

1. Introduction	2
1.1. Motivation.....	2
1.2. Problem Statement	3
2. Background	4
2.1. Overview of Hospital Management Systems	4
2.2. Existing Solutions and Limitations.....	4
2.3. Technologies Used	5
3. Specification	7
3.1. The Pocket Hospital (Consumer).....	7
3.2. Admin Panel	8
4. Design.....	9
4.1. System Overview.....	9
4.2. System Architecture	9
4.3. EER Diagram	10
4.4. Firebase Firestore Structure.....	11
4.5 Use Case Diagram	12
5. Implementation.....	14
5.1. Technology Stack and Development Tools.....	14
5.2. Development Approach	15
5.3. Challenges and Solutions	15
5.4. Testing and Deployment	16
6. Results and Evaluation	17
7. Future Work	18
8. Conclusions	19
9. Reflection on Learning	20
10. References.....	21

1. Introduction

Healthcare is one of the most critical sectors yet many hospitals still rely on manual systems for managing patient records, appointments and pharmacy services. This leads to long waiting times, misplaced records and operational inefficiencies.

Pocket Hospital is designed to digitize hospital services, making patient management easier for both hospital staff and patients. The system provides a mobile application for patients to book appointments, request ambulances, navigate departments, order medicines and access medical reports. On the other hand, the admin panel allows hospital staff to manage users, approve appointments, process pharmacy requests, make reports and oversee hospital analytics.

By integrating Android (Java), MySQL, Firebase, SQLite, Java EE and real-time notifications, this project aims to create a seamless and efficient hospital management system, reducing paperwork and improving overall healthcare accessibility.

1.1. Motivation

This project was inspired by a personal experience. My grandmother is a regular patient at Karapitiya Hospital and every visit feels like a battle long waiting times, misplaced records and endless paperwork. Seeing her and many other patients struggle with these inefficiencies made me realize how much a well-designed digital system could improve the hospital experience.

Currently, patients have to wait in long lines just to get an appointment, manually submit prescriptions for medicines and often deal with delayed or missing medical reports. The hospital staff also faces difficulties managing patient records, scheduling appointments and tracking pharmacy stock due to outdated manual processes.

I chose this project because I believe technology should make healthcare simpler. By creating Pocket Hospital, I aim to streamline hospital services, reduce waiting times and provide patients with easy access to appointments, pharmacy services and medical records; right from their mobile phones. This system isn't just a project for me; it's a way to make a real difference for people like my grandmother and countless others who depend on hospitals for their well-being.

1.2. Problem Statement

Hospitals are meant to provide care and support but at Karapitiya Hospital, the current system often makes things harder for both patients and hospital staff. Most of the processes like appointment booking, pharmacy requests and medical record management are still done manually. This leads to long waiting times, misplaced records and frustrated patients who have to spend hours navigating the system just to get the care they need.

For example, patients have to physically visit the hospital just to book an appointment which not only wastes their time but also overcrowds hospital lobbies. The pharmacy process is slow, requiring patients to stand in long lines and manually submit prescriptions while the hospital struggles to keep track of medicine stock. Emergency services like ambulances are not easily accessed, makes delays in critical situations.

Hospital staff also face challenges as they have to manage hundreds of patient records manually, increasing the chances of errors, miscommunication and lost documents. Without a proper digital system, keeping track of appointments, medical test reports and patient history becomes a stressful and inefficient process.

The Pocket Hospital system aims to solve these problems by automating hospital operations, giving patients an easy-to-use mobile app for booking appointments, requesting ambulances, navigating departments and accessing medical records while providing hospital staff with an admin panel to efficiently manage users, pharmacy stock and appointments. This will help reduce waiting times, paperwork and manual errors, creating a smoother and more accessible healthcare experience for everyone.

2. Background

2.1. Overview of Hospital Management Systems

Hospitals handle a huge amount of information every day patient records, appointments, prescriptions, test reports, billing and emergency services. A Hospital Management System (HMS) is designed to digitize and automate these processes, making healthcare services more efficient, accessible and error-free.

A good HMS helps in:

- Managing Appointments - Patients can book appointments online instead of waiting in long hospital queues.
- Handling Patient Records - Medical histories, test results and prescriptions are stored digitally, reducing paperwork and human errors.
- Pharmacy and Medicine Management - Ensuring hospitals always have the right medicines in stock and preventing delays for patients.
- Emergency Services Coordination - Allowing hospitals to quickly respond to ambulance requests and urgent medical needs.
- Billing and Payments - Automating payments and invoices to make transactions easier for both patients and the hospital.

Many hospitals around the world have already adopted Hospital Management Systems to improve their services. However, some hospitals like Karapitiya Hospital, still rely on manual processes. This project, Pocket Hospital aims to introduce a modern, user-friendly HMS specifically designed to address the challenges faced by Karapitiya Hospital and improve overall healthcare accessibility.

2.2. Existing Solutions and Limitations

In Sri Lanka there are a few Hospital Management Systems available but most of them are not specifically designed for government hospitals like Karapitiya Hospital. These existing systems often focus on private hospitals which have different operational needs and resources. While these solutions aim to automate certain hospital functions, they often fall short in addressing the unique challenges faced by public healthcare institutions.

For example, many of the available systems are expensive and complex, making them inaccessible for government hospitals with limited budgets. Moreover, these systems tend to be tailored to more developed healthcare environments and may not integrate well with local practices and infrastructure. As a result, many hospitals still rely on outdated, manual methods for handling appointments, patient records and pharmacy services.

At Karapitiya Hospital, the current system struggles with inefficiencies such as long waiting times, misplaced patient records and an over-reliance on paper-based processes. These issues are particularly problematic in a government setting, where the patient load is higher and resources are more limited. Existing HMS solutions have not been able to fully address these challenges, leaving gaps that result in frustrating experiences for both patients and staff.

In essence, while there are some systems available, none of them are specifically designed to meet the needs of government hospitals, like Karapitiya Hospital and help them improve accessibility, reduce operational delays and enhance patient care. This is where Pocket Hospital comes in, offering a modern, affordable and efficient solution tailored to the specific requirements of public healthcare institutions in Sri Lanka.

2.3. Technologies Used

The development of Pocket Hospital utilizes a combination of modern technologies to create a seamless and efficient hospital management system.

- **Android (Java):** The mobile app is built using Android with Java. This allows for a secure, user-friendly interface that can run smoothly on a variety of Android devices, making it accessible to a wide range of users.
- **MySQL:** For handling the core data storage (Users), Pocket Hospital relies on MySQL, a powerful relational database management system.
- **OkHttp:** OkHttp is used for handling HTTP requests and ensuring smooth communication between the app and the backend server.

- SQLite: For local data storage, SQLite is used. This allows the app to store certain data on the device itself, providing offline access to important features like appointment information.
- iTextPDF: Pocket Hospital uses iTextPDF to generate and manage PDF documents, such as medical reports. This helps in securely storing and sharing essential documents that need to be easily accessible by patients.
- MPAndroidChart: For visualizing data, such as hospital analytics and patient trends, MPAndroidChart is used. This library helps to create beautiful and interactive charts that display key insights.
- GSON: GSON is utilized to convert Java objects into JSON and vice versa, making it easier to handle and transmit data between the app and the backend in a readable format. This ensures smooth data exchange when fetching and submitting information.
- Google Maps API: The Google Maps API is integrated into Pocket Hospital to provide location-based services. It helps users navigate around the hospital and find specific departments or services, making it easier for patients to locate what they need.
- PayHere Payment Gateway: The PayHere Payment Gateway is integrated into Pocket Hospital to facilitate secure and seamless online transactions. It allows users to make payments for e-pharmacy purchases. The integration ensures a smooth checkout experience while maintaining transaction security and compliance with payment regulations.
- Firebase Firestore: Firebase Firestore is used as a cloud database in Pocket Hospital to store and manage real-time data efficiently.
- Firebase Storage: Firebase Storage is integrated into Pocket Hospital to handle secure file uploads and storage, such as medical prescriptions and reports.

3. Specification

3.1. The Pocket Hospital (Consumer)

The Pocket Hospital system offers a variety of services and operations, enhancing the hospital experience for both patients and hospital staff. Some of these services can be accessed by guests without the need for registration or sign-up, providing instant access to critical services.

- **Registration:**
Patients can easily register on the app by providing their personal details. After registration they can access to all the available services.
- **Make Appointment:**
Registered patients can schedule appointments directly through the app. They can choose the department and sub-department, select an available time slot and add the appointment without the need to visit the hospital in person.
- **Request Ambulance (Direct Call):**
Available to guests and registered users
In case of an emergency, users can request an ambulance directly through the app, connecting them to the hospital's emergency services for immediate assistance.
- **Locate a department:**
Available to guests and registered users
The app provides a hospital map for users to easily locate various departments ensuring a smooth and quick hospital visit.
- **E-Pharmacy:**
Registered Users can browse available medicines, view prices and place orders through the app. Users can also upload prescriptions and check the status of their orders streamlining the medication process.

- **Track My Steps:**
Registered patients can track their daily steps and monitor their health. This feature helps patients stay motivated and active.
- **About Us (Hospital Details and Contact):**
Available to guests and registered users
The app provides a section with important hospital details, such as the hospital's location, services offered and contact information. This feature ensures users can find the information they need without registration.

3.2. Admin Panel

The admin panel is designed for hospital staff to manage the system efficiently. Key operations available through the admin panel include:

- **User Management:**
Hospital staff can manage patients.
- **Appointment Management:**
Admins can review and cancel appointments and monitor appointment availability to avoid overbooking.
- **Pharmacy Management:**
Admins can oversee pharmacy operations, manage inventory and approve medicine orders to ensure that patients receive their prescriptions on time.
- **Reports Management:**
The admin panel allows staff to generate and manage various reports. This helps in analyzing hospital operations and improving services.

4. Design

The design of Pocket Hospital focuses on creating a smooth and efficient digital healthcare experience while ensuring user-friendliness, security and performance. This section explains how different components of the system work together to deliver the required services.

4.1. System Overview

The system is built as a mobile-based platform for both patients and hospital administrators. The front-end is developed using Android (Java), while the backend relies on MySQL, Firebase and SQLite for data management. Secure API communication is handled using OkHttp. The system includes three key interfaces:

- Patient Interface - Allows users to register, book appointments, access e-pharmacy and more.
- Guest Interface - Provides limited access to services like requesting an ambulance, finding a department and browsing about us.
- Admin Panel - Enables hospital staff to manage users, appointments, pharmacy and reports.

4.2. System Architecture

The system follows a client-server architecture where the Android app (client) communicates with a central backend (server) through APIs.

Frontend (Android App)

- Developed using Java
- Uses Google Maps API for hospital navigation
- Includes offline data storage using SQLite
- Visualizes health data using MPAndroidChart

Backend (Server and Database)

- MySQL stores user-related data
- Firebase Firestore handles real-time notifications and data
- OkHttp and GSON manage secure API calls and data transfer

Storage and Reporting

- ITextPDF is used to generate reports
- Shared Preferences and Internal Storage handle user session management

4.3. EER Diagram

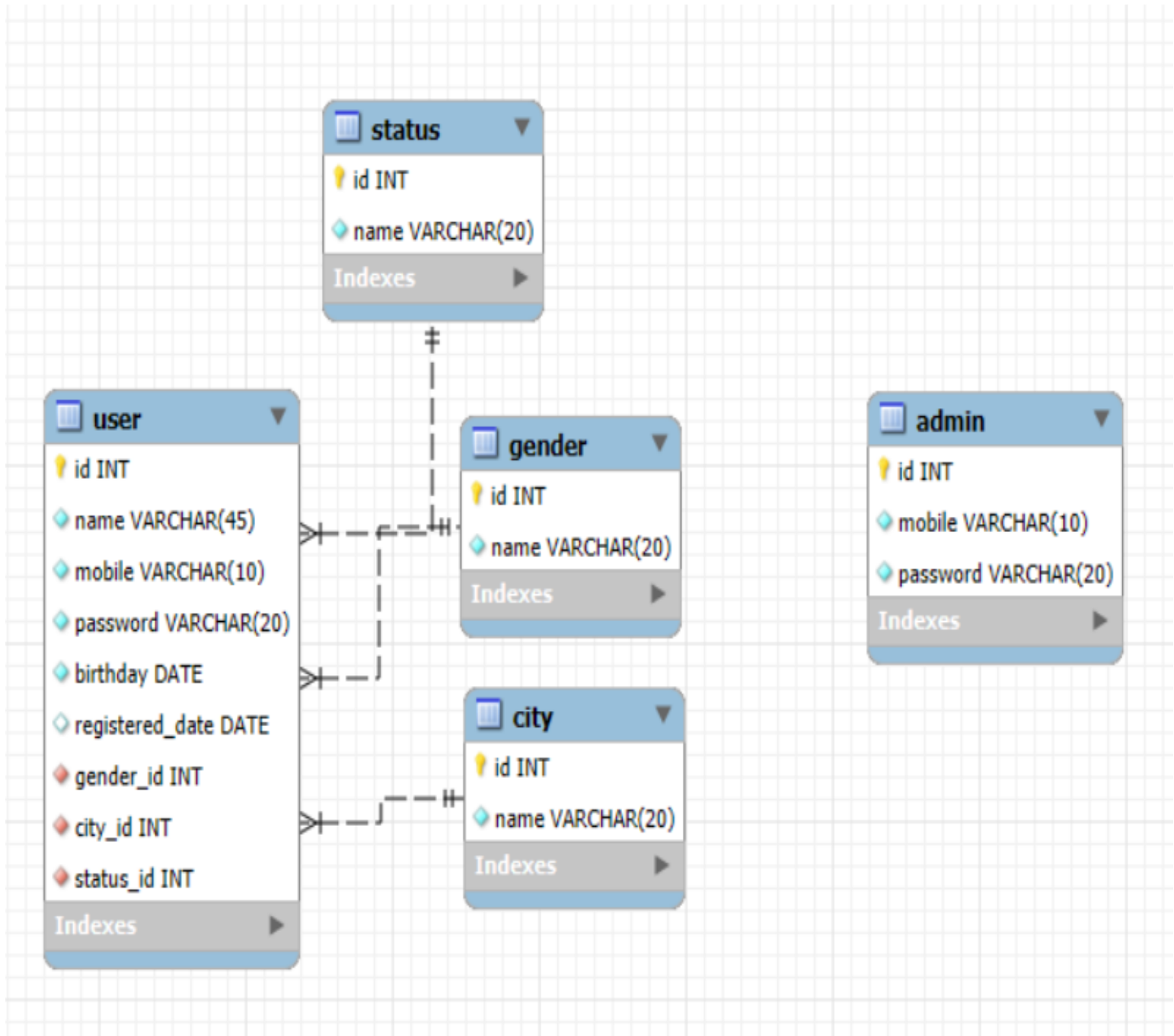


Figure 1: EER Diagram

4.4. Firebase Firestore Structure



Figure 2: Firestore Structure

4.5 Use Case Diagram

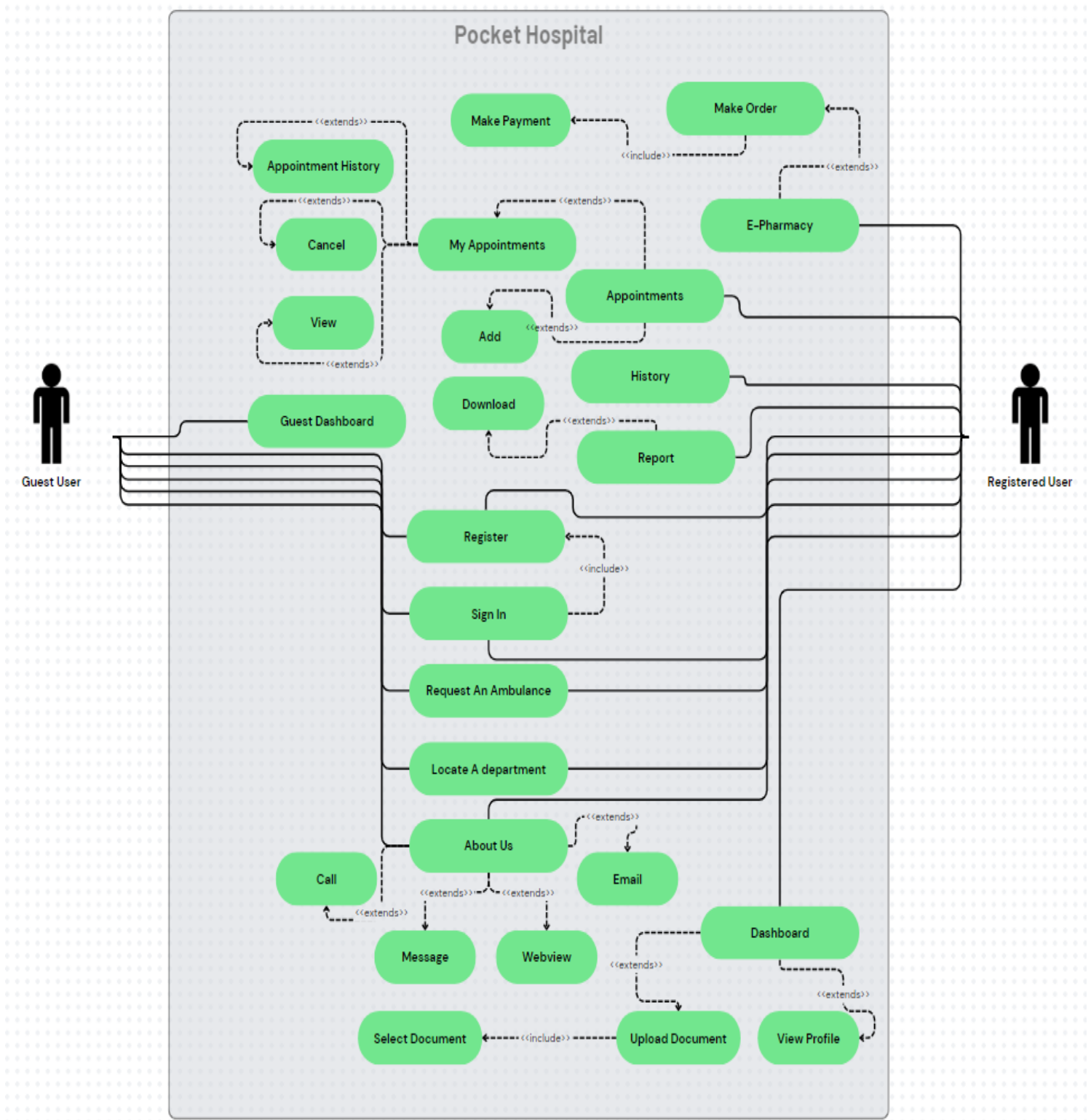


Figure 3: Pocket Hospital use-case diagram

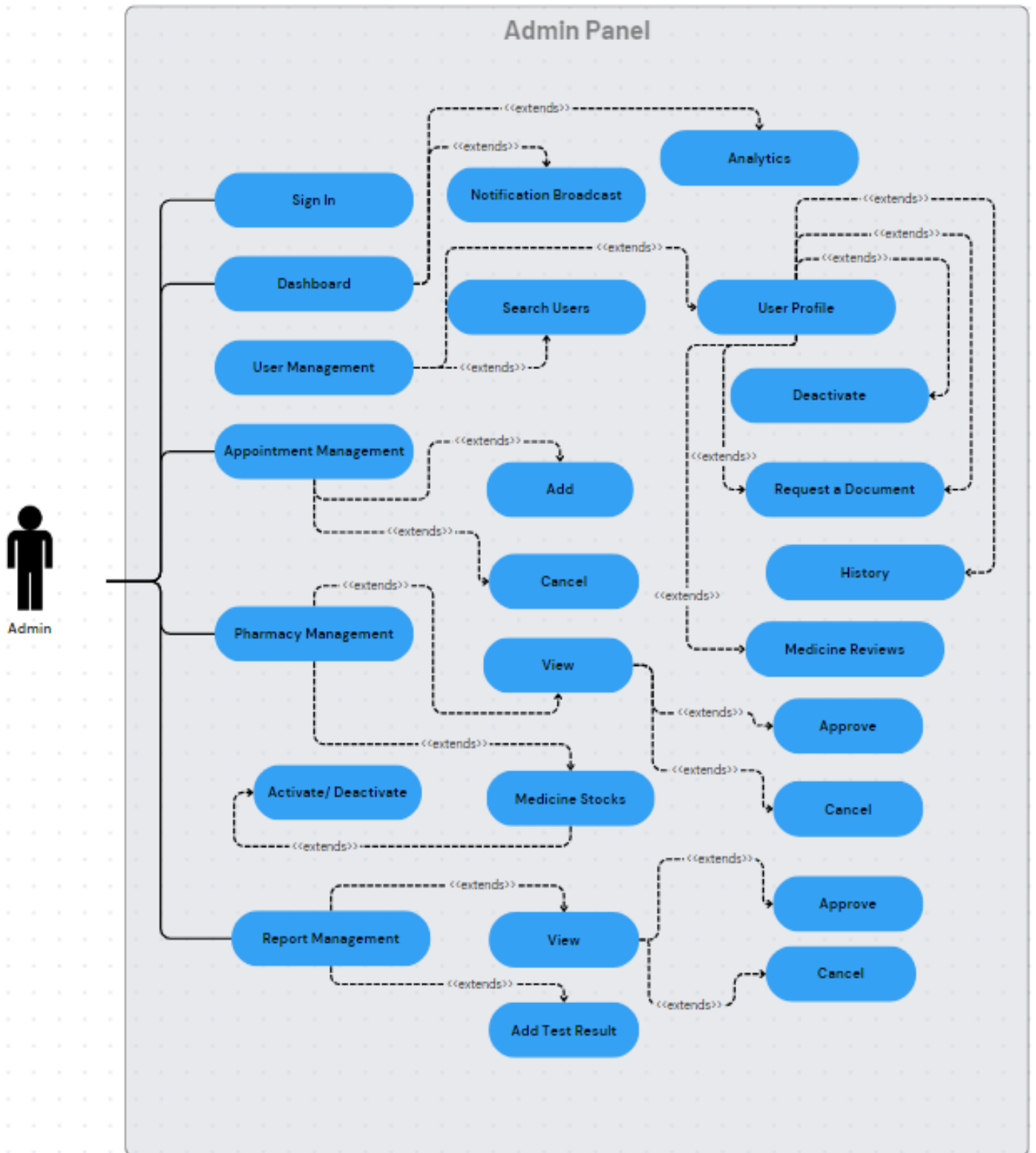


Figure 4: Admin Panel use-case diagram

5. Implementation

The implementation phase of Pocket Hospital was a structured yet dynamic process balancing frontend and backend development simultaneously. Since the project followed a hybrid development approach, I broke tasks down into manageable units-each interface and its corresponding operations, ensuring smooth progress across different features.

5.1. Technology Stack and Development Tools

The project was built using a combination of native Android technologies and backend services to ensure efficiency, scalability and security:

- **Frontend (Android App):** Java (Android), XML for UI, MPAndroidChart (for analytics), Google Maps API (for location-based features)
- **Backend:** Java EE Server using GlassFish, MySQL for structured data, Firebase Firestore for real-time data and services
- **Libraries Used:**
 - GSON - Efficient JSON parsing
 - iTextPDF - Generating and handling PDF reports
 - MPAndroidChart - Data visualization for analytics
 - Google Maps API - Hospital navigation
 - PayHere - Payment Gateway
- **Networking and Storage:** OkHttp for HTTP requests, SQLite and Shared Preferences for local storage
- **Tools:** Android Studio, Netbeans IDE, MySQL Workbench, SQLite query generator and Firebase Console (Google cloud)

5.2. Development Approach

Since the project required handling multiple user types (patients, admins and guests) with distinct functionalities, I followed a simultaneous development approach, ensuring both frontend and backend were evolving in parallel. The implementation started with a prototype, which helped in visualizing the app's flow and refining the core features before diving deep into development.

To streamline development:

- I tackled UI design and core functionalities first (e.g., appointment booking and pharmacy management).
- Simultaneously, backend development handled API creation, database management and request handling.
- As each module was completed, it was integrated and tested in smaller units before moving to the next phase.

5.3. Challenges and Solutions

No project comes without its hurdles and Pocket Hospital had its share of challenges:

API Integration Issues:

While integrating Google APIs, I ran into billing-related issues due to account verification. Resolving this required contacting the bank and ensuring the payment method met Google's requirements.

For other APIs, I referred to the official documentation and debugging tools to troubleshoot connectivity issues.

Time Management Struggles:

Managing time efficiently was a challenge, given the project's complexity.

I prioritized critical functionalities first, avoiding feature creep and sticking to essential features before refining secondary ones.

Paid Library Limitations:

Some third-party libraries required paid versions, which wasn't ideal.

I researched and implemented free, open-source alternatives that provided similar functionality without compromising quality.

5.4. Testing and Deployment

To ensure the Pocket Hospital app worked seamlessly, I used a combination of:

- **Unit Testing:** Testing core components like authentication, API responses and database queries.
- **Manual Testing:** Running end-to-end tests on a physical Android device to check real-world usability, responsiveness and user experience.

6. Results and Evaluation

After completing the development and testing phase the Pocket Hospital application successfully met its core objectives. All the essential features such as appointment scheduling, e-pharmacy, emergency requests and medical report access are fully functional and working as intended. However, some optional features like live chat and medical shopping without a valid prescription were removed due to practicality and efficiency concerns.

During testing, I had friends and relatives try out the app and the feedback was overwhelmingly positive. They found the interface easy to navigate and the core functionalities worked smoothly. Their feedback also provided valuable insights for future updates.

In terms of performance, the application runs well under normal conditions. However, I identified some potential scalability issues, particularly with Firebase and the Google Maps API. If the app reaches its usage quota limits, certain features could become temporarily unavailable. Additionally, while GlassFish was sufficient for initial development and testing, I believe a more optimized and scalable server setup would improve performance in a real-world hospital environment.

One of the biggest challenges was dealing with a higher number of bugs than expected. Some issues caused major frustration during debugging, but through rigorous testing and troubleshooting, I managed to resolve them all.

Overall, the Pocket Hospital project turned out to be a successful implementation, achieving its goal of digitizing hospital services and improving accessibility for both patients and administrators.

7. Future Work

Looking ahead, there are several ways to enhance and expand Pocket Hospital to make it even more efficient and impactful.

- **Scalability and Performance Upgrades**
While the app runs smoothly, switching to an enterprise-level server would significantly boost performance, especially as more users come onboard. Additionally, upgrading Google Cloud services (Firebase, Maps API, etc.) will help prevent quota limitations and ensure a seamless experience.
- **Feature Enhancements**
Several key improvements are planned, including:
 - **Better Appointment Scheduling** - Making it more intuitive and efficient.
 - **AI-Based Recommendations** - Providing users with personalized medical insights.
 - **Live Chat with AI** - Assisting users with medical inquiries in real time.
 - **Optimized E-Pharmacy** - Smoother ordering and verification process.
 - **More Medical Assistance Tools** - Expanding features like the step tracker and adding new health monitoring tools.
- **Security Improvements**
Strengthening authentication mechanisms and implementing better data encryption will be a priority to ensure patient data remains secure and private.
- **Future UI/UX Upgrades**
Based on current feedback, the UI/UX is already well-received. However, future updates will be driven by user feedback to ensure continuous improvement.
- **Expanding Beyond Karapitiya Hospital**
Once the app proves successful in Karapitiya Hospital, the next big step is to expand it island-wide to all government hospitals, medical schools and medical departments. This would significantly enhance digital healthcare operations across Sri Lanka.

With these future improvements, Pocket Hospital aims to become a game-changer in digital healthcare, offering a seamless, secure and user-friendly experience for both patients and medical professionals.

8. Conclusions

Developing Pocket Hospital has been a challenging yet rewarding experience. The project started with the vision of digitizing hospital services and making healthcare more accessible; and I'm proud to say that it successfully brings appointment scheduling, e-pharmacy, medical records access and hospital navigation into one seamless mobile platform.

Throughout the development process, I faced several hurdles, from API integration issues (especially with Google services) to time management struggles and paid library limitations. However, overcoming these challenges through research, official documentation and alternative solutions helped shape a good and functional application.

The app has been tested thoroughly and all core features work flawlessly. Some optional features, like live chat and medical shopping without a valid prescription were removed to streamline the experience. While the performance is stable, there's room for improvement, especially with server optimizations and cloud service upgrades.

The feedback from friends, family and early testers has been overwhelmingly positive, with some great insights for future upgrades. Moving forward, the goal is to enhance security, improve AI-driven features and eventually scale the project to hospitals across Sri Lanka.

Overall, Pocket Hospital is a strong foundation for modernizing healthcare services and with continuous improvements, it has the potential to revolutionize hospital management and patient care on a larger scale.

9. Reflection on Learning

Working on Pocket Hospital has been an incredible learning experience, pushing me beyond my comfort zone in both technical and project management aspects.

One of the biggest lessons I learned was the importance of planning and time management. Balancing multiple tasks; UI/UX design, backend development, API integration and database management was overwhelming at times and I realized that better task prioritization could have saved me from last minute stress.

From a technical perspective, I gained hands-on experience in hybrid development, breaking down tasks into interfaces and operations helped me manage the complexity of the system. Integrating third-party APIs (Google Maps, Firebase, PayHere, ITextPDF and MPAndroidChart) was challenging but referring to official documentation and troubleshooting unexpected issues (especially payment verification with Google API) taught me resilience and problem-solving.

I also faced real-world challenges like paid library restrictions, API quotas and server limitations. Finding alternative solutions to paid libraries and understanding server scaling made me appreciate the importance of cost-effective decision-making in software development.

Beyond coding, this project reinforced the value of user feedback. The early testers provided insights that helped refine the app and I realized that no matter how perfect a system seems, real world users will always highlight areas for improvement.

Overall, this project not only strengthened my technical skills in Java, Firebase, MySQL and Android development but also taught me adaptability, patience and the ability to tackle real world problems head-on. Moving forward, I feel much more confident in handling large-scale software projects and making strategic decisions for better performance and user experience.

10. References

- Pichon (Icons) - Flaticon
This resource provided free icons for the project, helping to enhance the visual appeal and user experience. I used various icons from Pichon to represent key functionalities throughout the app.
<https://www.flaticon.com>
- Google Fonts - Google Fonts
I used Google Fonts to ensure the app had a consistent and modern typography style. The fonts chosen contributed to the overall clean and readable UI.
<https://fonts.google.com>
- Google Maps API Documentation - Google Developers
This guide provided in-depth information on how to integrate Google Maps into the app for location-based services. The detailed instructions helped me understand key functionalities such as geolocation and department tracking.
<https://developers.google.com/maps>
- Firebase Documentation – Firebase
The Firebase documentation was instrumental in integrating various Firebase services like real-time database and push notifications. It provided clarity on how to manage and utilize Firebase for the backend operations.
<https://firebase.google.com/docs>
- MPAndroidChart Documentation – GitHub
This resource provided a charting library that I used to visualize data like patient appointment statistics and pharmacy usage. The guide helped with understanding the customization options for different chart types.
<https://github.com/PhilJay/MPAndroidChart>
- iText PDF Documentation – iText
This guide provided resources on how to generate and manipulate PDF files within the app. It was used for generating PDF reports, appointment details, and e-pharmacy receipts.
<https://itextpdf.com>

- **Android Developer Documentation – Android Developers**
The official Android documentation offered comprehensive information and best practices for developing Android apps. It was essential for understanding core Android concepts like views, activities, intents, and UI/UX guidelines.
<https://developer.android.com/docs>
- **Various Online Forums and Communities (StackOverflow, GitHub and GeeksforGeeks)**
These forums played a crucial role in resolving coding challenges and understanding best practices. They provided community-driven solutions and suggestions that helped overcome technical hurdles.
<https://stackoverflow.com>, <https://github.com>, <https://www.geeksforgeeks.org>