EasyGlucose

CMPT 276 – Group 01 – Glucinators
Assignment 2 – Design Document

Website: https://sites.google.com/view/cmpt276-summer2018/

Anmol Bajaj

Faisal Atif

Zhixin Huang

Tony Liu

Henry Yip

June 22th, 2018

1. Table of Contents

2. Revision History

Revision	Status	Publication/Revision Date	By
0.0	Rough draft created with requirement for each category	June 11, 2018	Henry Yip
1.0	Design document created	June 15, 2018	Everyone
1.1	Added rough writing with each part	June 15, 2018	Henry Yip
1.2	Assigned sections of plan for each group members	June 15, 2018	Everyone
2.0	Added “Guidelines”	June 17, 2018	Henry Yip
2.1	Added “System Diagram”	June 17, 2018	Faisal Atif
3.0	Modified “Guidelines” & “System Diagram”	June 18, 2018	Anmol Bajaj
3.1	Added “System Diagram”	June 18, 2018	Faisal Atif
3.2	Added “Data Requirements”	June 18, 2018	Henry Yip
3.3	Added “Feature Priority”	June 18, 2018	Everyone
4.0	Modified the report	June 20, 2018	Everyone
4.1	Formatted the file	June 22, 2018	Zhixin Huang

3. Development Guidelines:

The developers will abide the following technical and legal/ethical guidelines in the development process for EasyGlucose.

Technical guidelines:

1. Developers should use Xcode 9 as their main IDE to develop EasyGlucose.
2. Developers should use Git as their version control software.
3. Developers must write the source code for EasyGlucose in the programming language Swift.
4. Developers should follow the following naming style guidelines for code consistency

Subject	**Style**
Variable names	Lower case camel case Eg. camelCase
Method names	Upper case camel case Eg. CamelCase
Constants	All caps with underscores Eg. ALL_CAPS

5. Developers should use meaningful and descriptive words when following the above styling guideline to improve code readability.
6. Developers should reduce code complexity by modularizing functions.
7. Developers should produce concise and sufficient documentation for each file, function, class, and class method.

Ethical/legal guidelines:

1. Developers must not infringe existing copyrights when developing EasyGlucose or writing related documents.
2. Developers must not plagiarize existing information when developing EasyGlucose or writing related documents.
3. Developers must give credit to referenced work and external help where appropriate.
4. Developers should follow the IEEE code of ethics when developing EasyGlucose or writing related documents.
5. Developers should follow App Store requirements when developing EasyGlucose.

4. System diagrams

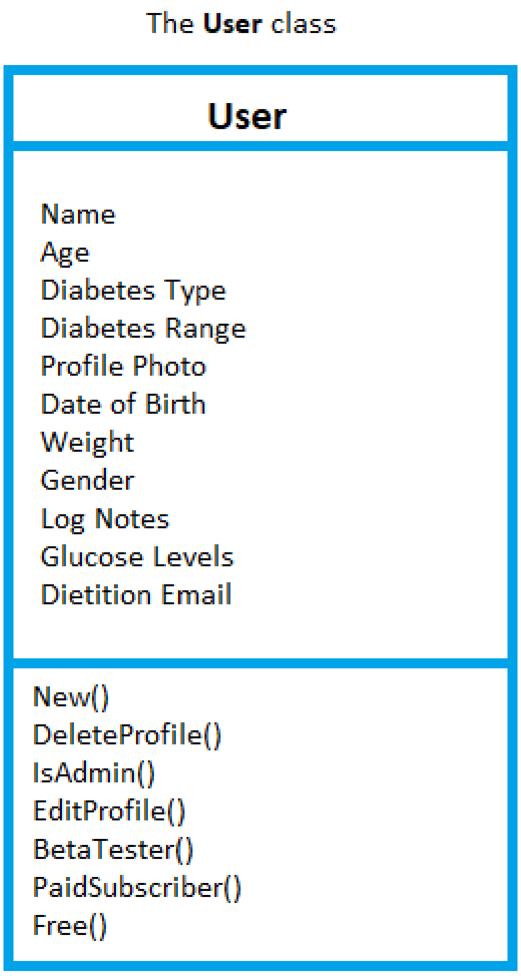

Figure 1.a

The User class is defined in Figure 1.a. This is a critical component of our application as it handles all user details. The data we intend to store is listed on the first half and the methods for the class are listed on the bottom half. Below is the explanation for the purpose of each method we intend to create:

New() - Initialized during a new user set up.
DeleteProfile() - Delete the profile and all associated data.
IsAdmin() - Allows important stakeholders special access to admin dashboard. It displays important information such as total number of sign ups or downloads of the app.
EditProfile() - Called when the user wishes to change their Target Glucose Range, Age, or Gender.
BetaTester() - Allows the users special access to features not pushed out to the public.
PaidSubscriber() - Ability to flag user as Paid Subscriber. This can potentially unlock special features of the app or access to dietician on-demand.
Free() - Describes whether the user is using the Free version of the app.

The **DiabetesData** class

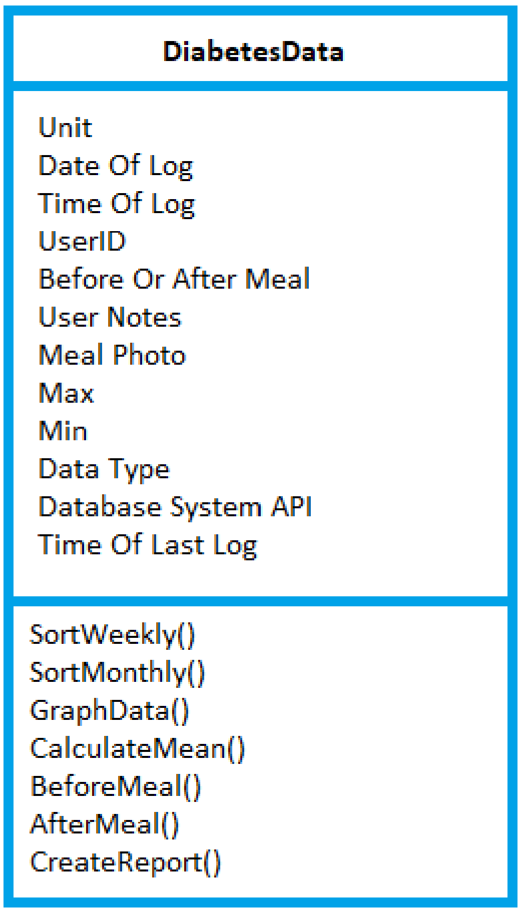

Figure 1.b

We have also created a DiabetesData class to accomplish the de facto purpose of our application: to track glucose. This class is displayed in the Figure 1.b. The DiabetesData class is responsible for storing the blood glucose data for each user, computing important characteristics and subsequently visualizing the data by creating graphs from associated data. We list the function and purpose of each method below:

SortWeekly() - Sort the diabetes data by last 7 days
SortMonthly() - Sort the diabetes data by last 30 days
GraphData() - Method to take the associated data and graph the blood glucose data on the y-axis and the date on the x-axis
CalculateMean() - Takes all blood glucose data, adds it and divides by the number of data elements, and prints the user's average blood glucose
BeforeMeal() - True/False method to tag whether the user recorded their glucose PRE-meal
AfterMeal() - True/False method to tag whether the user recorded their glucose POST-meal
CreateReport() - This method gathers all associated data, complies it neatly in a PDF format file and prepares it for email to a dietitian

Activity Model of the User Flow

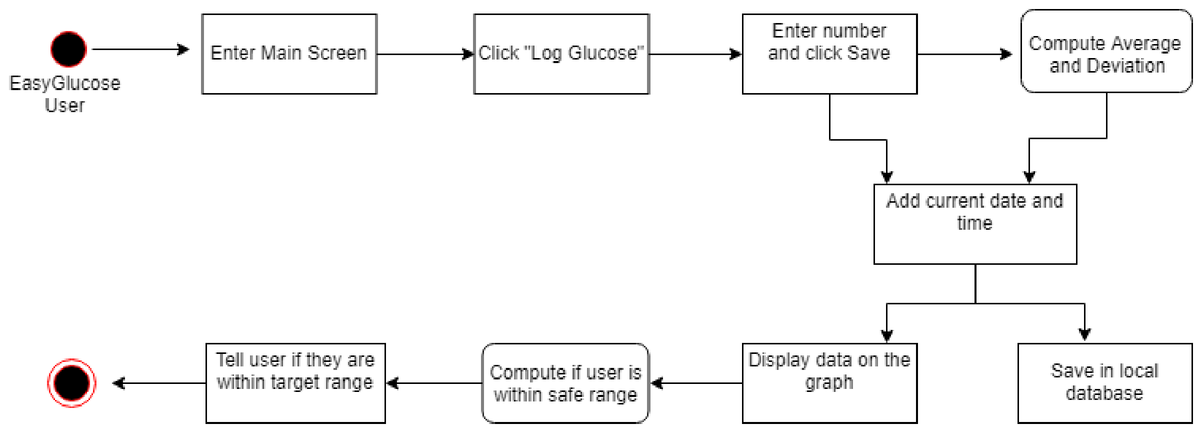
EasyGlucose User
Enter Main Screen
Click "Log Glucose"
Enter number and click Save
Compute Average and Deviation
Add current date and time
Save in local database
Display data on the graph
Compute if user is within safe range
Tell user if they are within target range

Figure 1.c

This activity model of the User Flow in Figure 1.c aims to visualize the typical actions that the user will take on the daily basis and how our application will react to their actions. The typical EasyGlucose user will want to log their blood glucose level as often as recommended by their physician or dietician.

They will start by clicking on the app icon on their iOS device and entering the main screen of our app. We plan to make the “Log Glucose” button large and minimize distractions before the glucose is logged. The user will enter the reading from their personal Diabetes Glucose meter into the text field in our application.

Our algorithm will then compute the average and deviation, including the newly input data. We will also associate their glucose data to the current data and time and save it to the local database. Our app will then call functions to visualize the data. If the user is within the safe range, we will display a “Great Job!” message. However, if the user has abnormal levels of blood glucose, we will alert the user with a “Please consult your physician or dietician” message and prompt them to email the glucose report to their dietician and/or physician.

Sequence Diagram of Report Creation

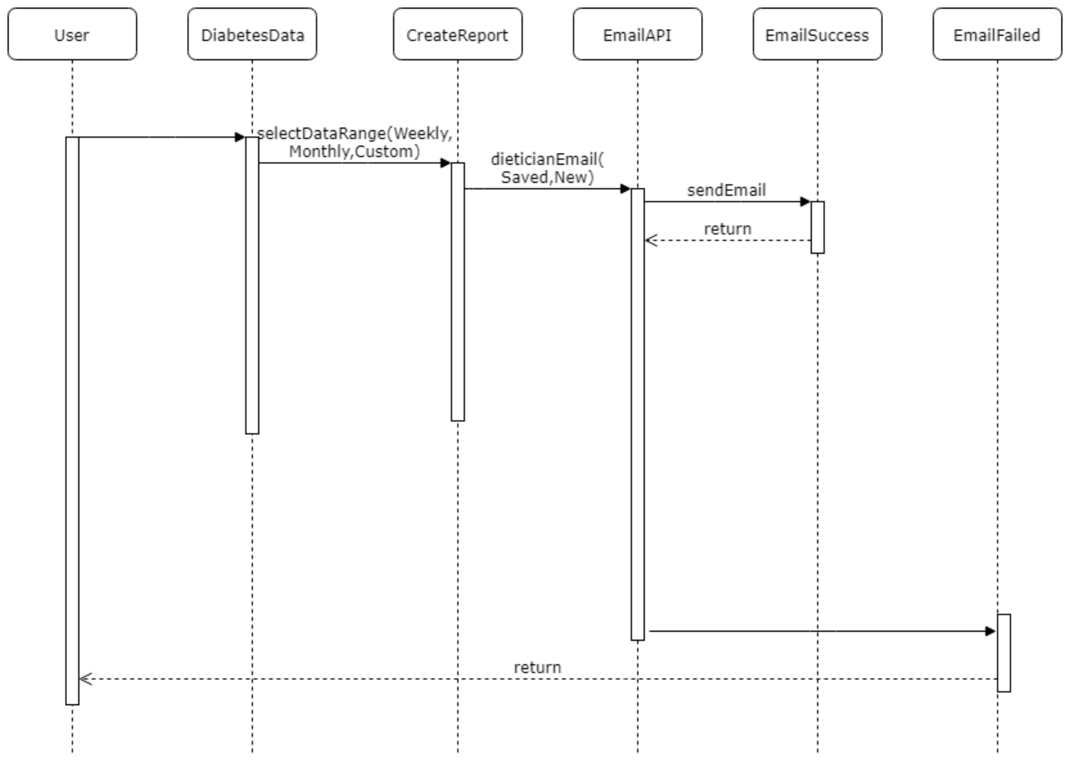

Figure 1.d

The purpose of Sequence Diagrams is to highlight interactions between users and the system and their system components. In Figure 1.d, we highlight the objects and methods that need to be concurrently running for the user to email their Blood Glucose Report to their dietician and/or physician.

The diagram starts with the User and continues to DiabetesData where the user is prompted to select the range of data that that they wish to include in their report. After they select ‘Weekly’, ‘Monthly’, or a ‘Custom’ range, the Create Report method is called. This method prepares the report in a PDF file format and calls the function send Email. If the email is valid and no network connection is lost, the email gets sent and EmailSuccess is called. However, if the network connection is lost during the time the email was being sent, the diagram reaches the EmailFailed state. This state returns the use back to the original window and prompts them to try again.

5. Data requirements

The main input method for EasyGlucose will be using the iPhone touchscreen, which provide software input methods such as built-in keyboard and drop-down selection forms. Other input methods will include built-in camera, access from system gallery, and get requests from the system clock. The application will collect data for purpose according to the following list of tables, with the list title being an app feature.

IO overview:

- EasyGlucose will use the following iPhone hardware for input: touchscreen display, microphone.
- Software input methods include built-in keyboard, swipe gestures, drop down selection forms, data access to system image gallery, and get requests from the system clock.
- EasyGlucose will use the following iPhone hardware for output: touchscreen display, on-board speakers, network card.
- Software output methods include PDF export, and sending of Emails
- The external system that EasyGlucose will interact is the Email system.

Detailed input and output methods in relation to features:

- The following list of tables are description of input and output methods based on features of EasyGlucose

Glucose level entries:

Type of data collected	**Input method and details**
Glucose level measurement	Built-in numeric keyboard
Time and date of entry	Drop down menu with time and date options, one of which is ‘now’. If user selects ‘now’, then the current time and date is obtained from the OS.

Diary log with pictures:

Type of data collected	**Input method and details**
Title field	Built-in alphabetic keyboard
Time and date of entry	Drop down menu with time and date options, one of which is 'now'. If user selects 'now', then the current time and date is obtained from the OS.
Diary entry	Built-in alphabetic keyboard
Picture	Picture from built-in camera or from system image gallery
Related tags	Drop down menu with given action types, with the bottom option being 'other'. If user chooses 'other', then a description field which take alphabetic keyboard input is added.
Add tag command	Operation initiated by tapping a "+" button.

Blood glucose level analysis graph settings

Type of data collected	**Input method and details**
Graph time frame resizing	Pinch and expand gestures from touchscreen.
Graph time-period adjusting command	Drag gesture from touchscreen.
Graph time frame finetune	Drop down menu with pre-entered time frames.
Toggle diary entry bubbles display	Button toggle by tapping the button on the touchscreen.
Diary entry access	Tap displayed diary entry thumbnail.

Profile initialization/modification

Type of data collected	**Input method and details**
Date of birth	Drop down menu with date options.
Type of diabetes	Drop down menu with diabetes type.
Weight	Built-in numeric keyboard with drop down menu to select unit of measurement.
Height	Built-in numeric keyboard with drop down menu to select unit of measurement.
Gender	Drop down menu with gender options.

Printable PDF export

Type of data collected	**Input method and details**
Graph time frame resizing	Pinch and expand gestures from touchscreen.
Graph time-period adjusting command	Drag gesture from touchscreen.
Graph time frame finetune	Drop down menu with pre-entered time frames.

Scrollable blood glucose level entry table and diary entry table

Type of data collected	**Input method and details**
Table positioning commands	Vertical swipe gestures from touchscreen.
Entry view selection	Tap gestures from touchscreen.
Change displayed table type	Button toggle by tapping the button on the touchscreen.

Database Format Definition:

Data inputted for glucose level entries, diary log with database and profile initialization/modification will be saved using SQLite database with format shown in figure below.

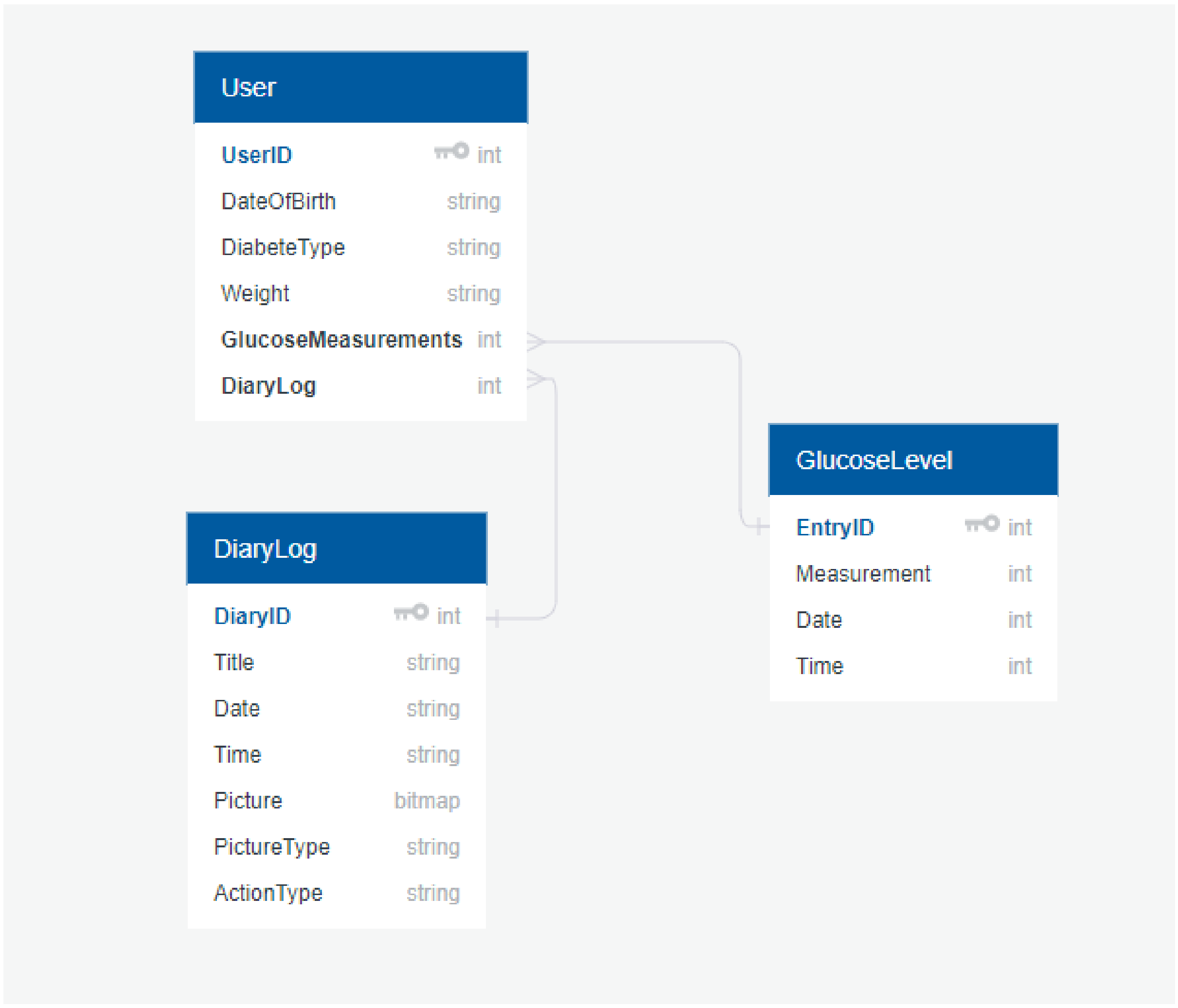

Figure 2. Diagram creation tool credit: QuickDBD

File Definition

EasyGlucose will interpret data from database formatted as the diagram above, and output as formatted viewable document to the app's filesystem. This file is defined as follows

- Format: .pdf file
- Size: less than 50 MB
- Saved location: app filesystem

6. Feature priority

EasyGlucose shall provide users with these features that are ranked based on their priority.

Version 1:

- 1- Manual input: Allow the user to manually input their glucose levels into the app.
- 2- Database: Create a local database to store all the raw data and the tables/graphs after analysis.
- 2.5- Analysis: Analyze the raw data and add it back to the database.
- 3- Startup screen: Create a startup screen that asks the user for their type of diabetes, age, gender, and weight.

Version 2:

- 1- Document food: Allow the user to log and to take pictures of their food.
- 2- Log notes: Allow the user to log notes or add tags along with their glucose input.
- 3- Email dietitian: Allow the user to send their dietitian a summarized report.
- 3- Improve aesthetics: Improve the user interface and pick a colour scheme.
- 4- Voice input: Allow the user to input their glucose levels through voice commands.

Version 3:

- 1- Create a settings page: Allow the user to update their weight and other personal information.
- 2- Unit conversion: The user can choose to use the imperial or metric system.
- 3- Create a profile page: The user will have the ability to customize their account more and add more information to it.
- 4- Change language: Give the user the option to change the application’s language to Chinese (Simplified).