InsuLink

Documentation

Produced by: Nicola Dean and Marco Fasanella

CP: 10674826,10617541

Table of Contents

This file was created for the benefit of

1

SECTION

Idea

When Type 1 Diabetes[1] is diagnosed, a patient starts a new life with different eyes. From now on, the conception of food is completely different from the normal one, and the patient has to assimilate the big change and learn how to handle the disease. One of the most difficult but at the same time important things that the patient must learn, is the ***carbohydrates count*** and subsequently the correct insulin dose for a bolus [2]. InsuLink has been designed with the main purpose of giving an hand to Type 1 Diabetes patient with the calculation of the correct ***insulin doses*** and storing Glycemia values.

1.1 Main Goal

InsuLink main goal is to give a first support to the patient but only if combined with the doctor supervision. It is important to underline that this application is only defined by an algorithm, and in this kind of diseases ***each patient needs ad hoc treatments***.

```
\usepackage
```

or

```
\usepackage{package}
```

2

SECTION

Functionalities

Insulink offers some useful tools to keep track of the daily routine of a patient.

2.1 Food Scan

It is possible to to scan a given Food BarCode and be redirectd to the FoodDetails page with all necessary data.

2.2 Glycemia

Keep track of your daily Glycemia with intuitive charts and easly with the glycemia insertion tool.

2.3 Insulin Calculator

An algorithm (inside Insulin Calculator class) will retrieve last Glycemia, total amount of carbohydrates, sport activity and all essential data to calculate the optimal insulin dose for the given meal. A more detailed explaintation can be fount in the Insulin Calculator Section.

2.4 Calendar

The user can see a well detailed sight of all previous data, just choosing a date from the InsuLink calendar, that will retrieve all the informations about that day from the database.

3

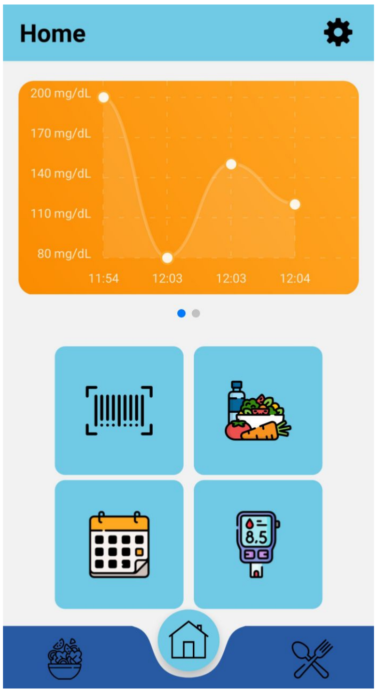
SECTION

Screens and Navigation

The following provides a screenshot of the pages with a brief description of their use.

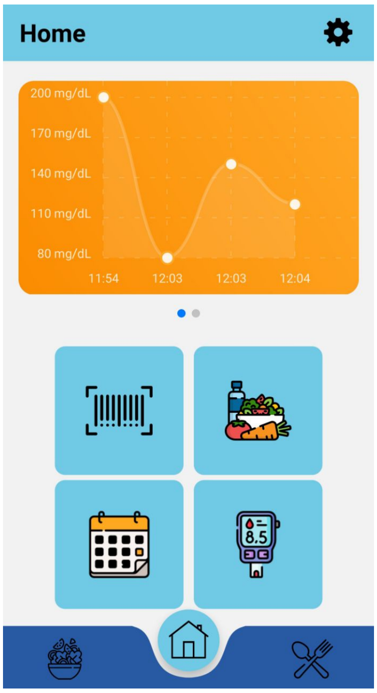
3.1 Home

Home menu offers shortcuts to the main functionalities and a quick sight of the today glycemia with its intuitive charts.



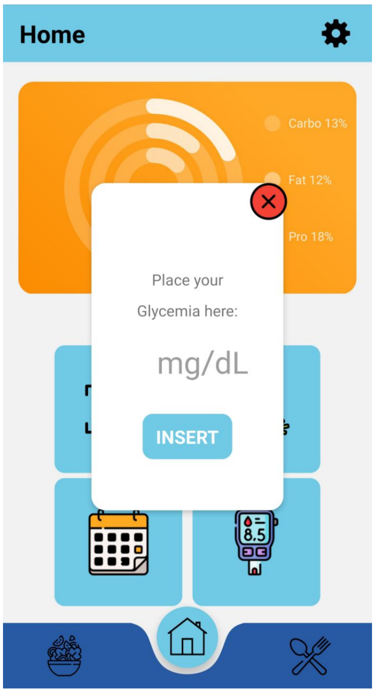
3.2 Search

Search food or recipe for nutritional details or to add it in meal diary. User can easly modify the unit measure and quantity of food.



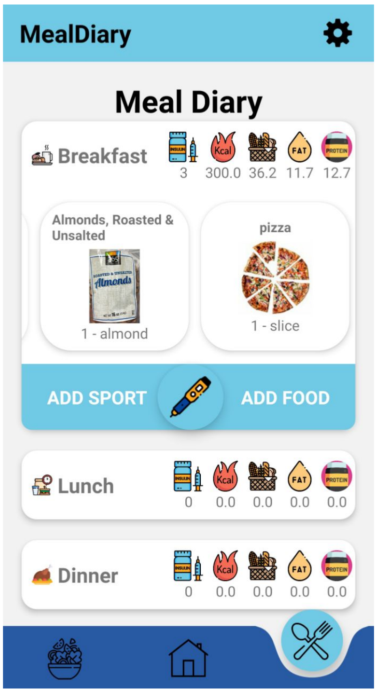
3.3 Glycemia PopUp

Add glycemia quickly just using the menu shortcut or during the insulin caluclation procedure. The value will automatically stored in Firebase.



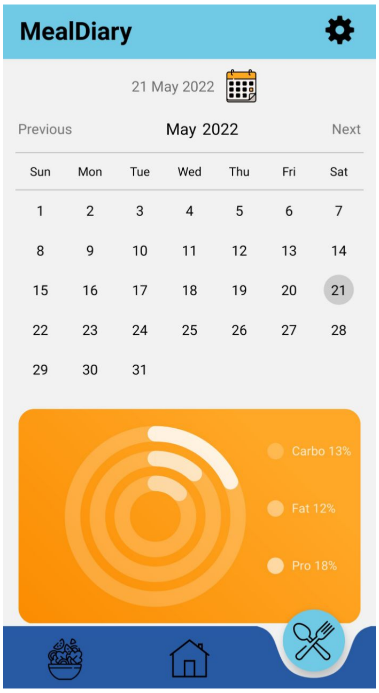
3.4 Meal Diary

Meal Diary can be used for both calulating daily total macro nutrients and insuline dose of each meal.



3.5 Calendar

In calendar it will be possible to retrieve historical data by clicking on a date.



Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

4

SECTION

Architecture

The technology used to make this app is react native [3].

4.1 Folder Structure

`\assets`

Contains all images and component with a proper mapping.

`\constants`

All constants concerning the design and states of the app.

`\customComponents`

All buttons, charts and pickers specifically designed for the pages.

`\pages`

Folder with all pages of the app, using the custom components

`\stateManager`

Redux States for data managing with actions and reducers: macroTracker for meals and userReducer for the patient.

`\utils`

Logic of API, authentication, Firebase and Insulin Calculator

5

SECTION

API doc

5.1 Nutritionix

Nutritionix [4] is the API used to have a food database-

5.2 Firebase

Firebase [6] is a Google serverless platorm for application development.

6

SECTION

Redux States

Inserting grap

7

SECTION

Insulin Calculator

Inserting graphics in your document requires the **graphicx package**, which you can find in the folder that accompanied this guide.

7.1 Easiest Way to Insert Images

The easiest, most straightforward way I have found so far to insert images into your LaTeX document is through TeX Studios include graphics feature. This features allows you to bypass having to constantly type and communicate with your computer where the folder containing your images is located. This really comes in handy once one has created several latex documents across several folders. If you decide to insert images through this method, as I presume most of those reading this will, **make sure to remove the graphics path command from the preamble of the templates in the folder containing this pdf.**

So to insert an image into your LaTeX file, go to the taskbar located at the top of your screen and click the menu labeled **LaTeX**. Inside the LaTeX dropdown, go down to **Input/Include Files** and then **includegraphics**. Once there, find where you put the image you wanted to insert into your document and then press ok.

7.2 Graphics Path

Most of the time, it will easier to insert images into a file using the method we mentioned in the previous subsection. However, the following allows for a cleaner, more organized, but a little more complicated at times possibly more advanced way to insert images into your LaTeX file.

```
\graphicspath{path}
```

This command prevents you from having to tell the computer where the image is stored each time you wanted to insert an image.

The easiest way to set up the graphicspath is to create a folder where you store all the images you want to use and label it **images**. You put this folder in the same area where you write your files. Then you input the following at the beginning of your document:

```
\graphicspath{{./images/}}
```

7.3 Inserting Images

Once you have set up the graphics path from above, you insert pictures by using the following command:

`\includegraphics`{*file-name*}

This command allows you to actually insert the image in the document. To alter the position and size of the image, see https://www.overleaf.com/learn/latex/Inserting_Images.

To change the scale of file, use the following

```
\includegraphics[scale= ]{image file name}
```

7.4 Image Additions

`\caption`{*text*}

Allows you to add a caption to an image.

```
\begin{wrapfigure}
 ⟨text⟩
\end{wrapfigure}
```

Allows you to wrap an image inside text. For more on this, see the wrapfigure.pdf inside the wrapfigure folder.

7.5 Positioning an Image

To alter an images position: see https://www.overleaf.com/learn/latex/Inserting_Images

8

SECTION

Testing

To perform automated and personalized testing it was used Jest [5]. It is a JavaScript Testing Framework that supports React Native. Tests were performed on:

`\__tests__`

Where to overcome some technological barriers, mock objects were used instead of not supported libraries.

8.1 Folders

`\api-test`

Local storage and API calls from Firebase and Nutritionix.

`\redux-test`

Redux and user actions such as: adding food to meal or removing it.

`\renders-test`

Checks correct Pages rendering. Makes snapshots of all pages and compares them with expected result.

`\utils`

Checks User input and Insulin Calculator

9

SECTION

Future Implementations

—-

10

SECTION

References

[1] Diabetes Definition
https://en.wikipedia.org/wiki/Type_1_diabetes

[2] Bolus Definition
https://en.wikipedia.org/wiki/Bolus_(medicine)

[3] React Native
https://reactnative.dev

[4] Nutritionix
https://www.nutritionix.com

[5] Jest
https://jestjs.io

[6] Firebase
https://firebase.google.com