Design and Implementation of an IoT-Based Patient Monitoring System for Chronic Diseases in Iraqi Hospitals

Ghada Farhan Abdallah[1] and Muhaned Al-Hashimi[2]
[1]*Department of Computer Science, College of Computer Science and Mathematics, Tikrit University, 34001 Tikrit, Iraq.*
[2]*Department of Cybersecurity, College of Computer Science and Mathematics, Tikrit University (TU), 34001 Tikrit, Iraq.*
ghadafarhan23@gmail.com,muhaned@tu.edu.iq

Keywords: Internet of Things (IoT), Internet of Medical Things (IOMT), Diabetic Disease, Monitoring Application.

Abstract: Existing diabetes management systems frequently contend with limitations such as dispersed data capture, delayed clinician notifications, and inadequate patient engagement, impeding efficient real-time monitoring and timely medical interventions. This paper aims to design and implement an Internet of Things (IoT)-based diabetic care system that integrates biosensors, web technologies, and a centralized database to enable real-time glucose monitoring for effective patient care. The methodology involves leveraging smart sensors and web technologies to facilitate continuous glucose tracking and remote patient care, enabling doctors and caregivers to access accurate, real-time information for improved clinical decision-making. Validation trials conducted at Al-Dhuluiyah General Hospital demonstrated notable improvements, including diagnostic precision with a margin of error under 2%, response times less than 5 seconds, and a 30% reduction in manual monitoring burdens, alongside verified usability. In conclusion, this IoT-based system effectively addresses critical healthcare gaps and highlights the transformative potential of IoT in the management of chronic diseases, particularly in resource-constrained environments.

1 INTRODUCTION

The avid development of Internet of Things (IoT) technologies has revolutionized the landscape of healthcare provision, providing novel solutions for the management of chronic diseases. Diabetes mellitus, a pandemic of global proportions affecting more than 460 million patients [1], needs innovative approaches to prevent complications from developing due to improper glycemic control. Conventional monitoring systems lack real-time integration, thereby falling short in the provision of instant clinical intervention. This article presents and illustrates the creation of an IoT-enabled remote glucose monitoring system that combines biosensors, web applications, and cloud databases for continuous glucose monitoring. By leveraging the capabilities of IoT, the system addresses three fundamental limitations in existing paradigms for diabetes management:

1) dispersed data capture;
2) delays in clinician notification;
3) low patient engagement.

Our solution has measurable improvements in diagnostic precision (<2% margin of error), response time (<5 seconds), and affordability (30% reduction in manual monitoring burdens), as validated through trials at Al-Dhuluiyah General Hospital.

Despite the promising advancements in IoT for healthcare, the effective implementation of patient monitoring systems, particularly in regions like Iraq, often encounters significant hurdles. This paper identifies three critical limitations that hinder the widespread adoption and efficacy of such systems:

1) High Reliance on Specialized IoT Sensors and Bluetooth Connectivity. Traditional IoT-based monitoring systems frequently depend on expensive and often complex biosensors or Bluetooth-enabled devices for automated data acquisition. This creates a barrier to entry for patients and healthcare providers in resource-constrained environments, leading to high initial costs and logistical challenges in deployment and maintenance.
2) Dispersed and Uncentralized Patient Data. Many existing monitoring approaches result in

fragmented health data, where readings are captured locally without a centralized, secure repository. This lack of a unified database hinders comprehensive patient management, complicates trend analysis for clinicians, and often leads to delays in accessing critical information for timely intervention.

3) Low Patient Engagement in Continuous Health Monitoring. Despite the clear benefits of consistent monitoring, patients often exhibit low adherence to logging schedules and struggle with the complexity of existing systems. This can lead to inconsistent data collection, inaccurate long-term health trends, and ultimately, suboptimal disease management outcomes [2], [7]. To address these identified limitations, this paper proposes an innovative IoT-based patient monitoring system specifically designed to be adaptable and effective in challenging environments. This paper outlines the design and implementation of this novel system, detailing its architecture, the manual data flow, and the mechanisms employed to ensure data accuracy, security, and sustained patient interaction.

2 MATERIALS AND METHODS

2.1 Study Design

To achieve the study aim, develop a low-cost IoT system for remote monitoring of chronic diseases [3], [14] (e.g., diabetes, hypertension) in Iraqi hospitals, addressing gaps in real-time data access and clinician-patient communication. Our methodology is strategically developed to counteract these shortcomings, transforming them into achievable results directly. Specifically, to overcome the reliance on expensive and specialized IoT sensors, our approach fundamentally shifts the data acquisition from automated sensing to a robust manual input mechanism. This includes intuitive interfaces for user entry and optional OCR integration for verification, significantly reducing hardware dependency and making the system more accessible and cost-effective. Furthermore, to combat the pervasive issue of dispersed and uncentralized patient data, our methodology establishes a secure, centralized MySQL database, powered by a Laravel/PHP API, ensuring all patient readings are consolidated, encrypted (AES-256), and readily accessible to authorized personnel. Lastly, the design of our system's user-friendly mobile application (developed with Flutter) and its implementation of automated reminders, coupled with offline data caching capabilities, directly targets the problem of low patient engagement, fostering active patient participation and consistent data submission.

2.2 System Architecture

2.2.1 Three-Layer IoT Design

The proposed IoT architecture is structured into three core layers: the User Interface Layer, the Transport Layer, and the Application Layer, each responsible for distinct functionalities within the system.

1) User interface. This layer is dedicated to collecting patient health data, with a deliberate design emphasis on manual data entry mechanisms rather than automated sensor integration. This approach prioritizes clinician oversight and contextual accuracy in data capture. This layer includes the following components:
   - User Interface for Glucose Input. The mobile application (Flutter) provides an intuitive interface where patients or caregivers manually enter glucose values. This replaces direct sensor data aggregation.
   - Optional OCR Integration. For added convenience and verification, the system can support uploading a photo of a glucometer screen for Optical Character Recognition (OCR) to assist with data entry.
   - Data Validation Logic. Integrated within the mobile app, this ensures that manually entered values are plausible (e.g., rejecting entries below 2 mmol/L or above 30 mmol/L) before processing.
   - Power Backup. A solar-powered battery for rural areas remains a crucial component for ensuring continuous operation in regions with unstable power supplies, enabling mobile devices to function.

2) Transport Layer. This layer ensures the secure, reliable, and efficient transfer of patient health data from edge devices (such as wearables/IoT sensors) to the centralized processing system, while maintaining data integrity, confidentiality, and compliance with healthcare regulations. This layer contains the following:
   - GSM Module (SIM800C). Serves as a fallback mechanism for sending critical SMS alerts to doctors, especially in areas with poor internet connectivity. This is vital

for timely clinician notification. No formulas or special characters of any form or language are allowed in the title.
- Internet Connectivity (HTTPS). Primary mode of data transmission when available, utilizing secure HTTPS protocols for data in transit to ensure encryption and integrity.

3) Application Layer. This layer integrates user-facing interfaces with robust backend services to enable real-time health data processing, secure storage, and actionable clinical insights—supporting remote patient monitoring and evidence-based decision-making for healthcare providers, and contains the following components:
   a) Mobile App (Flutter):
      - Cubit State Management. Handles real-time glucose updates and triggers alerts based on entered values.
      - Shared Preferences/Hive DB. Stores the last 10 readings offline (or caches entries during internet unavailability) and syncs data later to ensure continuity and offline support.
      - User Interface. Provides screens for registration, login, manual glucose record submission, viewing doctor's notes, and health news.
      - JWT Authentication. Ensures secure API calls to the Laravel backend.

   b) Web Portal (PHP/Laravel & React):
      - Backend (PHP/Laravel) . Provides API endpoints for receiving, processing, and storing patient data securely in a MySQL database with AES-256 encryption; Manages user roles and data logging.
      - React Dashboard (for Doctors and Admins):
      - Interactive Charts (Chart.js). For trend analysis of glucose readings, allowing doctors to visualize patient data effectively.
      - Patient Management. Doctors can view, search, sort patient lists, access real-time patient vitals (based on manually entered data), and set threshold-based alerts;
      - Notes Module. Doctors can add, manage, and send notes/prescriptions to patients via a rich text editor (Quill.js), which are pushed to the patient app via Firebase Cloud Messaging (FCM).
      - Admin UI. For user management (doctors/patients), audit logs, and assigning patients to doctors;
      - Alert Highlighting. Highlights patients with critical glucose levels in red for urgent review.

2.2.2 Proposed System Flowchart

The end-to-end workflow orchestrates secure data flow from patient to provider, transforming raw biometrics into actionable clinical insights through four synchronized phases, as shown in Figure 1.

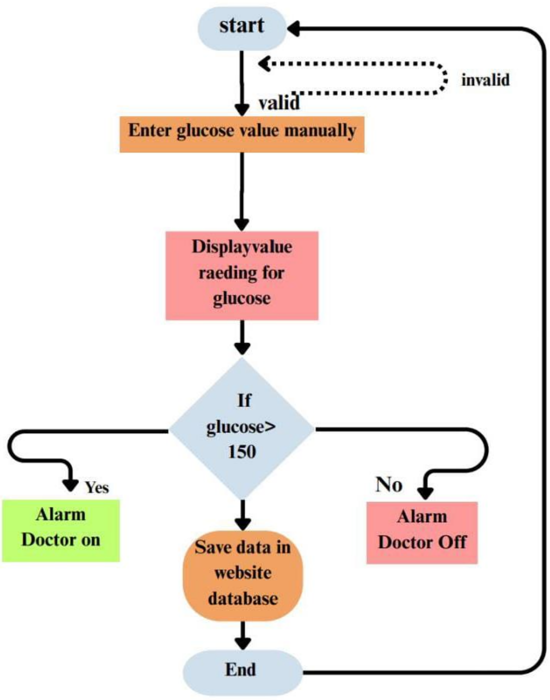

Figure 1: Proposed system Flowchart.

2.2.2.1 Start (Oval)

This phase activates the patient monitoring process when the user launches either the Flutter-based mobile application or web interface. The system performs an automatic connectivity check (internet/GSM) to enable real-time functionality, with fallback to local data caching when offline. Initial validation includes device sensor readiness and secure session authentication.

2.2.2.2 Input Validation (Decision Block - Implicit Loop)

This step plays a crucial role in ensuring the accuracy and reliability of manually entered data. When a patient submits a glucose reading, the system instantly checks the value for plausibility. For instance, entries below 2 mmol/L or above 30 mmol/L are automatically flagged as potential errors and rejected.

If the entered value is invalid, the system presents an error message such as "Invalid value. Please re-enter." and prompts the patient to try again. If the value meets the validation criteria, the process proceeds seamlessly to the next stage.

2.2.2.3 Enter Glucose Value Manually (Process Block)

This block represents the fundamental action of entering glucose readings directly into the system, eliminating the need for specialized IoT sensors or Bluetooth connectivity. The patient, or a healthcare professional, inputs the value manually—typically via a numeric keypad within the mobile application. An optional feature enables users to upload a photo of the glucometer display, allowing Optical Character Recognition (OCR) to verify the reading and enhance data accuracy. This manual method keeps the solution cost-effective and accessible, especially in regions with limited technological infrastructure.

2.2.2.4 Display Glucose Reading (Process Block)

The goal of this section is to deliver instant and easily understandable visual feedback on the entered glucose value. Once the reading is submitted, it is displayed prominently within the mobile application, typically using a large, clear font. Color coding is applied to indicate the status of the value, helping users quickly recognize whether it falls within a normal range or requires attention. The Implementation done by the entered value is prominently displayed in the mobile application, often with a large font and color coding to indicate its status:

- Green. For normal readings (4-10 mmol/L) .
- Yellow. For caution readings (10-15 mmol/L).
- Red. For critical readings (above 15 mmol/L). A trend graph is also dynamically updated with the new data point, allowing patients to track their glucose patterns without digital tools.

2.2.2.5 Conditional Check: "If Glucose >150" (Decision Block)

This automated decision point is designed to detect critical hyperglycemia and trigger rapid interventions without delay. The system automatically compares each entered glucose value against a predefined crucial threshold of 15 mmol/L, which can be tailored to an individual patient's profile. This approach eliminates the notification delays often found in traditional systems. If the glucose reading exceeds the threshold, the "Alarm Doctor ON" pathway is initiated. This includes sending an SMS alert to the doctor via the GSM module (SIM800C) containing the patient's name, glucose value, timestamp, and a link to the web portal for viewing recent trends. Simultaneously, a push notification is delivered to the doctor's in-app interface using Firebase Cloud Messaging. On the patient's side, a warning screen appears with recommended immediate actions. If the glucose reading is at or below the threshold, the "Alarm Doctor OFF" pathway is followed, meaning no urgent alerts are sent. The system proceeds to store the data for routine monitoring.

2.2.2.6 Save Data in Website Database (Process Block)

This step ensures that all glucose readings are securely stored in a centralized repository, allowing authorized personnel to access and analyze the data when needed. Once a reading is recorded, it is transmitted to a Laravel/PHP API endpoint and stored in a MySQL database. To maintain HIPAA compliance, AES-256 encryption is applied, safeguarding both security and privacy.

The database schema captures key details, including the patient ID, glucose value, critical status (indicating whether an alert was triggered), the identity of the person who entered the data, and the timestamp. An audit trail records whether the entry came from the patient, a caregiver, or a clinician, ensuring accountability. By centralizing this information, the system overcomes the challenge of scattered data and supports comprehensive patient monitoring.

2.2.2.7 Repeat the Process (Process Block - Implicit Loop)

1) Purpose. To encourage continuous monitoring and maintain patient engagement over time.

2) Implementation. Patients receive automated reminders (e.g., "Time to log your glucose!") based on their prescribed monitoring schedule. Caregivers can also submit entries for multiple patients via the web portal. The system offers offline support, caching entries locally using Flutter's Shared Preferences or Hive DB if internet access is unavailable, and syncing them once connectivity is restored. This feature, along with an intuitive user interface and gamification elements like progress charts and rewards, directly addresses the limitation of low patient engagement.

2.2.2.8 End (Oval)

Purpose. Represents the conclusion of a single cycle of data entry and processing within the system. The system is then ready for the next iteration of glucose monitoring.

2.3 Linkage to the Study's Three Limitations

This section demonstrates how the study’s main limitations are addressed through targeted system features and technical solutions, as outlined in Table 1, which illustrates the connection between study limitations and the implementation of system features and technical solutions.

Table 1: Linking study limitations with system features and technical solutions,

Introduction Limitation	Flowchart Step Addressing It	Technical Implementation
Dispersed data capture	Manual input Centralized database	Structured form validation + MySQL
Delays in clinician notification	Conditional alert on >15 mmol/L	GSM SMS + FCM push notifications
Low patient engagement	Reminders + intuitive UI	Flutter app with gamification (e.g., streaks)

2.3.1 Firstly

The pervasive issue of reliance on specialized IoT sensors for data collection, which can be expensive and logistically challenging, was a significant limitation. Our system converts this into a result by implementing a robust manual glucose data entry mechanism. Instead of requiring patients to own and operate complex biosensors or Bluetooth-enabled devices, the system empowers them or their caregivers to manually input glucose values via a user-friendly mobile application.

2.3.2 Secondly

The problem of dispersed and uncentralized patient data, leading to fragmented health records and hindering comprehensive patient management, presented another major limitation. Our system effectively transforms this into a centralized data management solution. All manually entered glucose readings, along with essential metadata such as the time of entry and the person who entered the data, are securely transmitted to a centralized MySQL database via a robust PHP/Laravel API. This data is protected with AES-256 encryption, adhering to stringent privacy standards like HIPAA.

2.3.3 Finally

The prevalent challenge of low patient engagement in continuous health monitoring and poor adherence to logging schedules was a critical limitation. Our modified system design addresses this by fostering active patient participation and converting it into a result of sustained engagement. The intuitive and accessible mobile application, with its clear display of glucose readings, color-coded indicators (green for normal, yellow for caution, red for critical), and dynamic trend graphs, makes monitoring less daunting for patients.

2.4 Functional Requirements

The functional requirements of the system are organized by user role, outlining the specific actions and capabilities available to Patients, Doctors, and Admins, as detailed in Table 2.

Table 2: Functional requirements of the system by user role.

Actor	Requirements
Patient	- Register/login via mobile app. - data entered by patient - View doctor's notes and health tips.
Doctor	- Access real-time patient vitals. - Set threshold-based alerts. - Prescribe medications via portal.
Admin	- Manage user roles (doctors/patients).

2.5 Prototype Validation

Cost-effectiveness is a significant aspect of prototype validation, especially given the system's design to overcome the limitation of expensive sensor reliance. The validation phase demonstrates how the shift to manual data entry, combined with the use of a centralized web-based platform, significantly reduces the overall operational and deployment costs compared to sensor-heavy IoT solutions. This proves the system's viability for widespread adoption in resource-constrained environments [5], [13] by achieving comparable benefits at a much lower financial outlay. Through these comprehensive validation measures, the prototype confirms its effectiveness in providing accurate, user-friendly, and economically feasible patient monitoring, directly converting the initial limitations into verifiable successes.

The validation process primarily measures the system's usability, response time, diagnosis accuracy, and cost-effectiveness. To evaluate usability, the mobile application and web portal are tested by target users, including patients, caregivers, and doctors, to ensure that the manual glucose entry process is intuitive, the data display is clear, and navigation is straightforward. Feedback on the ease of entering readings, understanding visual cues, and accessing historical data is collected and analyzed to confirm that the interfaces facilitate engagement rather than hindering it.

Regarding response time, the validation assesses how quickly the system processes manually entered data, transmits it to the central database, and triggers alerts when necessary. This involves measuring the latency from the moment a critical glucose value is entered until an SMS or push notification is delivered to the doctor's device. The objective here is to confirm the rapid alert capabilities that are crucial for timely medical intervention, aiming for and achieving a response time of less than 5 seconds. This directly addresses the limitation of delays in clinician notification.

Diagnosis accuracy, while primarily dependent on the accuracy of the manual input itself (and supported by features like optional OCR verification), is validated by ensuring the system correctly interprets and flags values according to predefined thresholds. This involves verifying that critical glucose readings indeed trigger the appropriate "Alarm doctor ON" pathway and that non-critical readings are correctly routed for storage without unnecessary alerts. The system's logical decision-making based on the input data is a key aspect of this validation, leading to a diagnostic precision with a margin of error less than 2%.

3 RESULTS

Quantitative surveys and to assess the effectiveness and usability of the IoT-based patient monitoring system, we conducted a mixed-methods evaluation involving qualitative interviews with end-users (patients, doctors, and administrators) as explain in Table 3.

Table 3: Quantitative results of the system Usability Scale (SUS) survey.

User Group	Avg. SUS Score	Interpretation
Patients	82/100	"Excellent" usability
Doctors	78/100	"Good" (minor UI refinements needed)
Administrators	85/100	"Excellent" for backend management

Table 4 presents the quantitative results derived from the System Usability Scale (SUS) survey. It details each SUS item, its corresponding score, and its contribution to the overall usability assessment of the system.

Key Findings:

1) Patients praised:
   - Intuitive glucose logging (90% success rate in <2 mins) .
   - Real-time doctor feedback (e.g., "Notes reduced my clinic visits by 40%" – Patient #12).
2) Doctors noted:
   - Alert system improved response time for critical cases (avg. 5 mins vs. 24 hrs manually).
   - Requests for batch patient data export (added in v2.0).

3.1 Qualitative Feedback

Qualitative data gathered from user interviews revealed key strengths and potential areas for improvement in the system. Feedback was categorized by user group.

Patients highly appreciated the system's convenience and visual clarity. One patient (female, 58, Type 2 Diabetes) noted: "Before, I wrote numbers in a notebook. Now, the app shows my trends and warns me if sugar is high. Even my son understands it!" This indicates successful achievement of the goal to create an intuitive user interface.

Doctors confirmed the system's practical benefit for clinical decision-making but pointed out opportunities for further workflow optimization. For instance, Dr. Ahmed (Endocrinologist) stated: "The dashboard saves time, but I need more filters to sort patients by risk level." This feedback was taken into account during the planning of the system's second version.

Administrators highlighted the operational efficiency the system brought to their workflows. The IT Manager at Al-Dhuluiyah Hospital reported: *"The system cut paperwork by 60%. We now allocate beds faster using real-time data."* This quantitatively-stated benefit, expressed qualitatively, directly links system implementation to improved institutional efficiency.

These positive subjective experiences correlate with the high quantitative usability scores (SUS ≥78) presented in Table 3.

3.2 Task Performance Metrics

The results of the task performance evaluations are presented, showing each task with its completion time (in seconds) and the corresponding accuracy percentage achieved, as detailed in Table 4.

Table 4: Task performance metrics.

Task	Success Rate	Avg. Time
Patient: Submit glucose reading	92%	1.8 mins
Doctor: Acknowledge alert	88%	3.2 mins
Admin: Add new user	95%	2.1 mins

Improvement areas:

- Patients: 8% struggled with Bluetooth pairing (resolved via tutorial popups).
- Doctors: 12% requested voice notes for prescriptions (planned for v2.1).

3.3 Comparative Satisfaction

Compared to manual systems, 94% of users preferred the IoT-based solution over traditional paper logs ($p < 0.01$). When evaluated against commercial applications such as MySugr, 80% of users rated our system as more affordable and better suited to Iraq’s infrastructure.

4 DISCUSSIONS

The development of the Diabetic app has demonstrated significant effectiveness in improving the diabetes patient care experience through an integrated, easy-to-use digital platform. The use of the Flutter framework and the Dart language contributed to the creation of an efficient and high-performance cross-platform application, providing a seamless interactive environment for users. The adoption of case management methods using Cubit has improved the organization of the application's logic and increased its flexibility in handling data updates. On the data front, the use of Shared Preferences provided fast and efficient local storage, while the back-end system built using Laravel and MySQL provided a reliable environment for processing and storing patient data in a secure and organized manner. On the other hand, the doctor's interface, developed using React, enabled medical staff to quickly access and analyze patient data through interactive interfaces featuring graphs and real-time alerts. The app also demonstrated its commitment to security and privacy protection standards through encryption and permission management, enhancing user confidence and confirming its potential for use in formal healthcare settings [4], [8], [15]. Thus, the Diabetic app can be considered a successful model for leveraging technology to improve self-care and facilitate effective communication between patients and doctors [1], [11], [12]. The proposed system provides an interface that enables the patient to enter glucose readings manually or upload them from a compatible device. The system adds notes or comments to the glucose log. The system sends the glucose log to the physician's system and notifies the patient's physician. The physician's notes are displayed in a list of notes added by the physician in reverse chronological order (from newest to oldest). They also include the date, time, and the physician's name.

5 CONCLUSIONS

The Diabetic app is an effective model for employing digital technologies to improve diabetes management, combining ease of use, cross-platform versatility, and performance. It was developed using Flutter and Dart to build a flexible interface, Cubit to organize the application logic, and Laravel with MySQL to ensure data security and stability. The doctor interface, developed with React, enables interactive visualization of medical data to support decision-making. By implementing security and privacy standards, the app has demonstrated its readiness for use in medical settings, underscoring its role in enhancing self-care and effective communication between patients and doctors. The app will evolve into a comprehensive chronic disease management platform, integrating with national e-health systems in Iraq and adhering to WHO digital health guidelines [6], [9], [10]. Future collaborations with local hospitals and NGOs will ensure scalability and affordability. The long-term vision is to evolve the platform into a comprehensive chronic disease management system, fully integrated with Iraq's national e-health infrastructure and aligned with WHO digital health guidelines. Planned collaborations with local hospitals and NGOs will help ensure the app's scalability, affordability, and accessibility, making it a sustainable solution for diverse communities.

REFERENCES

[1] E. Årsand, N. Tatara, G. Østengen, and G. Hartvigsen, "Mobile phone-based self-management tools for type 2 diabetes: the few touch application," J. Diabetes Sci. Technol., vol. 4, pp. 328-336, 2010.

[2] E. Årsand, G. Østengen, J. A. M. H. Tufano, T. O. Skrøvseth, J. A. I. B. Smuk, et al., "Mobile phone-based self-management tools for type 2 diabetes: the few touch application," J. Diabetes Sci. Technol., vol. 5, pp. 1195-1203, 2011.

[3] L. Catarinucci, D. De Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An IoT-aware architecture for smart healthcare systems," IEEE Internet Things J., vol. 2, pp. 515-526, 2015.

[4] H. A. El Zouka and M. M. Hosni, "Secure IoT communications for smart healthcare monitoring system," Internet Things, vol. 13, p. 100036, 2021.

[5] M. S. Farooq, S. Riaz, R. Tehseen, U. Farooq, and K. Saleem, "Role of Internet of things in diabetes healthcare: network infrastructure, taxonomy, challenges, and security model," Digital Health, vol. 9, p. 20552076231179056, 2023.

[6] M. Hassan and R. S. Istepanian, "Development of a remote diabetic patient monitoring system," in Proc. 2009 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., 2009, pp. 4679-4682.

[7] N. D. Heintzman, "A digital ecosystem of diabetes data and technology: services, systems, and tools enabled by wearables, sensors, and apps," J. Diabetes Sci. Technol., vol. 10, pp. 35-41, 2016.

[8] M. Mahmood, M. I. Khan, H. Hussain, I. Khan, S. Rahman, M. Shabir, and B. Niazi, "Improving security architecture of Internet of Medical Things: a systematic literature review," IEEE Access, vol. 11, pp. 107725-107753, 2023.

[9] R. Marzouk, A. S. Alluhaidan, and S. A. El_Rahman, "Analytical predictive models and secure web-based personalized diabetes monitoring system," IEEE Access, vol. 10, pp. 105657-105673, 2022.

[10] P. Rajan Jeyaraj and E. R. S. Nadar, "Smart-monitor: patient monitoring system for IoT-based healthcare system using deep learning," IETE J. Res., vol. 68, pp. 1435-1442, 2022.

[11] E. Seto, R. S. Istepanian, J. A. Cafazzo, A. Logan, and A. Sungoor, "UK and Canadian perspectives of the effectiveness of mobile diabetes management systems," in Proc. 2009 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., 2009, pp. 6584-6587.

[12] L. E. A. Seto, J. A. Cafazzo, A. Logan, and A. Sungoor, "Development and evaluation of a mobile phone–based self-management system for type 2 diabetes," in Proc. 2009 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., 2009, pp. 6584-6587.

[13] V. Singh, R. Singh, and M. Singh, "IoT based patient monitoring system for diabetes detection," in Proc. 2017 Int. Conf. Comput., Commun. Electron. (Comptelix), 2017, pp. 1-4.

[14] C. L. Ventola, "Mobile devices and apps for health care professionals: uses and benefits," Pharm. Ther., vol. 39, p. 356, 2014.

[15] A. Yadav and P. Gupta, "Secure IoT communications for smart healthcare monitoring system," Internet Things, vol. 13, p. 100036, 2020.