Pilot 3: Diabetes

1. Key Information

1.1 Involved Partners

- Huawei
- Nissatech
- Royal College of Surgeons in Ireland

1.2 Involved Countries

- Germany, Ireland

1.3 Keywords

- Complex event processing
- Detection of problematic patients
- Blood sugar monitoring
- Glucose level monitoring

1.4 Task Description

The aim of this pilot is to develop an integrated self-monitoring and self-management system for Gestational Diabetes Mellitus (GDM). The proposed solution is a response to the rising number of women with GDM which causes pressure on an overstretched Health Care (HC) system.

GDM can result in serious complications during pregnancy for both the baby and the mother, such as, pre-eclampsia, macrosomia, birth injury, premature delivery, newborn complications (hypoglycaemia, jaundice) and at a later stage there is a risk of developing type II diabetes if not adequately treated. The process of monitoring GDM patients is currently overly manual (most of the HC organizations), time consuming and prone to human error. In addition to running the clinic, midwives receive approximately 100 calls/week from patients to dictate their sugar levels over the phone, then they have to manually transcribe and store the information. In addition, existing monitoring methods are based on periodic visits to the hospital for those pregnancies complicated by GDM which are managed alongside routine antenatal care in an outpatient department with limited staff and resources.

Moreover, there is a lack of a standardized approach to deal with GDM, even across different hospitals within the same country. Thus, the pilot has proposed to use a flexible and explainable approach, the fuzzy inference system which enables incorporating the thresholds range measurements established by different hospitals which are used as a baseline to define the fuzzy inference rules.

In this pilot this monitoring process is improved by developing a system which automatically analyzes blood sugar level streams, in combination with other relevant information [1]. This will allow healthcare professionals to focus on the patients who are at greater risk, and reduce the visits to the hospital of patients who are having a non-problematic pregnancy.

2. Building Blocks

2.1 Architecture

2.1.1 System Architecture

The architecture consists of three main components, namely the patient smartphone, the analytics server, and the hospital server.

Patient smartphone

Represented in the schema as the phone icon. The patient takes measurements in the glucometer, and the mobile app reads them automatically. The mobile app can also collect patients' food intake information (food information is a manual input from the patient). Additional information which is currently not being sent to the server includes: exercise log, weight tracking, and pedometer (step-counter).

Figure 1. Some of app's screenshots (Homepage and Diary menu)

This app has two purposes:

- Send the information to the hospital server [1] via HTTP/S after recording the glucose level readings

- Collect the results from the analytics, provide motivational tips for improving patient's conditions, general GDM advice and some historical information, and present it in a way that is useful and easy to understand for patients.

Analytics server

Represented as 'Cloud Analytics' in the schema. It is an instance in the cloud, and it has two tasks.

The model is trained on the cloud-based server. To create the model, anonymized historical data is sent, and stored while the model is being trained. The first iteration of the model uses the original batch of data collected by the hospital, and as we collect more data during the pilot, this dataset will increase in size. After the training is finished, the model will be deployed into the final application, and the data will be destroyed.

After the model has been trained, the analysis of new incoming data is also performed on this server. The model is deployed into a java application that will run the analysis, and return the results to the hospital server, where they will be decoded. No information is stored in the analytics server. The analytics server hosts only the model. After the analysis, the application automatically deletes the information. The application receives de-identified data as explained in 2.2.3.

Hospital server

Represents the remaining elements of the schema. A server inside the hospital physical campus hosts it, and within the secure digital environment of the hospital, i.e. VPN. The following systems are also deployed in the hospital server:

- Kafka is the broker of the application. It operates as the messaging center, i.e. it receives messages from different parts of the application, and the application consumes the messages from the broker.
- Broker service is a service which receives data from the app, and forwards it over to Kafka.
- Database is where the data is stored. The Kafka consumer sends the data. It hosts personal information. Only developers and the software middleware can access.
- Kafka consumer is the middleware of the application. The consumer has multiple functions. It de-identifies the data, and then sends it to analysis. It writes to the database both the raw data and the results from the analysis.
- Medical portal is a webpage. Only authorized medical personnel can access it, it is only accessible through the hospital VPN. It is the tool for the medical team to see the historical data from the patient, and the results from the analytics. Doctors can use the web portal to send notifications to patients (e.g. ask the patient to come to checkup).
- KeyCloak is the technology used for OAuth 2.0 authentication.

- Security proxy is a simple piece of software used as reverse proxy (Nginx), to route requests coming to the server to appropriate services, and to hold SSL/TSL certificates for each service.
- Firebase is the technology used to send notifications to the phone.

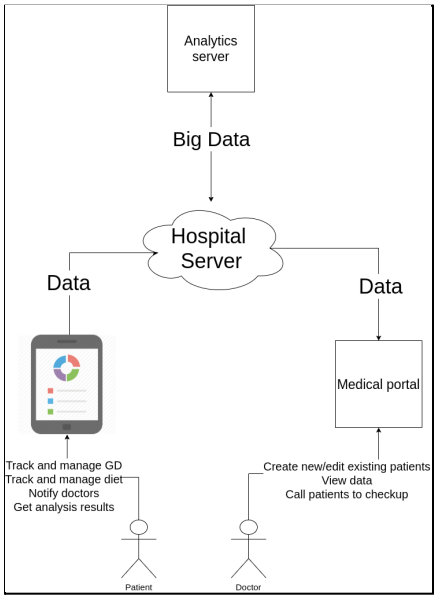

Figure 2. Schematic Representation of usage of the application

Figure 2 describes how the patients use the application to record, track and manage their glucose levels. The data can be sent via the mobile app to the hospital server, and the information is displayed to medical staff via the medical portal. In this use-case, the HCPuses the portal to create new patients, review their data, and directly communicate with patients by sending notifications to their app to call them for checkup. Only after being enrolled by assigned HCPs, patients are allowed to log into the app and use it.

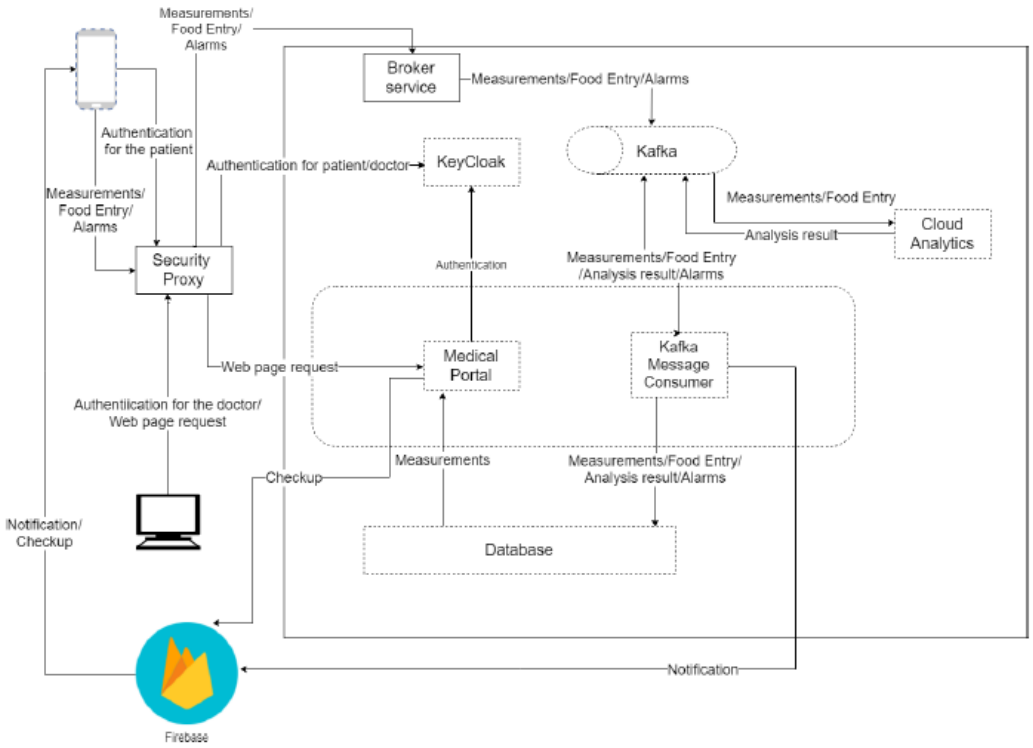

Figure 3. Detailed architecture

Figure 3, shows an overview of the overall architecture described in the above-mentioned sections and the data flows.

2.1.2 Data Flow & Interoperability of services

2.1.3 Necessary Hardware

Patients: a glucometer with Bluetooth and a smartphone, preferably Android from Android 6 and above, or iPhone from iOS 12 and above. In this pilot, two different glucometers have been used, i.e. Contour PlusOne meter, or Nipro 4SureSmart meter.

For the hospital, necessary hardware is minimal, a server with the following hw specifications: 4 core processor, at least 8GB of RAM, and at least 10GB of HDD/SDD (needs to be adaptable to scale up with incoming data).

2.1.4 Software Components

Software components used on server side are mostly concerned with back-end performing as intended, but there are some tools used to ease the deployment and communication itself. Firstly, the full list of server's third-party software components (used, not developed by Nissatech) is as follows:

- Ubuntu 18.04 OS x64-86

- Docker
- Docker-compose
- Nginx proxy
- PostgreSQL
- Kafka message broker
- Keycloak

Firebase is another service we used for delivering notifications to the patient's smartphone, but it is Google's Cloud service, so it is not installed on our server, only it's API is called to achieve needed functionality.

2.2 Data Sources

2.2.1 General Information

The aim of this pilot is to increase the frequency of blood sugar level monitoring for those patients that were diagnosed with GDM (gestational diabetes mellitus) until the pregnancy is completed. The overall approach is to collect daily sugar levels (primary measure), thereby obtaining a lengthy review of glycaemic control, which ensures a prompt recognition of high sugar levels. In addition, the solution proposed will improve patient awareness regarding the effect of diet, as the patient can also provide this information using the mobile application developed for the pilot. At a later stage, other relevant parameters will be included such as exercise.

During the course of the project, more data will be available (both historical and real-time) which will serve to improve the selected models for predicting risk situations for the patients. This will be an iterative process until the end of the project. From the technical point of view, two different types of data sets will be used:

Table 1. Data sources

Data Source	**Description**	**Acquisition**	**Characteristic** **(Size, Patients, Years, Origin/Region)**
Monitoring thresholds data	Glucose thresholds used to identify when a patient's sugar level has reached a threshold that may cause a risk for the patient. It consists of a range of thresholds for fasting and postprandial measurements This data was used to generate the fuzzy thresholds	Communication with different hospitals, guidelines.	30 hospitals

Pre-study data	The hospital collected data from pregnant women diagnosed with GDM. All data comes from patients with GDM, as non-diagnosed patients do not take measurements of their glucose levels. It consists of two parts: Patient background data (medical data, antenatal ultrasounds, abdominal circumference) and a glucose level time series for every patient. In Table 3 the data elements are further described. For the fuzzy models, this data is used for validation of the model. In the predictive part, it is used for training.	Previous studies done by the hospital	First round with 64 patients
Real time data	Once the pilot is running in a real operational environment, the proposed RPM (remote patient monitoring) system will collect relevant information from the real time usage. The data includes sugar levels measured with a glucometer sent via a mobile app.	The pilot main stream of data is the sugar levels of the patients that will be measured with a glucometer, and sent to analyse automatically via a mobile app. The mobile app also generates more data, namely the food intake of the patient. This data needs to be manually introduced by the patients, and therefore is less reliable.	-

Table 2 presents a synthesis of some key aspects of the pilot.

Table 2. Data sources key aspects

Pilot	**Multiple sources**	**Integration to data warehouse**	**Data access**	**Data stored in cloud**	**Multi-party architecture**	**Secure environment**	**Transform raw / unstructured data**
3	yes	yes	Edge processing + key access permission controls for identity authentication based on IAM	yes	no	yes	yes

2.3 Data Processing

2.3.1 Multi-velocity processing of heterogeneous data streams

This pilot faces input of data at uneven intervals, and with varying frequency. Some types of data are measured once at the beginning of the process, whereas some others are measured daily, with different frequencies. Table 3 presents different streams in a more detailed fashion.

Table 3. Data streams descriptions

Stream name	Contents of stream	Stream velocity	Description of the stream
Background data	Medical data (variable-value)	Initial patient input which can be updated in additional visits to the doctor	The doctors will collect this data into the system. When a doctor registers a new patient into the pilot, there are different measurements that can be introduced (BMI, age, ethnicity, results of initial glucose test, smoking status, employment status, etc)
Antenatal ultrasounds and other specific data	Medical data (variable-value)	Periodic visits to the hospital during pregnancy	Realization of ultrasounds (e.g. macrosomia predicted, polyhydramnios), gestational weight gain, need for insulin, etc
Abdominal circumference	Medical data (variable-value)	Initial input, then measured every few weeks	During the pregnancy, the doctors monitor the abdominal circumference, as well as the centile.
Glucose monitoring	Glucose levels (variable-value)	From 4 to 7 times per day	Glucose monitoring constitutes the main stream of the project. The patients will use a glucometer connected to a mobile app to report their blood levels. The mobile app will send the values from the stream into the computing infrastructure.
Activity data	Patient activity (exercise) (variable-value)	Every time the patient does sport (4-7 times per week)	Exercise done by the patient. It needs to input this data manually into the app, so its availability will be much sparser.

Food monitoring	Food intake (variable-value)	Every time the patient eats regular food and drinks	Food and drink consumed by the patient. The patient needs to input this data manually into the app, so its availability will be much sparser.

Combination of different streams:

There are two main objectives to combine different streams in this pilot.

The models that we will use for monitoring the patients benefit greatly from the additional information generated by each one of the different streams. By combining them, the models will have access to a richer set of information, helping them to improve their recommendations to the patients.

The doctors will have a visualization screen that they can use to follow the evolution of the patients. This screen will present the data in a way that is useful to the doctors, as well as present the results of the models. This will allow doctors to take the most informed decision, both using raw data and the model results.

Challenges combining different streams:

The data that we are working with, especially the background data, is extremely sensitive. We needed to design a system that kept all the confidential information inside the hospital infrastructure, and that allowed communication with more powerful resources. As an additional difficulty, those resources are located in a different country, outside the hospital infrastructure. The design addresses this difficulty, making sure that it ensures users privacy.

Solution:

We decided to deploy an Apache Kafka broker on a server inside the hospital infrastructure.

This broker has a wide range of tasks to perform, namely:

All the messages generated go through the broker infrastructure. When a doctor introduces a new patient, it goes through the broker before going into the database. The broker and the doctor’s screen are hosted in the server inside the hospital infrastructure, and stay secured.

The mobile app sends data (glucose and food) to the broker. The broker cleans and organizes it, and then redirects it to the database (for storage), and to the outside server (to apply the model).

The data that is sent to the outside server is completely anonymized beforehand, and it consists of a time series of glucose levels. Before the data leaves the hospital infrastructure, we make sure that it cannot be traced back to any individual person. Then, the model is applied and the results are sent back to the broker. The broker distributes them to the database. From there, the doctors can consult the data through their interface.

The intermediate broker allows us to ensure data privacy in a scalable fashion. In Table 4 we can see the streams that are mixed, and in Table 5 the challenges we faced, and how we solved them.

Table 4. Combination of streams in the application

Streams mixed	**Technology used**	**Purpose of the process**	**Result (name if is a stream)**
Background data Glucose monitoring Food monitoring Antenatal ultrasounds Abdominal circumference	Java, Kafka, Flink	Prepare the data before sending it to the analytics model.	Stream (Analytics input).
Analytics input Analytics output	Java, Kafka, Flink	Store the results of the analytics, and show them to the medical team.	Database storage Visualization layer.

Table 5. Challenges in stream combination

Streams combination	**Difficulties**	**Solutions**
Analytics input	Data is being collected at different speeds, and with different frequencies. Organizing the data to serve as an input for the analytics model.	Considering the data by date, as our model needs the temporal information. Order by date, inform the model which variable is being sent.

2.3.2 Processing of large structured / unstructured data sources

2.3.2.1 De-Identification and anonymisation

As the training data includes personal information, as well as other relevant measurements, it undergoes a process of de-identification which eliminates any type of sensitive or personal information, but it keeps variables potentially useful for the model (range of age, ethnicity, number of pregnancies and any other relevant information). All recruited participants in the pilot will be assigned a study ID number and only authorized members of the medical research team will be able to link the study ID with identifiable patient information.

2.3.2.2 Acquisition

We acquire the data using our system, with patients that agreed to join the study.

2.3.2.3 Cleansing

The mobile app cleans the stream data. It sends only clean data into the hospital server. The medical introduces the static data (ethnicity, etc.) into the system at registration time. They use the web app front-end. This front end allows only a limited number of options. Therefore, all the data that the infrastructure receives is clean. Regarding missing data, the predictive system will be missing-values tolerant, since the patient can forget to take measurements of her blood sugar. This won't affect the system at any point, and it will continue monitoring the patient.

2.3.2.4 Data Integration

To perform the analysis, we need to format the data. The format used has the static data, and some historic data (current version has two weeks of data). It uses the glucose levels time series, which can't be traced back directly to a patient.

2.3.3 Natural Language Processing

Does not apply.

2.3.4 Image Processing

Does not apply.

2.3.5 Complex real-time event detection

2.3.5.1 Notifications

1. **Type of notifications**

Table 6. Types of notifications and alerts to be issued

Need for notification services			
Notification	**Warning**	**Alarm (automated / manual reaction)**	**Other**
Traffic light system – (red if above given value, green if normal, and orange if below). Risk level based on historical evolution (to medical team)	If red notification is issued two times in one day. If the risk level increases too much.	If red notification occurs more than 2 times in one day, issue an alarm. If the risk level arrives at dangerous levels.	-

2.3.5.2 Situations of Interest

Type of situations of interest

Table 7. Types of complex events to react on

Type of situations of interest			
Simple	**Trends (time-window / frequency based)**	**Complex (multiparameter / historical context)**	**Other**
If the value of glucose sugar is above the predefined threshold	If the value of glucose sugar is increasing continuously in several consecutive measurements (or during one day) Glucose level are high in a time window for a patient with tendency to low levels	If the value of glucose sugar is above a predefined threshold and the value of blood pressure is high (increased) Glucose level show patterns shared with previous patients that developed issues	-
When user takes 5th measurement of glucose today -> display motivational message	Each day at 9 am, 12 pm, 3 pm, 5 pm and 7 pm -> issue notification (reminder) to take glucose measurement	Doctors are alerted when they go to portal about patients whose glucose readings are above normal values (if around 30% of total glucose readings are above threshold) -> Medical staff can then issue notification with which they call patients to come to check-up	

2.3.5.3 Event Processing

2. **Type of event processing**

Table 8. Types of event processing actions

Filter	**Transform**	**Other**
Only the values in a range are valid	The average value of several measurements should be sent to e.g. medical doctor	Enrich – Add model predictions information, together with historical data.

2.3.5.4 Event Sources

3. **Event sources**

Table 9. Event source during complex-event processing

Stream name	Contents of stream	Stream velocity	Description of the stream
Glucose monitoring	Glucose levels (variable-value)	4 times per day	The patients will use a glucometer connected to a mobile app to report their blood levels. The mobile app will send the values from the stream into the computing infrastructure.

Food monitoring	Food intake (variable-value)	Every time the patient eats	The mobile app also allows the sending of food data. The patient needs to input this data manually into the app, so its availability will be much sparser.

2.4 AI Components

2.4.1 Prediction Algorithms

Task: Monitoring and categorizing of glucose levels

Data, Data Modelling: We used the thresholds and guidelines used in different hospitals as an input to establish the fuzzy inference range of thresholds for postprandial and fasting measurements. Prior studies from Rotunda and Drogheda hospitals to validate the baseline.

Size:

Patients: 67 patients

Features: Glucose levels

Model: The model in the first version uses fuzzy inference. It generates a risk level for each patient, on a scale from 0 to 100. Results are easy to interpret as it is a linear scale, and the reasoning behind the risk is explainable, as it generates from the rules used in the hospitals. Creating a range value allows for a more personalized diagnosis, as the decision changes from going/ not going to the hospital to an intuitive scale to decide which patients need more attention.

Fuzzy inference works in four steps. It first converts the information coming from the environment (glucose readings) in an internal format (i.e. through membership functions) acceptable by the knowledge-base and the decision-making unit (i.e. evaluating IF-THEN statements). This process is called fuzzification. Symmetrically, the defuzzification provides a conversion procedure to transform information coming from the decision-making module into the form acceptable by the modeling environment (i.e. interpreting the rule evaluation output as a numerical quantity, in this case, a risk scale from 0 to 100).

The model is explained with more detail in (Salort Sánchez, et al., 2019), as well as the experiments conducted to ensure its validity.

As more data becomes available, the model will be improved with new technologies.

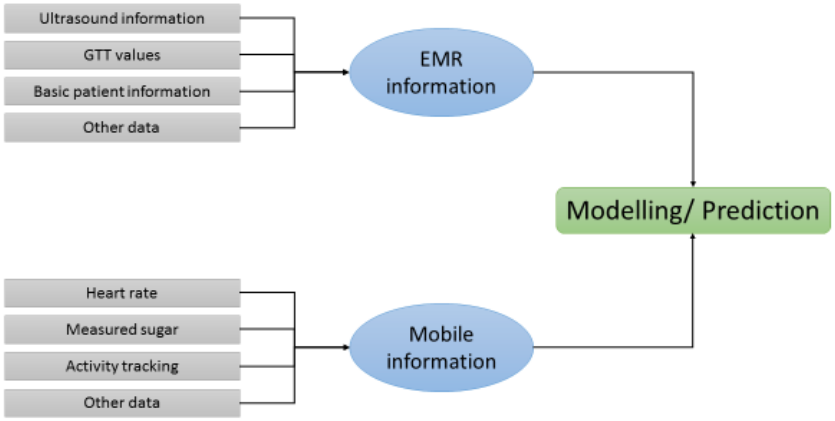

Figure 4. In this diagram, other data includes the possibility to include new sources of data, given that technology develops in the span of the project, and new hardware is manufactured that may be useful for a better development of the final solution.

2.4.2 Experimental Setup

2.4.2.1 Description

Membership functions were calculated for the fuzzy inference system based on the guidelines and current processes used in different hospitals. Once the fuzzy inference system was ready, the next step was validating the model using existing information based on previous well analyzed cases.

2.4.2.2 Evaluation

- **Description of Evaluation Setup** For validation, we calculated the distribution of data using the real observations, and generated a large set of patients following the real distribution. We then compare the results of the fuzzy inference, with the diagnosis that would take place following the hospital rules.
- **Size of Data:** Real data 30 patients, simulated 4000 observations
- **Evaluation Method** (e.g. ROC, F1): Accuracy
- **Results:** 94.68% accuracy. Border cases (the ones that had different values for the fuzzy inference and the current hospital method) were studied individually, and it showed that some cases where the fuzzy inference detected risk would benefit from a hospital visit, and the inverse when the fuzzy inference didn't detect but the current system did. This allows for better personalized attention.
- **Comparison to other related work** There aren't any similar works.

2.5 Security and privacy of data access and processing

There are multiple layers of security ingrained into the project. First off, we have usage of secure keychains of the iOS/Android operating systems. We primarily use them to store access and refresh tokens for sending data to the server in a secure way. Data in transit is encrypted with the latest TSL version, with SSL/TSL certificates and keys. Server with Database is firstly, inside of RCSI's VPN, secondly, the very Database is inside of a docker stack which requires root privileges to access, and thirdly, the data inside of the Database is encrypted, meaning, data at rest is quite safe.

Next, privacy of data access is resolved by respecting OAuth 2.0 protocol for resource access. Namely, we use Keycloak, which is an open source Oauth server solution, which respects aforementioned protocol, allowing only registered and logged in users to send/access data in Database. Also, only medical staff can register users, and they decide who can send data to the server.

Each of the services of the server communicates with each other in a confined Docker network, meaning, no one but the root user of the server can access their communication, nor data.

To summarize, for security we use multiple protocols on multiple levels, to make sure data is safe during transit, and at rest. Data access is managed by Oauth 2.0 specification of roles and resources via Keycloak server.

2.5.1 Access Control

2.5.1.1 Authentication

On the patient side, the app has authentication and maps who is the owner of the data

On the medical site, each member of the medical team has a user for the portal, and actions are also recorded. Furthermore, to connect to the VPN an additional login is needed.

Developers need to authenticate on the VPN and on the server.

2.5.1.2 Authorization

Medical team members can only see their patients, and each patient can only view their own data. Authorization is done via Keycloak OAuth server and its resource management functionality

2.5.2 Data Protection

2.5.2.1 Data at rest

Data is encrypted in a database. Only developers have access directly to the database, and even that requires first access to the server, then root user credentials, and after that credentials

for the database, which are hidden via docker-compose secret sharing concept. The frontend is only accessible through authentication

2.5.2.2 Data in transit

For data in transit, there are limited choices when it comes to available protocols, so we went with the currently most secure one, the TLSv1.2 (there exists TLSv1.3 but it is not yet supported by many countries). To achieve TLSv1.2 we use certificates for which only we have the keys.

2.5.3 Auditory and logs

2.5.3.1 System Auditory

Files get written locally into the server.

2.5.3.2 Services Auditory

Database audit logs exist within its docker container as well as on the host machine.

Other service's logs are within their respective directories.

2.5.4 Privacy measurements

2.5.4.1 Data Privacy Impact Assessment (DPIA)

The clinical partner had to write and get approved for a DPIA. A Data Protection Impact Assessment (DPIA) is a process that helps the clinical partner identify risks to the privacy of data subjects and ensure legitimate best practices are followed when a new project is planned, or when changes are made to an already existing product or service. The purpose of a DPIA is to ensure that privacy related risks that arise during data collection, use and disclosure are mitigated using appropriate plans and measures, while allowing the objectives, outputs or deliverables of a project which involves the use of personal data, to be met.

2.5.4.2 Legal/Ethical process

We applied for approval of the medical investigation. This process included multiple parts:

- Approval of the study by an independent ethics board
- Completion of multiple documents explaining in detail every detail of the application
- Compliance with legal entities. As the pilot is based in Ireland, this corresponding authority body is the HPRA (Health Products Regulatory Authority). HPRA has classified the mobile app as a Class IIA according to the MDR (Medical device regulation). Application to be submitted to the authorities is in process to obtain the corresponding approval.

2.5.4.3 Processes for complying with the current legislation

The processes have been described in the previous section. The documents include an extensive list of tasks that needed to be completed and documented. A non-comprehensive list follows:

- Clinical investigation plan. Including purpose, scientific, technical or medical grounds, scope, number of patients, etc.
- Investigator's brochure. Including literature review, comprehensive risk list, clinical evaluation, clinical development strategy etc.
- Preclinical testing. Testing of the system, validation testing, description of device components, technical explanations, etc.
- Documents and statements. This includes approval by the ethics board, data agreements, informed consent of the patients, etc.

2.6 Trustworthy AI

2.6.1 Technology/user adoption and establishing trust

For the patients, using the app with the glucometer eases the process as it is entirely remote and easy to use

For the medical team, the proposed system allows the HCPs to digitalize the overall process (monitoring BSL, data storage and communication) with adequate security levels. This process is currently overly manual, time consuming and prone to human error. Furthermore, this process can greatly optimize the workflows for HCPs, hospital costs and resources.

Moreover, the analytics model developed based on Fuzzy inference systems, is a simple and explainable model, totally transparent to HCPs which offers an interpretable output tailored for each patient where the final decision making is done by the HCP.

2.6.2 Ethical principles

- Respect for human authority. The fuzzy inference generates a risk score, which is meant to help medical teams take decisions, and in no way override them.
- Prevention of harm.
- Fairness. The model does not have any obvious discrimination bias. The roadmap includes making sure that the model hasn't created any bias when we introduce new cohorts.
- Explicability. Fuzzy inference is a type of modeling that can be explained, as it can be interpreted as a complex set of rules.

2.6.3 Key requirements

- Human agency and oversight. The fuzzy inference generates a risk score, which is meant to help medical teams take decisions, and in no way override them.
- Technical Robustness and safety. As the model it's explainable, it is robust in the sense we know exactly how it would behave in a given case. Data is also available for the medical team, which can override decisions at any point if they are unsure about any result.
- Privacy and data governance. Privacy was fundamental in the design of the architecture, and patients can't be recognized from their data.
- Transparency. The architecture and the functioning of the model are shared.
- Diversity, non-discrimination and fairness. The model does not have any obvious discrimination bias. The roadmap includes making sure that the model hasn't created any bias when we introduce new cohorts.
- Societal and environmental well-being. The model is designed in a scalable way, so multiple hospitals and countries can participate with minimal ecological footprint.
- Accountability. Actions within the system are monitored and can be audited.

2.7 System-Interaction

2.7.1 Human-Machine Interface / GUI

Our GUI is very simplistic, based on material design advice and recommendations. The goal of our design is to be as appealing to patients as possible, to help with engagement of patients in the project. The detailed description of the app's interface can be found in the document "GDMApp user scenario guide", here we will just briefly go over a couple of main views.

Firstly, when users open the app, and whenever refresh token expires, they will be asked to log into the application via following screen:

Figure 5. App login page

After that, users are presented with the homepage of the app, which contains a short greeting and link to PDF with detailed explanations about the GDM (GDM Booklet).

Figure 6. App homepage

In the bottom bar there is the main navigation bar, with the following options Home, Blood Sugar, Diary and Connect. Each of the options is linked to corresponding views with different functionalities.

For example, Blood Sugar levels displays the following information on the screen:

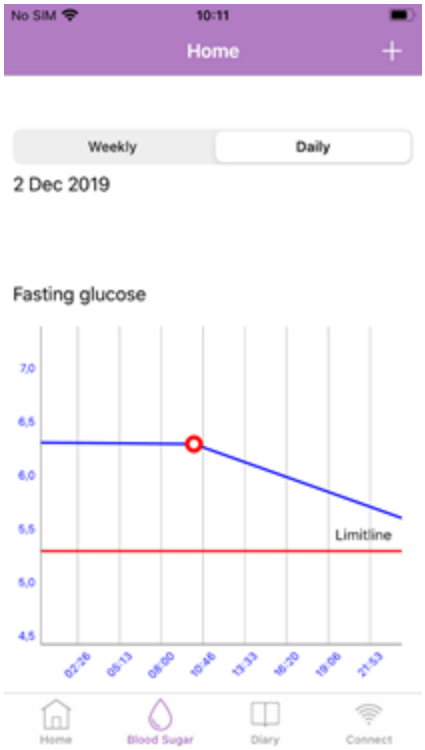

Figure 7. Blood glucose page

On this page the user can review our glucose readings via charts, as well as manually input new ones via the plus icon in the upper right corner.

Diary page is, in a way, a menu of the application's other functionalities such as Food Diary, Exercise log, Pedometer, Contact page and more, as shown in following figure (Figure 8).

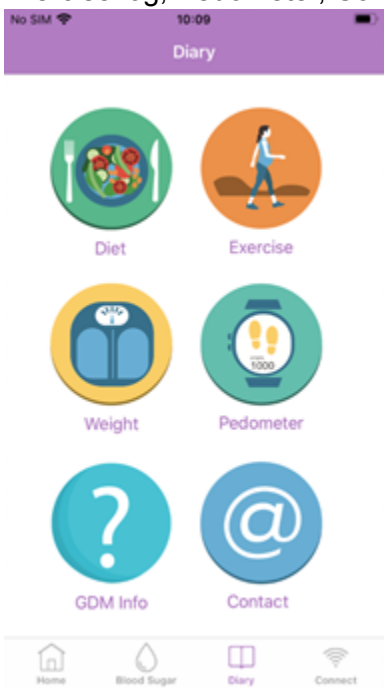

Figure 8. Diary

Finally, Connect screen (Figure 9) is used to connect a patient's glucose meter with the application via Bluetooth 4.0 standard (BLE). Connect is just a list of nearby glucose meters, which allows the patient to select one of meters, connect with it and receive glucose readings from the meter.

Figure 9. Connect view

With these four main screens, patients can easily access and manage the application's most important functionalities.

2.7.2 Education

Patient Education: Patients can use the app to record and monitor their blood glucose levels. They can also use the app to record their diet and lifestyle routine, including recipes and healthy options so they can go back to it when they need to. The information is saved on the app, all in one place. There are also educational links which the patients can use to educate themselves further on understanding gestational diabetes and how to monitor it better throughout their pregnancy.

Health care worker Education: The HCW has access to the portal through which they can view the patient recordings and see if the patient has high blood sugar levels. They will be able to see if the patient is compliant or if they are having any issues. They can also review patient feedback.

3. Learnings

3.1 Challenges & Barriers

- Architecture
    - One of the challenges of such architecture is orchestrating. We have many services which need to function as a unit, to fully achieve results. The solution to this challenge was, of course, docker stack. By using docker-compose to manage docker containers it was even easier for us to orchestrate our services and deployment.
    - Scalability. How to design an architecture that would easily scale into different hospitals, while preserving data privacy. The option chosen was a centralized analytics server, where multiple hospitals can connect for obtaining predictions, and multiple individualized data servers for each hospital. This central server does not store any data, which is stored locally in each one of the hospitals, inside their secure infrastructure.
- Processing of large structured / unstructured data sources
- Multi-velocity processing of heterogeneous data streams
    - All streams of data trigger their own system for cleaning and organization, independently on their arrival time. Some of them also trigger analytics every time a new measure arrives.
- Complex real-time event detection
    - Ensure that new data is being continuously analyzed. We solved it by having each new measurement as an independent event, meaning it used its own resources, and thus we allow scalability.
- Natural Language Processing
    - NA
- Image Processing
    - NA
- Prediction Algorithms
    - Deciding which model to use that would be explainable and enable better monitoring for the medical team. We decided to use a model that puts together all the knowledge used in different hospitals, while still being explainable for the medical team.
- Security and privacy of data access and processing
    - Finding a good CA to provide us with appropriate SSL/TSL certificates was quite a barrier, considering how many CA's there are. Namely, not all of the CA's offer the same level of insurance and security, and finding a good balance between price and quality was quite tough.
    - Also, data encryption with open source databases is very scarce. Implementing our own solution is costly, and using paid DB can be expensive. Luckily we found that PostgreSQL can understand basic C language if appropriate flags are turned on, so we changed the way PSQL reads and writes data by adding a seamless Transparent Data Encryption (TDE) to the PostgreSQL database.
- Trustworthy AI
    - As medical applications are extremely sensitive, it is challenging to ensure that the system is understandable to everyone involved. The above-mentioned

approach is transparent to HCPs and easily explainable. In addition, the ultimate decision-making is done by the HCPs.

- System-Interaction
    - Medical practitioners tend to adapt slowly to new, unproven technologies. Building trust and creating a tool they deem positive may be challenging.

3.2 Lessons Learned

- Architecture
    - Designing the architecture in a horizontal and distributed way allows easy to keep data privacy. Local servers inside hospital infrastructure allow for a safe management of data, and most of the anonymization effort must be conducted in the vertical part of the process, i.e. the prediction part of the system.
- Processing of large structured / unstructured data sources
    - In some particular cases, structuring data in a .csv and importing it into PostgreSQL can be faster than writing record by record/bulk to the DB.
- Multi-velocity processing of heterogeneous data streams
- Complex real-time event detection
    - Analyzing multiple users independently eases distribution and allows multiple analysis to be done in a faster way
- Natural Language Processing
    - NA
- Image Processing
    - NA
- Prediction Algorithms
- Security and privacy of data access and processing
    - PostgreSQL has no official support for data encryption but that can be circumvented with TDE.
    - During our exploration phase, where we searched for optimal solutions for data access management, we found that a service named auth0.com is probably the best security platform today, it offers the widest variety of options, easily understandable console, and even better prices than AWS IAM or Azure AD.
- Trustworthy AI
- System-Interaction
    - Final users may have needs that are not obvious from the developer points of view. If possible, it is valuable to interact with them, as it will show unforeseen needs for the system.

3.3 Main (quantifiable) achievements

- Highly reducing the number of hospital admissions which will reduce hospital costs and staff resources' while also increasing patient care at home.
- Better understanding of patient health conditions and predicting healthcare complications earlier.
- Significant cost savings in terms of healthcare such as costs related to hospitalization of patients and improve the efficiency in the clinic processes.

- Secure transfer, storage and analysis of health information to greatly improve the current hospital system.

4. Output

4.1 Papers

C. Salort Sánchez *et al*., "Fuzzy Inference System for Risk Evaluation in Gestational Diabetes Mellitus," *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, Athens, Greece, 2019, pp. 947-952, doi: 10.1109/BIBE.2019.00177.

4.2 Open Source & Resources (refer to ELG)

Code for model training and validation: https://github.com/csalort/fuzzy-gdm

4.3 Demos

The study team (HCP) provides patients with a video link which shows a demo of the app and the portal. It goes into detail about the aim of the study and what it entails for the patient. The video includes images of the app and the portal to show where their information is being stored. It also goes through a step by step guide with the patient on how to use the app.