Requirements Elicitation - Carb Capture

Functional Requirements

System Functionality

1. What will the system do?
    - The system will use API's and user entered information, such as carb-to-insulin ratios and an image of a food item, to give the user an estimation of how much insulin they should take for that meal.
2. When will the system do it?
    - The system will be able to do it whenever the user provides a photo of a food item, whenever they are eating.
3. Are there several modes of operation?
    - No, there is just one mode of operation, one where the user will enter information. They must enter a carb-to-insulin ratio, then, it is optional whether they want to submit a photo whenever they please of the food item they would like to get an estimation for.
4. What kinds of computations or data transformations must be performed?
    - The system will have to communicate with Dexcom's API and OpenAI's API in order to get blood glucose levels and the estimated carbohydrates, respectively. The data received from these API's may need to be transformed from text to integers for computation. It will then need to multiply the amount of carbohydrates in the food item by the Dexcom blood glucose level and the carb-to-insulin ratio that the user provided to give an estimate.
5. What are the appropriate reactions to possible stimuli?
    - There should be two main stimuli in the application. For the first one, it would be entering in a carb-to-insulin ratio. The application should accept this number or decimal and store it for later, letting the user know that they successfully updated their information. The second one would be taking a photo of food to send to get an estimate. The application should take this photo, use OpenAI's API to decipher the photo and get an estimate from the food, do the computation to convert it to an insulin amount, then send back the estimated insulin amount. The application should receive blood glucose levels from Dexcom, but should not need to manually enter these numbers if a user has a compatible sensor.

Data

1. For both input and output, what should be the format of the data?
    - For input, the data should be in the format of an integer or decimal for the carb-to-insulin ratio, a and a photo (jpg, png, etc.) of the food item.
2. Must any data be retained for any period of time?
    - Data may be retained for the user to keep a log of their meals and consult their doctor for updated carb-to-insulin ratios. This is optional.
3. Design Constraints
    - The design should be user friendly, able to be used by a variety of ages from teenagers to middle ages. It should be intelligent enough for doctors to derive enough information to provide educated advice.

Physical Environment

1. Where is the equipment to be located?
    - The app will run on user smartphones or mobile devices.
2. Is there one or several locations?
    - The app can be used anywhere the user carries their phone.
3. Are there constraints on size of the system (Handheld/Server/PC etc)?
    - Yes, the system must be optimized for mobile devices.
4. Are there any constraints on programming language, OS because of existing software components?
    - The app should be developed using cross-platform frameworks like Flutter or React Native to support both Android and iOS.

Interfaces

1. Is input coming from one or more other systems (“upstream”)?
    - Yes, input is coming mainly from Dexcom’s API and OpenAI’s API. These systems are assumed to be working at all times for the use of the application.
2. Is output going to one or more other systems (“downstream”)?
    - No, there are no systems outside our application that should be receiving the data from our application, as it deals with health information.
3. What is the protocol for the upstream and downstream systems?
    - The upstream data will most likely be received through a secure channel like HTTPS, using GET, POST, or other requests relevant to this protocol.

End-Users

1. Who will use the system?
   - The primary users are people with diabetes, particularly those requiring insulin management (Type 1 and some Type 2 diabetes patients).
2. Will there be several types of users?
   - Yes, there will be users from all demographics, health backgrounds, and intentions.
3. What is the skill level of each user?
   - They must be capable of reading text and have basic smartphone knowledge to open the application. However, the app should also be accessible to those with minimal technical expertise.

Quality Requirements

Performance

1. Are there constraints on execution speed, response time or throughput?
   - There are no constraints on execution speed. The application’s most resource intensive and time constraining feature will be the API pulling data from OpenAI. This process is relatively quick, and the user should not be in a life-or-death situation requiring immediate output from the application.
2. How much data will flow through the system.
   - For the first version of our application, the data being input into the system is an image, or multiple images depending on the user, and DexCom text data. Further versions may allow for a user log-in text inputs, data reading from a user’s connected DexCom device, and healthcare provider interactions (which may involve images, videos, or text). The return data will be through our connected API, pulling text from OpenAI.
3. How often will data be received or sent?
   - Data will be collected and received at the user’s discretion, meaning every time they upload an image. Data will be sent via an image upload and the corresponding output will be received shortly after. DexCom data will either be uploaded to the app via background pulls, upon image processing execution, or upon app startup. There will likely be a limiting factor, preventing the user from uploading a picture before the previous input returns data.

Usability and Human Factors

1. What kind of training will be required for each type of user?
    - The application will not require training to get an understanding of the functions. An optional tutorial or introduction can be provided by the healthcare provider.
2. How easy should it be for a user to understand and use the system?
    - It should be fairly easy for a user to understand and use the system. There will be very minimal interaction needed to fulfill the purpose of using the application.

Security

1. Must access to the system or information be controlled?
    - Access to the system does not need to be controlled. The application will take a photo from the user and output how many carbs are in the food. There is no private health information on file or other sensitive information. If an additional logging system is added to the app, we plan to store that data on the user’s phone.
2. Should each user’s data be isolated from the data of other users?
    - The user’s data will always remain independent of other users’ data.
3. Should user programs be isolated from other programs and from the OS?
    - No, the user program does not need to be isolated from other programs and the OS. There is no sensitive information flowing into or out of the application.
4. Address Cyber security issues
    - With our current plans, we expect no cybersecurity issues as a threat to the user. This will change if we decide to implement a log-in feature, meal or health record loggings, or healthcare provider information.

Reliability and Availability

1. Must the system detect and isolate faults?
    - No, due to the nature of our application using API's to retrieve information from OpenAI, a fault detection system is not necessary. We can implement fault detection code into the API call, like outputting "Please try another image, the provided picture is not recognized as food" if the user uploads an inevaluable image. The results provided from the application are approximate and have minimal chance to cause harm.

2. What is the prescribed Mean Time between Failures?
    - Our estimated mean time between failures is estimated to be no more than 24 hours. The system is using Dexcom and OpenAI's APIs in order to ensure maximum uptime and reliability.
3. Is there a maximum time allowed for restarting the system after a failure?
    - The maximum time allowed for restarting the system after a failure should be around 30 minutes. This is because when making a request to the Dexcom and OpenAI API's there is usually a message that prompts you to try again in a few minutes should there be any errors. To ensure the failure is resolved, there should be a maximum of 30 minutes allowed for restarting the system.
4. How often will the system be backed up?
    - Our initial version of the system will not require a back up because no data needs to be stored. If data needs to be stored in the future, for example, interactions between a user and a healthcare provider, a daily system backup should be enough to minimize data loss.
5. Must backup copies be stored at a different location
    - Backup copies can be stored in the cloud to prevent user error backing up data to their device and limit resource demand. Storing backups to the cloud provides added security, scalability, and accessibility.

Maintainability

1. When and in what ways might the system be changed in the future?
   - Some possibilities for expansion in the future include the implementation of a user log-in, Dexcom connectivity, healthcare provider access, and a meal or health logging system. There is not a current time to expect any changes. Changes will be made according to user demand and resource availability.
2. How easy should it be to add features to the system?
   - It should be relatively easy to add features to our system. We would implement different pages or functionality within the app, making a seamless transition between version updates. As well as our first product will cover the basic goals of our idea, it will be easy to expand the complexity.
3. How easy should it be to port (or migrate) the system from one platform to another?
   - It will be easy to port the system across platforms. The team is going to use software that requires little-to-none effort transitioning the system across other platforms.