EDMON - A System Architecture for Real-Time Infection Monitoring and Outbreak Detection Based on Self-Recorded Data from People with Type 1 Diabetes: System Design and Prototype Implementation

Sverre Coucheron[1,*], Ashenafi Zebene Woldaregay[1,*], Eirik Årsand[2], Taxiarchis Botsis[3], Gunnar Hartvigsen[1]

[1]Department of Computer Science, University of Tromsø -The Arctic University of Norway, Tromsø, Norway, ashenafi.z.woldaregay@uit.no

[2]Norwegian Centre for E-health Research, University Hospital of North Norway, Tromsø, Norway

[3]The Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University School of Medicine, Baltimore, MD, USA

[*]Authors contributed equally

Abstract

Infection incidences in people with diabetes can create sever health complications mainly due to the effect of stress hormones, such as cortisol and adrenaline, which increases glucose production and insulin resistance in the body. The proposed electronic disease surveillance monitoring network (EDMON) relies on self-recorded data from people with Type 1 diabetes and dedicated algorithms to detect infection incidence at individual level and uncover infection outbreaks at population level. EDMON incorporates four major modules; patient modules, mobile computing modules, computing modules (cloud backend), and end user modules. This paper presents the patient and computing module prototypes along with various essential design choices and challenges together with their solution. At the time of writing, development of the EDMON infection and outbreak detection algorithms are already completed and the next phase of the study involves integration of the prototype along with the EDMON algorithms, developing end user visualization mechanism and performing a pilot study.

Keywords

Type 1 diabetes, Infection detection, Outbreak detection, Patient module, Cloud backend solution

1 INTRODUCTION

The Emerging and re-emerging infectious disease outbreaks are still a major threat to public health security requiring instant detection and response [1]. The transition from the classical surveillance to the syndromic surveillance has improved the timeliness for optimal public health response. Syndromic surveillance often uses clinical features that precedes diagnosis and various activities triggered by the onset of symptoms as an alert of change in disease activity [2]. Patient information is acquired from secondary sources of information primarily built for other purposes, which includes google searches, twitters, pharmacy drug sells, school and work absenteeism, and others. Due to the ever changing social and biological environment dynamics, new surveillance systems integrated with novel aberrant detection mechanisms are required to meet the demand of the rapidly changing world [2].

Diabetes mellitus is a chronic metabolic disorder, which results in abnormal blood glucose levels. Blood glucose control is mainly maintained through self-management practices involving active tracking of blood glucose levels, administering proper diet and medication, and performing balanced physical activity [3]. The number of people with diabetes was estimated to be around 425 million in 2017 and yet forecasted to reach 642 million by 2040 [4]. The incidence of infection in people with diabetes often creates sever health complications, mainly due to increased glucose production and insulin resistance developed in the body [5, 6]. This profound physiological alteration is occurred mainly due to the effect of stress hormones, such as cortisol and adrenaline, which in turn is triggered by the onset of infection in the host body [7-9]. As a result, it is common in people with diabetes to experience elevated blood glucose levels and increased insulin consumption upon infection incidences. The proposed Electronic disease surveillance monitoring network (EDMON) is an event-based digital infectious disease outbreak detection system, which relies on self-recorded data and dedicated algorithms to detect infection incidence at individual level (micro-events) and uncover outbreak detection at population level (macro-events) [10]. The EDMON system architecture mainly incorporates four major modules; patient module, mobile computing module, computing module (private cloud backend), and end user module, as shown in the **Figure 1**. The patient module is a standalone mobile health app that integrates different sensor reading including blood glucose levels, insulin injection, and diet information along with Global Positioning System (GPS) coordinate of the patient in terms of geographical latitude and longitude. The mobile computing module handles the communication and

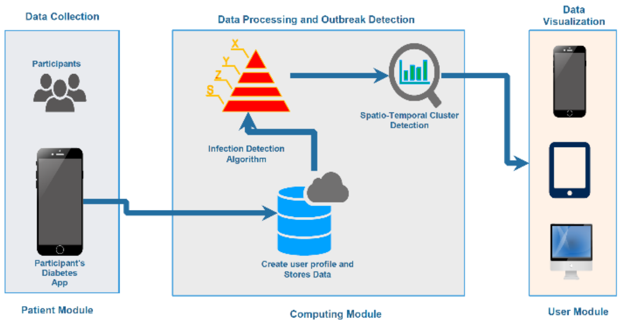

Figure 1 EDMON System Architecture.

transmit the user's collected data from the mobile health app to a remote private cloud for further processing. The main purpose of the computing module (cloud backend) is to securely store users' data and further processing of these data so as to uncover any potential threat of infectious disease outbreaks. The infection detection algorithm tracks individual diabetes patients and detect any incidence of infection using the key diabetes parameters; blood glucose levels, insulin injection and diet information. The spatio-temporal clustering algorithm scans the entire geographical region under surveillance looking for a group of infected people with diabetes within a specified time frame [11]. The end user module disseminates the status information for the concerned bodies and ordinary citizens under the surveillance regions based on both a standalone mobile application and through a dedicated website. It incorporates various means of information visualization techniques including maps, tables and graphs, where the end user can easily access through various devices including smartphones, tablets and desktops. To this end, this paper presents the patient and computing module prototypes along with various essential design choices and challenges together with their solution including standardization issues, privacy, security and data confidentiality compliance with General Data Protection Regulation (GDPR). Moreover, the paper presents different system design features including application programming interface, data transfer mechanism, authentication and access control mechanism, and system database. It worth mentioning that at the time of writing, development of the EDMON infection and outbreak detection algorithms are already completed and the next phase of the study involves integration of the prototype along with the EDMON algorithms, developing end user visualization mechanism (front end application) and performing a pilot study.

2 STATE-OF-THE-ART SYSTEMS

The widespread use of mobile devices, IoT devices and wearables and sensors have enabled people to collect vast amount of health data through quantified self and lifelogging for the purpose of personal informatics - self-tracking, self-quantifying, and self-surveillance [12, 13]. The amount of self-collected data is growing daily and is becoming apparent that it requires technologies such as cloud computing to process such a huge volume of data. However, collecting health related data to a central cloud brings various challenges including security, privacy, data confidentiality compliance, standardization issues, and other related ethical and motivational issues [12, 14- 18]. The storage of sensitive information like health data in a cloud requires strict compliance to regulations like HIPPA and GDPR along with other related ethical issues. This demands strict mechanism to ensure authentication and authorization, identity and access control, de-identification and anonymization of user data, and encryption [14, 16]. Moreover, it is necessary to make sure the system is secured against any attacks by thoroughly capturing enough provenance information [14]. Furthermore, collecting of data from heterogeneous devices to the cloud brings standardization challenges, which calls to follow a certain standard such as HL7. There are fewer literatures, which describes a cloud (server) based solution in relation to people with diabetes [16, 19-27]. For example, Mougiakakou et al. [23] developed SMARTDIAB, which is a platform to support diabetes monitoring, management and treatment. The platform has relied on packet enciphering with Triple Data Encryption Standard (3DES) algorithm and data hashing by means of secure hashing (SH-1) algorithm to provide data privacy and data integrity [23]. Data exchange is performed based on Extensible markup language (XML) messages using the Simple Object Access Protocol (SOAP). A secured communication between remote server is ensured relying on IPSec on the network layer, secure socket layer (SSL) on the transport layer, and Hypertext Transfer Protocol Secure (HTTPS) for secure data transmission. Role-based access control along with an application layer firewall– intrusion detection system (IDS) is used to secure the central database [23]. Standardization is ensured following Health Level 7 (HL7) standard when designing the central database and also making the database ICD10 compliant [23]. Moreover, Al-Taee et al. [22] developed a platform to support self-management of diabetes based on internet of things (IoT). A communication between clients and remote server is secured using HTTPS protocol [22]. A Model- View-Control (MVC) pattern is adopted to design the application that handles users' related functionality considering the security and scalability aspects of the system. The central databased is based on SQLite and PostgreSQL database instance to store patients' data and other system related information [22]. Furthermore, Huzooree et al. [16] developed a platform to support remote real-time monitoring of people with diabetes based on a wireless body area network (WBAN). The proposed system relies on Bluetooth and ZigBee communication protocols to transmit the sensor data to the mobile phone. Further, the communication between the mobile phone and access point is based on IEEE 802.11 /Wi-Fi/GPRS [16].

3 MATERIALS AND METHODS

The prototype of the patient module and computing module (cloud backend solution) were developed using various tools, libraries, software and programming language. Both of these prototypes were developed bearing in mind the functionality required by each other during integration. The patient module was developed using TypeScript v3.4.5 [28], React-native v0.59.5 [29], and SQLite based on agile software development with an incremental approach. The prototype was run and tested using an Android-emulator running Android version 9.0(pie). The minimum functional and non-functional requirement were derived directly from the existing diabetes mobile apps, especially Diabetes diary [30] and mySugr app [31]. However, a more advanced functionalities required by a disease surveillance system like the EDMON system were integrated.

Figure 2 A prototype for EDMON patient module [32].

The development of the backend cloud solution relied on different service tools such as Vertelo [33] and Draw. Io [34], which were used to visualize the database and to design and illustrate figures respectively. Moreover, Golang [35], a programming language designed by Google, and Python programming languages were used. Apart from this, libraries and software were used such as Gin library [36], a framework used for HTTP requests and debugging, Postgres [37], an open source database used as the systems database, Gin-jwt [38], used as a middleware to handle JSON web tokens, and Openssl [39], which is used to create a certificate and public key for HTTPS requests. The performance of the cloud backend solution was evaluated using data from the Ohio T1 Diabetes datasets [40]. The Ohio T1 Diabetes datasets contains real datasets from six individuals with type 1 diabetes. It incorporates blood glucose, insulin, carbohydrate, physical exercise, sleeping time, heart rate, and other important physiological measurements. Performance evaluation was carried out for various features and functionality including the average time of execution for a user, average time of concurrent requests to the system, profiling in Golang, and system throughput**.** Moreover, the computational performance of the system was tested under the execution of the EDMON infection detection and cluster detection algorithm, assuming fetching and computing a very large dataset at a time from so many participants. The average performance of the system was calculated by running the task for a certain number of times. Different tasks such as creating user, login with user credentials, storing medical records with location and timestamp, fetching and deleting all user medical records were used to test and evaluate the system performance. Average time of execution for a user was evaluated to determine the average time it takes to perform a given task while the server is under load. Average time of concurrent requests to the system was evaluated to determine the time it takes for a given concurrent requests from a given number of concurrent clients to the system. The throughput of the system was evaluated through spamming the server with requests from the Apache benchmark [41]. This is mainly performed to evaluate the performance of the server under heavy load and also to showcase the maximum number of requests per second the server can handle along with the maximum number of concurrent client requests it can handle.

4 THE PATIENT MODULE

The patient module has similar functionality with the existing well-known diabetes apps, such as Diabetes Diary and mySugr, as shown in the **Figure 2**. It can record key diabetes parameters such as blood glucose levels, insulin injection (both bolus and basal units separately), diet information, and physical activity load. However, it is designed to include additional functionality required by disease surveillance system like EDMON. The added functionality was capability to estimate and record the user geographical location along with the user log, recording weather information, and record events of infection and medication taken. Moreover, the prototype has a feature that enables to push newly created logs with added metadata to a remote EDMON private cloud, giving the user a freedom to pull out of the system at any time. Upon new event, the client can connect to the EDMON cloud, encrypt and transmit the added metadata along with the device GPS position and weather information. The user positional coordinates are collected through the Android API and requires post-processing in order to convert fine accuracy position to a course accuracy position.

5 THE CLOUD BACKEND SOLUTION

5.1 Functional requirements

The main functional requirements of the cloud backend solution is selected based on the necessary requirement for a digital infectious diseases outbreak detection system, such as EDMON, and include the following [42]:

- It should provide an open API so that an API call can be made to the system.
- A new participant could join the system at any time by creating a new account on the system.
- It should store the user’s self-recorded data from the self-monitoring diabetes mobile app (patient module).
- It should store the user’s location upon new data registrations.
- It should be able to fetch the individual user data and run the EDMON algorithm on each individual user’s data at each hour of the day or at the end of each day to look for any infected individuals and mark them as sick.
- It should protect user privacy and confidentiality of the data by providing authentication and access control mechanisms for user’s data.

- It should secure user data by encrypting all the data sent back and forth between the user and the system.
- Furthermore, it should let the user store small videos of medical recordings, which can be notes, and other important things the user wants to revise at later time.

5.2 System component design

Database design

To handle the stable storage of the medical data, a Postgres database is used. The database is designed focusing on being as general as possible to support later development of applications and exporting data from other applications into this system. The main goal is to support all types of medical data, while safely storing sensitive user information. All database access is done on individual user's credentials, which is based on JWT access control. The database incorporates three tables; User, Medical records, and locations.

User

User of the system are participants, who has type 1 diabetes. Each user is represented by a unique key identifier, and each user table contain many fields such as username, password, token, location, medical records, and health status. A user sets a password and username upon signing up a new account. The password is stored in the database and used for comparison purpose when the user tries to log in to the system or to fetch the JSON web token. The passwords are hashed and salted so as to ensure that the user security, privacy and data confidentiality are preserved at all times. As a result, whenever user tries to login to the system, their passwords are compared to the hashed and salted version stored in the database. For each user, once an account created a MedRecs ID is assigned, which is used to store and fetch user data upon request. Moreover, a field containing the health status of the individual is also assigned, which indicates the hourly and daily infection status of the individual.

Medical Record

A medical record in the system is a recording of health-related data at a given time. It is defined by its LOINC code, which has been called HL7 in the database. Each recording is unique but has a foreign key identifier to the user. The value of the recording is an integer which could be the heart rate per minute, blood glucose level, insulin injection, food intake or stress level recorded in scale from one to six. Upon registering data in the database, the data is tagged with a timestamp by the built-in now() functionality of Postgres. This time value is used to categorize the temporal aspect of the data and is required by the EDMON system to pinpoint the timing when an outbreak is detected in the surveillance region. For categorizing the spatial aspects of the data, each recording is assigned with geographical location coordinates used by the EDMON system to detect a cluster of infected people with type 1 diabetes on a certain geographical region. Each medical record also contains fields that let the user record and save either a picture or a short video showing how their condition and feeling at the time is.

Location

In the database, both the user and medical record table contain a reference to a location. The location holds information regarding user geographical location coordinates expressed in terms of latitude and longitude. The location table is separated from the user and medical record table for a security reason and also to give the possibility of changing the location format, if necessary, without affecting the other. Each user can have two different locations; static and dynamic location. The static location is defined as the user hometown, home address, and postal codes. The dynamic location specifies the approximate current location of the user upon registering a data as compared to the static location. These locations information are used by the EDMON algorithms to detect clusters of people based on geographical region. Moreover, this geo-information can also be used for other purposes such as to locate a patient with e.g. hypoglycemia (critical low blood glucose levels) and needs help, through alerting mechanism in the system.

Relations between the tables

The foreign keys in the database defines how each table interact with each other, as shown in the **Figure 3**. For example, a user has two different foreign keys to the location table; a static and dynamic location, which set the user geographical position at all times. However, even if the user has two different locations, he/she can only have one medical record ID. The relationship between a user and a medical record is a one-to-many relationship, which meant that a user can have many medical records. In turn, the medical record has a foreign key to a location, which indicates where the user is located at the time of recording the data.

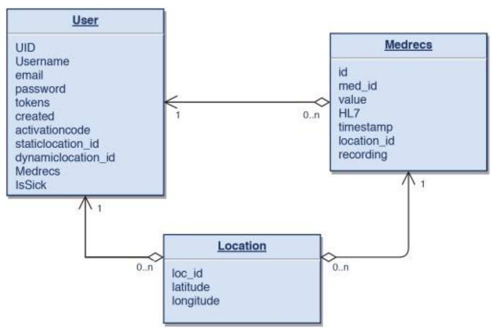

Figure 3 Database design.

5.3 Authentication and access control

Authentication and access control in the system is handled by JSON web tokens. These tokens are session-based, which mean that as long as they are refreshed before the expiration by the application, a user does not need to log in using their username and password. When joining the system, a user is expected to sign up and create an account by providing a username, email and password, which is used for authentication purposes. However, once the account setup is complete, the user email is automatically removed and deleted as part of the de-identification process and a unique key identifier is set in place. The token provided by the server is the users' way of authenticating all other HTTP requests.

5.4 Application programming interface

The application programming interface (API) is used to create an interface and connect the cloud backend solution with the necessary EDMON applications. Regarding the patient module, the API is designed giving a due consideration for the users to have a complete control of their stored data, where the user anytime can leave the system by deleting all the records. In connection with the data transfer design feature, the API is built with a RESTful design [43], which incorporates four different possible request to either fetch, delete and store data. These includes GET, PUT, POST and DELETE requests and the data sent from the user to the system is in JSON-format. For example, the transfer of data to the system is executed by the POST request, which contains JWT body with the JSON formatted data, as shown in the **Figure 4**. The authorization token should also be included in the header so that the user is given with the right privilege to edit the data. Upon completion of the data transfer, ID is created in the database to identify the object and the system acknowledges the user.

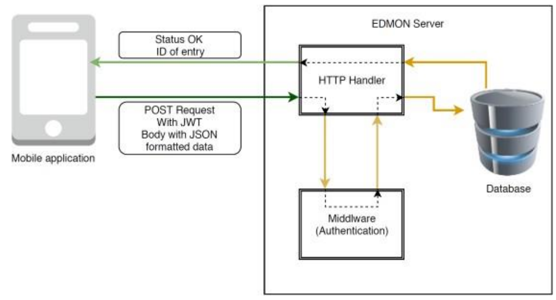

Figure 4 Example of a POST request to the system.

5.5 The overall system design

Generally, the designed cloud backend solution incorporates an API in which a user's patient module can initiate a send request to the system upon new data registration, a middleware that can authenticate the user, and a stable storage of database, as shown in the **Figure 5**.

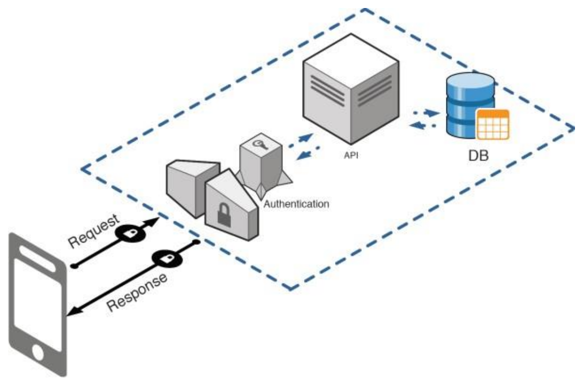

Figure 5 The overall system design.

Each request from a user goes to the systems through the authentication service which validates the token that is sent with the request. If a user has the correct authorization token the system will perform the requested task and send it back to the requester, as can be seen in Figure 6. On the other hand, if the user cannot provide the correct authorization token, the middleware denies further access.

6 SYSTEM ILLUSTRATION

The interaction between different components of the designed system is illustrated, with an example, as shown in the **Figure 7**. The depicted example shows the component interaction, when a patient module application access request passes the authentication phase and initiates a request to fetch medical records stored in the system database. As can be seen, when a user places a request, it is first directed to the middleware for authentication through the HTTP request.

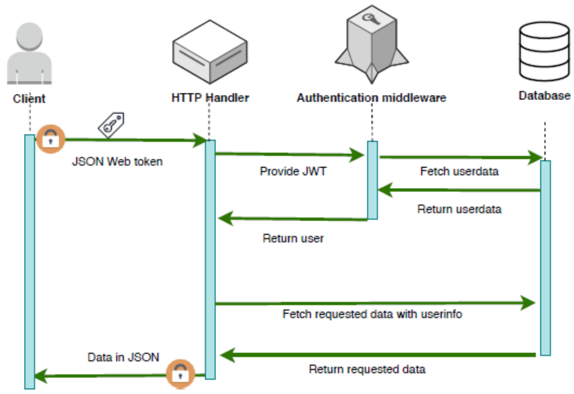

Figure 6 An example of granted access.

Here, the user access to the system is decided whether the user has provided the correct access token or not. If the user has provided the correct access token, then he/she will be granted access and the request goes through the user part of the API and directed to the model for the medical records. Then the requested data is returned to the user API, where it is parsed with JSON and sent to the requester.

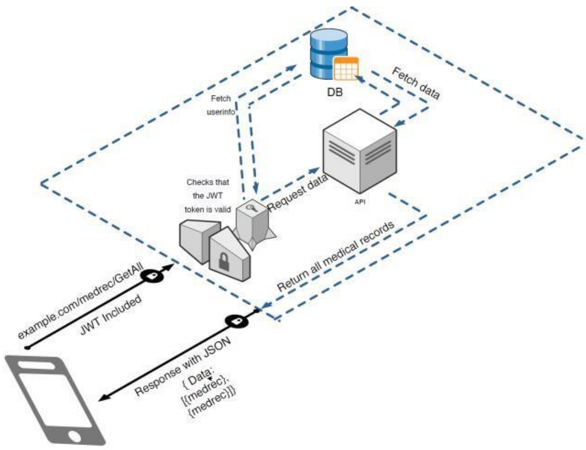

Figure 7 Example of a request to the system.

7 RESULT

The performance test, which is based on real datasets from OhioT1DM dataset, indicates that the server can handles up to 1000 requests per second depending on the request type and one can conclude that the requests per second are sufficient for a real-time system like EDMON with a lot of participants. Moreover, the test conducted to evaluate the server performance during execution of EDMON detection algorithms performs well, however, the test reveals that an approach that could create a pool of workers in the form of Goroutines, that fetches the users' data continuously and spreads the workload over an hour is preferred, instead of having a heavy algorithm that run once per hour. Furthermore, the test conducted to evaluate the performance of the server in regard to execution time under a fair amount of traffic reveals that a user can upload 600 medical records along 300 different locations almost under seven seconds. Apart from these performance test, a stress test is conducted to measure the liability and maximal throughput of the server using Apache benchmark to create a small denial of service attacks. Almost close 7100 concurrent clients sent 10000 requests, which averaged around 800 requests per second. The test indicates that the server can withstand a high number of requests, only around 2% of the request failed.

8 DISCUSSION AND CONCLUSION

We have presented a prototype for a patient and computing (backend cloud solution) module, which are components of EDMON system towards infection monitoring in people with type 1 diabetes and detecting infection outbreaks based on a spatio-temporal cluster detection. Generally, the purpose of EDMON system is to collect diabetes related self-recorded data from individual with type 1 diabetes to a cloud and perform computation to detect infection incidences on an individual level and further use this information to spatio-temporal clusters for a possible confirmation of outbreak incidences. The system components are designed with due consideration of vital features and implemented in accordance with the state-of-the-art techniques of ensuring security, privacy, data confidentiality in compliance with GDPR along with standardization approaches. To secure the data transferred from the individual patient module, encryption, hashing and salting, and secure authentication is used on the server. In this regard, HTTPS is used to encrypt the data sent from a patient module to the server. This provides the security of a man-in-the-middle attack on the server where the certificates provide authentication from the server. Moreover, upon creation of new account user's password are hashed and salted before being added to the database so as to ensure optimal protection. As a result, there are no parts of the system that interacts with the password after a user is created. Furthermore, JSON web tokens are used to ensure the identity of a user upon authentication. This gives the user minimal interaction with the authentication part of a patient module application, where the application itself can refresh the token if used frequently.

As far as our knowledge is concerned, this is the first system that considers the use of self-recorded data from people with diabetes to detect the incidence of infection outbreaks. We believe such kind of cloud-based system might benefit other similar system built to provide decision support to the patient, diabetes patient monitoring and patient empowerment system, and most importantly provoke further thought in the challenging field of real time digital infectious disease outbreak detection systems.

9 FUTURE WORK

At the time of writing, development of the EDMON individual infection detection and spatio-temporal clustering algorithms are already completed and the next phase of the study involves:

- Integrating both prototypes; patient and computing module.
- Providing functionality such as automatic entry of sensors reading to the patient module.
- Integration of the computing module prototype with the individual infection detection algorithm and spatio-temporal clustering algorithm.
- Developing the end user visualization mechanism (dashboard) based on both a standalone mobile app and a dedicated website.
- Integration of the end user app/website with the computing module to fetch timely infection status of the region under surveillance.

10 REFERENCES

[1] Choi, J., Y. Cho, E. Shim, and H. Woo, Web-based infectious disease surveillance systems and public health perspectives: a systematic review. BMC public health, 2016. 16(1): p. 1238-1238.

[2] Hope, K., *Syndromic surveillance: is it a useful tool for local outbreak detection?* Journal of Epidemiology & Community Health, 2006. **60**(5): p. 374-374.

[3] Woldaregay, A.Z., E. Årsand, S. Walderhaug, D. Albers, L. Mamykina, T. Botsis, and G. Hartvigsen, *Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes.* Artificial Intelligence in Medicine, 2019.

[4] Ogurtsova, K., J.D. da Rocha Fernandes, Y. Huang, U. Linnenkamp, L. Guariguata, N.H. Cho, D. Cavan, J.E. Shaw, and L.E. Makaroff, *IDF Diabetes Atlas: Global estimates for the prevalence of diabetes for 2015 and 2040.* Diabetes Res Clin Pract, 2017. **128**: p. 40-50.

[5] Botsis, T., A.M. Lai, G. Hripcsak, W. Palmas, J.B. Starren, and G. Hartvigsen, *Proof of concept for the role of glycemic control in the early detection of infections in diabetics.* Health Informatics J, 2012. **18**(1): p. 26-35.

[6] Marcovecchio, M.L. and F. Chiarelli, *The effects of acute and chronic stress on diabetes control.* Sci Signal, 2012. **5**(247): p. pt10.

[7] Bosarge, P.L. and J.D. Kerby, *Stress-induced Hyperglycemia.* Advances in Surgery, 2013. **47**(1): p. 287-297.

[8] Kajbaf, F., M. Mojtahedzadeh, and M. Abdollahi, *Mechanisms underlying stress-induced hyperglycemia in critically ill patients.* Therapy, 2007. **4**(1): p. 97-106.

[9] Rayfield, E.J., M.J. Ault, G.T. Keusch, M.J. Brothers, C. Nechemias, and H. Smith, *Infection and diabetes: The case for glucose control.* The American Journal of Medicine, 1982. **72**(3): p. 439-450.

[10] Woldaregay, A.Z., E. Årsand, T. Botsis, and G. Hartvigsen, *An Early Infectious Disease Outbreak Detection Mechanism Based on Self-Recorded Data from People with Diabetes.* Studies in health

technology and informatics, 2017. **245**: p. 619-623.

[11] Yeng, P.K., A.Z. Woldaregay, T. Solvoll, and G. Hartvigsen. A systematic review of cluster detection mechanisms in syndromic surveillance: Towards developing a framework of cluster detection mechanisms for EDMON system. in Proceedings from The 16th Scandinavian Conference on Health Informatics 2018, Aalborg, Denmark August 28–29, 2018. 2018. Linköping University Electronic Press.

[12] Rodriguez-Rodriguez, I., J.V. Rodriguez, and M.A. Zamora-Izquierdo, *Variables to Be Monitored via Biomedical Sensors for Complete Type 1 Diabetes Mellitus Management: An Extension of the "On-Board" Concept.* J Diabetes Res, 2018. **2018**: p. 4826984.

[13] Rodríguez-Rodríguez, I., M.-Á. Zamora- Izquierdo, and J.-V. Rodríguez, *Towards an ICT- Based Platform for Type 1 Diabetes Mellitus Management.* Applied Sciences, 2018. **8**(4): p. 511.

[14] George, M., A.M. Chacko, S.K. Kurien, and N. Ali, Diabetes care in cloud - research challenges, in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. 2019, ACM: Limassol, Cyprus. p. 160-162.

[15] Azeez, N.A. and C.V. der Vyver, Security and privacy issues in e-health cloud-based system: A comprehensive content analysis. Egyptian Informatics Journal, 2018.

[16] Huzooree, G., K.K. Khedo, and N. Joonas. *Wireless Body Area Network System Architecture for Real-Time Diabetes Monitoring*. 2017. Cham: Springer International Publishing.

[17] Woldaregay, A.Z., E. Årsand, A. Giordanengo, D. Albers, L. Mamykina, T. Botsis, and G. Hartvigsen. EDMON-A Wireless Communication Platform for a Real-Time Infectious Disease Outbreak De-tection System Using Self-Recorded Data from People with Type 1 Diabetes. in Proceedings from The 15th Scandinavian Conference on Health Informatics 2017 Kristiansand, Norway, August 29–30, 2017. 2018. Linköping University Electronic Press.

[18] Woldaregay, A., D. Issom, A. Henriksen, H. Marttila, M. Mikalsen, G. Pfuhl, K. Sato, C. Lovis, and G. Hartvigsen, *Motivational Factors for User Engagement with mHealth Apps.* Studies in health technology and informatics, 2018. **249**: p. 151.

[19] Mamykina, L., E. Mynatt, P. Davidson, and D. Greenblatt, MAHI: investigation of social scaffolding for reflective thinking in diabetes management, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2008, ACM: Florence, Italy. p. 477-486.

[20] Alhazbi, S., M. Alkhateeb, A. Abdi, A. Janahi, and G. Daradkeh. *Mobile application for diabetes control in Qatar*. in *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*.2012.

[21] Luo, Y., C. Ling, J. Schuurman, and R. Petrella. GlucoGuide: An Intelligent Type-2 Diabetes Solution Using Data Mining and Mobile Computing. in 2014 IEEE International Conference on Data Mining Workshop. 2014.

[22] Al-Taee, M.A., W. Al-Nuaimy, A. Al-Ataby, Z.J. Muhsin, and S.N. Abood. Mobile health platform for diabetes management based on the Internet-of- Things. in 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). 2015.

[23] Mougiakakou, S.G., C.S. Bartsocas, E. Bozas, N. Chaniotakis, D. Iliopoulou, I. Kouris, S. Pavlopoulos, A. Prountzou, M. Skevofilakas, A. Tsoukalis, K. Varotsis, A. Vazeou, K. Zarkogianni, and K.S. Nikita, *SMARTDIAB: A Communication and Information Technology Approach for the Intelligent Monitoring, Management and Follow- up of Type 1 Diabetes Patients.* IEEE Transactions on Information Technology in Biomedicine, 2010. **14**(3): p. 622-633.

[24] Baskaran, V., F. Prescod, and L. Dong, *A Smartphone-Based Cloud Computing Tool for Managing Type 1 Diabetes in Ontarians.* Canadian Journal of Diabetes, 2015. **39**(3): p. 200- 203.

[25] Poorejbari, S., H. Vahdat-Nejad, and W. Mansoor, *Diabetes Patients Monitoring by Cloud Computing.* 2017: p. 99-116.

[26] Salama, M. and A. Shawish. A novel mobile-cloud based healthcare framework for diabetes. in Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies-Volume 5. 2014. SCITEPRESS- Science and Technology Publications, Lda.

[27] Hsu, W.C., K.H.K. Lau, R. Huang, S. Ghiloni, H. Le, S. Gilroy, M. Abrahamson, and J. Moore, Utilization of a Cloud-Based Diabetes Management Program for Insulin Initiation and Titration Enables Collaborative Decision Making Between Healthcare Providers and Patients. Diabetes technology & therapeutics, 2016. **18**(2): p. 59-67.

[28] *Typescript - javascript that scales*. [cited 2019 9]; Available from: https://www.typescriptlang.org/.

[29] *React native github*. [cited 2019 9]; Available from: https://github.com/facebook/react-native.

[30] Årsand, E., M. Muzny, M. Bradway, J. Muzik, and G. Hartvigsen, *Performance of the First Combined Smartwatch and Smartphone Diabetes Diary Application Study.* Journal of Diabetes Science and Technology, 2015. **9**(3): p. 556-563.

[31] *mySugr App*. [cited 2019 8]; Available from: https://mysugr.com/.

[32] Johnsen, A., H.B. Borhaug, H.B. Borhaug, J.A. Johansen, J. Blomkvist, and K.A. Johansen, *A prototype of Diabetes mobile health app for EDMON System*. 2019, Institute of Technology: University of Tromsø.

[33] *Vertelo, design of the database.* [cited 2019 9]; Available from: https://www.vertabelo.com/.

[34] *Draw.io*. [cited 2019 8]; Available from: https://www.draw.io/.

[35] *A Tour of Go*. [cited 2019 8]; Available from: https://tour.golang.org/concurrency/2.

[36] *Gin HTTP library*. [cited 2019 8]; Available from: https://github.com/gin-gonic/gin.

[37] *PostgreSQL: The World's Most Advanced Open Source Relational Database*. [cited 2019 8]; Available from: https://www.postgresql.org/.

[38] *JWT Middleware for Gin framework*. [cited 2019 8]; Available from: https://github.com/appleboy/gin-jwt.

[39] *OpenSSL*. [cited 2019 9]; Available from: https://www.openssl.org/.

[40] Marling, C. and R.C. Bunescu. *The OhioT1DM Dataset For Blood Glucose Level Prediction*.

[41] *Apache HTTP server benchmarking tool*. [cited 2019 8]; Available from: https://httpd.apache.org/docs/2.4/programs/ab.html.

[42] Loum, L. and D.A. Mokoena, *The Role of Cloud-Based mHealth Disease Surveillance System in Regional Integration: A Case of the Ebola Crisis in ECOWAS, in Innovation, Regional Integration, and Development in Africa: Rethinking Theories, Institutions, and Policies*, S.O. Oloruntoba and M. Muchie, Editors. 2019, Springer International Publishing: Cham. p. 171-190.

[43] *What is REST?* [cited 2019 8]; Available from: https://restfulapi.net/.