# Enhancing Hospital Efficiency with Healthyline: A Digital Solution for Registration, Appointment Reminders, and Queue Managemen


## ABSTRACT

Registration and queue management are essential for efficient healthcare services. However, the process has some potential drawbacks: long queues, long waiting times, difficulty in getting real-time information on queues, and missed doctor which negatively impact patient experience and hospital operations. So, this research addresses specific challenges in hospital operations, including inefficient queue management, high patient anxiety due to long waiting times, and frequent missed appointments. The study focuses on the development of the Healthyline application, which will provide online registration for patients, real-time queue information, and notification in regard to appointments with the doctor. The development process includes needs analysis, system design, development of web and mobile-based applications, testing, and maintenance. With the Healthyline application, online registration is possible, saving patients' valuable time and reducing anxiety related to waiting times. The application also complements the operational efficiency of hospitals by advancing queue management and minimizing no-shows for appointments among patients, consequently enhancing the productivity of the hospital. Such an innovative solution is hypothetically bound to improve the lots of patients and healthcare providers in creating a more structured, efficient, and user-friendly healthcare environment. In a nutshell, Healthyline has been struggling to revolutionize the patient registration process, make queue management easy, and ensure good health for every stakeholder. This study is significant as it addresses critical issues in healthcare delivery, such as long queues and appointment management. By providing real-time updates and automatic reminders, the application not only enhances patient satisfaction but also improves hospital operational efficiency by reducing the number of no-show patients. While the study demonstrates significant improvements, it also acknowledges limitations such as the need for further evaluation of the application's performance during peak hours and potential security concerns in handling patient data. It is important to note that the Healthyline application has not yet been implemented or deployed in a hospital setting; thus, the research is limited to the stages of development, testing, and maintenance.



*Keywords: hospital registration;, real-time queue information; doctor appointment reminders; web application; healthcare application.*


## 1. INTRODUCTION

Health care services that are effective and efficient are among the major pillars leading to the improvement of living standards in quality of life for the community [1]. The application of health information technology (HIT) is crucial for enhancing the quality, safety, and efficiency of health and care services. HIT can support the development of national learning health and care systems, which continuously use data-enabled infrastructure to support policy and planning, public health, and personalization of care[2]. In regard to registration and queue management process in hospitals, it has usually been facing many different challenges: long queues, inability to present real-time information on queues, and issues for patients in remembering doctors' appointments. Lack of interoperability not only disturbs the patient experience but also

reduces the operational efficiency of the hospitals [1]. Long waits, and uncertainty about queue information, may lead to wastage of time for the patients and medical staff and also increase the possibility of administrative negligence.

The application of queuing theory has gained popularity in healthcare as it helps identify bottlenecks in operations and evaluate portions of the operation in detail for optimal solutions. Operational research techniques, particularly queuing theory, play a crucial role in resource allocation and management of overcrowding, providing solutions that optimize the use of scarce and costly resources. The mathematical description of queuing systems allows healthcare organizations to identify bottlenecks and evaluate operations for optimal solutions [3].

Additionally, issues were found in data input, such as operator negligence that caused data duplication and slowed down the service [4]. Human errors like this will certainly affect data efficiency, which can ultimately lead to a decrease in the operational efficiency of the hospital. Data integration issues lead to outdated information, data duplication, and increased storage requirements. This will certainly impact data management in hospitals[4].

Therefore, the use of health information technology is necessary to ensure the stability of the healthcare system. Various health information technologies have been developed to transform health information from paper-based to electronic health information over the past few decades, and many hospitals around the world have implemented health information systems [5]. It was found that the use of information technology such as SIMRS can have a positive impact on the performance of medical staff, thereby improving the overall quality of hospital services [4]. Based on the Regulation of the Minister of Health of the Republic of Indonesia Number 24 of 2022 concerning Medical Records, healthcare facilities that use electronic medical records are required to register the electronic system used, which includes system documentation, available features, and data of the healthcare facilities using the system. Patient registration includes filling out identity and social data for outpatient, emergency, and inpatient care [6]. This regulation highlights the importance of an integrated and structured registration system in healthcare services.

This research aims to develop a hospital registration application that provides real-time queue information and appointment reminders to address these challenges. The urgency of this research lies in the need to address the persistent challenges of long waiting times and missed appointments in healthcare settings. This study fills a critical research gap by developing an innovative hospital registration application that leverages information technology to enhance operational efficiency and patient satisfaction. The availability of real-time queue information has many positive impacts on patients in hospitals, such as reducing anxiety and uncertainty. Real-time information provides patients with a clearer understanding of how long they have to wait. This can reduce the anxiety and frustration that usually arise due to the uncertainty of waiting times. Improving patient satisfaction, by knowing when they will be served, patients feel they have more control over their time. Real-time information allows patients to make better use of their waiting time [7].

Additionally, the presence of a doctor appointment reminder feature will have a significant impact on missed appointments and an increase in patient attendance at the hospital. Appointment reminders are likely to reduce missed appointments by an average of 41%, helping make the patients more consistent in their attendance. When reminders were used, there was an average increase in patient attendance to the clinic of about 34% [8]. This will enhance operational effectiveness in a home of health care service provision.

It would make the process of patient registration easier, the queue information for the patients more accessible in a quicker manner, and the number of no-shows among patients for doctor appointments lesser. An application developed on these features will not only be advantageous for the patients, in terms of an improved experience regarding registration and queue management, but it will also help the hospital to work better with efficiency in operations and management.

Health information systems (HIS) have been shown to have a significant impact on healthcare provider performance. Research shows that system quality, record quality, and service quality positively contribute to the effectiveness of HIS use, which in turn improves healthcare provider performance. Knowledge quality and effective use of health information systems are key factors in improving healthcare provider performance. Previous research in Malaysia showed that good knowledge quality of EHR systems significantly contributes to clinical performance [9].

Dhagarra's study shows that technology acceptance in healthcare in India is highly influenced by factors such as trust and privacy concerns. Good quality health information systems positively contribute to effective use and performance of healthcare providers. However, privacy concerns can hinder technology adoption if not addressed properly. Therefore, it is important for healthcare providers to build trust among patients and address privacy concerns to increase technology acceptance in healthcare [10].

Health Information Systems (HIS) in Indonesia have shown significant progress since regulations requiring the implementation of Hospital Management Information Systems (SIMRS) in 2013. However, challenges remain, such as resistance to change among staff accustomed to old ways of working, lack of skills and training in information technology, and uneven infrastructure between private and government hospitals [11].

This research aims to develop a hospital registration application with real-time queue information features and doctor appointment reminders while also prioritizing security measures to protect patient data. Given the significant concerns regarding privacy and data security highlighted in studies from Malaysia, India and Indonesia, it is crucial that this application incorporates robust security protocols to safeguard sensitive health information. By addressing potential vulnerabilities, the application is expected to not only resolve the issue of unstructured long queues but also enhance trust among patients regarding the handling of their personal information. This innovation will have a positive impact on both hospitals and patients, with improvements in operational efficiency and an overall enhancement in the quality of healthcare services.

## 2. RESEARCH METHODS

### 2.1 Related Work

The research conducted by Pomalingo and Tobing focuses on optimizing the patient registration process through the development of an Online Admission Application using the Scrum Method approach. This method was chosen because it can offer innovative and responsive solutions to the needs of patients and healthcare staff. The application development process follows the Scrum stages, such as Product Backlog, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective, and is supported by UML (Unified Modeling Language) design to ensure smooth development. Application testing is conducted using the black box method to ensure the alignment between design specifications and user requirements. Research results show that the use of Scrum has proven effective, with a success rate of 90% in completing sprints on time. The developed application is also assessed to meet quality standards and is capable of providing innovative and responsive digital solutions to user needs [12].

Fajar conducted research focused on optimizing outpatient services through user interface (UI) design for a mobile-based online registration application. This research uses the Research and Development (R&D) method with a case study approach at Nurhidayah Hospital. Data was collected through Focused Group Discussion (FGD) and observation of existing websites. The research results show that although the hospital has provided online registration through the website, its usage is not yet optimal. Therefore, the development of mobile applications is considered more efficient and capable of increasing patient satisfaction in the registration process [13].

Asworowati conducted research on the design of a mobile-based outpatient registration information system at Gizar Mother and Child Hospital. This research uses the waterfall software development model, but only two stages are utilized, namely requirements analysis and design. This system is designed using UML, which includes ERD, LRS, Class Diagram, Sequence Diagram, and user interface. (UI). Testing is conducted on the front-end by potential users to ensure that the login function, data management, and report printing work well, as well as back-end testing by administrators to test the efficiency and accuracy of data management [14].

The conclusion from several studies shows that the development of an online registration system in hospitals has proven capable of increasing service efficiency. Through approaches like Scrum used by Pomalingo and Tobing, the development of the online admission application can be completed on time and provide innovative solutions that meet the needs of patients and healthcare workers [12]. Fajar emphasized that the use of mobile applications for online registration is more effective than websites, increasing efficiency and patient satisfaction [13]. Asworowati also demonstrated that the design of a mobile-based registration system

using the waterfall model can optimize data management and improve administrative efficiency [14]. Overall, these innovations play a crucial role in accelerating the registration process, improving service quality, and effectively meeting the needs of hospitals.

## 2.2 Method

This research uses the waterfall method for application development [15]. The Waterfall method is a linear and sequential software development model. In this model, each stage of software development must be fully completed before moving on to the next stage.This research focuses on the development of a hospital registration application with doctor appointment reminders and real-time queue information.
The waterfall method was chosen for the development of the Healthyline application due to its structured and systematic approach, allowing each development stage to be completed well before proceeding to the next stage. This is very suitable for the Healthyline project, whose needs are already clear and stable, namely the addition of doctor appointment reminder features and real-time queue information. Thus, it supports risk control, ensuring that system specifications can be met accurately [15], [16].

Comparison of the waterfall method with agile methods such as Extreme Programming (XP), Scrum, and Dynamic Systems Development shows that agile methods are indeed more flexible and adaptive, allowing for changes in requirements during the development process. However, this flexibility can result in a lack of documentation and clear structure, which are important in the development of healthcare applications. Unlike the agile method, which emphasizes minimal documentation, the waterfall method produces comprehensive documentation at each phase, which is crucial for regulatory compliance [17]. Although agile emphasizes continuous user involvement, the waterfall method ensures that all user requirements are clearly identified and documented before development begins to reduce the risk of misalignment.

There are five development phases using the waterfall method, as shown in Figure 1, namely: requirement analysis, system design, implementation, testing, and maintenance [18]. Qualitative data was collected through a literature review needed in the application. Qualitative data was gathered through interviews with hospital staff to understand their needs and challenges in the registration process. In addition, data can also be obtained through direct observation of the registration process and queue management at the hospital. The data was then mapped to determine user needs, starting from functional needs and non-functional needs. That information serves as the basis for application development during the system design phase. In this system design phase, it produces DFD (Data Flow Diagram), software architecture design, and web and mobile application prototype designs. Next, in the implementation phase, everything produced from the system design phase is coded into the form of application programs. The application program produced will be tested in the testing phase. This stage is important, which is to test whether the program meets user needs and can be used effectively. The testing method uses blackbox testing. The test results will be used for improvements during the maintenance phase.

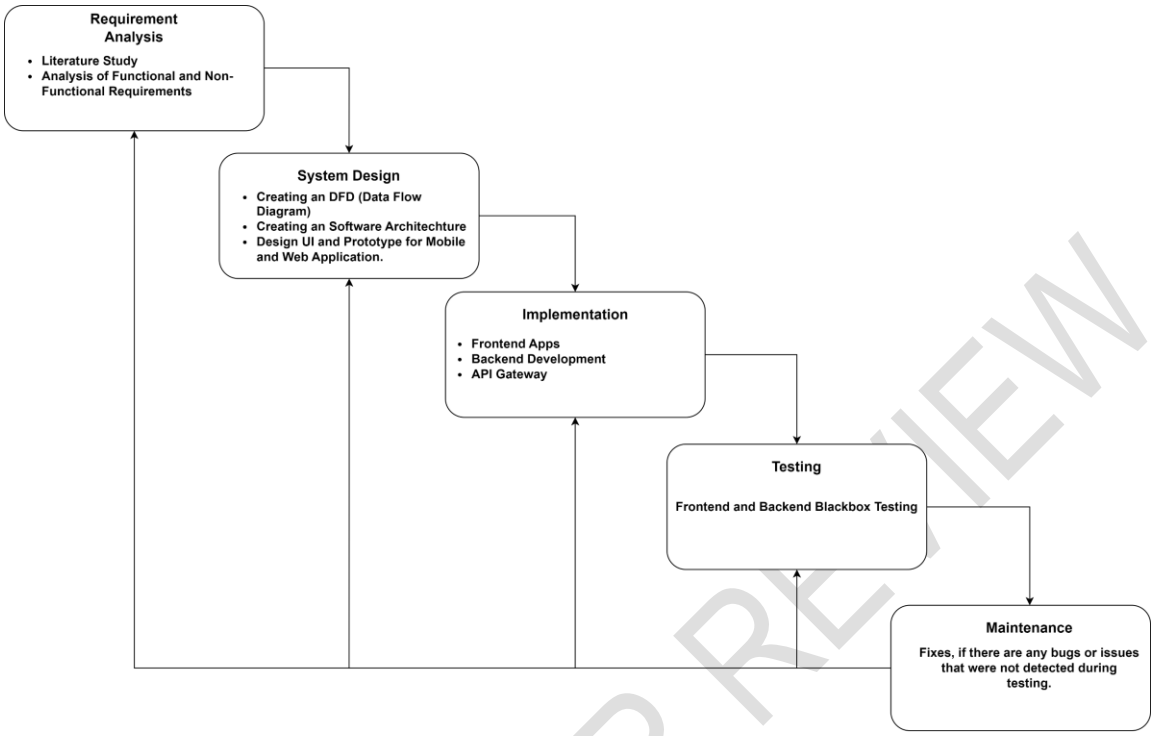


**Fig. 1. Workflow of Waterfall Method**

## 2.2 Research Framework

The research framework for Healthyline begins with problem formulation and goal setting. In this stage, the goal determination is divided into two phases: identifying the problem and collecting qualitative data through literature review. In identifying the challenges, a major issue was found: information regarding patient queues may not always be available in real-time, causing patients to often be unaware of their position in the queue or the estimated waiting time required before registration. This can result in long patient waiting times, making it less effective. Therefore, it is important to have a system that can connect various healthcare services. As stated by Ketut Agus Saputra et al., "A system is needed that can act as a link between health systems and provide information related to the patient's health history. This includes the patient's personal information and medical history compiled from various storage sources in healthcare services, commonly referred to as a Personal Health Record (PHR) " [19].

After identifying the challenges faced, a literature review and observation were conducted, focusing on functional and non-functional requirements. The result of this stage is the achievement of the development goals for the Healthyline application with doctor appointment reminders and real-time queue information. The proposed system model for hospital registration aims to address the main challenges faced as described in Figure 2. In this case, it is important to note that the presence of HMIS (Hospital Management Information System) can reduce the complexity of healthcare services by improving the effectiveness and efficiency of hospitals as organizations through innovation in information system development [20]. In addition to real-time queue information features, doctor appointment reminder features will also be implemented. After the patient successfully registers, the system will automatically send reminders to the patient before their doctor's appointment time. These reminders can be in the form of push notifications through the application or text messages directly to the patient's device. With the presence of this reminder feature, it is hoped that the rate of patient absenteeism at the scheduled appointments can be reduced, thereby helping to improve the operational efficiency of the hospital and ensuring smoother and more effective healthcare services [8]. Thus,

online registration systems like Healthyline are expected to improve efficiency and effectiveness in queue management and patient registration.

After obtaining a clear formulation of the problem and objectives, the next stage is to design a conceptual framework solution that defines relevant objectives for the research process, which can help researchers draw logical conclusions from the available data. Next is the system design phase, which will produce the DFD (Data Flow Diagram) design, software architecture design, and prototypes of web and mobile applications as explained in Figure 1 of the Waterfall Method Workflow. The design phase ensures that all application components are well-structured and meet user needs. The prototypes resulting from the system design phase will be coded in the development phase, where the web application will be coded using Express.js and the mobile application using Kotlin. Implementation also includes API creation; this structured approach ensures that all functions, such as patient registration, real-time queue updates, and appointment reminders, are developed effectively. To support the application's security, an authentication feature will also be developed. This feature ensures that only registered users such as patients and admins can access certain information based on their access rights. The use of technologies such as Firebase and Express Validator will be employed for secure user identity management. Additionally, business logic on the backend, such as appointment reminder notifications, will be developed so that the application can send automatic appointment reminders to patients before their appointments with the doctor. Real-time queue management is also being developed using business logic that updates the queue status directly, allowing patients to see their position in the queue and the estimated waiting time.

After the development stage, the application undergoes testing to ensure it functions correctly and meets user expectations. Testing will be conducted on the front-end and back-end as explained in Figure 1. Blackbox testing will be used to evaluate the main features, including patient registration and queue management. Performance assessments will be carried out to evaluate the application's responsiveness under different scenarios, ensuring that it can manage load and operate efficiently.

In the final stage, application maintenance will be carried out if errors or bugs are found during the testing phase. The application will be fixed to produce a stable and efficient application. The expected final condition after the implementation of this application is the creation of a more efficient and structured registration and queue management system in the hospital. With the presence of this hospital registration application, it is hoped that long and unstructured queues can be significantly reduced. Patients will be able to register independently through the application or website, reducing congestion and confusion in the physical registration area.

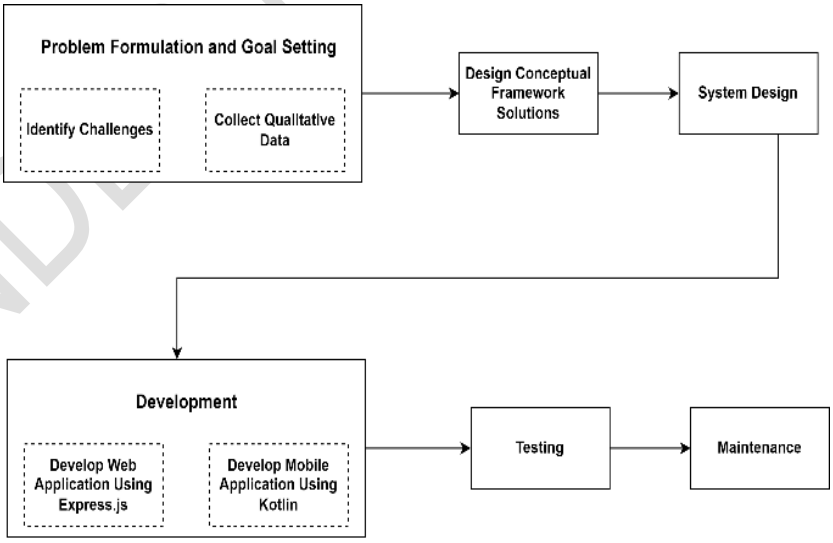


**Fig. 2. Research Framework**

## 2.3 System Recommendations

In this section, the hospital registration system before and after the development of the Healthyline application will be explained.

The condition before the system was created in Figure 3, patients registered by sending data to the hospital's WhatsApp, after which the registration staff manually entered the data into the system. This was inefficient and could lead to data entry errors, such as incorrect patient data entry or mistakes in managing queues. After that, the confirmation of the registration will be sent to the patient, which contains the patient's queue number. However, the patient does not know when their turn will arrive, resulting in an inability to know their position in the queue or the estimated waiting time needed.

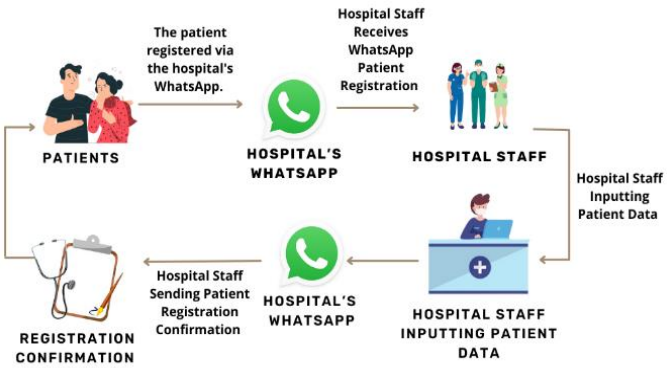


**Fig. 3. Hospital Registration System Before The Development of The Healthyline Application**

Through online patient registration, patients can create an account on the Healthyline application. After creating an account, patients can register at the hospital by filling in the required data directly in the application. Registration will be confirmed by the hospital and the application will send a registration confirmation containing information about the patient queue. The application displays real-time queue information currently happening at the hospital so that users can know their position in the queue or the estimated waiting time required. In addition, the application will also send notifications to remind users of their scheduled doctor appointments, as well as notifications if there are any schedule changes as illustrated in Figure 4.

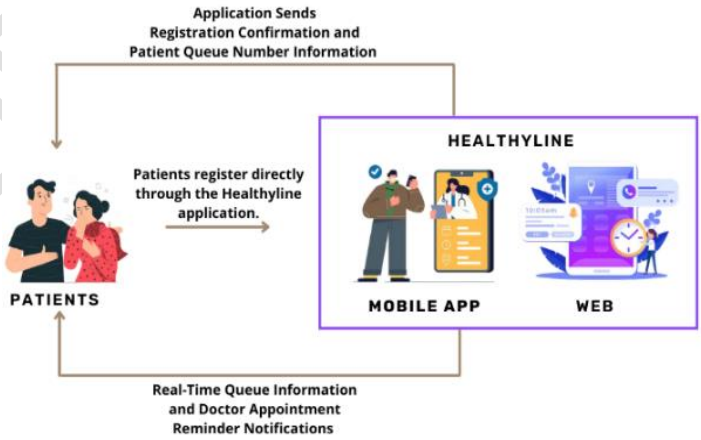


**Fig. 4. System After the Development of The Healthyline Application**

## 3. RESULTS AND DISCUSSION

At this stage, the application is developed, tested, and evaluated. Testing is conducted to assess the performance and functionality of the application.

### 3.1 Result

In problem identification and needs analysis conducted through qualitative data collection using the observation method. As explained in Figure 1, the stage through the literature study produces data on functional and non-functional requirements.

As in Table 1, users are required to enter personal data such as ID number, name, date of birth, and contact information to schedule a doctor's appointment through the application. This functionality is designed to streamline the registration and scheduling process, enabling users to arrange medical consultations quickly and efficiently. The application is required to handle these registrations promptly, ensuring minimal wait times and a smooth user experience, and users should be able to search for doctors according to their needs through the search feature. The application will provide output in the form of notifications to users, such as real-time queue status and appointment reminders. This ensures that users always receive up-to-date information.

**Table 1. Functional Requirements**

| No. | Functional Requirements | Description |
|---|---|---|
| **1** | Input Requirements | Users can enter personal data, such as name, date of birth, and contact information, as well as schedule appointments with doctors through the application. |
| **2** | Process Requirements | The application must be able to process user registrations, as well as allow for the search of doctors that match the patient's needs. The system must run smoothly without any obstacles. |
| **3** | Output Requirements | The application will provide notifications to users in the form of appointment confirmations, real-time queue status, and reminders for upcoming appointments. |

In the development of the Healthyline application, a supporting operating system is required, namely Windows 10 64-bit. For development, Visual Studio Code IDE and Android Studio are used for integrated Android environment development. The database will be hosted on MySQL, with user database management conducted through Phpmyadmin and Firebase.

**Table 2. Non-Functional Requirements**

| No. | Non-Functional Requirements | Spesification |
|---|---|---|
| **1** | Software Requirements | • Operating System: Windows 10 64-bit |

| | | |
|---|---|---|
| | | • IDE: Visual Studio Code and Android Studio<br>• Hosted Database: MySQL<br>• User Database : Phpmyadmin, Firebase |
| **2** | Hardware Requirements | • RAM: Minimum 8GB<br>• Prosesor: Intel Core i5-6200U<br>• Storage: SSD 256GB and Hardisk 500GB |

As explained in Figure 1, at the System Design stage, the data model design uses DFD (Data Flow Diagram) where the DFD represents the flow of data processes or systems [21]. Figure 5 is a Level 0 Diagram or commonly referred to as a Context Diagram, which provides an overview of a system within an organization, showing the system boundaries, interactions between external entities and the system, and the general flow of information between entities and the system. Patients have the right to register after logging in.

The patient registration data is then stored in the system. Patients have access rights to register after logging in and filling out the patient registration data. Patients also gain access to real-time queue information, allowing them to monitor their position in the queue. In addition, patients will also receive appointment reminder notifications with the doctor, aimed at reducing patient no-show rates and ensuring that patients arrive on schedule.

Patient Registration Clerk has access rights to manage patient registration data, doctor data, doctor schedule data, queue data, polyclinic data, and medical record data by logging in as an admin. This includes the ability to manage patient registration data, such as new registrations and updates to existing patient profiles. They are also responsible for overseeing doctor data, such as ensuring the availability of doctors is recorded correctly. Additionally, they handle outpatient clinic data, coordinating information related to various departments or available medical services. Doctors can view patient registration data and queue data by logging in as a doctor. unavoidable.

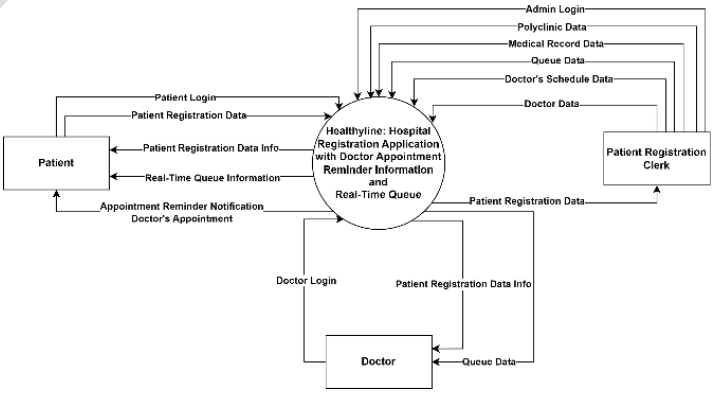


**Fig. 5. Context Diagram**

The hierarchical diagram of the patient registration system development is a graphical representation that illustrates the stages and processes in the development of that system. As explained in Figure 6, this diagram includes login, master data, transactions, and reports. At the initial stage, users must log in to ensure secure access to the system; both patients and hospital staff must enter their credentials to access the system. This feature includes user authentication, which is important for maintaining the security of patient data. Master data serves as the primary storage for information and data related to patient registration, such as patient data, doctor data, polyclinic data, queue data, etc. This data will later become the foundational basis for the transaction processes that run within the system. The Transaction stage includes key operational activities, such as patient registration, real-time queue management, and appointment notification delivery. After this process is completed, the system can generate a Report that provides data such as the number of registrations and queue status. By using this diagram, system developers can clearly visualize the workflow and the interconnections between each stage, making it easier to plan and execute each step efficiently and effectively.

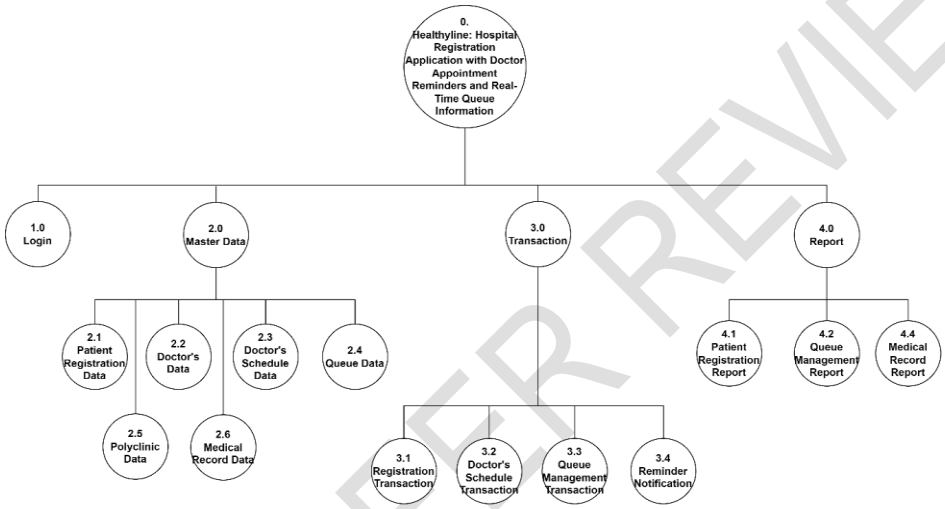


**Fig. 6. Hierarchical Diagram**

In Table 3, there is a database model for patient registration. The appointment database table uses a relational database structure [22], and the table is designed with various foreign keys to maintain integrity between the tables related to the appointment table. This table serves to store patient registrations with doctors, including important information such as patient identification (patient_id), polyclinic (department_id), and doctor (doctor_id), all of which are related to other tables in the system. The relationship will later be used for retrieving data such as patient data, doctor data, and department data to be processed further in the patient registration process.

**Table 3. Appointment Database Model**

| Column | Requirements |
|---|---|
| **id** | As a Unique Identifier for each medical appointment. |
| **nik** | Storing the Nomor Induk Kepeundudukan (NIK) of patients who make appointments. |
| **name** | Storing the full name of the patient who made the appointment. |

| | |
|---|---|
| **date_of_birth** | Storing patients' birth dates for verification and reference purposes. |
| **date** | Saving the scheduled medical appointment date. |
| **time** | Storing the scheduled medical appointment time. |
| **address** | Storing the patient's complete address for reference and administrative purposes. |
| **email** | Storing patients' email addresses for notifications or appointment reminders. |
| **phone** | Storing patients' phone numbers for reminders or notifications. |
| **pasien_id** | Storing the patient ID as a reference to the patient table. |
| **department_id** | Storing the department ID relevant to the medical appointment (misalnya, departemen spesialis). |
| **doctor_id** | Storing the ID of the doctor responsible for the medical appointment. |

In Table 4, there is a database model for the Queue. Where this table is connected to the patient table and the registration table, with foreign keys pasien_id and appointment_id. By connecting the queue table with these two tables, it ensures data integrity between the tables and ensures that changes to the registration data or doctor's appointment will automatically be updated in the queue table.

**Table 4. Queue Database Model**

| Column | Requirements |
|---|---|
| **antrian_id** | As a primary key or unique identifier for each queue. |
| **pasien_id** | Storing the ID reference of the patient who is in line, connected to the patient table. |
| **appointment_id** | Storing the appointment ID reference, linked to the appointment table. |

| | |
|---|---|
| **queue_number** | Storing a unique queue number for patients on that day. |
| **status** | Storing the patient's queue status, with options such as 'Waiting', 'In Process', or 'Completed'. |
| **created_at** | Storing the date and time when the queue is created, with a timestamp that is automatically updated with each data change. |

In the software architecture diagram in Figure 7, it explains how the workflow of the Healthyline application, built to facilitate the hospital registration process, operates. This application consists of two main components, namely the Client Side (user side) and the Server Side (server side), which interact with each other to provide information and data to patients. On the user side, the mobile application is developed using Kotlin and utilizes Firebase services for authentication and real-time notification delivery. The reason for using Kotlin in the development of the Healthyline application is that Kotlin is a modern programming language developed by JetBrains and recognized by Google as the primary language for Android application development since 2017. This makes it a very suitable language because the main platform for the Healthyline application is Android. In addition, Kotlin has a simpler and safer syntax compared to Java, which reduces the likelihood of coding errors [23]. The mobile application built with Kotlin is connected to FCM (Firebase Cloud Messaging) to receive notifications, such as appointment reminders or queue status [24]. FCM (Firebase Cloud Messaging) is a cross-platform notification service provided by Google through Firebase. FCM allows app developers to deliver messages, notifications, or data to app users in real-time and cost-effectively. The mobile application sends requests through a REST API developed with Express.js API on the server side [25]. Express.js is a Node.js-based web application framework. Express.js is used to build web applications and APIs more easily, providing various features for handling HTTP requests, middleware, and routing. These requests include appointment scheduling, queue status, and access to information such as doctors, patients, and clinics.

On the server side, Express.js functions as a backend framework that handles various requests from mobile and web applications. The API built with Express.js provides RESTful endpoints to receive and process requests from mobile and web applications. For the web application being developed, EJS (Embedded Javascript) is used as the template engine. This web application allows hospital staff to manage patient, doctor, and clinic data, as well as monitor appointments and queues in real-time. The data source contains data such as patient data, doctor data, and polyclinic data stored in a MySQL Database accessed through Phpmyadmin as a Database Management System. (DBMS). Here are the functions and features of the system or business logic on the server side:

a. User Authentication & Authorization, Firebase is used here for user authentication in the mobile application, while the Express.js API ensures that only authorized users can access the system features.
b. Appointment Scheduling & Queue Management, this feature allows patients to schedule appointments with doctors and monitor their queue status in real-time.
c. Doctor & Patient Data Management, on the server side, hospital staff can manage doctor, patient, and clinic data.
d. Real-time Notification Logic, the system sends direct notifications to users via FCM whenever there are important changes, such as appointment reminders or queue updates.
e. API Handling, with Express.js provides API endpoints accessed by mobile and web applications to retrieve data from the MySQL Database and process user requests.

The development phase of Healthyline begins with back-end development on the server side. Back-end development starts with designing the API structure and architecture. The back-end is implemented using Express.js, which will provide endpoints for CRUD operations. (Create, Read, Update, Delete). The next step is to design the database using MySQL and PhpMyAdmin as Database Management Systems. (DBMS). Tables are linked using relationships to maintain data integrity, such as the appointment, queue, doctor

tables, etc. API error handling is implemented to manage various types of errors, such as authentication errors or failures when communicating with the database. For the development of business logic using the Model-View-Controller (MVC) architecture [26], where the application logic is separated from the user interface (front end), making it easy to maintain and enhance. MVC (Model-View-Controller) is a software architecture used to separate applications into three main components: Model, View, and Controller. This approach helps developers organize code, making applications more modular, maintainable, and flexible.

The authentication process on the back-end uses Express Validator as middleware that combines the Express.js model and Validator.js to validate input data. Data exchange between the back-end and front-end, including inter-service exchange, uses JSON Data. JSON Data is chosen for several reasons: it is lighter than other data exchange formats, easier to implement, and does not depend on the user interface [19].

Front-end development on the server side using EJS (Embedded Javascript) is an approach where the HTML sent to the client is generated and rendered directly from the server, based on existing dynamic data. By using EJS, it allows the insertion of Javascript logic directly into the template, so that the HTML can be created to adjust to the data retrieved from the backend. Front-end development using EJS falls under the SSR (Server-Side Rendering) approach. (Server Side Rendering). SSR (Server-Side Rendering) is a rendering technique where the content of a web application or web page is first generated on the server, then sent in complete HTML form to the user's browser. SSR tends to provide faster page load times for users, especially when the page is heavier or more complex. SSR sends a complete HTML page to the client, allowing users to see the content faster compared to CSR (Client Side Rendering), which requires time to download and execute Javascript before displaying the content [27]. In the MVC architecture, the application logic is divided into three components: Model, View, and Controller. The Model is responsible for managing data and business logic. With SSR, data from the Model is rendered on the server and sent to the client in the form of complete HTML, allowing users to see the content faster [28].

The next development is the front-end development on the client side. This phase begins with the development of the User Interface (UI) on the client side, focusing on creating mobile applications. The design that was previously created using Figma is now being coded using Kotlin. The main features developed include inputting personal data for patient registration, real-time queue information, and appointment reminder notifications through Firebase Cloud Messaging. (FCM). Firebase is used as a solution for user authentication and data storage that is directly integrated with the back-end.
At the application testing stage, Healthyline uses the blackbox testing method with the aim of ensuring that every function in the application runs as expected. This testing focuses on the input provided and the output generated. Blackbox testing on Healthyline includes testing of main features, such as patient registration, real-time queue, and appointment reminder notifications. Here are the sample blackbox testing results that can be seen in Table 5.

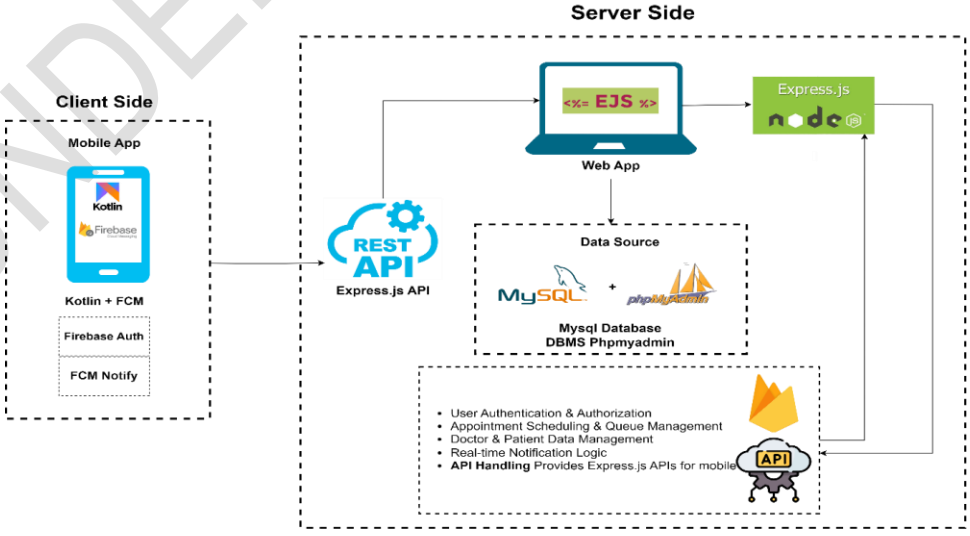


**Fig. 7. Software Architecture**

The test results showed that all major features are working correctly: from the patient's registration to the real-time queue and appointment reminders. It responded properly, proving that management of data and integrity of data in the backend process was successful, from testing new patient registration down to sending the reminder notifications. It successfully rejected invalid/incomplete data, provided error messages, and ensured the timely delivery of notifications.

Next, a performance test was executed with JMeter. It is capable of doing user simulations simultaneously, test on a graphical interface, and save tests in XML format.

**Table 5. Blackbox Testing**

| Feature | Testing Scenario | Expected Outcome | Actual Results | Status |
|---|---|---|---|---|
| **Patient Registration** | Adding a new patient with complete data | Patient data has been successfully saved, the server responded with a 201. | Patient data saved, 201 response | Passed |
| **Patient Registration** | Patient registration with an already registered NIK | The system rejects the registration and sends the message "NIK already registered." | An error message appears. | Passed |
| **Real-Time Queue** | Taking a queue number for registered patients | The system automatically assigns queue numbers in the JSON response. | Queue number given | Passed |
| **Real-Time Queue** | Updating the queue status from "Waiting" to "In Progress" | The queue status has been successfully updated and returned a 200 response. | Status updated | Passed |
| **Doctor Appointment Notification** | Sending appointment reminder notifications to patients | The system sends notifications to users' devices with appointment information. | Notification sent | Passed |
| **Doctor Appointment Notification** | Displaying an appointment reminder 10 | The system displays automatic reminders on the patient app before | Reminder appears | Passed |

| | | | | |
|---|---|---|---|---|
| | minutes before the schedule | the appointment starts. | | |
| **Doctor Appointment Notification** | Displaying a notification message when the appointment is finished | Appointment notifications have been sent to patients with relevant information. | Notification sent | Passed |

The performance testing of the Healthyline application is done in accordance with some measured metrics that identify the responsiveness and efficiency of the application interface in handling main features of the Healthyline application. The three important parameters contained within the test results are average UI time, raster time, and average FPS for each feature. Testing the patient's login ensures authentication will be smooth and truly work efficiently. The graph represents the average UI time taken by the Patient Login feature, which is 180 ms; the average raster time is 22 ms, while the average FPS is 58 fps, indicating good performance and stability of the interface in the above-mentioned login procedure. Testing Patient Registration and Data Management has to be performed to evaluate the speed of the application itself in storing and updating the data regarding patients in the database. UI time for Patient Registration Data takes a bit longer, 195 ms, with a raster time of 24 ms. The FPS maintains at a constant 57 fps, which still shows a quite good performance about the processing of patient registration data. Real-Time Queue testing assesses the performance of the application in terms of updating the queue in real-time and is quite critical, especially when many patients queue up at one time. With the Real-Time Queue feature, UI time increased to 210ms with a raster time of 25ms where FPS was slightly lower at 56 fps because updates in real-time on the queue require more processing. This is still, however, within the bounds of good responsiveness. The next feature is Appointment Reminder, the lowest UI Time of 175ms, with 19ms of Raster time and an FPS of 59 fps, which represents very good and responsive performance to provide appointment reminders to the patients. Meanwhile, the search for a Doctor/Polyclinic recorded a UI time of 190 ms and a raster time of 23 ms while the frames ran smoothly at an average of 58 fps, depicting good responsiveness in the search feature.

Overral, the Healthyline application is quite consistent and responsive; the average FPS for all main features is above 55 fps, which means the application can provide a smooth user experience. For different features, UI and raster times were over 200 ms but still within the limits of acceptability for a health-based application with various real-time updates in data. Conclusion: This testing could prove that the application can handle the user's interactions smoothly on every main feature, waste little time, and shows stability, especially in features updating data directly.

**Table 6. Performance Testing**

| Feature | Average UI (ms) | Average Raster (ms) | Average FPS (fps) |
|---|---|---|---|
| **Patient Login** | 180 | 22 | 58 |
| **Patient Registration Data** | 195 | 24 | 57 |
| **Real-time Queue** | 210 | 27 | 56 |
| **Appointment Reminder** | 175 | 19 | 59 |

| | | | |
|---|---|---|---|
| **Doctor/Polyclinic Search** | 190 | 23 | 58 |

In the patient registration flow in Figure 8 Before logging in, patients can create an account through the register menu, where they fill in their personal data so that during the appointment registration process, they do not need to fill in their personal data again, patients can log in to the Healthyline application after creating an account through the register menu. After logging in, patients will be directed to the main page of the Healthyline application. Patients can then select the Online Registration menu. Next, patients can choose the date for their examination and select the intended polyclinic, then the doctors at that polyclinic will appear and can be chosen by the patients. After selecting a doctor, patients can enter their complaints and the appointment time they wish to convey and select the register button to register. Patients will receive a queue number that can be monitored through the queue menu and will receive a notification for their appointment with the doctor.


**Fig. 8. Flow Patient Registration**

In the notifications reminder flow as shown in Figure 9, to access the notifications menu, patients must first log in and fill in the required credentials. The notification menu can be directly selected by patients from the main menu after they log in. The notification display is in the form of a card that contains the notification title and the notification content. Patients can also mark notifications as read and delete read notifications. Notifications in the form of in-app notifications and push notifications. These reminder notifications will be sent to patients 3 days before the appointment, 1 day before the appointment, and 1 hour before the appointment. With the presence of this quite intensive reminder notification feature, patient attendance rates can be improved, thereby increasing the productivity of the hospital.

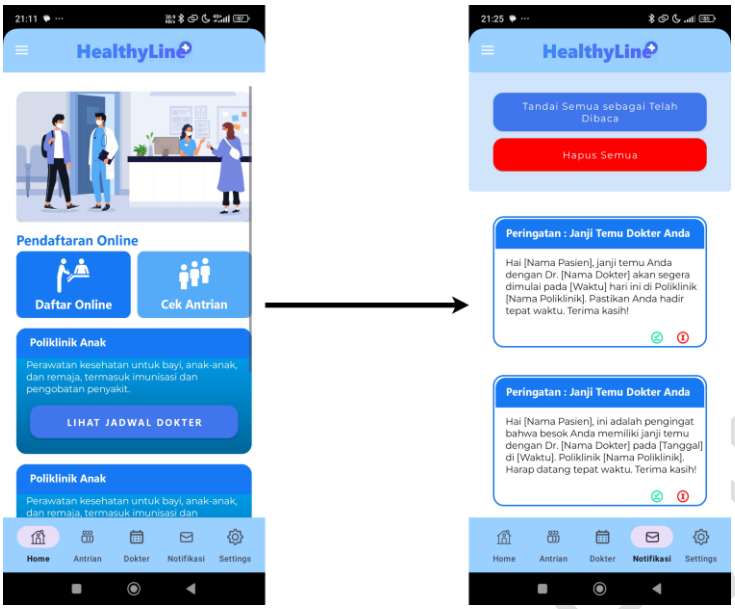


**Fig. 9. Flow Notifications Reminder**

To access the real-time queue menu like Figure 10, patients must first log in. Patients can choose the Queue menu through the bottom menu or the main menu. For the real-time queue, there are 2 types of queues, namely the registration queue and the polyclinic queue. The queue will be updated by hospital staff if there are any changes. With the presence of this real-time queue feature, it is hoped that patient waiting times will become more efficient, thereby improving the patient experience and enhancing the operational performance of the hospital.

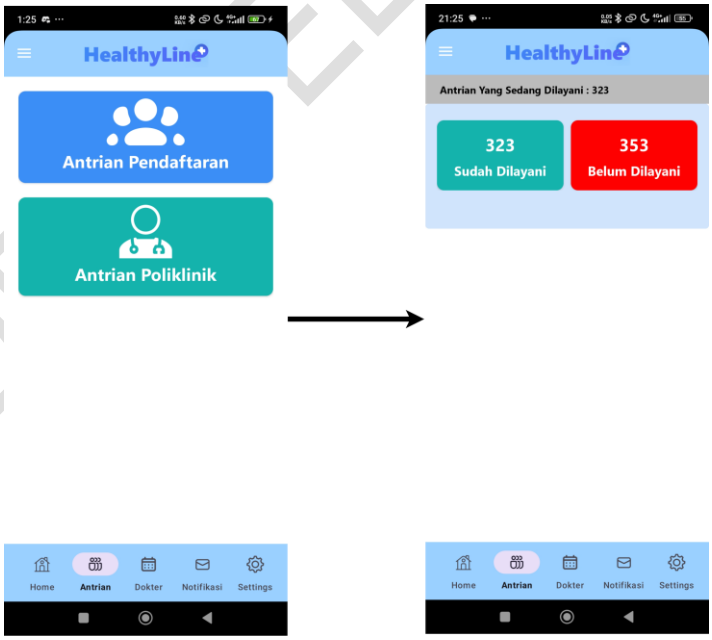


**Fig. 10. Flow Real-Time Queue**

The polyclinic queue will show the real-time queue for each polyclinic in the hospital as shown in Figure 11, making it easier for patients to see the queue in their respective polyclinics.

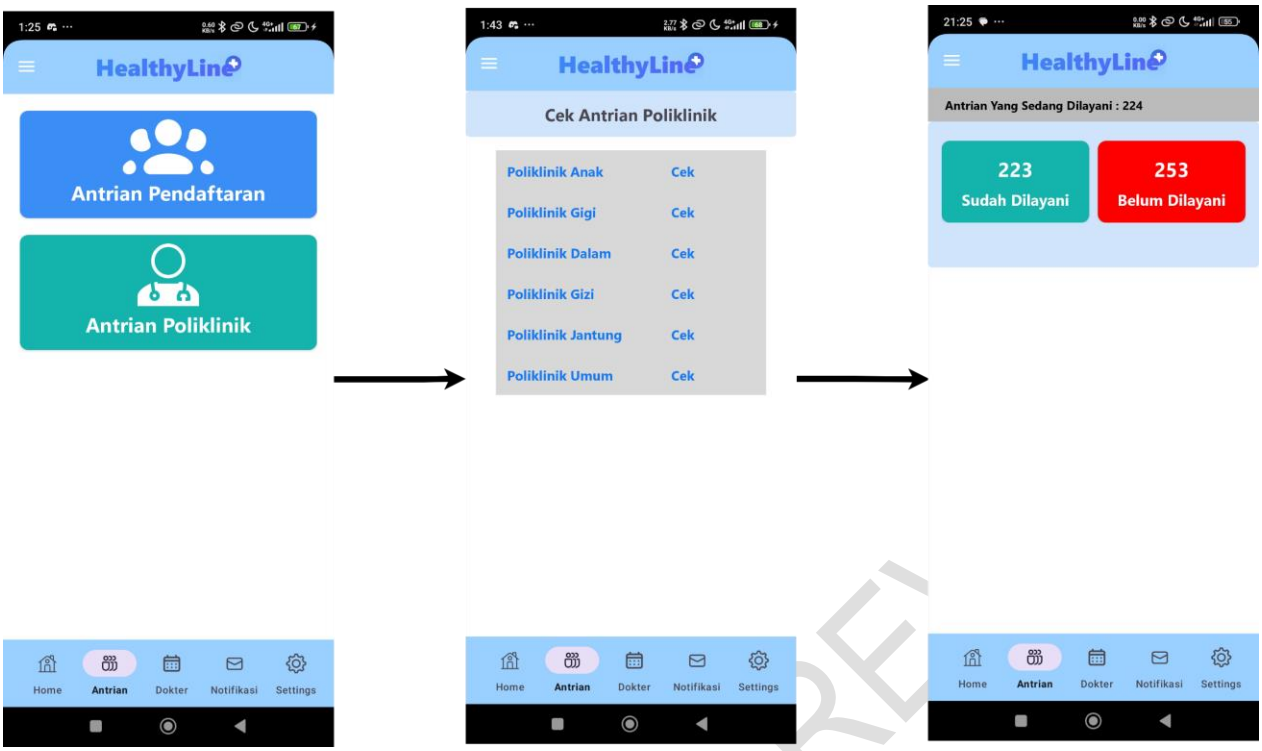


**Fig. 11. Flow Real-Time Polyclinic Queue**

## 3.2 Discussion

The development of the Healthyline application in facing significant challenges in the hospital registration system and queue management. The results of this study indicate that the Healthyline application successfully meets functional needs such as patient registration, real-time queue information, and doctor appointment reminders. This is in line with research on the effectiveness of digital solutions in improving operational efficiency in healthcare settings. Similar to findings in other studies, the application of queuing theory has proven to be a valuable tool in optimizing patient flow and resource utilization in healthcare settings. With real-time information, patient anxiety related to waiting times can be reduced, thereby increasing patient satisfaction. Additionally, the appointment reminder feature is crucial for maintaining hospital productivity in terms of reducing patient no-shows.

However, several aspects require further examination. For example, although the application functions well under normal conditions, its performance during peak hours remains uncertain. Although the application effectively handles user data, potential security issues in the healthcare environment must also be addressed proactively. From the performance testing results, the average response is acceptable, but the author again reminds that the high user load needs to be evaluated to improve reliability. Overall, the findings indicate that Healthyline has the potential to significantly improve hospital registration efficiency, but continuous monitoring is crucial to enhance its effectiveness.

## 4. CONCLUSION

Based on the research conducted, it can be concluded that the development of the Healthyline application effectively addresses the challenges related to hospital registration and queue management. This application facilitates online patient registration, provides real-time queue information, and sends appointment reminders, thereby enhancing patient experience and operational efficiency of healthcare services. There are findings that show Healthyline can significantly reduce patients' anxiety levels regarding waiting times and increase patients' attendance rates for medical appointments, which is crucial in maintaining hospital productivity.

The implications of this research suggest that integrating digital solutions into healthcare practices can create a more structured and efficient system, streamlining operations and improving patient satisfaction. Additionally, technology-based health solutions are developed to align with global trends.

For future research, it is essential to conduct further evaluations of the Healthyline application, particularly its performance during peak hours, to ensure it can handle high user loads effectively. Addressing potential security concerns in handling patient data is also critical, and future studies should explore robust security measures to protect sensitive information. Moreover, the application has not yet been implemented in a hospital setting, which limits the current findings to the stages of development, testing, and maintenance. Therefore, real-world implementation studies are necessary to assess the application's impact on hospital operations and patient outcomes over time.

Other studies may still discover the additional features that can be integrated into Healthyline to further improve it. The inclusions of telemedicine, for example, might provide the opportunity for patients to consult even from remote areas could expand health service to more people, especially those in isolated and resource-poor communities. Future research should measure the long-term impact of Healthyline on hospital operations and patient outcomes. Secondly, the application needs to be closely monitored and updated so that it responds well to evolving patient and health professional needs. Generally, this research substantiates that the Healthyline application has a capacity to make enormous improvements in hospital registration processes, improvement in patient engagement, and finally contribute to improved health outcomes.

**Disclaimer (Artificial intelligence)**

The author(s) hereby declare that generative AI technologies, specifically ChatGPT and Blackbox AI, have been used during the writing and editing of this manuscript. The following details outline the usage of these AI technologies:

1. **Technology Used**: ChatGPT (OpenAI)
   a. **Version**: GPT-3.5
   b. **Purpose**: To assist in generating text, refining arguments, and improving clarity in various sections of the manuscript.
2. **Technology Used**: Blackbox AI
   **Purpose**: To enhance the quality of the writing and ensure coherence throughout the manuscript.
3. **AI Check**: The manuscript was checked for AI-generated content using GPT Zero, which indicated an AI contribution of approximately 8.6%.

The author(s) confirm that the use of AI technologies was aimed at improving the overall quality of the manuscript while ensuring that the final content reflects their original ideas and research findings.

## REFERENCES


[1] A. Torab-Miandoab, T. Samad-Soltani, A. Jodati, and P. Rezaei-Hachesu, "Interoperability of heterogeneous health information systems: a systematic literature review," *BMC Med Inform Decis Mak*, vol. 23, no. 1, p. 18, Jan. 2023, doi: 10.1186/s12911-023-02115-5.

[2] A. Sheikh *et al.*, "Health information technology and digital innovation for national learning health and care systems," *Lancet Digit Health*, vol. 3, no. 6, pp. e383–e396, Jun. 2021, doi: 10.1016/S2589-7500(21)00005-4.

[3] M. Tyagi *et al.*, "Impact of application of queuing theory on operational efficiency of patient registration," *Med J Armed Forces India*, vol. 79, no. 3, pp. 300–308, May 2023, doi: 10.1016/j.mjafi.2021.06.028.

[4] M. Darwis, K. Nawangwulan, D. Ekawaty, A. Imran, Y. Yusfik, and C. Author, "International License Publish : Association of Indonesian Teachers and Lecturers International Journal of Health Sciences (IJHS) Volume 1 | Number 4 | Desember 2023 | International Journal of Health Sciences is licensed under a Creative Commons Attribution 4.0 International License Publish : Association of Indonesian Teachers and," *Lecturers International Journal of Health Sciences (IJHS)*, vol. 1, 2023, doi: 10.59585/ijhs.

[5] S. Tian, W. Yang, J. M. Le Grange, P. Wang, W. Huang, and Z. Ye, “Smart healthcare: making medical care more intelligent,” *Global Health Journal*, vol. 3, no. 3, pp. 62–65, Sep. 2019, doi: 10.1016/j.glohj.2019.07.001.
[6] Menteri Kesehatan Republik Indonesia, “Regulation of the Minister of Health of the Republic of Indonesia Number 24 of 2022 Concerning Medical Records.” [Online]. Available: https://yankes.kemkes.go.id/unduhan/fileunduhan_1662611251_882318.pdf
[7] M. Ahmad Baballe, A. Mansur Gambale, A. Shehu Bari, A. Sani Lawan, and R. Jafar Suleiman, “Issues With Our Hospitals’ Queue Management Information Systems 1* Tmp Universal Journal Of Research And Review Archives Volume 1 │Issue 2│Year 2022│Oct-Dec 2022 Received Date Revised Date Accepted Date,” 2022. [Online]. Available: www.tmp.twistingmemoirs.com
[8] S. O. Opon, W. M. Tenambergen, and K. M. Njoroge, “The effect of patient reminders in reducing missed appointment in medical settings: a systematic review,” 2020, *Pan African Medical Journal*. doi: 10.11604/pamj-oh.2020.2.9.21839.
[9] M. I. M. Salleh, R. Abdullah, and N. Zakaria, “Evaluating the effects of electronic health records system adoption on the performance of Malaysian health care providers,” *BMC Med Inform Decis Mak*, vol. 21, no. 1, p. 75, Dec. 2021, doi: 10.1186/s12911-021-01447-4.
[10] D. Dhagarra, M. Goswami, and G. Kumar, “Impact of Trust and Privacy Concerns on Technology Acceptance in Healthcare: An Indian Perspective,” *Int J Med Inform*, vol. 141, p. 104164, Sep. 2020, doi: 10.1016/j.ijmedinf.2020.104164.
[11] P. Dyana Christasani, Y. Wijoyo, T. Siwi Hartayu, and A. Widayati, “A multifaceted review journal in the field of pharmacy Implementation of Hospital Information System in Indonesia: A Review,” *Systematic Review Pharmacy*, vol. 12, no. 7, pp. 499–503, 2021, [Online]. Available: http://garuda.ristekbrin.go.id
[12] S. Pomalingo and F. A. Twince Tobing, “Optimizing Patient Registration Process through Online Admission Application: A Scrum Approach,” *Jurnal Bumigora Information Technology (BITe)*, 2023, doi: 10.30812/bite/v5i1.3007.
[13] M. M. Fajar, L. R. Ilmi, I. Sevtiyani, B. Wicaksono, and D. Mardiyanti, “Optimizing Outpatient Services: User Interface Design of Mobile-Based Online Registration Application,” *Indonesian of Health Information Management Journal (INOHIM)*, vol. 11, no. 1, pp. 29–36, Jun. 2023, doi: 10.47007/inohim.v11i1.494.
[14] R. D. Asworowati, D. Mustomi, P. R. Adawia, A. D. Suhendra, A. Natong, and M. Cahya Ningrum, “Design of Mobile-Based Outpatient Registration Information System at Gizar Mother and Child Hospital,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 4, pp. 542–549, Oct. 2023, doi: 10.47233/jteksis.v5i4.1059.
[15] M. I. H. -, “Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management,” *International Journal For Multidisciplinary Research*, vol. 5, no. 5, Sep. 2023, doi: 10.36948/ijfmr.2023.v05i05.6223.
[16] F. Fatmariani, A. Saputra, and L. G. Sari, “Web-based Ukp Public Health Center Services System Using the Waterfall Method,” *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 1, pp. 302–314, Jan. 2024, doi: 10.47709/cnahpc.v6i1.3380.
[17] P. K. Yeng, S. D. Wolthusen, and B. Yang, *Comparative Analysis Of Software Development Methodologies For Security Requirement Analysis: Towards Healthcare Security Practice*. 2020.
[18] B. V. Thummadi and K. Lyytinen, “How much method-in-use matters? A case study of agile and waterfall software projects and their design routine variation,” *J Assoc Inf Syst*, vol. 21, no. 4, pp. 864–900, Jul. 2020, doi: 10.17705/1jais.00623.
[19] K. A. Seputra, A. A. G. Y. Paramartha, G. A. Pradnyana, and K. Y. E. Aryanto, “A Middleware Applications Design for Health Information Sharing,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 3, pp. 321–332, Jun. 2024, doi: 10.29207/resti.v8i3.5707.
[20] M. Realis Murphi, R. Sidauruk, H. Heryanto, and F. Aziz, “The Role of Management Information Systems (MIS) in Improving Hospital Service Effectiveness and Efficiency,” 2021. [Online]. Available: http://www.openjournal.unpam.ac.id/index.php/SNH
[21] Dr. S. Suriya and N. S., “Design of UML Diagrams for WEBMED - Healthcare Service System Services,” *EAI Endorsed Transactions on e-Learning*, vol. 8, no. 1, p. e5, Feb. 2023, doi: 10.4108/eetel.v8i1.3015.
[22] A. Rozaq and R. Fauzan, *Konsep Perancangan Sistem Informasi Bisnis Digital*. Poliban Press, 2020. [Online]. Available: https://books.google.co.id/books?id=o68mEAAAQBAJ
[23] Tatiana Grin, “Kotlin programming language.” Accessed: Oct. 12, 2024. [Online]. Available: https://kotlinlang.org/assets/kotlin-media-kit.pdf

[24] “Firebase Cloud Messaging.” Accessed: Oct. 12, 2024. [Online]. Available: https://firebase.google.com/docs/cloud-messaging
[25] “Express - Node.js Web Application Framework.” Accessed: Oct. 21, 2024. [Online]. Available: https://expressjs.com/
[26] E. Tang, “Investigating the Effect of the Entity Relationship View on Model-View-Controller (MVC) Web Development,” 2020.
[27] Carl, “Examensarbete 30 hp Comparisons of Server-side Rendering and Client-side Rendering for Web Pages,” 2023.
[28] J. Eriksson, “Migration Of The User Interface Of A Web Application-From Thymeleaf to Angular,” 2022.