Università degli Studi di Padova

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Development of a mobile platform prototype for data gathering and monitoring of type 1 diabetes individuals during clinical trials.

Student: Luca Cossu

Supervisor: Andrea Facchinetti

Co-supervisors: Giacomo Cappon
Giovanni Sparacino

Corso di Laurea Magistrale in Bioingegneria
16 Dicembre 2019 A.A. 2018/2019

UNIVERSITA' DEGLI STUDI DI PADOVA
DIPARTIMENTO INGEGNERIA DELL'INFORMAZIONE (DEI)
Via Gradenigo 6/b 35131 - Padova Italy
Tel. +39 049 827 7600
www.dei.unipd.it

Abstract

Diabetes is a chronic disease which affects more than 60 million people only in Europe and is set to become one of the mass diseases in the near future. This disorder has serious short and long-term implications and requires life-long attention to many aspects, blood glucose monitoring and meal planning and insulin injections. This is true especially for patient with Type 1 Diabetes, which have to manage their disease the whole life.

The therapy is based on multiple daily exogenous insulin administrations tuned according to blood-glucose (BG) measurements, made by the patient himself or with an autonomous sensor. In recent years many research projects have focused on easing the patient's quality of life while increasing the control on glucose levels. Many projects involve the development of new algorithms that help the patient taking therapy decisions. All this new techniques have to be tested in order to verify their actual usability and improvements in blood glucose control.

The Biomedical engineering research group in Padua has been developing some new algorithms and there was the need for a platform that would have allowed an easy testing in clinical trials.

In this thesis we developed a prototype of this such platform, in the form of a mobile application where the patient can track all his data and the algorithm can be deployed to be tested. We also developed a website for clinicians to track patients levels in real time during trials. This platform allows also to save all gathered data for later use in simulations to develop even more optimized algorithms. The whole platform has been developed to fulfill requirements in privacy and usability.

Contents

Chapter 1

Mobile health applications for diabetes management

During the last few years, the interest for the development and creation of medical applications to be used in mobile technologies has increased more and more due to the wide availability and diffusion of smartphones. So far, total of 80% to 90% of adults in the United States and 75% of adults worldwide currently own a smartphone [1]. Nowadays, a smartphone is the preferred device among the adult people to access an app, 53% of adults using them for such a purpose, while desktop computers are used one-third of the time or less for the same purpose [2].

mHealth, which stands for mobile health, is the term used to define mobile communication devices for health support activities [3]. mHealth has many different purposes, which span from remote monitoring to decision support, from easier and enhanced access to patient data to reminders of therapy need. From the literature point of view, mHealth topic has had a three-fold increase in publications between 2010 and 2016 remaining stable after that year, this is due to the increasing interest by research groups, doctors and patients on the topic [4].

The main target of mHealth and telemedicine is to keep healthy patients at home, reducing the need to move to healthcare facilities for control visits, and to give sick patients early and tempestive care. This allows to reduce healthcare costs and to improve the success rate of prevention campaigns and survivability of sick patients by increasing focused efforts by doctors. Some evidence shows that the use of mHealth increases new patient capture decreasing at the same time, health care costs as the number of patients that are actively treated is reduced [5]. Other findings have shown that pediatric telemedicine significantly reduced emergency ward and urgent care usage [6].

According to the latest e-health report “From innovation to implementation – e-health in the WHO European Region” [7] by World Health Organization Europe, health authorities of about half of the countries in the European region promote the development and adoption of mHealth in the health sector. The report further shows that when governments sponsor mHealth programs, they are also more likely to provide incentives and guidance on innovation and evaluation, as well as regulation for their use. Many telehealth projects in the European Region are now progressing from pilot projects to broad-scale implementation. Realizing the public demand for telehealth, larger regional telehealth initiatives are emerging. The main fields of actual telehealth working projects are teleradiology, telepathology and remote patient monitoring.

In Italy a recent article about telemedicine remarks how healthcare has a major impact on state expenses, up to 10% of GDE goes to healthcare [8]. This value is going to raise due to increasing of average age of the population and increasing incidence of chronic diseases. Investments in mHealth, in particular in telemedicine, can avoid an enormous impact on state finances, keeping costs stable and even increasing care quality and equality. Mobile medical application can result very useful in the management of chronic diseases, reducing the effort needed by patient to manage their condition. In Italy health app and wearable devices are widely spreading. Italy is the second most frequent worldwide user of new technology, particularly wearables, after the United States and before Germany and France [9]. A recent survey shows that, in Italy, health apps and wearable devices are sufficiently known and used by the adult population and are considered potential supports for greater involvement in health management. Mobile applications are valued especially for their usefulness to improve patients’ engagement and compliance with treatment, while wearables are mainly used to check physical activity and glycemia. The main obstacles for the diffusion of mobiles and mobile applications are personal or technical reasons, with a high concern on overmedicalization in the long period [10].

1.1 Type 1 diabetes mellitus

Diabetes Mellitus is a chronic metabolic disease caused either by deficiency in production of insulin by the pancreas’ cells or by the ineffectiveness of the insulin produced or both. Diabetes Mellitus causes an inefficient or absent control of glucose levels in blood, thus leading to hyperglycemia. Symptoms of marked hyperglycemia include polyuria (frequent urination), polydipsia (thirst), weight loss, sometimes with polyphagia (continuous sense of hunger), and blurred vision. Impairment of growth and susceptibility to certain infections may also accompany chronic hyperglycemia. Acute, life-threatening consequences of diabetes are hyperglycemia with ketoacidosis (uncontrolled production of ketone bodies causing blood PH fluctuations), most severe forms can include coma and even death. Diabetes Mellitus complications can be either acute or chronic. The chronic hyperglycemia of diabetes is associated with long-term damage, dysfunction, and failure of various organs, especially the eyes, kidneys, nerves, heart, and blood vessels [11]. Overall, complications are far less severe in people with well-controlled blood sugar levels [12].

The majority of cases of diabetes fall into two broad categories. In one category (type 1 diabetes), the cause is an absolute deficiency of insulin secretion. In the other, much more prevalent category (type 2 diabetes), the cause is a combination of resistance to insulin action and an inadequate compensatory insulin secretory response. In the latter category, a degree of hyperglycemia sufficient to cause pathologic and functional changes in various target tissues, but without clinical symptoms, may be present for a long period of time before diabetes is detected. There is also a third category which affects women during pregnancy and is called gestational diabetes, but has a much lower incidence than the two cited categories.

Type 1 Diabetes (T1D), formerly known as insulin-dependent, in which there is the destruction of pancreas' β-cells, whose role is insulin production. T1D is an autoimmune disorder and affects about 10% of patients: usually it appears during childhood or adolescence, but it can begin in adults too. All the patients need to take insulin injections for the rest of their life (insulin-dependency) and to follow a special diet. Despite active research, T1D has no cure, but it can be managed.

Type 2 Diabetes (T2D), formerly named insulin-independent, results from the body's inability to respond properly to the action of insulin produced by the pancreas. It develops most frequently in adults, when the body becomes resistant to insulin or if the pancreas doesn't produce enough insulin to keep blood glucose at normal levels, causing hyperglycemia. The risk of developing T2D increases with obesity, sedentary life and wrong diet. Patients are not required to do lifelong insulin treatment but can control blood glucose with healthy food and exercise, combined with oral drugs or addition of insulin as needed at least in the early phases of the disease [11]. T2D is more common and accounts for around 90% of all diabetes cases worldwide. It occurs more frequently in women with prior Gestational Diabetes and in individuals with hypertension or dyslipidemia (high fat levels in blood), and its frequency varies in different racial/ethnic subgroups [11]. There is also a genetic predisposition to the disease but it's form is complex and not well defined yet.

Gestational Diabetes (GD), affects women during pregnancy but usually resolves itself after childbirth. Since blood with high glucose levels circulates through the placenta to the baby, this form of diabetes must be controlled to protect the baby growth. Gestational diabetes increases the risk of congenital malformations, breathing problems and increased weight at birth.

This work focuses mostly on T1D because it is object of several studies for its high risk and mortality rate.

1.1.1 T1D therapy

Type 1 diabetes get diagnosed usually during childhood or early adolescences and from that point on the patient has to learn how to cope with it.

The standard T1D therapy is an "open-loop" control approach that tries to maintain the blood glucose concentration in the euglycemic range (70-180 mg/dL) during the day. Particular attention is given to avoid the hypoglycemic episodes given their dangerousness for the patient. The control is obtained with multiple injections of insulin, usually at meal time, in quantity based on the meal, physical activity and current blood glucose concentration. This is measured by the patient with self-monitoring blood glucose (SMBG) samples taken with a fingerstick device (shown in Figure 1.1) three to five times a day.

Insulin injection's quantities are estimated with some rules that have to be fine tuned for each patient. This rules are usually kept easy in formula to allow patients to perform the calculation on the fly, but there are some devices and mobile applications that perform the calculations and advice the patient on insulin doses. This applications are the basic format of decision support systems.

The fingerstick process is burdensome for the patients and finger skin gets harder and thicker the more the procedure is repeated. This is why in recent years there is a new technology that is being used more and more by the patients: Continuous Glucose

Figure 1.1: Fingerstick device

Monitoring (CGM) devices. This devices are small sensors that are applied on the skin of the patient and thanks to a micro-needle sensor placed in the subcutis are able to measure the blood glucose (BG) concentration every few minutes.

This higher frequency of BG values allows the patient to keep track of his levels easier and without the need of manually prick himself. It is also possible to have some informations about trends and change the insulin dose accordingly. Having more data allows also to detect dangerous events that would have otherwise been missed as can be seen in figure 1.2 where the hyper and hypo-glycemic events would not have been detected with SMBG only.

1.1.2 Problems of the standard therapy

The standard therapy has some disadvantages and problems which leave space of improvement. As said, patients have to live with the burden of the measurement need multiple times every day and may have a lack of motivation and education to understand the results of the measure. The patient has to complete at least 3 sequences of tasks everyday, involving for each meal two measures (before and after the meal) and at least one insulin injection (meal bolus and corrective if needed). The amount of tasks performed by a patient in therapy exceeds 500.000 during his whole life. CGM allows to reduce the number of this tasks and allows to have measures during nighttime, which would otherwise be unmeasured.

Insulin bolus calculation has high chance of error. The used formulas optimally tuned in a given time instant could be suboptimal later on because of the natural variability

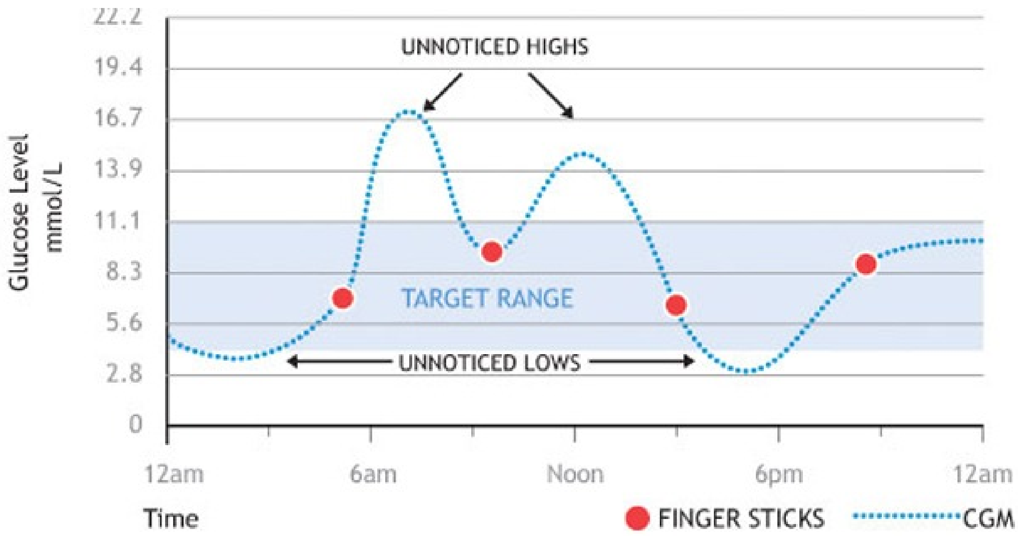


Figure 1.2: 24 hours of glycemic monitoring using SMBG (red dots) and CGM (blue line). CGM allows to detect hyper/hypo events which are not tracked by SMBG. [13]

of patient's physiology (e.g. variantions in insulin sensitivity), whose fluctiations could happen also within a single day. The standard formula for meal insulin bolus does not take into account all parameters influencing glucose use during different activities in addition, meal absorption times and carbohydrates estimate are key parameters whose wrong estimation can lead to large errors in insulin dose. Finally, physical activity, sleep and stress are other influencing factors which are not taken in consideration in standard control approach. New approaches and more advanced techniques to evaluate different data sources are needed to acquire such important information, to improve e.g. insulin dosing and perform a better control. Mobile health is one of the most promising solutions in which most of the new techniques can be summarized in decision support systems.

1.2 mHealth and diabetes

In Europe there are 60 million people living with diabetes and 32 million more are at risk [14], thus diabetes has been a major target for medical software companies. Diabetes is one of the future mass diseases, thus the research and development of mHealth techniques targeting affected patients aims to improve their lifestyle and management of the disease.

mHealth for diabetes is mainly focused on:

- Real time monitoring of patients' status
- Decision support for patients on activities and insulin doses
- Decision support fro clinicians on therapy corrections
- Better customization of patient-specific values in therapies according to long term trends and habits
- Easier data gathering in clinical trials of new therapies
- Psychological support and pressure ease of patient stress, due to awareness of continuous monitoring

There had been evidence that virtual coaching can substitute or even improve in-person visits for both children and adults with type 1 diabetes. Benefits of telemedicine in diabetes range from increased attendance at visits to lower overall costs and less time spent attending visits [15]. mHealth and telemedicine in particular can also overcome future shortage of diabetes clinicians and give them better chance to work and attend patients in critical conditions, leaving follow-up patients to virtual approach sessions.

Users report high levels of satisfaction with diabetes apps and increased app usage has been correlated with improved disease management, both in disease knowledge and in glucose levels control [16, 17].

1.2.1 State of art of mHealth in diabetes

During the last decade, with the increasing of smartphone popularity and diffusion, mobile app stores have seen an increase in availability of medical applications. Soon there had been the need of critically analyze the application's features in a medical accuracy perspective.

IDF Europe (International Diabetes Federation European Region) has categorized the existing applications in a recent article [18] according to their main target audience and strategies to prevent and manage diabetes:

Tracking/Logging tools to document relevant information including BG, meals, activity. This data is used to discover patterns and modify medications.

Nutrition These Apps help patients with food choice, carbohydrate and calorie counting and also help in bolus calculation. This relieves the patients of a a boring and repetitive task, while keeping lower the error in manual calculations.

Fitness This apps track physical activity, giving objectives and helping to control weight.

Connectivity with Platforms Many new apps connect with existing medical devices, from BG meters, CGM devices and pumps.

Coaching Many applications focus on education and promotion of a healthy lifestyle, giving general guidance and in some cases, customized coaching.

Social Network Blogs and chats where patients can share their experiences and receive tips, improving their management operations.

In 2013, Klonoff has summarized the features and problems of the main diabetes applications at the time [3]. Already in 2013 there were more than 500 applications resulting in diabetes filtering on the Apple Store [19] and in 2017 the number went up to 1200 [15]. Despite the high number of available applications, many of them present various issues and many have only a blood glucose tracking feature. There is also a general trend of diminished interest by users in diabetes-related app over time. Huckvale et al. performed in 2015 a systematic review of insulin dose calculators, with a focus on data validation and used model consistency [20]. An interesting outcome of this work is that many calculators have issues, all related to lack of coherent informations that could be solved with support during development by a clinician.

One of the most successful applications dealing with diabetes management is mySugr, which has many features and got international prices and certifications [21].

1.2.2 mySugr mobile application

Launched in 2012, the mySugr mobile app was designed to support patients in the diabetes self-management areas recommended by the American Association of Diabetes Educators 7 Healthy Behaviors curriculum [22]: healthy eating, being active, monitoring, taking medication, risk reduction, problem solving, and healthy coping. During the last years it has become a reference among mobile applications for diabetes care.

The application has some key features like, automatic uploading of data from SMBG and CGM devices, either directly via Bluetooth or, for instance, Apple Health, and can also be synced between devices via a cloud-based service. Additionally, insulin data can be entered manually into the app. Each new entry can be enriched by the user with pictures, other information about meals, medications, activities and current mood. Other information like physical activity levels can be imported automatically from other apps, adding context to the clinical data. The user has always access to all data with the ability to download it as CSV, Excel spreadsheets, or PDF report, giving patients, as well as clinicians, an accurate therapy overview that presents statistics and detailed logs.

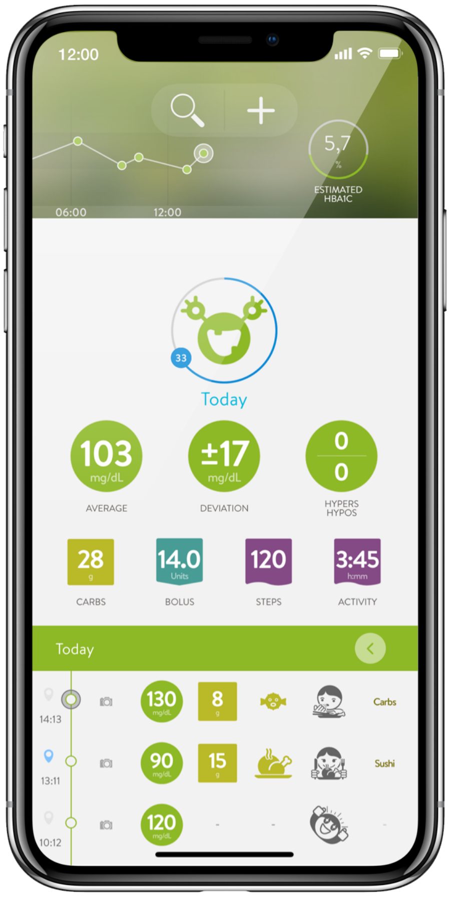


Figure 1.3: mySugr today page with statistics, graph and logbook

The application has also a proactive action: it uses pattern recognition algorithms to highlight areas of control where the patient should focus his attention. It also has a bolus calculator in it which has been CE-marked (Conformité Européene). It assists in bolus calculations and carbohydrate corrections in case of a low blood glucose prediction.

mySugr power and success resides in the mix of psychology and assistance given to users, not only focus on diabetes data. The whole app has been designed to be visually pleasing and to give feedbacks to the users of their daily management successes. [21] To keep patients involved in their disease management the application has a game mechanics which uses a reward system for each action taken by the patient each day on the app. The application uses a digital avatar of the disease that the user combats with each measure and entry uploaded.

There is also a form of self teaching: users can search and look up their past data checking which impact each event had on glucose levels. Using this information users can change their behavior and overcome future difficult situations better.

1.3 Fundamental usability aspects and open issues

To recap what said in this chapter, mobile health will have a big impact on health management and health system of countries in near future. As said many applications, not only for diabetes management, are available on mobile stores, but with rare exceptions those have big usability and clinical relevance issue. Developers of new platforms will need to take in consideration all previous assessments in order to give to the users a well built and functioning application that can help their disease management and improve their quality of life. There is also the need of cooperation between developers and clinicians, but also clinic and hospitals' managers will have to understand and promote mobile applications and virtual coaching to improve patients care and reduce maintenance costs. The development process will have to consider user needs, disease-specific and general, with a long period of application testing on various groups, in order to guarantee high specs in the final product.

Another key aspect is that are experiencing an always more connected environment, where all data gathers in data hubs. This will need to be considered also for health data, with a particular attention to privacy and security. More data from various sources will allow to build custom therapies for patients, improving even more their lifestyle with minimum impact. Remote monitoring will be a big part of this process: it will allow lower waiting times for in-person visits and a direct line between patients and clinicians.

Mobile applications will become a perfect platform for decision support systems, that will need to be tested in clinical trials. Clinical trials themselves will change the methodology of data gathering, using modified versions of existing applications or, in case of algorithms, ad hoc developed applications.

This is the context where my thesis was set, with the aim of creating a platform (mobile app, cloud database and web interface) for data gathering during clinical trials involving T1D individuals. In particular, once created, this platform will allow also the testing of algorithms developed by the research group in DEI in Padua, which will build in the end a decision support system for T1D patients.

Chapter 2

Project overview and thesis objectives

During the last year there had been a continuous research of new diabetes management algorithms (bolus calculators, glycemic prediction, decision support systems...) by the Bioengineering group at Department of Information Engineering (University of Padua). In the near future these algorithms will be tested on T1D patients during clinical trials, so there is the need for an easy to use platform that will enable their use and will allow gathering data for later analysis.

The main objective of this thesis is to develop a platform prototype to fulfill this need. The main requirement of the platform is thus to allow data gathering during clinical trials of new possible algorithms for diabetes therapy support, and to create a database that will allow to gather those data for later use, e.g. in further simulations and studies. This can be divided in four smaller objectives:

1. Develop a mobile application for data gathering by the patient during clinical trials, with ability to easily add modules for future testing (e.g algorithms for decision support);

2. Give the patient during platform usage an enriching therapy management experience, with statistics and trends that will allow the patient to learn from the past.

3. Give the clinician the ability to monitor in real time all his patients.

4. Create a database that will allow for later use of the gather data for later use, e.g. as dataset for future analysis and studies.

By gathering data from the patient we allow him to know his statistics and improvements in the glycemic level control. With the data we can potentially give him some decision support information via newly developed algorithms and thus making him feel safer and more monitored in his everyday life.

Uploading all data to a cloud database, it enables the clinician to monitor trends and, in case of need, call the patient for adjusting his therapy and analyze his progress in the trial.

With specific regard of clinical trials the main need is the ability of choosing which kind of data to gather for each patient, while keeping everything anonymous and easy to analyze later.

2.1 Platform overall structure

The whole platform to be developed is composed of three different main sub-systems: a mobile application, a cloud database and a website. As shown in Figure 2.1 the project has a common model interface, which will be the common basis of the entities involved between all project parts. The cloud database will conain all gathered data during trials, with all patient specific variables and will be communicating through a REST API with the mobile application and the web interface, the other two main parts of the project. The mobile application will allow the patient to upload new data to the cloud database and to review his previous data. It will also have a local database for the patient to save data in case of network unavailability. The web interface will allow the clinician to monitor all patients in the trial in trial time.

2.1.1 Thesis structure

Starting from the next chapter we will describe the development progress, with focus on requirements definition for each platform, usability choices and privacy preservation.

Before delving into project specific development, we will introduce, in Chapter 3, the framework and language that has been used and why it has been chosen. In the Appendix there will be some code samples of the language and implementations of small pieces of logic. In Chapter 4, we will show the chosen model for data and all relations between data

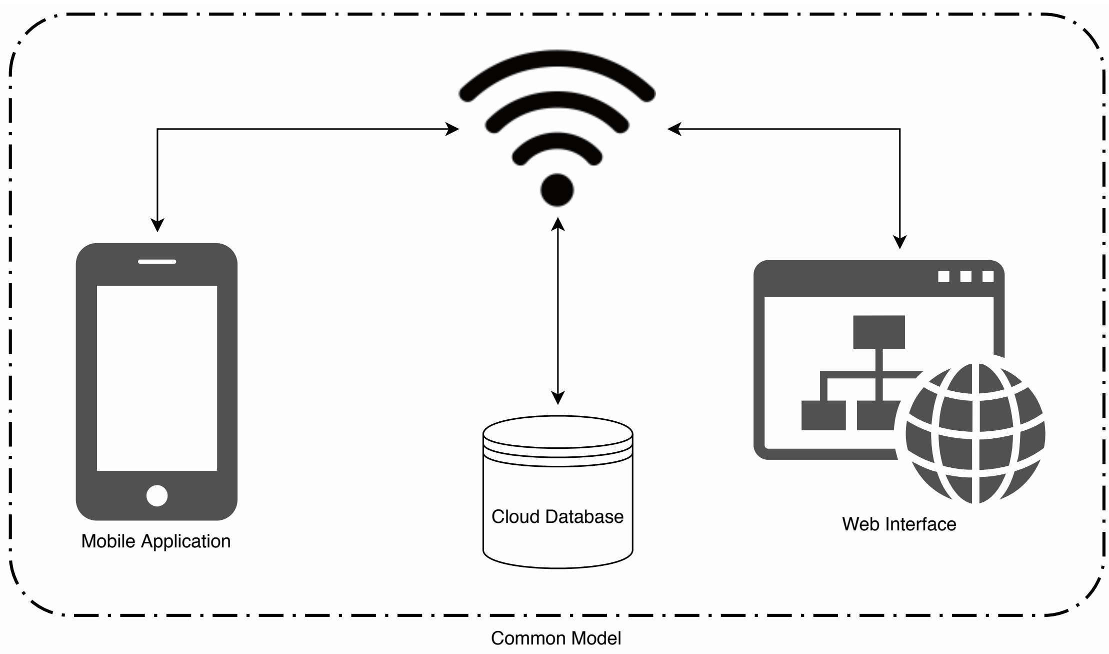


Figure 2.1: Platform structure overview. There are three elements all communicating through a restful API: the mobile application, the cloud database and the web interface

types. There will be a section on how the security and privacy problem was dealt and a section describing the API that allows communications between all platform elements. Chapter 5, will be focused on a description of the mobile application with emphasis on user interaction and interface. In Chapter 6, we will describe the clinician monitoring website, with an introduction on patient creation process. In Chapter 7, there is an analysis of a small beta testing session that allowed to try the application in real life scenario. Finally in Chapter 8, results will be summarized with a look on possible future development possibilities.

Chapter 3

Development framework and language

In the following chapter we will discuss the decision process behind the choice of the development framework and language to be used. First we will describe the chosen language, then the framework that uses that language in development.

3.1 Rationale of framework choice

Before starting the actual development, we needed to choose which language to use for mobile applications. One of the main problems of medical mobile applications is that not all patients may have the needed smartphone to support the app [10], thus we wanted to have the widest spectrum of mobile systems supported. There was the possibility of writing two different codes, one for Android and one for iOS, with the same functionality. It would have meant to develop the same features in both platforms and to later maintain two different source codes.

In early 2015 Facebook announced React Native, a framework for building native applications writing a single Javascript code. In late 2018 Google announced its own framework, called Flutter, which uses Dart as language.

The choice ended on Flutter because of Dart's support for web and desktop applications, allowing wide support and multi-platform development. This meant that writing a single code, one could have working applications on mobile, desktop and a working website. It was exactly what we needed for a fast and stable development. Moreover Flutter has many Widgets that mimic the native look and functionality of the native UI elements, and Flutter doesn't need any platform specific optimization, as React might need [23], because it is already compiled in native code. Another advantage is that usually to see a change in the code take effect the application must restart and rebuild from scratch, meaning long periods of inactivity because of code compilation. With "hot reload", all changes take effect in seconds, just saving the source code gives a visible effect on the application. Flutter also supports state preserving hot reload, which means no application state is lost.

3.2 Dart

Dart is an object-oriented, class defined, garbage-collected language. The Dart virtual machine features a just-in-time (JIT) execution engine[1]. While writing and debugging an app, Just In Time compilation allows for "hot reload", with which modifications to source files can be injected into a running application.[24]

Dart's execution engine can also be AOT (Ahead Of Time). Dart's code is compiled to a fast, predictable, native code, which allows almost all of Flutter to be written in Dart. This not only makes Flutter fast, but virtually everything (including all the widgets) can be customized at very low level. Dart makes it easier to create smooth animations and transitions that run at 60fps. Dart can do object allocation and garbage collection without UI locks and freezes.

Moreover, Dart is declarative and thus it allows to avoid the need for a separate declarative layout language like JSX or XML, or separate visual interface builders. All the layout is done in one language and in the same file, thus it is easy to have advanced tooling that makes layout designed and built very fast.

A layout example code can be found in Appendix A.

3.3 Flutter framework: a brief introducion

3.3.1 Everything is a Widget

The 4th December 2018, Google has announced the first stable release of Flutter framework. Flutter allows a fast development for both mobile platforms and, lately also for

[1] Just In Time compilation means that the compiler runs during the execution, not before starting the application

web and desktop. The fundamental brick of Flutter is the Widget: everything is a Widget. Flutter development consists in describing the layout of the application on the device, composing, assembling and creating Widgets from other Widgets. All Widgets are organized in a tree, which has as root the build() method of the main application Widget. Each leaf of the tree is child of his parent widget, which builds itself after his children. A documentation page says: "A widget is an immutable description of part of a user interface." [25]. However, there are many types of Widgets in Flutter, and during common development one cannot see or touch all of them. A string is a Widget (`Text`), but so is its style (`TextStyle`), which defines things like size, color, font family and weight. There are Widgets that represent things, others that represent characteristics (like `TextStyle`) and even others that perform actions. Complex widgets can be created by combining many simpler ones, and an app is actually just the largest Widget of them all (usually called `MyApp`) which contains all the other Widgets. In Figure 3.3.1 it is shown the widget tree that underlays the UI element shown. The UI is a simple row containing three buttons with an icon and a description each. This is translated in the Widget structure by having a repetition of three identical structures, building the button itself, all set in a row contained in a Container Widget that allows for some dimension properties to be set.

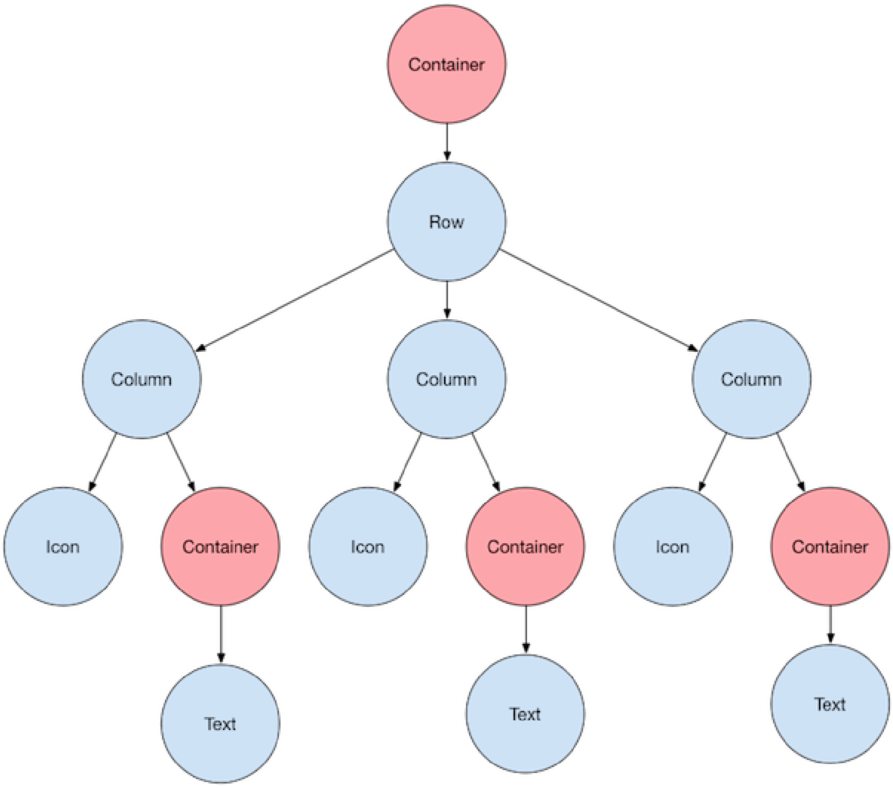
3.3.2 Hello World

The code in Listing 3.1 builds, for example, the standard Hello World application, which consists in application which displays a string in center of the screen. It is evident how the "everything is a widget" statement is applied here. The `HelloWorldApp` is the root Widget, and builds the application UI with a bar (i.e. `AppBar`) containing the title (which will be positioned on the screen according to the target platform style) and a body containing the centered text. Widget in code can be easily identified by a capital letter.

The power of Flutter resides also in the always increasing packages library. Packages are pieces of code, usually consisting in custom Widgets providing custom functionalities, layouts and styles. There are some official packages, such as Material, Cupertino and IO, which provide basic functionalities (i.e. predesigned UI elements in accordance to Material design, iOS design guidelines, platform file integration, respectively), but everyone can build his own package and distribute it through pub.dev website.

There are two main categories of Widget in Flutter: Stateless and Stateful. Widget themselves don't contain any state information, being immutable and having all their

(a)



(b)

Figure 3.1: UI element (a) and its Widget tree description (b). In (b) it is evident how each Widget can have a child and a parent: this ends up in a tree structure.

fields as final. To add a state information to a Widget, the Stateful Widget has to be used. It creates a State object, which stores information and gives the Widget itself the information needed for building and displaying. Stateless Widget are built the same every time with the configuration provided. Stateful Widget on the other hand, can be built several times during execution, each time containing different informations and possibly displaying differently.

```
import 'package:flutter/material.dart';

void main() => runApp(HelloWorldApp());

class HelloWorldApp extends StatelessWidget {
 @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Hello World App',
            home: Scaffold(
                appBar: AppBar(
                    title: Text('Hello World App'),
                ),
            body: Center(
                child: Text('Hello World'),
                ),
            ),
       );
    }
}
```

Listing 3.1: Hello World app code

3.3.3 Using Flutter for web app development

During the Google IO19 in May 2019, Google announced Flutter support for web. Flutter for the web is a code-compatible implementation of Flutter that is rendered using standards-based web technologies: HTML, CSS, and JavaScript. This means that existing Flutter code written in Dart can be compiled and executed in a web browser, and be deployed to any web server. All features of Flutter can work also in web environment. With Flutter version 1.9, the Flutter SDK actually contains web support in it, but is still in alpha phase, thus not all functionalities are working and will certainly change with future releases. As for version 1.9, Flutter for Web is not ready for production, but the developers can start developing websites written in Flutter and Dart.

In the next chapter we will analyze and show how the model of the data for the platform has been developed.

Chapter 4

Definition of the common model used in the project

In this chapter we will define the model used in the platform. The model has been defined once for all platform elements, in order to have consistent data. The model has to respect some requirements, such as high security level and modularity.

4.1 A fundamental requirements: security and privacy

Medical data are very sensitive, thus there is the need of some considerations in order to ensure that patient's data is not exposed and in case of stolen data, it wouldn't be possible to know which patient logged the data.

This means that all data must be anonymous and data communication has to be secured. This is obtained with, as will be explained in the next section, identification of all information by an id, which is assigned to the patient during its registration in the clinical trial.

To guarantee these aspects, as it will be better described later, the medic registers the patient in the trial via his web interface, inserting only patient's birth date and gender. The system returns an id that will allow the medic to later map the real patient to the data, and a password that will be used by the patient to log in on the mobile application. This way only the medic knows the correspondence between id and patient, saving it by himself.

4.2 Database structure: E-R Schema

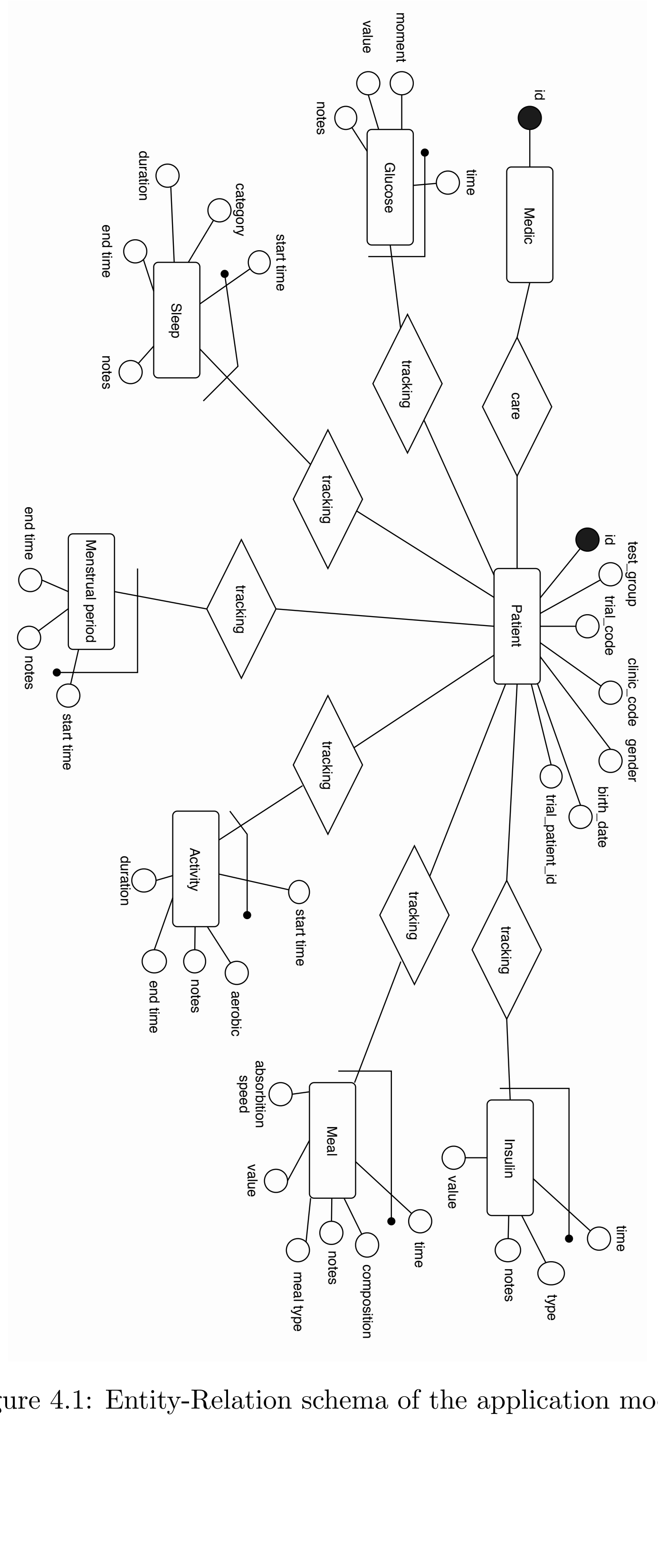


Figure 4.1: Entity-Relation schema of the application model

4.2.1 Entities

Because of the target of this project, there are some subjects that are elements of the problem: the Medic and the Patient.

Of course the data to be gathered is an element of the project and needs some formalization in the structure of each kind of data and we needed to decide on which kind of data to focus. As will be discussed later, the data types needed for sure are:

- Glucose measure
- Insulin dose
- Meal

To this some other came into play later in the development because of their importance in glucose concentration control:

- Activity
- Sleep
- Menstrual period

Trial was not considered an Entity at this time of development, because of the need to formally define its characteristics and properties. It will be inserted in the schema in future versions.

4.2.2 Relations

Entities relations are:

- Medic and Patient are in a n-1 relation. In fact the same medic can follow many patients at once. This means that medic field in patient entity is a foreign key.
- Patient and all data types are in a 1-n relation. This means that patient field for each data entity is its foreign key.

4.3 Database implementation using SQL

The complete tables structure considering all attributes is the following:

Medic id

The Medic entity has only the id field, which is its primary key.

Patient id, birth date, gender, medic, trial_code, test_group, clinic_code, trial_patient_id
The Patient entity has the id field, its primary key and internally used, patient's gender and birth date are useful informations for data analysis. Patients have also information about the trial.

As seen all data types have the same structure with the timing, the patient and a value. There are also some metadata specific of the single type of data and a note field which is present but not always allowed for editing.

Glucose time, patient, value, moment, note
Glucose's moment field describes the meal relative timing of the glucose measure:

- Pre-meal: the measure has been performed before eating
- Post-meal: the measure has been performed after eating
- Fasting: the measure has been performed hours distant from any eating

Insulin time, patient, value, type, velocity, note
Insulin's type field describes whether the injection is a correction bolus or made at meal time:

- Correction bolus: the insulin injection is a correction made to better control blood glucose levels
- Meal time: the insulin injection has been done corresponding to a meal

while the velocity is for the effective speed of used insulin:

- Fast: the insulin action speed is fast
- Slow: the insulin action speed is slow

Meal time, patient, value, meal type, absorption speed, composition, note
Meal's type field describes the kind of meal:

- Breakfast: a meal in the early morning
- Lunch: a meal at mid-day
- Dinner: a meal in the evening
- Snack: a meal in another time of day
- Night-time snack: a meal during the night

- Hypo-treatment: a meal needed to recover from or avoid a hypoglycemic event

absorption speed of the meal:

- Fast: usually simple carbohydrates
- Slow: usually complex carbohydrates

composition is useful for defining if the meal was carbohydrates only or had a mixed composition:

- Carbohydrates only
- Mixed, one or more can be selected among:
  - Carbohydrates
  - Fat
  - Proteins

Activity start time, end time, patient, aerobic, duration, note
Activity's aerobic field is a flag for whether the activity was aerobic or not.

Sleep start time, end time, patient, duration, category, note
Sleep's category field is based on a standard description:

- In bed
- Sleeping
- Awake

Menstrual period start time, end time, patient, note

The database has been setup with the code shown in Listing 4.1.

```
CREATE TABLE patients (
 id INTEGER(11) NOT NULL AUTO_INCREMENT,
 birth_date VARCHAR (191) NOT NULL,
 gender VARCHAR (191) NOT NULL,
 medic INTEGER(11) NOT NULL,
 PWD INTEGER(11) NOT NULL,
 created_at TIMESTAMP NULL DEFAULT NULL,
 updated_at TIMESTAMP NULL DEFAULT NULL,
 trial_code VARCHAR (191) DEFAULT NULL,
```

```
 test_group TINYINTEGER(1) DEFAULT NULL,
 clinic_code VARCHAR (191) DEFAULT NULL,
 trial_patient_id INTEGER(11) DEFAULT NULL,
 PRIMARY KEY (id)
);

CREATE TABLE medics (
  id INTEGER(11) NOT NULL AUTO_INCREMENT,
  name VARCHAR(191) NOT NULL,
  PWD VARCHAR(191) NOT NULL,
  PRIMARY KEY (id)
);

CREATE TABLE measures (
  time DATETIME NOT NULL,
  patient INTEGER(11) NOT NULL,
  glucose DOUBLE NOT NULL,
  moment VARCHAR(191) DEFAULT NULL,
  note VARCHAR(191) DEFAULT NULL,
  PRIMARY KEY (time),
  FOREIGN KEY (patient) REFERENCES patients (id) ON DELETE
     CASCADE
);

CREATE TABLE insulins (
  time DATETIME NOT NULL,
  patient INTEGER(11) NOT NULL,
  insulin DOUBLE NOT NULL,
  type VARCHAR(191) DEFAULT NULL,
  speed VARCHAR(191) DEFAULT NULL,
  note VARCHAR(191) DEFAULT NULL,
  PRIMARY KEY (time),
  KEY insulins_patient_foreign (patient),
  FOREIGN KEY (patient) REFERENCES patients (id) ON DELETE
     CASCADE
);
```

```
CREATE TABLE meals (
  time DATETIME NOT NULL,
  patient INTEGER(11) NOT NULL,
  carbohydrates DOUBLE NOT NULL,
  mealTime VARCHAR(191) DEFAULT NULL,
  assorbitionSpeed VARCHAR(191) DEFAULT NULL,
  note VARCHAR(191) DEFAULT NULL,
  composition VARCHAR(191) DEFAULT NULL,
  compositionElements VARCHAR(191) DEFAULT NULL,
  PRIMARY KEY (time),
  FOREIGN KEY (patient) REFERENCES patients (id) ON DELETE
     CASCADE
);

CREATE TABLE activities (
  startTime DATETIME NOT NULL,
  patient INTEGER(11) NOT NULL,
  endTime DATETIME NOT NULL,
  aerobic TINYINTEGER(1) NOT NULL,
  duration INTEGER(11) NOT NULL,
  note VARCHAR(191) DEFAULT NULL,
  PRIMARY KEY (startTime),
  FOREIGN KEY (patient) REFERENCES patients (id) ON DELETE
     CASCADE
);

CREATE TABLE sleeps (
  startTime DATETIME NOT NULL,
  patient INTEGER(11) NOT NULL,
  endTime DATETIME NOT NULL,
  duration INTEGER(11) NOT NULL,
  category VARCHAR(191) NOT NULL,
  note VARCHAR(191) DEFAULT NULL,
  PRIMARY KEY (startTime),
  FOREIGN KEY (patient) REFERENCES patients (id) ON DELETE
```

```
     CASCADE
);

CREATE TABLE menstrual_periods (
  startTime DATETIME NOT NULL,
  patient INTEGER(11) NOT NULL,
  endTime DATETIME NOT NULL,
  note VARCHAR(191) DEFAULT NULL,
  PRIMARY KEY (startTime),
  FOREIGN KEY (patient) REFERENCES patients (id) ON DELETE
     CASCADE
);
```

Listing 4.1: Database tables creation queries

In the following chapter we will see how this model has been used in the mobile application development.

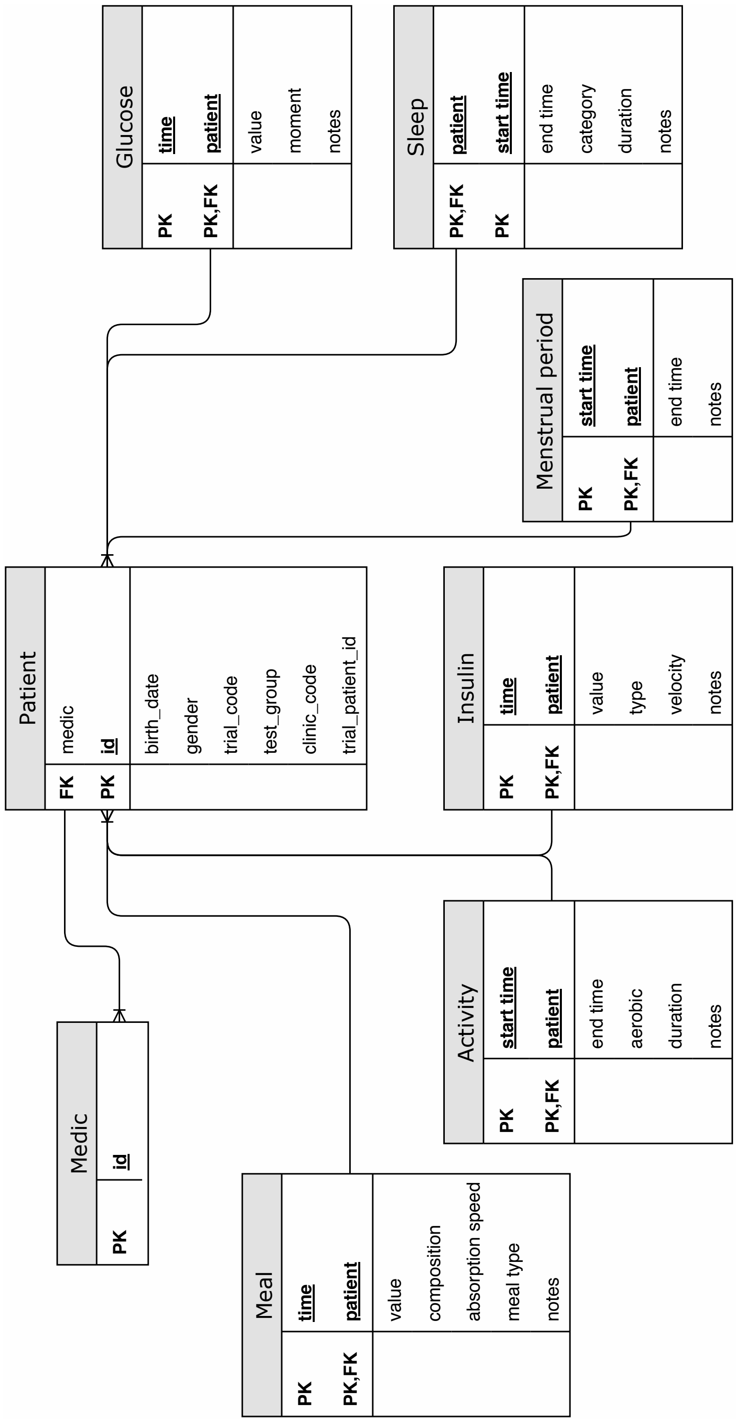


Figure 4.2: Created tables schema with all attributes, keys and relations shown

Chapter 5

SuGluLog: the mobile application

In this chapter we will describe the mobile application that has been developed, with a description of his screens and focusing on the user interaction possibilities.

5.1 Mobile application requirements

The mobile application is the core of the platform. It has to be built around patients and their experience.

The user experience needs to be as smooth as possible, and allow fast and precise data insertion while having a robust infrastructure behind. Before starting the development, some state of art research was needed and had been performed.

5.1.1 Requirements

Starting from Table 7 in [3], reported in Table 5.1 we have set up a list of needed features:

- The basic data types connected with diabetes are of course blood glucose levels, insulin doses and meals intakes, to those we have added activity, sleep and menstrual period tracking
- Every event needs to be pinned in time in order to be able to be used in statistics and trends analysis. Each event cannot be only a numerical value but is useful to

1.	Glucose tracking
2.	Carbohydrate tracking
3.	Insulin/medicine tracking
4.	Activity tracking
5.	Weight tracking
6.	Blood pressure tracking
7.	Meal-time tagging
8.	Preset notes
9.	Custom notes
10.	Food database
11.	Color coded for hypoglycemia/hyperglycemia
12.	Trend chart length
13.	Widescreen mode
14.	Logbook view
15.	Direct entry from logbook
16.	Averages
17.	Standard deviation
18.	Email composer
19.	Target range settings
20.	Background themes
21.	Email (comma-separated values)
22.	AutoSync to Website

Table 5.1: Data Management Features of the 12 Diabetes Mobile Applications with the Highest Ratings at the Apple iTunes Store as of October 9, 2009

attach to it some data type specific tags and custom notes

- A fast and easy data representation can help the patient understanding his trends in the short and long period, potentially leading to the decision to edit his lifestyle to improve the glucose control
- Basic statistical information, such as averages, variability, maximum and minimum, counters, can be useful if the patient knows his control target
- A logbook listing all data gathered allows the patient to edit wrong entries improving data quality
- Therapy specific data, such as target ranges and basal insulin, can improve the quality of data and tracking them is useful to the patient for reference
- A cloud data synchronization allows for remote monitoring by the clinician and easier access to dataset by researchers
- A printable and sharable Pdf report of the last 14 days can improve the communication between patient and clinician, reducing the need of in person control visits.

5.2 Design choices

5.2.1 Platform name

The name of the platform had to allow for a website URL that was not already used and the name had not to be linked to any existent organization or project in any field.

After some research on the internet, the choice ended up on **SuGluLog**, which stands for Sugar Glucose Logger.

5.2.2 Platform logo

The logo is the first element of the application that is visualized by the user, so it has to be clear and descriptive of the application. This platform's logo was chosen to be the stylized icon of a sugar container with the name of the platform as label (Figure 5.1).

Figure 5.1: Platform logo

5.2.3 General user interface design

The User Interface (UI) design during the development and prototype release phase had been focused on functionality and will be refined and restyled in future development phases.

The main interface color is light green with white as emphasis color. This allows for a high contrast and easy to read text. Most elements of the application respect the platform specific elements' style, while custom buttons and informative elements had been designed from scratch.

A big influence in the design of the mobile application design had come from mySugr and a famous menstrual cycle tracker app Clue.

The website got a restyle in the final development phase with a card oriented design that will be discussed in chapter 6.

The whole platform is only localized in Italian because the target of use is in Italy, it will be localized at least in english in future releases.

5.3 Screens

The application screens structure is organized as in Figure 5.2.

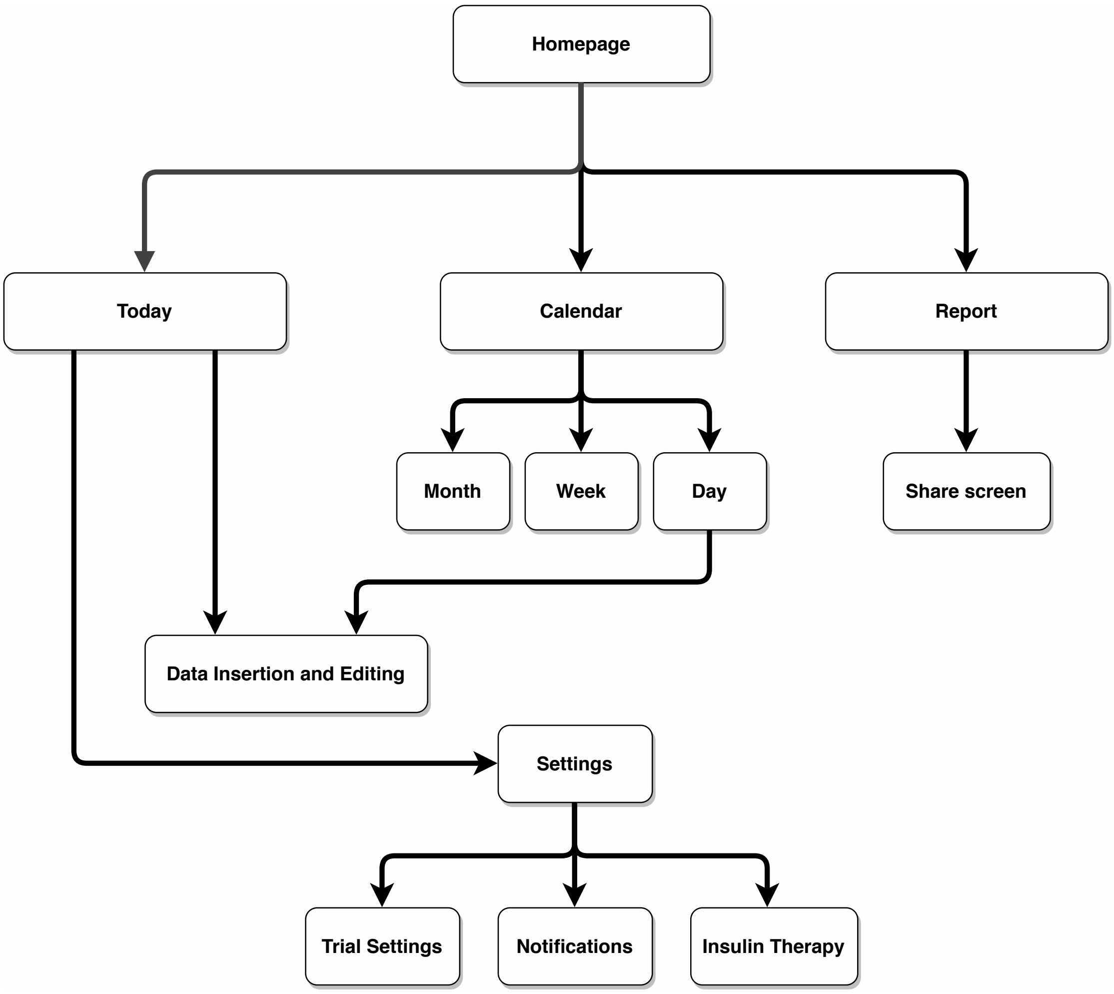


Figure 5.2: General screens organization. The user has the possibility to navigate between the tree main screens (Today, Calendar and Report) from the homepage and from those to the other screens

As can be seen the user lands on the Homepage which then is divided in three subscreens that will be described in the next sections.

Login The login page, not shown, is a simple form with a birthday and password fields. The patient logs in using as password the code generated by the web interface as will be described later. In case of successful login the application downloads all previously uploaded data of that patient: this is useful in case of smartphone change or log out.

5.3.1 Today

The today screen is the most used one, because from here the patient can add new data and see some statistics of the last 24 hours.

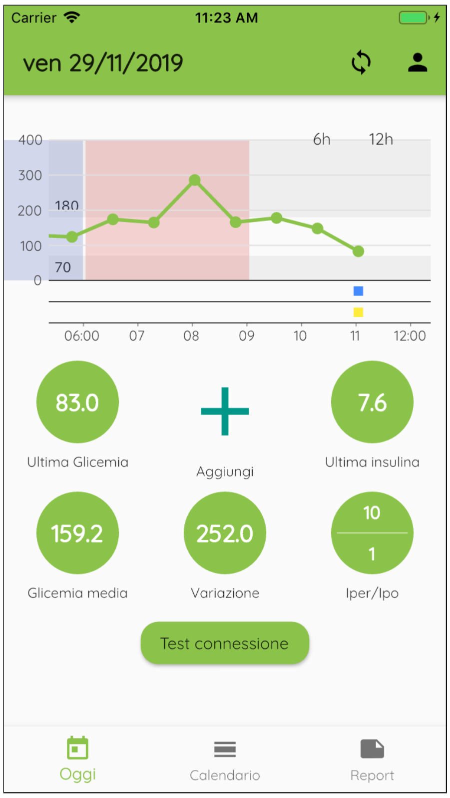


Figure 5.3: Today Screen, the most used screen by the user during the typical application usage

The layout is simple, yet informative: on top there is the AppBar, where the date is shown with two buttons, one for manual syncing of data, and one for patient informations and settings.

Below there is a graph showing the last 24 hours data of most the trackable data types.

The graph shows in green the glicemic curve with in grey the suboptimal ranges, insulin as blue dots below and meal as yellow dots. Sleep time is marked as blue area and activity period as red area.

Under the graph there is a grid with some statistics of the period and the plus button that leads to the Data insertion screen. The shown statistics are: the last glucose and insulin measures, an average on the period, the number of hyper-hypo events and the maximum glucose variation. This statistics has been chosen because they allow a fast understanding of the last day curve.

5.3.2 Data insertion and editing screen

The user reaches this screen after tapping on the plus button in the Today screen or pressing on an entry in the log view, which allows the patient to edit the selected entry.

The graphical structure of this screen has been inspired by the adding screen of Clue, a well-known menstrual cycle tracking application.

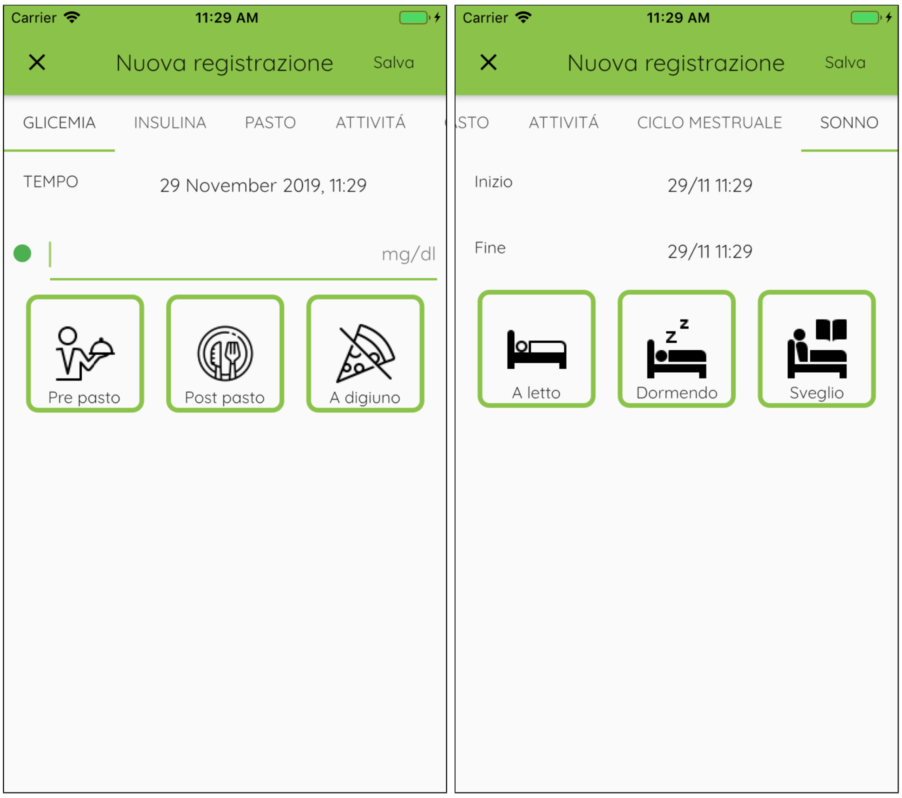
Having multiple data types, the page is divided in six tabs, that can be accessed from the TabBar below the AppBar. The general layout of each tab is similar. At the top there is the date and time of the event on which, by tapping on it, the patient can choose another time and day. Below there is a textfield for the measure's numeric value and below there are some icons to select the metadata needed for the measure. The metadata icon has been chosen to be informative at first sight, but there is a description below each button. All inserted data is validated during insertion and before saving.

As said, this screen can be reached also from the Day Calendar View by pressing on a log entry. This fills all fields of the pressed log entry. This allows the patient to edit a previously inserted data. The editing time is saved in order to recognize during analysis which data has been later modified.

After all data has been inserted the user can tap the button in the Appbar on top to save it locally and automatically the application tries to upload it to the cloud.

5.3.3 Calendar

The calendar tab is useful and informative to the patient because it shows statistics of past periods in an understandable way, which allows the patient to keep track of his



(a) Glucose measure insertion tab (b) Sleep data insertion tab

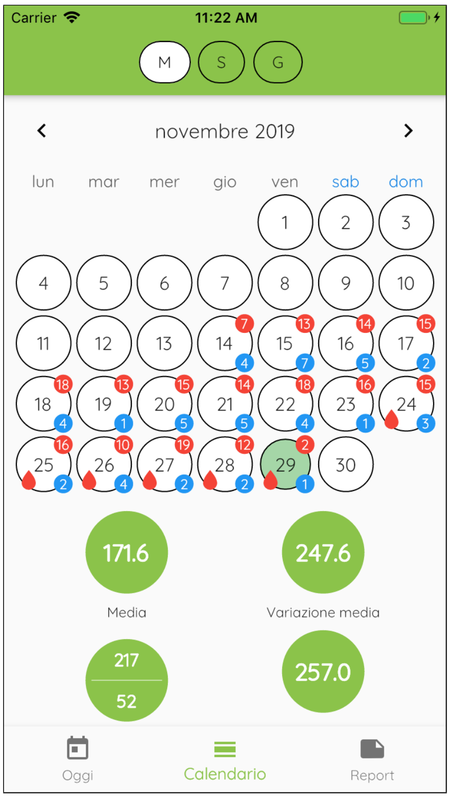
Figure 5.4: Data insertion screen tabs, one for each tracked data type and all with the same layout

improvements in the diabetes management and therapy progression. The patient can choose which view to see using the tab selector in the Appbar.

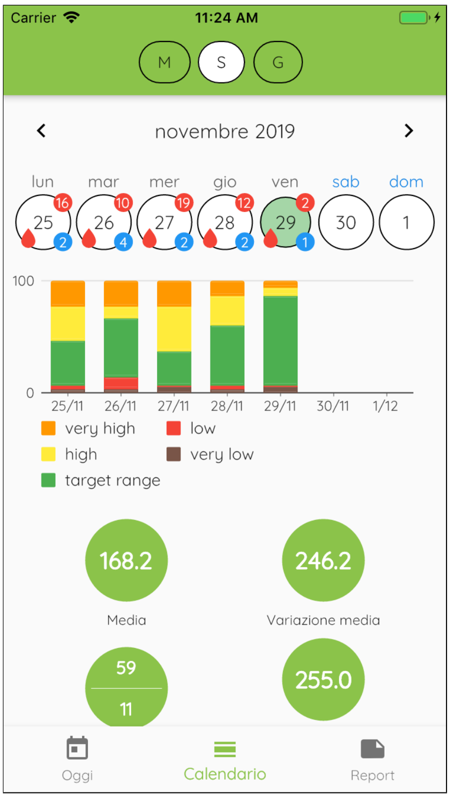
Month view The month view is the starting view of the calendar tab. In this subtab there is the month calendar and below it some statistics.

Week view The week view is the second view of the calendar tab. In this subtab there is the week calendar, a graph and below it some statistics.

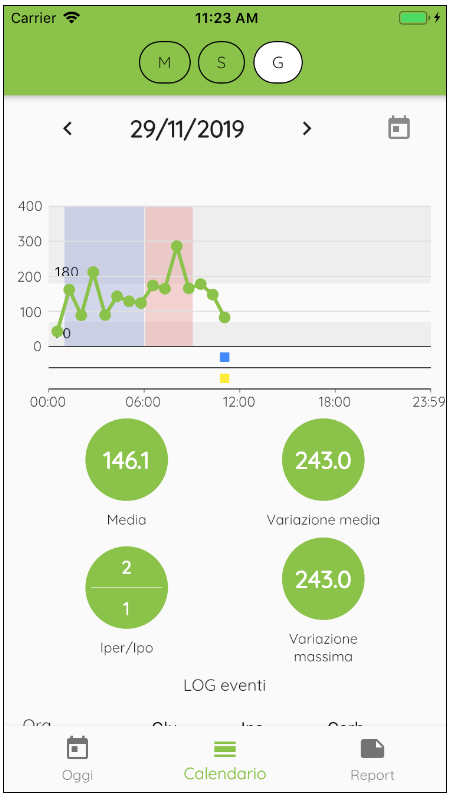
Day view The day view is the third view of the calendar tab. In this subtab there is the selected day graph and below it the day logbook.



(a) Month view



(b) Week view



(c) Day view

(d) Log view

Figure 5.5: Calendar views showing different layout and statistics of the selected period

5.3.4 Report

The pdf report page allows the patient to export and send a PDF report that summarizes the last 14 days of monitoring. The report is structured as proposed in [26]. An example is shown in Figure 5.7. The structure has been designed in order to give the medic a easy to read report with all information needed to understand patient control quality in the period.

On top there are some patient informations, below it there are on the left the metrics and their target values used in the report along with some statistics on the gathered data. On the right there is a bar chart showing the percentage of time spent by the patient in each glucose level range. Last there is a time graph, showing a 24 hours period where it is plotted the median of the glicemic value in that same hour of day throughout the period with quartiles. This is very important because it allows to recognize patterns during the last 14 days and thus analyze and solve them.

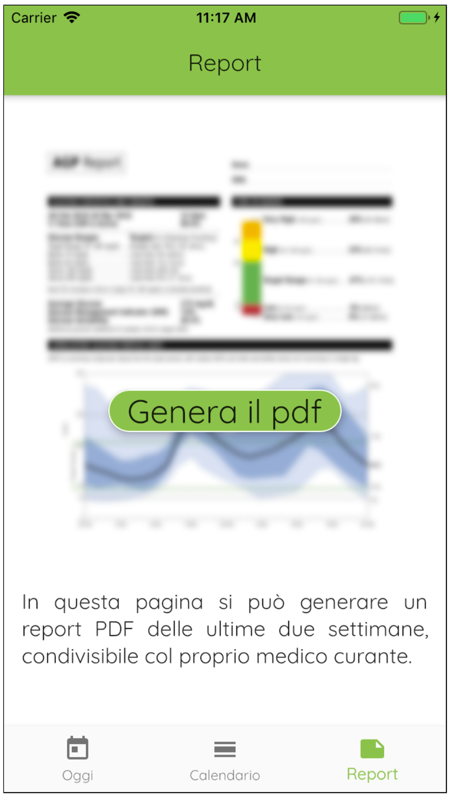
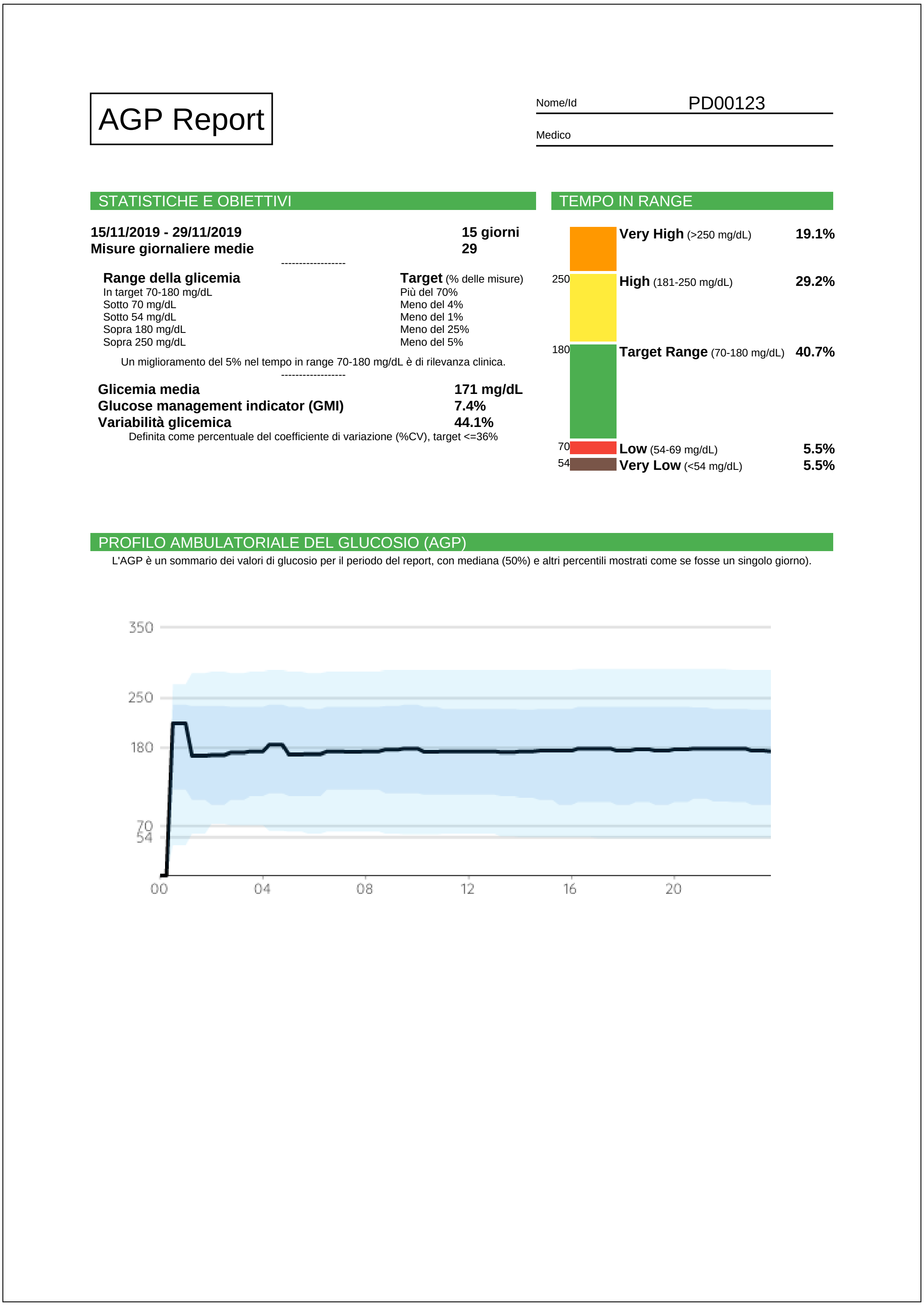


Figure 5.6: PDF report generation screen. Here the patient can gather all needed data to produce a report to share with the clinician


AGP Report

Nome/Id PD00123

Medico

STATISTICHE E OBIETTIVI

15/11/2019 - 29/11/2019 **15 giorni**
Misure giornaliere medie **29**

Range della glicemia	**Target** (% delle misure)
In target 70-180 mg/dL	Più del 70%
Sotto 70 mg/dL	Meno del 4%
Sotto 54 mg/dL	Meno del 1%
Sopra 180 mg/dL	Meno del 25%
Sopra 250 mg/dL	Meno del 5%

Un miglioramento del 5% nel tempo in range 70-180 mg/dL è di rilevanza clinica.

Glicemia media	**171 mg/dL**
Glucose management indicator (GMI)	**7.4%**
Variabilità glicemica	**44.1%**

Definita come percentuale del coefficiente di variazione (%CV), target <=36%

TEMPO IN RANGE

	Very High (>250 mg/dL)	**19.1%**
250	**High** (181-250 mg/dL)	**29.2%**
180	**Target Range** (70-180 mg/dL)	**40.7%**
70	**Low** (54-69 mg/dL)	**5.5%**
54	**Very Low** (<54 mg/dL)	**5.5%**

PROFILO AMBULATORIALE DEL GLUCOSIO (AGP)

L'AGP è un sommario dei valori di glucosio per il periodo del report, con mediana (50%) e altri percentili mostrati come se fosse un singolo giorno).

Figure 5.7: PDF report example produced with sample data

5.3.5 Settings

The settings screen can be reached from the Today page using the button on the top right. This screen allows the patient to change some settings like which kind of data to gather, specific to each trial, the hyperglycemia and hypoglycemia thresholds and the kind of insulin therapy. This screen is also where the patient can access the notification menu and can log out of the application.

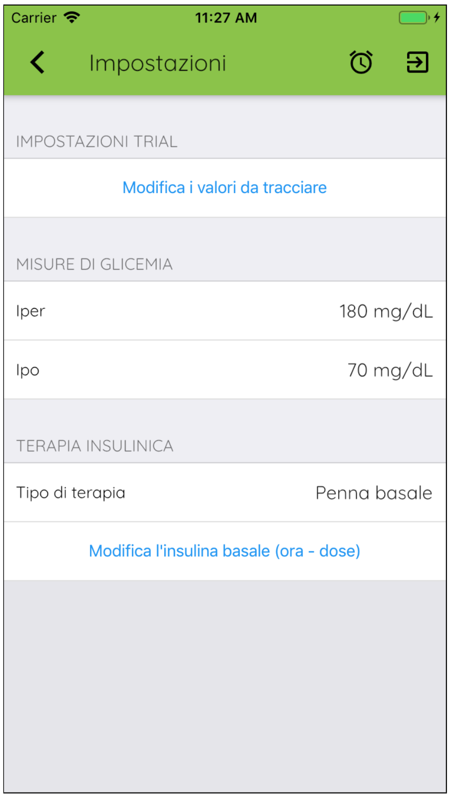


Figure 5.8: Settings screen. From this screen the patient can set his glycemic control targets and the insulin therapy type

Trial Settings Each clinical trial can have different kind of data required to be gathered. This screen allows the patient to choose which one he wants to track. This will possibly chosen by the clinician responsible for the trial.

In future versions the clinician will choose what to track for the patient and will not allow personal choices to track less data kinds than the one needed for the trial.

Notifications During first tests, it has been noticed how it is critical that the patient remembers to log data during the day. To increase the appliance to the trial and the logging procedure, it has been implemented a local notification system, that can be set up by the patient himself. This notifies the patient in specific times during the day to log important data.

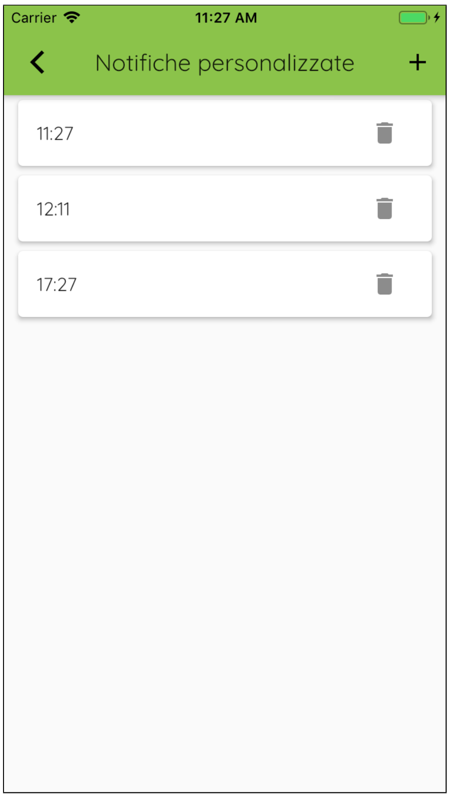


Figure 5.9: Notifications edit screen. From this screen the user can set up notifications to remember himself to add new data on the platform

In future versions there will be the possibility to improve the system, setting up a learning algorithm which will notify the patient based on habits and, when implemented, glucose variations read by the CGM sensor.

Insulin therapy There are few different insulin therapy types. The main purpose of this menu is to allow the patient to insert and track his basal insulin doses or pump infusions. If the patient has selected pump or basal pen, a new button comes up on the screen, which leads to a new page where the patient can add the pair time-quantity for basal insulin. By now this is only used for tracking purpose, in future versions it will be integrated in algorithms and in gathered data.

Figure 5.10: Insulin therapy screen where the patient can insert and edit his basal insulin therapy

5.4 Automatic data synchronization

The monitoring activity by the clinician is possible only if the data inserted by the patient lands on the server database. This is done automatically each time a new data is added by the patient, but the patient has the ability to manually synchronize all missing data. At current Flutter development state, the background activity of the application is not yet stable, and thus it was not implemented and automatic synchronization in background. It will be implemented easily when the background application task management will be optimized.

In the next chapter we will show how the data added by the patient with the application can be seen by the clinician in real time.

Chapter 6

Website

In the following chapter we will describe the online part of the platform, defining the requirements, the infrastructure that allows all communications and the actual web platform.

6.1 Web requirements

On the web platform the main objective is to allow the clinician to have under control all his patients at a glance. The website will allow the clinician to add a new patient in the system and, in future, set his tracking objectives for the specific clinical trial.

On load, the website shows a login form, where the clinician can log in with his identifier code and password. After log in the clinician is redirected to the main interface.

In order to communicate with the server to query all data needed an API has been developed.

6.2 Restful API

The restful API has been written using Lumen, a micro-framework of Laravel.

The main role of the API is to allow connection between the underlaying cloud database and the external environment, as shown in Figure 6.1. The API will be used by both the application and the website to update and request data on the cloud database.

Lumen allows for some Middleware, a class that intercepts requests and performs ac-

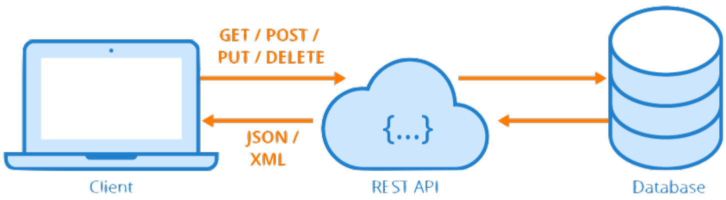


Figure 6.1: Restful API typical structure and functionality. It allows communications between devices and cloud data with HTTP requests made by each client

tions before the actual request is handled. In this project a Middleware was used for authentication of requests and was provided by an external source [27].

The language used is PHP and the code needs to represent the data model, describe the database tables, with migrations, and all queries needed. In Lumen this is done with the following modules:

Model In order to make the API know the data structure, a model of each table has to be defined. This allow Controllers to know how the data is presented and how to work with it.

Migrations Migrations are where there is the definition of tables, with their columns, keys and relations. This migrations are executed on the database and stored in a separate table to know all structural modifications of the database.

Controllers Controllers are the classes responsible for querying the database. In a controller there are all commands needed for a specific data request, insertion, update and deletion. Here it is where all the logic is written. Lumen uses Eloquent for faster and better query writing. Eloquent is an Object-Relation Mapping language which uses a model for each table to build queries in a more readable and easier to write way.

Routes Routes are the definition or the URL needed for a certain query. In Lumen this are defined in a file appending strings to the main URL. This can be done in groups to have a better control over the routes and here can be defined the HTTP method to be used for each route. In this project most re quests are made with POST method to hide informations in the URL and allow complex bodies to the requests.

In appendix A.2 there are some code examples of the previously described elements.

6.3 Interface

The prototype interface is built around the clinician experience.

As can be seen in Figure 6.2 it can be divided in three main sections.

1. The title bar with on the right the clinician profile button. It allows the clinician to log out, and in future updates to check and edit his profile and trials he attends to.

2. A navigation menu where the clinician can select the patient to monitor and add new one. Patients are identified with their unique id. In future updates it will be possible to filter patients by trial.

3. The selected patient informative section which contains a graph and some statistics of the day selected with the arrows above. Clicking on the date a pop up is shown containing a calendar for faster day selection.

    - The chart has the same characteristics of the one in the application, in fact, thanks to Flutter framework, the code is the same with only few aesthetics changes.

    - A statistics collection showing blood glucose mean and maximum variation, the number of hyper-glycemic and hypo-glycemic events and the total insulin and carbohydrates of the day.

6.3.1 Responsive layout

A website has to be accessible from different platforms, ranging from a desktop computer to a smartphone. Each one of this devices has different screen dimensions, thus the layout has to change. Three different layouts have been identified with the screen width (in pixels) as distinction criterion, which will be discussed in the next paragraphs.

Desktop screen (width $\geq$ 1200px), Figure 6.2 The layout for the widest screen is horizontally organized. The navigation bar is always available on the left of the screen, while in the center there is the information panel with the chart and statistics cards side by side.

Tablet screen (width between 800 px and 1200 px), Figure 6.3 Medium screen's layout focuses the attention more on the information panel, moving the navigation menu in a drawer, that can be open with a button at the top-left of the screen. Statistics card are moved below the chart to increase readability.

Phone screen (width $<$ 800px), Figure 6.4 Small screen layout is vertical oriented and is intended for smartphone visualization. Because of the small screen dimension, the chart has been removed and only statistics are shown. This has been chosen to allow clinician to consult the platform from their smartphone as needed, but the main usage of the platform is intended to be on a desktop or tablet environment.

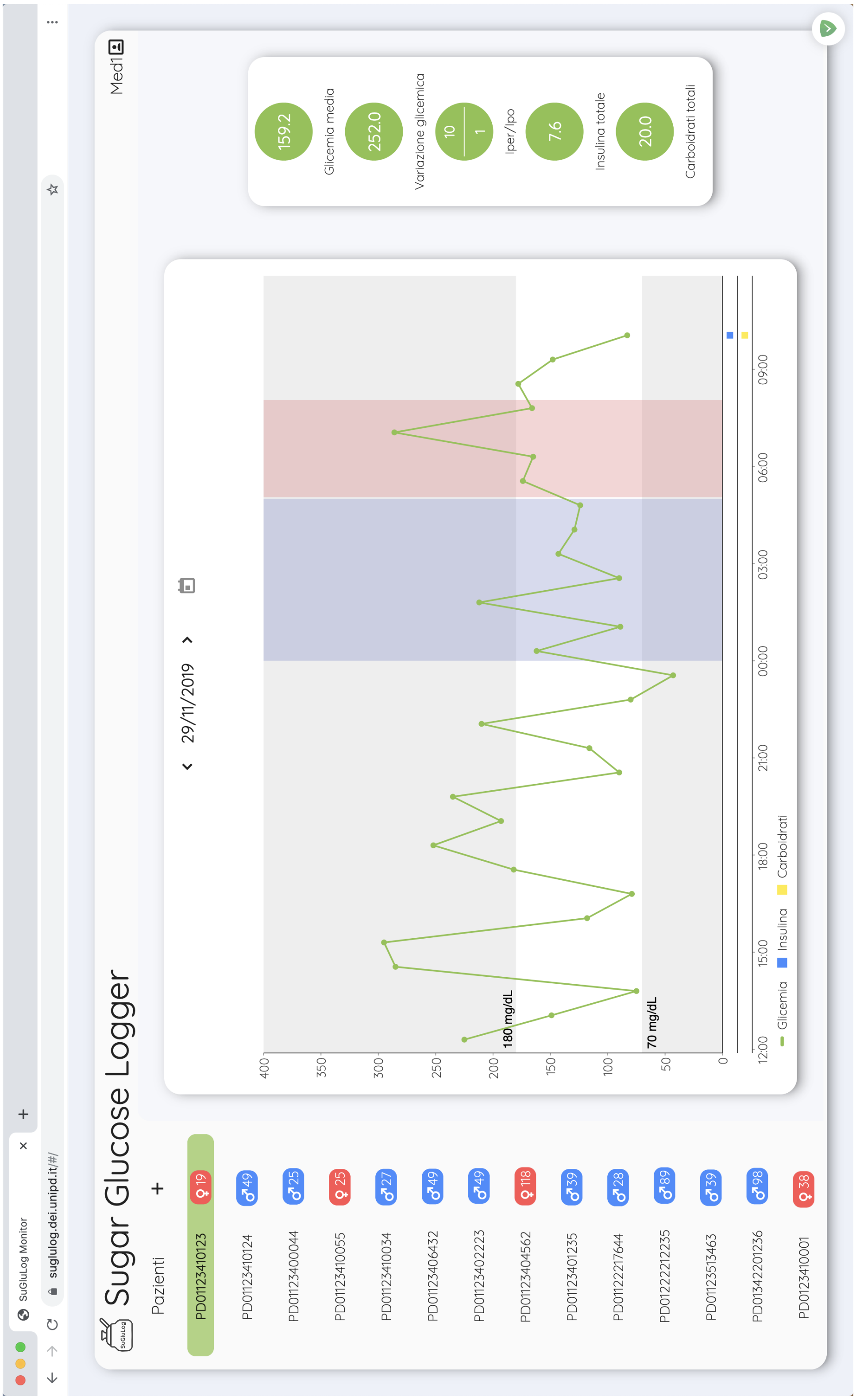


Figure 6.2: Large screen site layout with all UI element always shown: the list of patients is always available for the clinician on the left of the screen

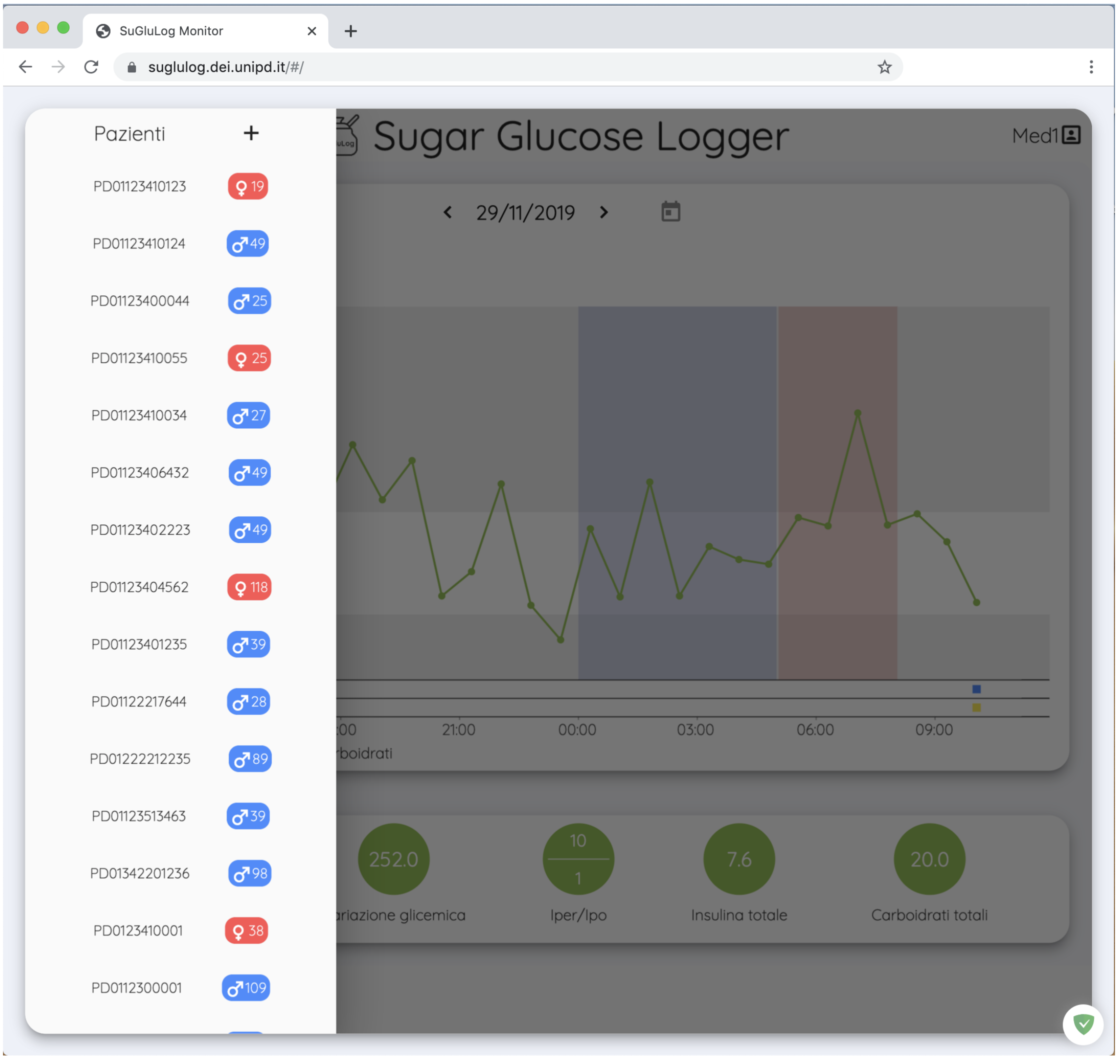


Figure 6.3: Medium screen site layout, where the patient list has been moved to a drawer in order to accomodate for smaller screen size

Figure 6.4: Small screen layout, where the only shown element is the statistics box. This has been decided to allow the clinician to see patient's data from his phone too, but because of the screen size it was not possible to show all information

6.3.2 New patient dialog

The clinician has the ability to add new patients in the platform for a specific clinical trial. Clicking on the plus button on the navigation menu a dialog comes up.

Figure 6.5: New patient dialog asking all needed data of the specific clinical trial and patient informations

The dialog has a form asking some informations to the clinician: the birthday, gender of the patient. Alongside this information the clinician has to insert the identifier code of the clinical center where he is in care and the trial code the patient will attend.

On submit all inserted data is validated and then a new patient is inserted in the database and a new dialog is shown to the clinician. It contains a generated password code and the trial-id of the patient built as CCCTTTFPPP where CCC is the identifier of the medical clinic, TTT the identifier of the trial, F is a flag for control or test group and PPP is the identifier of the patient.

6.4 Login and patient creation

On the website the clinician has his own personal profile and can monitor only the patients he has in care. To log in the clinician uses his unique identification number and a numerical password. Furthermore all requests made by the website are authenticated internally and made with the POST method. After patient creation, the clinician receives the id of the patient, which he has to save somewhere to remember the binding between code and physical patient, and the password which will be used by the patient.

The patient logs on the application with his birth date and the password generated by the system. This process sets the id of the patient internally to the application, that will use the id to attach each new data to the patient setting the id as foreign key of the data type table as described before.

In the next chapter there will be the space for the analysis of a brief beta testing session, made in order to proof that the platform is usable in all its parts.

Chapter 7

Beta testing

The last phase of the project has been a small beta testing session. It involved a diabetic subject who has been using the application for seven days. This short testing session allowed to prove the usability of the platform and solve bugs that would come out during long term use of the application. During the session few bugs have been encountered and reported to be solved. this included error in the login procedure and synchronization after first login. Report generation also had some issues that have been easily solved.

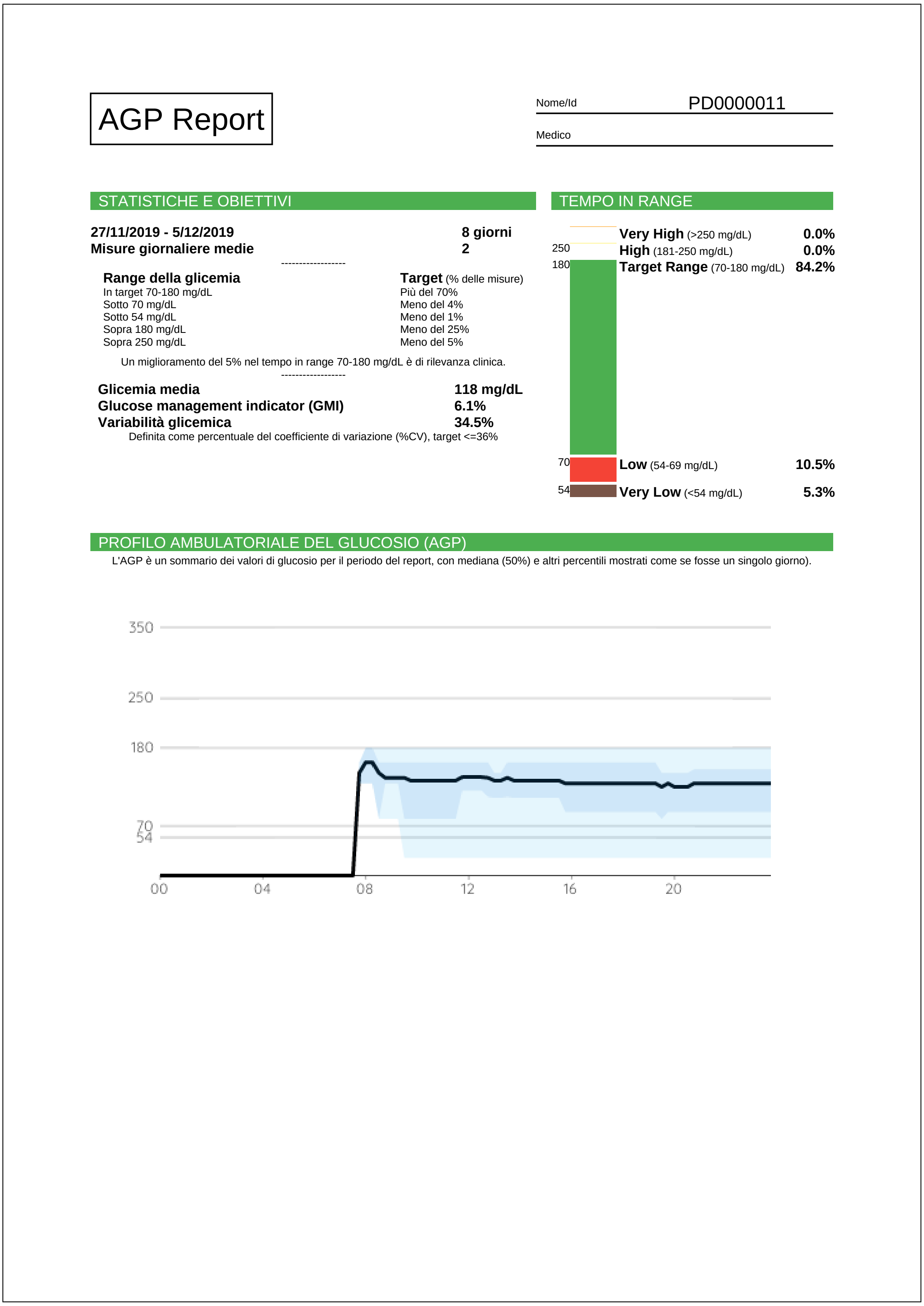
In Figure 7.1 it is shown the final report with all data of the testing session. The report shows an excellent glicemic control, with a high time in target. The graph in the report has data only during the day and an average of two measures a day. The graph shows some variability in glucose levels during the day and proves that the report building works.

The general feedback on the application is positive. The usability is good and all shown statistics allowed the subject to understand more his levels throughout the testing session. The whole application has been reported to be intuitive in use, but a helper or showcase function could be useful to discover every feature of the application. The subject reported a good experience in using the interface for data uploading, giving some useful advices for future improvements. For example insulin velocity metadata should be automatically chosen according to the specific patient settings, in fact pump or micro-infuser users usually get slow fast acting insulin, while pen users slow acting insulin. The subject reported also that during real life usage the metadata requested in meal insertion are usually the same and rarely considered. It has been proposed to show those metadata during specific trials only. Furthermore has been observed that the time selection for past data insertion could be improved setting a five minutes step in the selector, while at the moment the selector has one minute step. This would allow for faster time selection and consequent past data update. Some better explanation of some metadata could be added in order to give the user better understanding of some of them, such as "aerobic" activity and "fasting" time.

The only problem encountered is consistency in use. Notifications are implemented but not automatically set up, it has been suggested to add default notifications at every meal time. The subject was already using a micro-infuser, thus it was not important for him to track data manually on another device. This resulted in a small amount of data uploaded in the platform. The integration with health applications has to be developed as the highest priority feature in the near future.

With regards of monitoring, the system works in real time, but at current state the patient has no ability to know whether the clinician has checked new data. It will be implemented a system for the patient to check if it has been monitored and for the clinician if there is new data.

In conclusion the user reported that the application is really useful and helpful for those who use a manual diary during everyday life. If some new feature, such as automatic data insertion from other devices (CGM, fitness tracker), will be implemented the application will become even more useful.



AGP Report

Nome/Id PD0000011

Medico

STATISTICHE E OBIETTIVI

27/11/2019 - 5/12/2019	**8 giorni**
Misure giornaliere medie	**2**

Range della glicemia	**Target** (% delle misure)
In target 70-180 mg/dL	Più del 70%
Sotto 70 mg/dL	Meno del 4%
Sotto 54 mg/dL	Meno del 1%
Sopra 180 mg/dL	Meno del 25%
Sopra 250 mg/dL	Meno del 5%

Un miglioramento del 5% nel tempo in range 70-180 mg/dL è di rilevanza clinica.

Glicemia media	**118 mg/dL**
Glucose management indicator (GMI)	**6.1%**
Variabilità glicemica	**34.5%**

Definita come percentuale del coefficiente di variazione (%CV), target <=36%

TEMPO IN RANGE

Very High (>250 mg/dL)	**0.0%**
High (181-250 mg/dL)	**0.0%**
Target Range (70-180 mg/dL)	**84.2%**
Low (54-69 mg/dL)	**10.5%**
Very Low (<54 mg/dL)	**5.3%**

PROFILO AMBULATORIALE DEL GLUCOSIO (AGP)

L'AGP è un sommario dei valori di glucosio per il periodo del report, con mediana (50%) e altri percentili mostrati come se fosse un singolo giorno).

Figure 7.1: Beta testing session patient generated report with data of eight days

Chapter 8

Conclusions

mHealth has proven to be extremely useful in T1D therapy and management, in fact in recent years there has been a boost in research in this field. The aim of this thesis was to provide a new platform prototype that allows for patient monitoring and data gathering.

This project has obtained some important results: a new working platform prototype, usable for clinical trials, has been built and is ready for deployment. We have created a system that comprehends a cloud database, a mobile application, a web interface and the infrastructure that allows all communications between this elements. With this platform the clinician has real time monitoring capabilities of his patients with statistics and the patient has a mobile application that allows him to track his values and check some statistics of different time periods. This has been achieved by allowing the patient to add data manually in the application. All data is automatically uploaded to a cloud database for later reference and real time monitoring. The application was built trying to respect all the requirements identified and while giving him an exhaustive access to all data, the patient interaction is kept at minimum. The website has been developed to allow the clinician to check all his patients condition in a single place and in an easy way.

The whole system is modular and built so that it is easy to expand and reshape to the need of each specific trial. For example adding a new module for a new algorithm in testing, is as easy as creating a new file that builds the page and some data logic around it. The communications are one way only, from the patient to the clinician but it will be interesting to implement some alerts and suggestion that the clinician could send to his patient or change his parameters.

This platform prototype can be the starting point of a separate application that could be widely released which could implement working techniques to give patients a more integrated decision support system.

8.1 Future development

This project got the platform to a working point, but the whole project is not completed and many future possible features need to be designed and added.

At current state the application accepts only manual input by the patient. Many patients have some automatic sensors that measure their blood glucose continuously and it would be interesting to integrate this sensors in the data gathering platform. The Bioengineering research group at DEI has recently signed an agreement with Dexcom, Inc., a leader corporation in CGM sensors and diabetes care, which will provide all libraries and software to transfer in real time data from the sensor to the application. This will also allow to later implement all algorithms that compose a decision support system that use real time data.

Many smartphones, smartwatches and wearables nowadays allow tracking of physical activity, sleep cycle and many other types of information. It would be beneficial for the dataset and the patient to automatically retrieve this informations from the devices and integrate them in the platform. The integration would be possible using the health applications of each operative system. This would allow more complete data for later analysis and for decision support algorithms.

The platform has been developed and thought because there was the need of a consistent and easy method for data gathering and patient monitoring during trials of algorithms, thus all the algorithms will have to be implemented and those will integrate all data gathered and more information giving to the patient some outputs which will lead to a decision support system.

Each trial is different in the duration and type of data needed, in the current state the patient has to choose which data to track. In future versions it would be better to give the clinician this choice and allow the patient only to add more data types to the tracking. Moreover the clinician could send notifications to the application with changes in therapy, thresholds and other information. On the other hand the application could alert the patient if the blood glucose level are are below or above the safety thresholds and send to his clinician a notification to signal the event. This would implement a more integrated system allowing better patient care and disease treatment.

Appendix A

Code samples

A.1 Layout example

The following UI element is designed and coded as in Listing A.1.

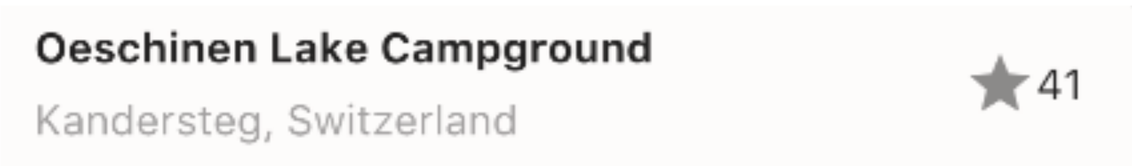


Figure A.1: Example layout of an UI element

The layout consists in a tree structure with a Container as root Widget. The Container Widget allows for some customization of the child Widget, in this case some padding all around the child Row Widget. The Row Widget describes an horizontal sequence of Widget: the first is a Column (a vertical sequence of Widgets) of two Text Widgets, the second is an Icon and the third is another Text widget. In order to have the second part of the row flushed right, the Column Widget is wrapped in an Expanded environment, which allows the Column to take all the horizontal space available.

```
Widget titleSection = Container(
  padding: const EdgeInsets.all(32),
  child: Row(
    children: [
      Expanded(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Container(
              padding: const EdgeInsets.only(bottom: 8),
              child: Text(
                'Oeschinen Lake Campground',
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
            Text(
              'Kandersteg, Switzerland',
              style: TextStyle(
                color: Colors.grey[500],
              ),
            ),
          ]),
      ),
      Icon(
        Icons.star,
        color: Colors.red[500],
      ),
      Text('41'),
    ],
  ),
);
```

Listing A.1: Layout example code

A.2 API examples

A.2.1 Model example

The following code shows the patient model that allows the API to know how to work with data from the patient table. In the model there is the description of they and the list of attributes, the password in this case, that are not returned in the JSON format to a request .

```
class Patient extends Model{
    protected $primarykey = 'id';
    public $incrementing = false;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'id', 'medic', 'gender', 'PWD', 'birth_date', '
            trial_code', 'test_group', 'clinic_code', '
            trial_patient_id'
    ];

    /**
     * The attributes excluded from the model's JSON form.
     *
     * @var array
     */
    protected $hidden = ['PWD'];
}
```

Listing A.2: Patient model

A.2.2 Migration example

This following code creates the glucose measures table. It has two functions, one for upgrade the database schema and one to downgrade it. In the up() function, it defines all columns needed with their properties and then it defines the primary key and the foreign key to the patients table.

```
class CreateMeasuresTable extends Migration{
    public function up(){
        Schema::create('measures', function (Blueprint $table) {
            $table->dateTime('time');
            $table->integer('patient');
            $table->double('glucose');
            $table->string('moment')->nullable();
            $table->string('note')->nullable();
            $table->timestamps();
        });
        Schema::table('measures', function ($table){
            $table->primary('time');
            $table->foreign('patient')->references('id')->on('patients')->onDelete('cascade');
        });
    }

    public function down(){
        Schema::dropIfExists('measures');
    }
}
```

Listing A.3: Migration example code

A.2.3 Query example

The following code selects all patients of the specified clinician.

```
public function showMedicPatients(Request $request){
    return response()->json(Patient::where('medic', $request
        ->input('medic'))->get());
}
```

Listing A.4: Query example code

This is equivalent to the SQL query

```
SELECT * FROM patients WHERE medic = medic
```

where medic variable is extracted from the request body. The result of the query is formatted in json for better usability and conversion in the application and website.

A.2.4 Routing example

The following code defines the URL "/api/patients/showAllMedic" to return the result of the query described in A.2.3. In fact on line 7 there is the definition of the route using the POST method which calls the PatientController function showMedicPatients.

```
1  $router->group(
2   ['prefix' => 'api', 'middleware' => 'client'],
3      function () use ($router) {
4          $router->group(
5              ['prefix' => 'patients'],
6              function () use ($router) {
7                  $router->post('showAllMedic', ['uses' =>
                       'PatientController@showMedicPatients']);
8              }
9          )
10     }
11 )
```

Listing A.5: Route example code

Bibliography

[1] Pew Research Center. 10 facts about smartphones as the iphone turns 10. [Online]. Available: https://www.pewresearch.org/fact-tank/2017/06/28/10-facts-about-smartphones/

[2] comscore.com. Global digital future in focus 2018 global digital future in focus 2018. [Online]. Available: https://www.comscore.com/Insights/Presentations-and-Whitepapers/2018/Global-Digital-Future-in-Focus-2018

[3] D. C. Klonoff, "The current status of mhealth for diabetes: Will it be the next big thing?" Journal of Diabetes Science and Technology, vol. 7, no. 3, pp. 749–758, May 2013. [Online]. Available: http://dx.doi.org/10.1177/193229681300700321

[4] pubmed website with search term "mhealth". [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/?term=mHealth

[5] G. Flodgren, A. Rachas, A. J. Farmer, M. Inzitari, and S. Shepperd, "Interactive telemedicine: effects on professional practice and health care outcomes," Cochrane Database of Systematic Reviews, Sep 2015. [Online]. Available: http://dx.doi.org/10.1002/14651858.CD002098.pub2

[6] S. Vyas, J. Murren-Boezem, and P. Solo-Josephson, "Analysis of a pediatric telemedicine program," Telemedicine and e-Health, vol. 24, no. 12, pp. 993–997, Dec 2018. [Online]. Available: http://dx.doi.org/10.1089/tmj.2017.0281

[7] "From innovation to implementation – ehealth in the who european region (2016)," OMS Europe, Tech. Rep., 03 2017.

[8] D. Marino, A. Miceli, D. N. Carlizzi, and G. Quattrone, "Telemedicina, cos'è e come farla in italia: tecnologie e finalità, un modello possi-

bile," March 2018. [Online]. Available: https://www.agendadigitale.eu/sanita/telemedicina-come-farla-in-italia-le-tecnologie-le-finalita-un-modello-possibile/

[9] D. Wright, "Kantar launches quarterly report for wearable technology," Kantar Worldpanel ComTech, Tech. Rep., 2016. [Online]. Available: https://www.globenewswire.com/news-release/2016/05/04/1088823/0/en/Kantar-Launches-Quarterly-Report-for-Wearable-Technology.html

[10] P. Mosconi, S. Radrezza, E. Lettieri, and E. Santoro, "Use of health apps and wearable devices: Survey among italian associations for patient advocacy," JMIR Mhealth Uhealth, vol. 7, no. 1, p. e10242, Jan 2019. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/30664455

[11] American Diabetes Association, "Report of the expert committee on the diagnosis and classification of diabetes mellitus," Diabetes Care, vol. 21, no. Supplement 1, pp. S5–S19, 1998.

[12] The Diabetes Control and Complications Trial/Epidemiology of Diabetes Interventions and Complications (DCCT/EDIC) Study Research Group, "Intensive diabetes treatment and cardiovascular disease in patients with type 1 diabetes," New England Journal of Medicine, vol. 353, no. 25, pp. 2643–2653, Dec 2005. [Online]. Available: http://dx.doi.org/10.1056/NEJMoa052187

[13] Diabetes Care Community. Understanding continuous glucose monitoring: A tool to help you manage your glucose levels. [Online]. Available: shorturl.at/jJVX1

[14] "Diabetes atlas - 8th edition," October 2019. [Online]. Available: https://diabetesatlas.org

[15] N. Ramchandani, "Virtual coaching to enhance diabetes care," Diabetes Technology & Therapeutics, vol. 21, no. S2, pp. S2–48–S2–51, Jun 2019. [Online]. Available: http://dx.doi.org/10.1089/dia.2019.0016

[16] S. Harrison, M. Stadler, K. Ismail, S. Amiel, and A. Herrmann-Werner, "Are patients with diabetes mellitus satisfied with technologies used to assist with diabetes management and coping?: A structured review," Diabetes Technology & Therapeutics, vol. 16, no. 11, pp. 771–783, Nov 2014. [Online]. Available: http://dx.doi.org/10.1089/dia.2014.0062

[17] R. Offringa, T. Sheng, L. Parks, M. Clements, D. Kerr, and M. S. Greenfield, "Digital diabetes management application improves glycemic outcomes

in people with type 1 and type 2 diabetes," Journal of Diabetes Science and Technology, vol. 12, no. 3, pp. 701–708, Dec 2017. [Online]. Available: http://dx.doi.org/10.1177/1932296817747291

[18] K. J. Rose, C. Petrut, R. L'Heveder, and S. de Sabata, "Idf europe's position on mobile applications in diabetes," Diabetes Research and Clinical Practice, vol. 149, pp. 39–46, Mar 2019. [Online]. Available: http://dx.doi.org/10.1016/j.diabres.2017.08.020

[19] O. El-Gayar, P. Timsina, N. Nawar, and W. Eid, "Mobile applications for diabetes self-management: Status and potential," Journal of Diabetes Science and Technology, vol. 7, no. 1, pp. 247–262, Jan 2013. [Online]. Available: http://dx.doi.org/10.1177/193229681300700130

[20] K. Huckvale, S. Adomaviciute, J. T. Prieto, M. K.-S. Leow, and J. Car, "Smartphone apps for calculating insulin dose: a systematic assessment," BMC Medicine, vol. 13, no. 1, May 2015. [Online]. Available: http://dx.doi.org/10.1186/s12916-015-0314-7

[21] F. Debong, H. Mayer, and J. Kober, "Real-world assessments of mysugr mobile health app," Diabetes Technology & Therapeutics, vol. 21, no. S2, pp. S2–35–S2–40, Jun 2019. [Online]. Available: http://dx.doi.org/10.1089/dia.2019.0019

[22] Americal Association of Diabetes Educators. Aade7 self-care behaviors®. [Online]. Available: https://www.diabeteseducator.org/living-with-diabetes/aade7-self-care-behaviors

[23] "Flutter vs react native – what to choose in 2019?" September 2019. [Online]. Available: https://www.thedroidsonroids.com/blog/flutter-vs-react-native-what-to-choose-in-2019

[24] Wm Leler, "Why flutter uses dart," October 2019. [Online]. Available: https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf

[25] "Flutter api: Widget class." [Online]. Available: https://api.flutter.dev/flutter/widgets/Widget-class.html

[26] T. Battelino, T. Danne, R. Bergenstal, S. Amiel, R. Beck, T. Biester, E. Bosi, B. Buckingham, W. Cefalu, K. Close, C. Cobelli, E. Dassau, J. DeVries, K. Donaghue, K. Dovc, F. Doyle, S. Garg, G. Grunberger, S. Heller, L. Heinemann, I. Hirsch, R. Hovorka, W. Jia, O. Kordonouri, B. Kovatchev, A. Kowalski, L. Laffel,

B. Levine, A. Mayorov, C. Mathieu, H. Murphy, R. Nimri, K. Nørgaard, C. Parkin, E. Renard, D. Rodbard, B. Saboo, D. Schatz, K. Stoner, T. Urakami, S. Weinzimer, and M. Phillip, "Clinical targets for continuous glucose monitoring data interpretation: Recommendations from the international consensus on time in range," Diabetes Care, vol. 42, no. 8, pp. 1593–1603, 2019.

[27] Dusterio. Lumen passport. [Online]. Available: https://github.com/dusterio/lumen-passport/blob/master/README.md