this time it is common for the user to ignore push notifications because it is distracting to whatever they are doing.

If users ignore the prompts, this will create missing data for the researcher. It is very important for the user to answer these questions at requested time in order to get concrete and consistent data from the user.

By following our low-intrusion approach we will mitigate this risk as much as possible so that these prompts do not interrupt the user throughout their day. Researchers may choose to offer monetary incentives to further decrease the risk.

Fitbit integration

A fairly large risk to this project is that our application requires Fitbit integration. The method of collecting this data has multiple issues, and since it is done through a third party application there isn't much that we can do about it.

The current issues with the portal are:

1. Fitbit watches become desynced from server
2. If large chunks of data are not collected, then depending on the size of the pulled data, server times out when zipping folders.

The current solution for the first issue is to have someone physically re-sync the Fitbit watches back with the server. This is tedious and time consuming to do, and the more participants there are the longer it takes to do. This problem arises once every two months.

There is no current solution with the second issue. This forces the researcher to pull fragments of data at a time until all the data has been zipped and acquired.

Application not synced with server

One feature of our application is to allow the researcher view near real time data from the users using the web portal. This feature will not be possible if the application does not have any connection with the server. This means if a user is ever in an area without a connection the application will not be able to update the server.

This should not a critical issue for the most part as participants will be provided a 4G data plan. Just as long as the questions are preloaded within the application and the questions are given out at the set time, the researcher will be guaranteed to have their data once the client is connected.

6.0 Project Plan

In this section we outline the milestones that we shall accomplish to fulfill the requirements of this project. Our project plan is to divide our overall project into separate tasks, as displayed in Figures 2 and 3 below. Figure 2 shows the completed milestones that we have accomplished through November, and Figure 3 shows an estimated timeline for projected milestones that we will accomplished in the months ahead. You will notice that we will be embarking on at least two milestones at any given time. This is due to the short time period that we are given in order to design and implement this project. We will accomplish this process of multitasking by delegating each milestone between different members of our team.

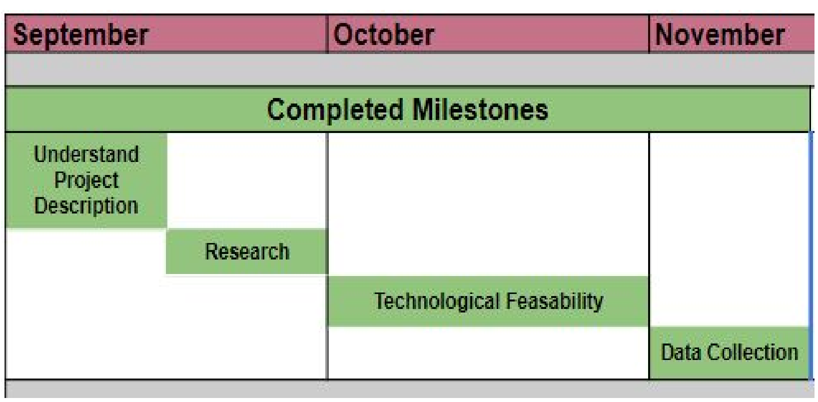

Figure 2: Completed Milestones

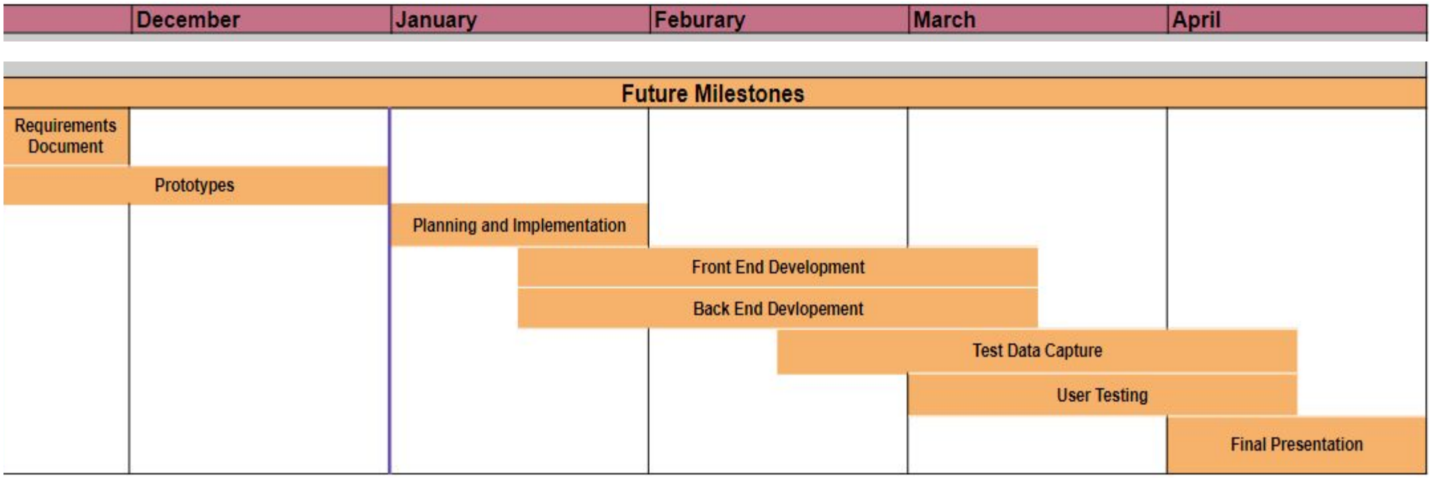

Figure 3: Future Milestones

In the list below we go more in depth on the future milestones displayed above, along with any subgoals that are associated with them.

Future Milestones:

1. **Prototype**
    a. Determine which components of our project will be the most difficult to implement.
    b. Identify a set of alternative solutions to these implementation challenges
    c. Perform research/read through code, and then make a final decision on which alternative we will proceed with.
    d. Create a technological demo to demonstrate the feasibility of these identified components.
2. **Planning and Implementation**
    a. Begin planning the implementation of our project.
    b. Start construction of our project's mobile application and web portal. This will be much more of a "skeleton" implementation.
    c. Identify overall color scheme of mobile application and web portal.
    d. Determine the format of data stored in MongoDB database.
3. **Front-end Development**
    a. Create one single CSS file for all elements, to keep uniform styling between whole project.
    b. Get Dr. Dmitrieva's approval on the user-end look of project.
    c. Finalize development on the user interface of mobile application and web portal.

4. **Back-end Development**
    a. Begin linking all components to central cloud hosted MongoDB database.
    b. Configuration of user account creation, time-based push notifications (based on specified protocol), and communication between mobile application and database.
    c. Research different methods on how to locally store different subgroup protocols onto individual participant's devices upon initial login.
    d. Make communication with database as efficient as possible.
5. **Test Data Capture**
    a. Deploy application
    b. Ensure that data is collected as expected into our cloud hosted MongoDB database.
    c. Test data collection between all components (mobile application/web portal) and different mobile platforms (iOS and Android).
6. **User Testing**
    a. Collect a small group of individuals that will test the functionality of the mobile application.
    b. Allow Dr. Dmitrieva to play the Admin role with this small group of testers.
    c. Finalize any changes requested or issues found.
7. **Final Presentation**
    a. Have completed product and finished presentation for end of semester.

7.0 Conclusion

As stated previously, approximately 10% of Americans have diabetes mellitus (32.3 million), with the highest rates observed among individuals who self-identify as American Indian and Alaska Native (15.1%). A large body of research has examined effective approaches to reducing diabetes incidence in this population. Currently there is an effective 16-session program called Special Diabetes Program for Indians Diabetes Prevention (SDPI-DP), however the retention rate is problematic, with only 74% of participants remaining in the program by the 16th session, and only 33% remaining by the 3-year follow-up. To increase retention of among American Indian patients with prediabetes in SDPI-DP, the long-term goal of this work is to develop a mobile application that will support attendance and engagement in SDPI-DP. Our Capstone project will create a prediabetes intervention mobile application that will be used to research psychosocial experiences and health behaviors among American Indians who are enrolled in SDPI-DP. This application will collect momentary data on various psychological, social, dietary, and physical activity experiences, while making sure that the data management is HIPAA compliant.

This requirements specification document clearly details our project's requirements, both functional and non-functional, and forms the contractual basis for the expectations to be fulfilled by our Capstone team. This document helped our developmental team line up our expectations with our project sponsor as well as identify key future milestones that we will face in the months ahead. We are excited to finalize the key requirements for our project and continue any additional research to further prove the feasibility of our proposed implementation.

8.0 Glossary

AWS - Amazon Web Services, which offers reliable, scalable, and inexpensive cloud computing services.

Chart.js - Simple, clean and engaging HTML5 based JavaScript charts. Chart.js is an easy way to include animated, interactive graphs on your website for free.

D3 - D3 is a JavaScript library for visualizing data with HTML, SVG, and CSS.

Fitbit - Activity trackers, wireless-enabled wearable technology devices that measure data such as the number of steps walked, heart rate, quality of sleep, steps climbed, and more.

HIPAA - The Health Insurance Portability and Accountability Act of 1996 is United States legislation that provides data privacy and security provisions for safeguarding medical information.

Meteor - Meteor is a complete platform for building web and mobile apps in pure JavaScript.

MongoDB - Mongo database is an open source database that uses a document-oriented data model.

PHI - Protected health information (PHI) under US law is any information about health status, provision of health care, or payment for health care that is created or collected by a Covered Entity (or a Business Associate of a Covered Entity), and can be linked to a specific individual.

User Interface (UI) - The junction between a user and a computer program. An interface is a set of commands or menus through which a user communicates with a program.

Web2.0 - Websites that emphasize user-generated content, usability (ease of use, even by non-experts), and interoperability (this means that a website can work well with other products, systems, and devices) for end users.