

Development and Implementation of an Ai-Assisted Web-Based Clinic Management and Clinical Decision Support System for Malaysian Public Universities

Mailasan Jayakrishnan^{1*}, Nur Athirah Nabila Mohd Idros², Ana Nabilah Sa'uadi³, Muhammad Sharilazlan Salleh⁴

¹Department of Software Engineering, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

^{2,3}Department of Information Systems, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar, Perak, Malaysia

⁴Department of Applied Data Engineering, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

***Corresponding Author**

DOI: <https://doi.org/10.47772/IJRISS.2026.100900013>

Received: 10 September 2026; Accepted: 15 September 2026; Published: 29 September 2026

ABSTRACT

In Malaysia, most public university students depend on on-campus clinics for their healthcare needs. Good healthcare facilities are indispensable and need to be emphasized by the university in order to ensure the sustainability of student health. Analyzing the university's clinical database of health conditions in the health service system will identify common health problems of the students, which could be useful in further appropriate and specific health service planning for the students. However, many of these clinics do not have any digital system that allows students to check the clinic status, view queue information, or manage their visit remotely. Moreover, increasing demand for these services has sparked conversation about the university's role in student health promotion. More students today come to campus with significant health needs, including mental health challenges. Students often have to walk or take a ride to the clinic without knowing whether it is open or how long the queue is. This situation causes wasted time, unnecessary travel costs, and delayed medical attention, especially for students who are already feeling unwell. This research study aims to develop an AI-Assisted web-based Smart Clinic Management System that addresses this problem. The prototype system will allow students to check the real-time clinic status, join a remote queue, and use an AI-powered symptom checker before deciding to visit the university clinic. The AI module will analyse the student's symptoms and classify the urgency level as urgent, moderate, or self-treatable, and it will automatically recommend whether the student should join the queue or not. The method employs a system development research approach using the waterfall model, which was selected for its suitability in developing structured and systematically documented systems. The findings indicate that the developed system framework is capable of improving the efficiency of the university clinic management system and supporting data-driven decision-making.

Keywords— Artificial Intelligence, Decision Support System, Public University, Smart Healthcare System, Web-Based Development

INTRODUCTION

Student satisfaction is an important factor in assessing the quality of healthcare facilities and services provided by the university. With the increasing trend of enrollment in tertiary education worldwide,

healthcare services for university students are important [1]. Today, driven by a broader and, in our judgement, better understanding of health and its impact on learning, many institutions of higher education provide much more. Psychological research has found that elevating healthcare in university is a key factor in advancing each student's success, meaning it should be a key focus this and every academic year [2]. Previous studies have shown that disease or illness in university students affects their learning, relationships, and quality of life both during academic years and after graduation [3], [4], [5]. More students today come to campus with significant health needs, including mental health challenges. In Malaysia, the issue is complicated by nationally expensive health care rates, and universities must invest in making these necessary services more available for students [6]. University health clinics play a vital role in the wellbeing of university students and campus populations by providing primary and urgent medical care for free or at a low cost on a walk-in basis. Students who need to visit the clinic must physically travel to the university clinic to register their presence by providing their matric number and identification number to the clinic staff.

This registration is done manually, with no digital platform in place to support the process. The absence of a digital clinic management system creates several practical difficulties for public university students. A student experiencing illness may travel to the clinic using a ride-hailing service, only to arrive and find that the clinic is closed for a break, the doctor is unavailable, or the queue is too long to wait. This results in wasted time, unnecessary transportation costs, and further physical discomfort for a student who is already feeling unwell. The manual queue process also means there is no mechanism for students to monitor their position in the queue or plan their arrival more efficiently. Beyond queue management, public university students currently have no access to a digital record of their own health visits at the clinic. Visit history, treatment details, and AI-assisted health information are not stored in any system accessible to the student. As a result, students cannot track patterns in their health or refer to past treatment records without requesting physical records from the university clinic. These scenarios can be justified as a lack of university clinic status information, where the existing public university clinic system does not provide any digital channel for students to check the clinic availability before making the trip. Students have no way of knowing whether the clinic is open, on break, or closed, nor can they view the current queue count or estimated waiting time. The existing manual process causes students to make unnecessary trips to the clinic, resulting in wasted time and transport costs, particularly for students residing in hostels located far from the public university clinic.

Besides the scenario of inefficient manual queue management process, the existing walk-in system at the public university clinic requires students to queue physically at the clinic without any remote access to the queue. Students must be present at the clinic from the start of their wait, and there is no notification system to inform them when their turn is approaching. This traditional approach is disruptive to students' academic schedules, particularly those who need to attend classes while waiting for a clinic appointment, and results in prolonged physical waiting times at the clinic. Furthermore, there is an absence of a pre-visit health assessment tool, as the existing clinic system provides no mechanism for students to assess the severity of their symptoms before deciding to visit. As a result, students with minor conditions that could be managed at home still visit the clinic, contributing to unnecessary congestion and longer waiting times for students with more urgent medical needs. The absence of any AI-assisted guidance means students cannot make informed decisions about whether a clinic visit is necessary before traveling to the public university clinic. Moreover, lack of digital health records and analytical data for students and clinic management, as the public university clinic does not maintain any digital system for recording student health visits or generating analytical reports. Students have no access to their own visit history, treatment records, or health trends. At the same time, clinic staff and admin have no data-driven tools to monitor queue performance, patient visit patterns, or symptom severity distribution. The absence of such records distribution. The absence of such records and analytics limits both student health awareness and the clinic's ability to make informed management decisions. To address these gaps, this study proposes the development of a Smart Clinic Management System (SCMS). SCMS provides students with an up-to-date view of the university clinic status, a remote queue management feature with QR code check-in, an AI-powered symptom checker, and a personal student health profile with analytical features. The system is designed to be accessible on both desktop and mobile devices, requiring no installation. Public university clinic staff will

manage daily operations through a dedicated admin panel, ensuring a smooth and organized patient flow throughout the day.

LITERATURE REVIEW

Technology has changed the way many organizations manage their daily operations. In the healthcare sector, web application systems have become an important tool to help clinics run more efficiently, reduce waiting times, and improve the overall experience for patients. Healthcare is increasingly facilitated by Information Systems (IT) via a managed IT service model that can reduce the probability of technology issues like data breaches and downtime [7]. Universities in particular can benefit greatly from this kind of digital solution, as they serve a large number of students who need quick and easy access to health services. Good services provide a supportive environment, which directly improves student health and well-being, and may also indirectly improve academic success. A clinic management system is a software system that supports the administrative and operational functions of a clinic, including patient registration, appointment scheduling, medical record keeping, and treatment documentation. According to [8], such management systems are common in hospitals but are rarely implemented in smaller clinic settings, where traditional paper-based methods continue to cause problems such as data redundancy, missing data, loss of data, and wasted time. In the Malaysian university context, [9] reported that the manual handling of student health records at a university health center leads to staff missing appointment requests and frequent loss of paper-based records, which makes record retrieval slow and inefficient. Likewise, [10] observed that a clinic relying on manual processes struggles with lost or duplicated patient records, and argued for a web-based clinic management system to store and retrieve patient data securely. In addition, [11] argued that a digital clinic service system gives both patients and providers rapid access to comprehensive medical records and improves the overall operational efficiency of the clinic. These findings support the need for a comprehensive digital clinic management system such as Smart Clinic Management System (SCMS).

Existing System

Several existing systems and studies related to SCMS were reviewed and analyzed to identify their features, technologies used, and limitations relevant to SCMS. [8] Developed Cliniro, a web-based clinic portal built with PHP and MySQL that supports three (3) user roles, namely doctor, patient, and assistant. The system provides patient registration, digital prescriptions, billing, and a data analytics dashboard. However, Cliniro focuses on records and scheduling and does not include any queue management, QR code arrival verification, or AI-based symptom assessment. [9] Developed the PKU system, an online appointment system for the UTHM university health centre, built using Java and a Firebase database. The system allows students to book appointments online and helps health-centre staff manage those appointments. However, it is an appointment booking system that does not provide real-time clinic status, remote queue joining, QR check in or any AI-assisted health guidance. [11] Designed a medical clinic service system supported by a relational database that lets patients retrieve electronic prescriptions, preview medical examinations, and schedule appointments. While comprehensive in record management, the system centres on prescriptions and appointments rather than on queue management or symptom triage. [12] Developed a web-based clinic management system for Klinik Medi Ehsan in Malaysia using PHP and XAMPP, following the evolutionary prototype model. The system serves three users through six (6) modules covering login, registration, appointment and scheduling, notification, patient monitoring, and reporting and records. However, the system is built around appointment booking and does not provide real-time clinic status, remote queue management, QR code arrival verification, or AI-assisted symptom assessment.

[13] developed QKlinik, a web-based queue management system for outpatient clinics implemented with PHP, HTML, and phpMyAdmin using the Agile methodology. The system allows patients to reserve queue numbers online using QR codes, track their real-time wait status, and make pharmacy purchases, achieving an average usability rating of 4.78 out of 5 in user acceptance testing. However, QKlinik was evaluated only in a simulated setting rather than a real clinic, and it does not include an AI symptom checker or a personal student health profile. The CLIQUE system described by [14] is a web-based queue management system for the Angeles University Foundation Office of the University Registrar, providing real-time queue tracking and notifying staff when a client joins the queue. The study reported strongly positive usability

results, but CLIQUE requires clients to enter their details at a counter or kiosk rather than joining remotely, and it has no QR arrival confirmation or healthcare-specific features. [15] proposed a hospital queue control system that uses a QR code to verify a patient's arrival. Patients register and book a queue online, download a QR code, and scan it on arrival at the hospital to confirm their presence. The study concluded that QR-based verification speeds up the administrative process and removes the need for patients to re-register on arrival, which directly supports the QR check-in mechanism used in this study. Beyond complete systems, [16] reviewed the application of queueing models in the hospital sector and concluded that queueing theory plays an important role in predicting patient flow and reducing waiting time. [17] showed that combining machine learning with a queueing model delivered through a web-based decision support application can produce accurate predictions and interpretable insights for service planning. These works reinforce the value of building a digital, data-aware queue management system of the kind for this study, as shown in Table 1.

Table-1 Comparison of Existing Systems with Smart Clinic Management System (SCMS).

References	Platform	Clinic Status	Remote Queue	QR Check-in	AI Checker	Health Records
[15]	Web	No	Yes	Yes	No	Yes
[9]	Mobile / Web	No	No	No	No	Yes
[14]	Web / Kiosk	Yes	No	No	No	No
[8]	Web	No	No	No	No	Yes
[12]	Web	No	No	No	No	Yes
[11]	Web / Mobile	No	No	No	No	Yes
[13]	Web / Mobile	Yes	Yes	Yes	No	No

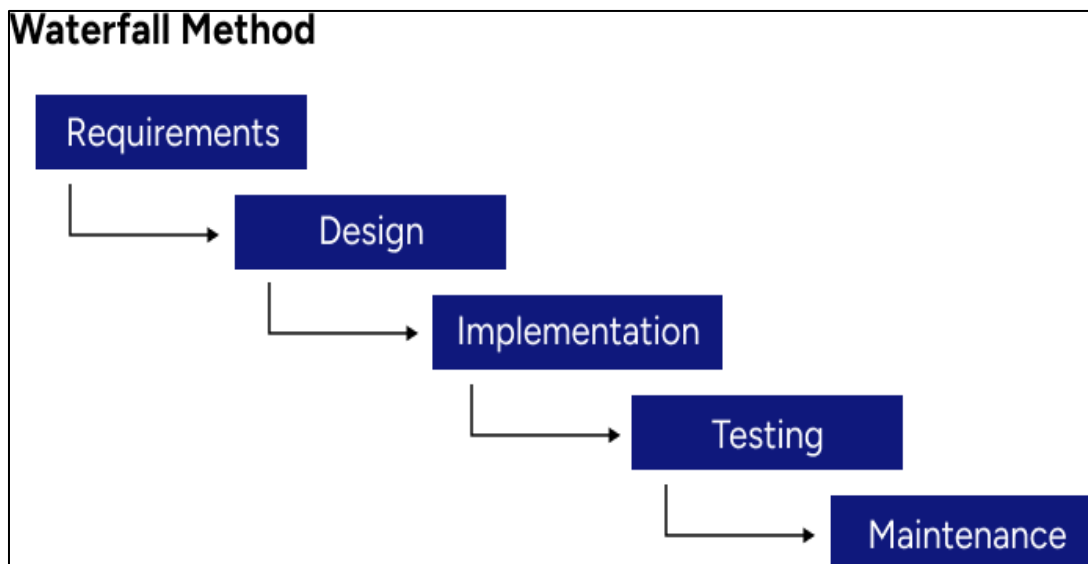
Based on the comparison in Table 1, none of the existing systems combines all six (6) features in a single platform. SCMS is the only proposed system that integrates clinic status monitoring, remote queue joining, QR code check-in, AI-powered symptom checking, and a comprehensive student health profile. This confirms the research gap that this study aims to contribute to.

METHODOLOGY

The main objective of this study is to provide high-quality software products at low cost. Therefore, proper planning and proper management are required for the software development process. Thus, a proper methodology is important for achieving this type of objective by organizations. A software development methodology is a framework used to structure, manage, and plan the software development process [18]. Each methodology offers a unique approach to managing the software development life cycle, defining project steps, roles, activities, communication standards, deliverables, and responsibilities. Its goal is to ensure the delivery of software in accordance with project specifications, within time and budget, and with minimized project risks [19]. Besides, software development methodologies follow a particular design philosophy that helps software developers to align these processes with the features of the software product. Therefore, this study adopts the Waterfall Software Development Life Cycle (WSDLC) model. Waterfall methodology is one of the traditional methods, which consists of a popular classic approach, and it is also a popular version of the software development lifecycle in the field of software engineering [20]. Moreover,

the Waterfall model is also a sequential and linear approach to software development where each phase must be fully completed before the next begins. The five (5) main phases of the Waterfall model are (1) Requirement Analysis, (2) System Design, (3) Implementation, (4) Testing, and (5) Maintenance are shown in Figure 1.

Fig.1. The Waterfall SDLC Model



Based on Figure 1, the Waterfall model was adapted for this study for several reasons. [21] stated that the Waterfall model is more suitable for projects where the requirements and goals are clear, the solutions and work to be done are known, and the requirements will not change, while Agile is more appropriate for projects where requirements are uncertain and change frequently. Since the requirements of this study are well-defined and stable from the beginning based on observation at the university clinic, and the system has a fixed scope of seven (7) clearly defined modules with no anticipated changes during development, the Waterfall model is the more appropriate choice. Furthermore, as a solo development project with fixed milestones, the structured linear progression of the Waterfall model is more manageable than the iterative, team-based approach for which Agile is designed. In the (1) Requirement Analysis phase, the functional and non-functional requirements of this study are identified through observation at the university clinic and a review of related existing systems. The problems with the current manual clinic process are documented, and the seven (7) modules of the study were defined together with their intended functions, user roles, and system constraints. Use case diagrams, data requirements, and system flow are also produced in this phase.

In the (2) System Design phase, the overall system architecture is planned, and the database structure is designed in the form of an Entity Relationship Diagram (ERD) and a data dictionary covering all ten (10) tables. User interface designs are produced for all seven (7) modules to guide the development process and ensure the design is aligned with the requirements identified in the previous phase. In the (3) Implementation phase, the system was developed module by module on a local XAMPP environment using PHP as the backend language, MySQL as the database, Bootstrap 5 for the front-end interface, and JavaScript with AJAX for dynamic interactions. Development began with Module 1 (User Authentication) and progressed through each module in sequence, with PHPMailer handling email notifications, phpqr code generating the daily QR code, HTML5-qrcode supporting the QR scanning mechanism, and the LLM API powering the AI features. In the (4) Testing phase, each module is tested using black-box testing to verify that all functional requirements are met and that the system behaves as expected from the user's perspective. Finally, in the (5) Maintenance phase, any issues or bugs identified during testing are resolved, and improvements are made based on feedback received from the users. Documentation is reviewed, and any outstanding issues are addressed accordingly.

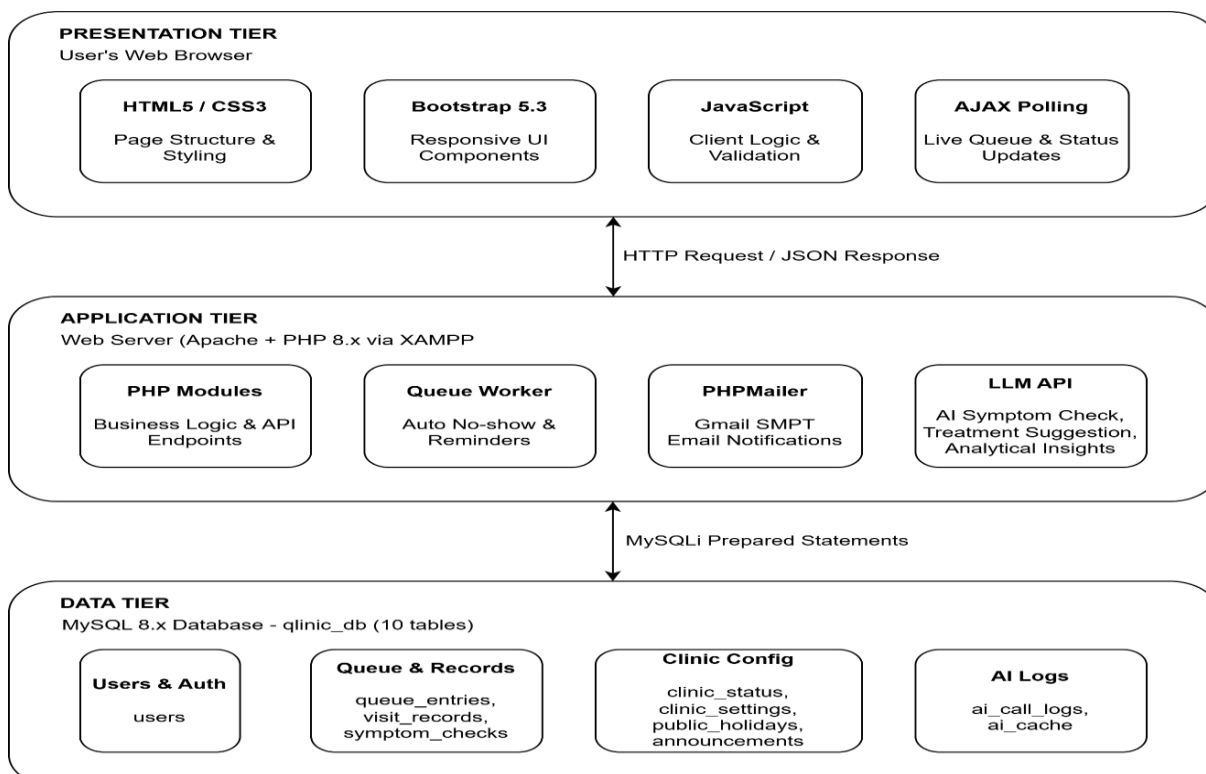
System Architecture

Architecture is a critical aspect of designing a system, as it sets the foundation for how the system will

function and be built. Moreover, system architecture is a key concept in technology and engineering, and is a blueprint defining a system design principle including structure, variability, interfaces, components, and modularity [22]. A well-defined system architecture gives development teams a shared mental model and a common language for decision-making across software development and operations. The architecture of a system reflects the way it is used and therefore changes as the system is used. Therefore, the study is designed as a three-tier web application consisting of the (1) Presentation Tier, the (2) Application Tier, and the (3) Data Tier. This architecture separates the user interface, the business logic, and the data storage into distinct layers, which improves maintainability and ensures that each component of the system can be updated independently. The (1) Presentation Tier is the front-end layer of the proposed system that users interact with directly through their web browsers. This layer is built using HTML, CSS with Bootstrap 5, and JavaScript. It is responsible for rendering the user interface pages for all four (4) user roles, handling user inputs such as form submissions, and displaying dynamic content such as queue status updates. AJAX is used in this layer to send asynchronous requests to the Application Tier and update specific parts of the page without requiring a full page reload, which is particularly important for the live queue status updates on the Clinic Status Dashboard and the queue polling on the student Queue Status page.

The (2) Application Tier is the backend layer of the proposed system that handles all server-side logic. This layer is implemented using PHP and processes user requests received from the Presentation Tier. It is responsible for enforcing role-based access control, validating user inputs, executing the business rules for each module such as queue deadline calculation, QR code generation, and queue ordering logic, and communicating with the external LLM API for the AI features. The Application Tier also handles email sending using PHPMailer for account verification and queue reminder notifications. In addition, a background queue worker script runs on a schedule, independently of any browser session, to automatically mark no-shows, send queue reminders, and recalculate queue statistics. All API endpoints in the system study are implemented as PHP scripts that return JSON responses. The (3) Data Tier is the database layer of the proposed system that stores all persistent data. This layer consists of a MySQL database named clinic_db, managed through phpMyAdmin during development. The database contains ten (10) tables covering users, clinic status, clinic settings, announcements, public holidays, queue entries, symptom checks, visit records, AI call logs, and AI cache. The Application Tier communicates with the Data Tier exclusively through MySQL prepared statements to prevent SQL injection, as shown in Figure 2.

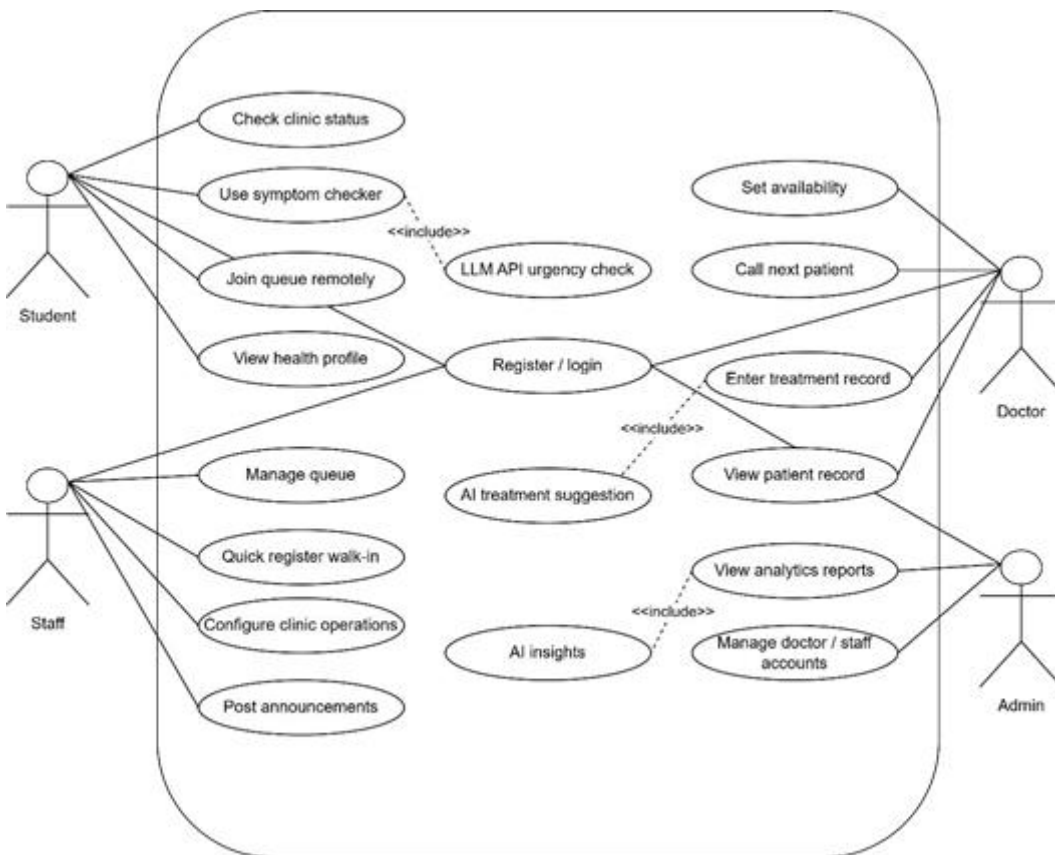
Fig.2. System Architecture Three-Tier Web Application



Target Users

The proposed Smart Clinic Management System (SCMS) has four (4) target users. (1) Students are the primary users of SCMS. They can register for an account using their student email and matric number. Once registered, students can check the clinic status, join the queue remotely from anywhere, scan a QR code upon arrival at the university clinic to confirm their attendance, use the AI-powered symptom checker to assess their condition, and view their personal health profile, including past visit records and treatment history. (2) Clinic Staff are the reception personnel at the public university clinic. They manage front desk operations through the Staff Dashboard, where they mark remote students as arrived, add and quickly register walk-in patients, flag priority patients, handle no-shows and cancellations, post clinic announcements, and configure the clinic operating hours, semester break periods, and public holidays. Clinic staff accounts are created by the administrator. (3) Doctors are the medical practitioners at the public university clinic. They access the Doctor Dashboard where they can set their availability status, call patients from the arrived pool, conduct consultations, and enter treatment records including diagnosis, treatment, Medical Certificate details, and referrals. Doctors can also search and view the complete treatment history of any patient and are assisted by AI features such as treatment suggestions and allergy conflict checks. Doctor accounts are created by the administrator and include a medical specialty field. (4) The administrator oversees the overall operation of SCMS. In addition to monitoring the clinic, the administrator can create and manage both doctor and staff accounts, view the admin dashboard with daily clinic insights, and access analytical reports on queue performance, patient visit frequency, and symptom triage distribution across weekly, monthly, and yearly periods. The administrator account is set up directly in the database as shown in Figure 3.

Fig.3. Use Case Diagram for Smart Clinic Management System (SCMS).



RESULT

Smart Clinic Management System (SCMS) implements a role-based navigation system where each user role has access to a distinct set of pages relevant to their responsibilities. Upon successful login, the system reads the user's role from the session and redirects them to their role-specific dashboard. Figure 4 shows

the navigation structure for the Student role, displaying the sidebar menu items available to a logged-in student, including Dashboard, Queue Status, AI Symptom Checker, and Health Profile.

Fig.4. Navigation Structure for Student.

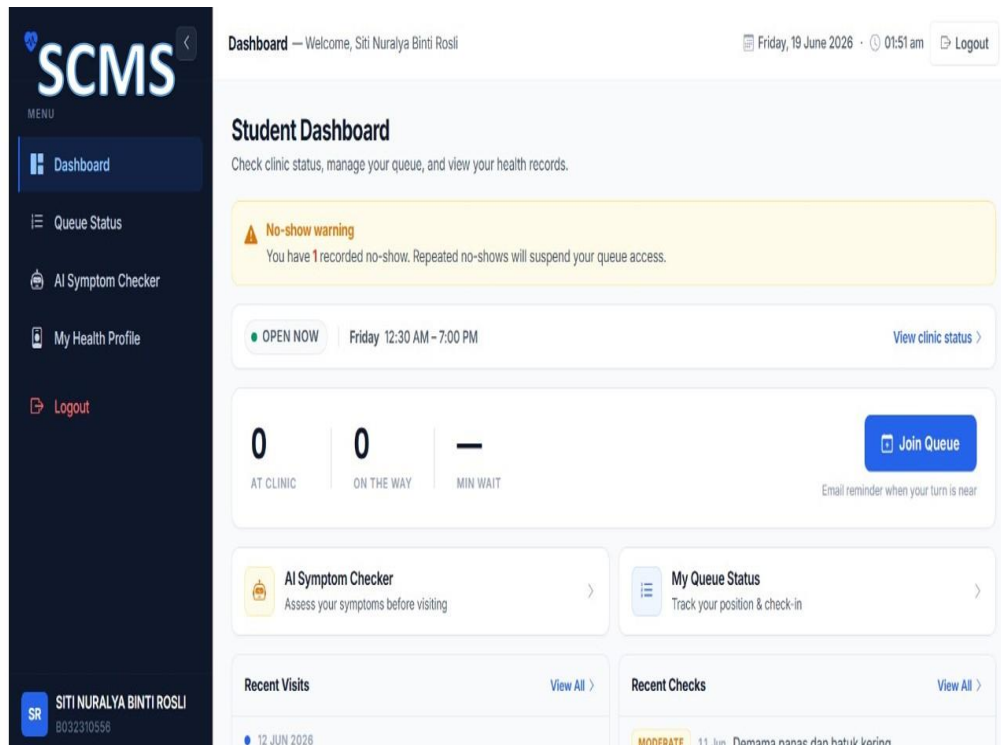


Figure 5 shows the navigation structure for the Clinic Staff role, displaying the sidebar menu items available to staff, including Dashboard, Quick Register, Announcements, Clinic Settings, and Profile.

Fig.5. Navigation Structure for Staff.

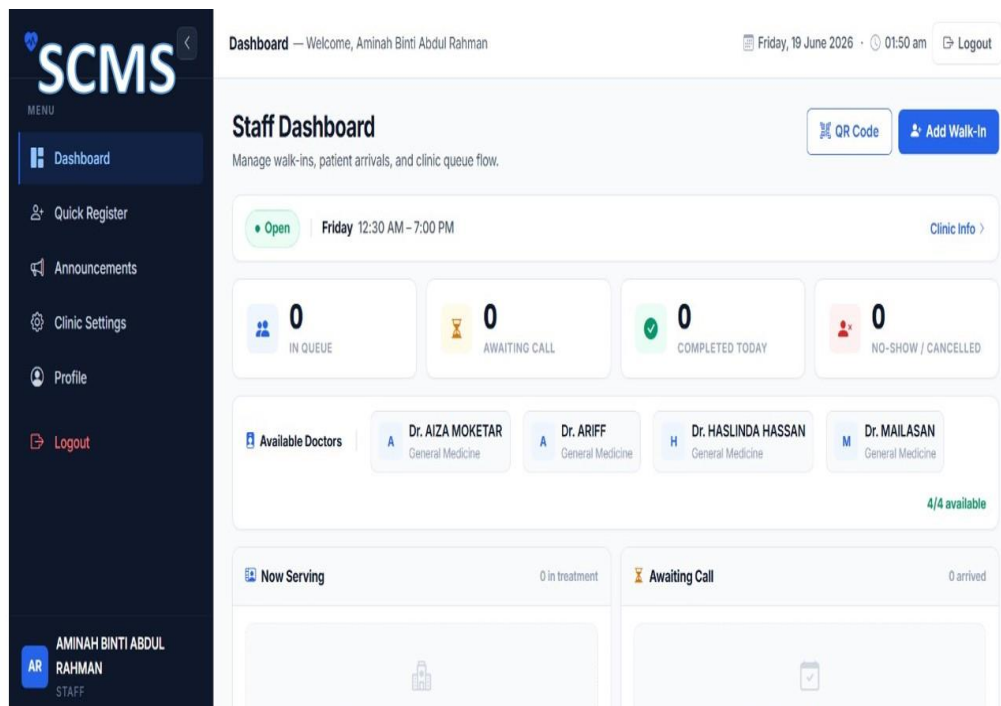


Figure 6 shows the navigation structure for the Doctor role, displaying the sidebar menu items available to a doctor, including Dashboard, Treatment, Patient Records, and Profile.

Fig.6. Navigation Structure for Doctor.

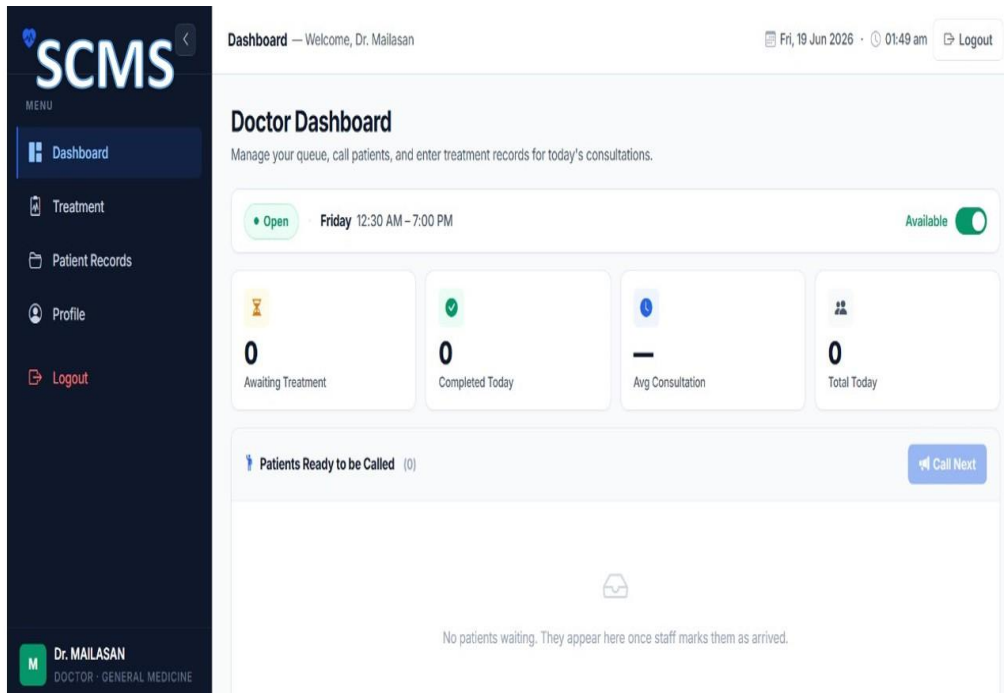


Figure 7 shows the navigation structure for the Administrator role, displaying the sidebar menu items available to the admin, including Dashboard, Analytics and Reports, Doctors and Staff Accounts, and Profile.

Fig.7. Navigation Structure for Admin.

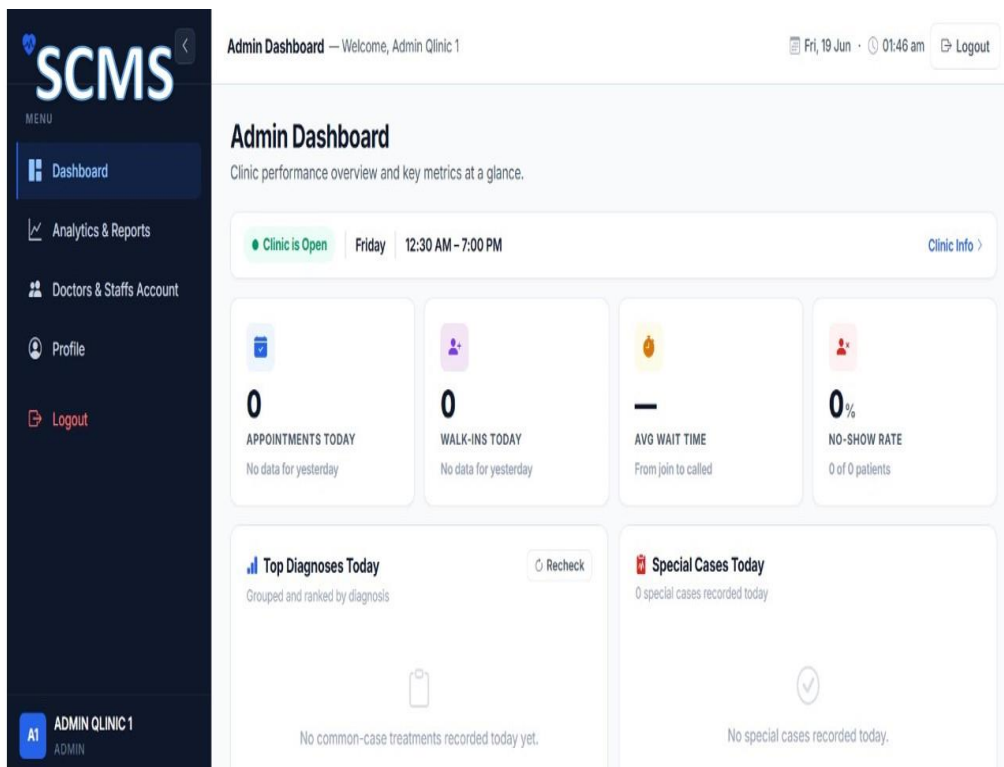
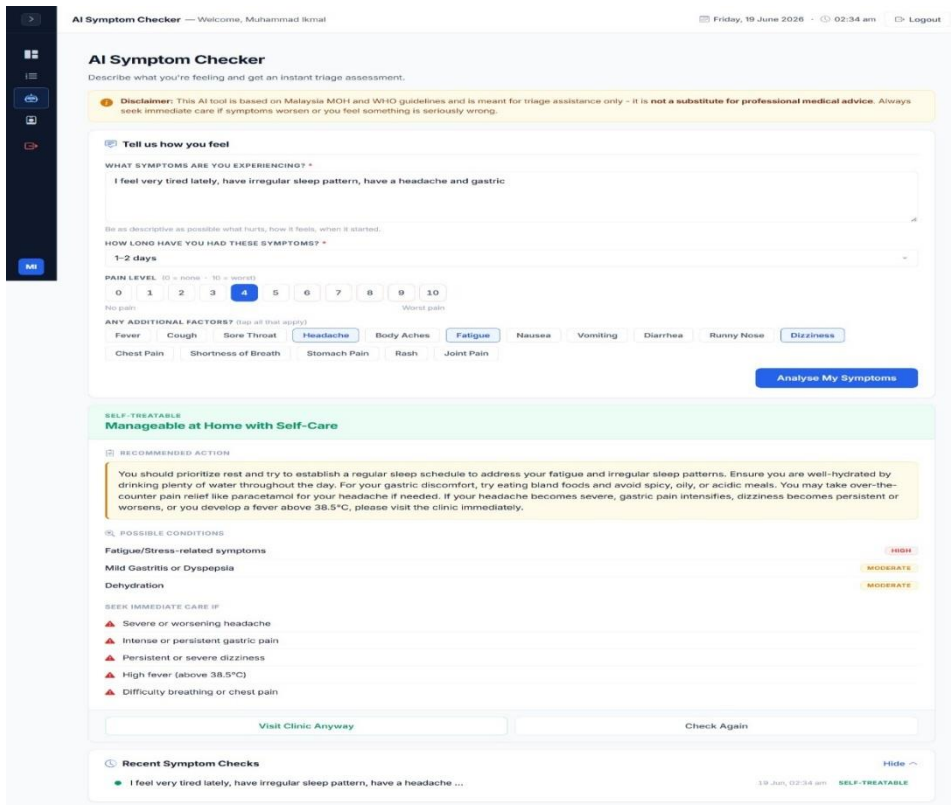


Figure 8 shows the result page of the AI Symptom Checker where the student can view the urgency level, likely conditions with likelihood ratings, an AI-generated summary, recommended action, and red flags based on their submitted symptoms.

Fig.8. Result of using AI Symptom Checker.



The screenshot displays the 'AI Symptom Checker' web application. At the top, it says 'Welcome, Muhammad Rimal' and shows the date 'Friday, 19 June 2026' and time '02:34 am'. A disclaimer states: 'This AI tool is based on Malaysia MOH and WHO guidelines and is meant for triage assistance only - it is not a substitute for professional medical advice. Always seek immediate care if symptoms worsen or you feel something is seriously wrong.'

The user input section shows: 'I feel very tired lately, have irregular sleep pattern, have a headache and gastric'. The 'HOW LONG HAVE YOU HAD THESE SYMPTOMS?' dropdown is set to '1-2 days'. The 'PAIN LEVEL' scale (0 to 10) has '4' selected. Under 'ANY ADDITIONAL FACTORS?', the following symptoms are checked: 'Headache', 'Body Aches', 'Fatigue', 'Nausea', 'Vomiting', 'Diarrhea', 'Runny Nose', 'Dizziness', 'Chest Pain', 'Shortness of Breath', 'Stomach Pain', 'Rash', and 'Joint Pain'. The 'Analyse My Symptoms' button is visible.

The results section is titled 'SELF-TREATABLE Manageable at Home with Self-Care'. It includes a 'RECOMMENDED ACTION' box with advice on rest, sleep, hydration, and diet. Below this, 'POSSIBLE CONDITIONS' are listed: 'Fatigue/Stress-related symptoms' (HIGH), 'Mild Gastritis or Dyspepsia' (MODERATE), and 'Dehydration' (MODERATE). A 'SEEK IMMEDIATE CARE IF' section lists warning signs like severe headache, intense gastric pain, persistent dizziness, high fever, and difficulty breathing. At the bottom, there are buttons for 'Visit Clinic Anyway' and 'Check Again', and a 'Recent Symptom Checks' log showing the current entry.

CONCLUSION

Smart Clinic Management System (SCMS) provides a practical and meaningful solution to the operational challenges currently faced by public university clinics. By digitalizing the clinic management process, SCMS benefits three main groups, which are the students, clinic staff, and the university as a whole. For students, SCMS significantly improves their healthcare experience at the public university clinic. Instead of making unnecessary trips to the clinic without knowing its current status, students can check whether the clinic is open and how long the queue is before leaving their hostel. The ability to join the queue remotely means that students no longer need to physically queue at the clinic, saving them valuable time, especially during busy academic periods. The AI-powered symptom checker further empowers students to make better-informed decisions about whether their condition requires immediate medical attention or can be managed at home, reducing anxiety and unnecessary clinic visits. For clinic staff and doctors, SCMS reduces the workload associated with manual queue management and record-keeping. The digital queue system allows staff to manage patient flow and reception more efficiently, while the dedicated doctor consultation workflow ensures that patients are called in an orderly manner and that treatment records, including diagnosis, Medical Certificate details, and referrals, are stored accurately and consistently in a centralized database. The AI-assisted treatment suggestions and allergy conflict checks further support doctors during consultation, and the analytical reports available to the administrator support better planning and decision-making for the clinic. For public universities as an institution, SCMS represents a step towards a more digitalized and modern campus environment. A well-functioning clinic management system reflects positively on the university's commitment to student welfare and operational efficiency. The system also serves as a replicable model that could be adapted for use by other public or private university clinics in Malaysia in the future.

ACKNOWLEDGEMENT

The authors would like to express gratitude to student Siti Nuralya Binti Rosli (B032310556) for developing the prototype of the system for this research study.

REFERENCES

1. Y. Fedorchenko, O. Zimba, M. K. Gulov, M. Yessirkepov, and M. Fedorchenko, "Medical Education Challenges in the Era of Internationalization and Digitization," *J. Korean Med. Sci.*, vol. 39, no. 39, 2024, doi: 10.3346/jkms.2024.39.e299.
2. A. M. Lederer, S. B. Oswalt, M. T. Hoban, and M. N. Rosenthal, "Health-Related Behaviors and Academic Achievement Among College Students," *Am. J. Heal. Promot.*, vol. 38, no. 8, pp. 1129–1139, Nov. 2024, doi: 10.1177/08901171241255768.
3. T. Baalman, "Health-Related Quality of Life, Success Probability and Students' Dropout Intentions: Evidence from a German Longitudinal Study," *Res. High. Educ.*, vol. 65, no. 1, pp. 153–180, Feb. 2024, doi: 10.1007/s11162-023-09738-7.
4. M. Luo, M. Hao, X. Li, J. Liao, C. Wu, and Q. Wang, "Prevalence of depressive tendencies among college students and the influence of attributional styles on depressive tendencies in the post-pandemic era," *Front. Public Heal.*, vol. 12, Jan. 2024, doi: 10.3389/fpubh.2024.1326582.
5. I. Moussa-Chamari et al., "The relationship between quality of life, sleep quality, mental health, and physical activity in an international sample of college students: a structural equation modeling approach," *Front. Public Heal.*, vol. 12, Jul. 2024, doi: 10.3389/fpubh.2024.1397924.
6. Y. M. Al-Worafi, A. M. Al-Shami, L. C. Ming, A. A. Dhabali, and K. M. Alakhali, "Medicine Education, Practice, and Research in Malaysia," in *Handbook of Medical and Health Sciences in Developing Countries*, Cham: Springer International Publishing, 2024, pp. 1–29. doi: 10.1007/978-3-030-74786-2_392-1.
7. M. J. Samonte, L. M. Achacoso, A. C. Amper, and A. L. Eco, "An In-Depth Analysis on Mitigating Data Breaches in Healthcare Systems through Big Data Structure," in *Proceedings of the 2024 12th International Conference on Computer and Communications Management*, New York, NY, USA: ACM, Jul. 2024, pp. 139–147. doi: 10.1145/3688268.3688289.
8. A. Kulkarni, V. Pansare, B. Patil, M., & Zionia, "A Window Based Portal for Doctors & Patients using PHP and MySQL.," *SSGM J. Sci. Eng.*, vol. 1, no. 1, pp. 174-178., 2023.
9. S. A. Tan, F. Y., & Mostafa, "The PKU System: Development of Online UTHM Health Center Appointment System.," *Appl. Inf. Technol. Comput. Sci.*, vol. 3, no. 2, pp. 973-989., 2022.
10. M. Lee, K. Kim, Y. Shin, Y. Lee, and T.-J. Kim, "Advancements in Electronic Medical Records for Clinical Trials: Enhancing Data Management and Research Efficiency," *Cancers (Basel)*, vol. 17, no. 9, p. 1552, May 2025, doi: 10.3390/cancers17091552.
11. K. Ivanov, M. Karova, and G. Spasova, "Design and Development of a Medical Clinic Service System," in *EEPES 2024*, Basel Switzerland: MDPI, Aug. 2024, p. 27. doi: 10.3390/engproc2024070027.
12. Y. M. . ZAMRI, N. S., & MOHMAD HASSIM, "Development of Web -Based Clinic Management System.," *Appl. Inf. Technol. Comput. Sci.*, vol. 3, no. 1, pp. 466-485., 2022.
13. Nayli Alissa binti Mohamad Fauzi and N. H. S. Suhaimi, "DESIGN AND EVALUATION OF QKLINIK: A WEB-BASED QUEUE MANAGEMENT SYSTEM FOR OUTPATIENT CLINICS," *Int. J. Softw. Eng. Comput. Syst.*, vol. 11, no. 2, pp. 147–159, Jan. 2026, doi: 10.15282/ijsecs.11.2.2025.12.0144.
14. D. S. Mallari, M. L. Z., Guintu, J. S., Magalong, Y. C., & Yap, "CLIQUE: A Web-Based Queue Management System with Real-Time Queue Tracking and Notification of Units for Angeles University Foundation Office of the University Registrar.," in *In Proceedings of the International Conference on Industrial Engineering and Operations Management.*, 2022, pp. 1988–1998.
15. R. H. Y. Perdana, H. -, M. Taufik, A. Eka, R. Muhamad, and Z. Arifin, "Hospital Queue Control System using Quick Response Code (QR Code) as Verification of Patient's Arrival," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 8, 2019, doi: 10.14569/IJACSA.2019.0100847.
16. R. Rathore, "A Review on Study of application of queueing models in Hospital sector," *Int. J. Glob. Acad. Sci. Res.*, vol. 1, no. 2, pp. 1–6, Jun. 2022, doi: 10.55938/ijgasr.v1i2.11.
17. B. Li, A. Castellanos, P. Shi, and A. Ward, "Combining Machine Learning and Queueing Theory for Data-Driven Incarceration-Diversion Program Management," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 21, pp. 22920–22926, Mar. 2024, doi: 10.1609/aaai.v38i21.30330.
18. J. He, C. Treude, and D. Lo, "LLM-Based Multi-Agent Systems for Software Engineering: Literature

-
- Review, Vision, and the Road Ahead,” *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 5, pp. 1–30, Jun. 2025, doi: 10.1145/3712003.
19. V. Bhoola, S. B. Hiremath, and D. Mallik, “An Assessment of risk response strategies practiced in software projects,” *Australas. J. Inf. Syst.*, vol. 18, no. 3, Nov. 2014, doi: 10.3127/ajis.v18i3.923.
20. S. Barman, P. Gupta, and S. Kashiramka, “Project Management Survey: A Review of Software Project Management Methodologies,” in *2024 IEEE 11th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, IEEE, Nov. 2024, pp. 1–6. doi: 10.1109/UPCON62832.2024.10983518.
21. S. Gupta et al., “Evaluating Waterfall vs. Agile Models in Software Development for Efficiency and Adaptability,” 2024, pp. 142–148. doi: 10.4018/979-8-3693-3318-1.ch008.
22. M. Mogahed and M. Mansouri, “Towards Governance of Socio-Technical System of Systems: Leveraging Lessons from Proven Engineering Principles,” *Systems*, vol. 13, no. 12, p. 1113, Dec. 2025, doi: 10.3390/systems13121113.