

ANDROID-BASED HEALTH POST MANAGEMENT APPLICATION DESIGN CLINICS IN INDONESIA

Dwi Kartinah

Article History

Received : June 2023

Revised : July 2023

Accepted : July 2023

Published : July 2023

Corresponding author*:

dwikartinah@yahoo.co.id

Cite This Article:

D. Kartinah, "ANDROID-BASED HEALTH POST MANAGEMENT APPLICATION DESIGN CLINICS IN INDONESIA", IJST, vol. 2, no. 2, pp. 36-41, Jul. 2023.

Abstract: This research aims to design and develop an Android-based application that can help manage Pos Sehat, especially in rural areas of Indonesia, in order to increase the efficiency and effectiveness of health services. The focus of this research is developing applications for Pos Sehat managers, with important features such as patient monitoring, viewing patient biodata, tracking disease history, and drug recommendations. The research method involves planning research, studying literature, developing applications using Android Kotlin for UI/UX, Node.js for backends, and cloud services or Firebase for data storage, as well as application testing and evaluation. The results of the study show that the application developed can increase the efficiency and effectiveness of the management of the Health Post, facilitate access to patient and drug information, and assist in the decision-making process regarding the management of the Health Post. This research is expected to be a reference for similar research and inspire further development in the field of technology-based health management.

Keywords: Application, Health Post, Android.

DOI:

<https://doi.org/10.56127/ijst.v2i2.780>

INTRODUCTION

Technological developments in this modern era have had a significant impact on various aspects of life, including the health sector. Health is an important aspect that requires efficient and effective treatment. On the other hand, many health facilities still rely on manual systems in managing patient data and drug stocks, which of course requires considerable time and effort. One of them is a health post managed by the writer's mother. The health post is a clinic that collects patient data and stocks of medicines manually. This manual method has many limitations, such as the possibility of losing documents, difficulties in finding data, and the potential for errors in data collection. In addition, this activity can also affect time efficiency and operational costs.

This project was initiated as a solution to this problem. The author is trying to develop an application that can help his mother manage patient data and drug stocks at the health post that he manages. This application is designed to optimize the process of data collection, management and monitoring of patients and drug stocks more easily, quickly and accurately. The features provided in this application include: displaying patient biodata such as name, age, address and history of stored diseases in databases; store and display data on the number of drugs in accordance with the amount of existing stock; provide drug recommendations to patients based on the type of disease and drug stock available; limiting the opportunity for patients to go to the clinic for treatment once a month; and displays the patient's medical history in the previous months along with their progress and symptoms.

In developing this application, the authors use various technologies and services that have been proven to be effective and efficient. Node.js is used for creating APIs, Google Cloud Platform (Cloud Run) is used for API deployment, Firestore is used as a database, and Firebase is used for authentication and various other needs. With this application, it is hoped that it can assist staff in managing health posts by better and more efficient, as well as assisting patients in obtaining more optimal services.

Scope of Problem

To ensure that the development of this application remains focused on the main goal, there are some limitations that need to be understood. This application is specifically designed for the internal management of healthy posts, not for general or larger-scale use. Therefore, this application only provides a login feature and does not include a registration feature, according to the needs of healthy post administrators as a single user. On the data side, this application focuses on patient biodata, their medical history, and drug stock. The medical history shown is only data from the previous months along with the associated developments and

symptoms. In addition, the application provides drug recommendations based on the type of disease and available drug stocks, but the final drug determination depends entirely on the medical considerations of the author's mother. Other aspects of clinical management, such as scheduling, personnel management, and other functions are not included in the scope of this application. In addition, this application relies on Node.js, Google Cloud Platform (Cloud Run), Firestore, and Firebase in its development process. Issues that may arise regarding these services and technologies are beyond the scope of the issues in this project.

Research Purposes

This research aims to develop an application that can help the efficiency and effectiveness of the health post management. The application developed in this study is expected to be able to help facilitate the management of patient data and drug stock at clinic health posts throughout Indonesia.

Application Overview

The Android-based Pos Sehat Admin mobile application is aimed at Pos Sehat administrators to make it easier for them to input patient data and receive the desired output more easily and efficiently. The admin healthy post Android application consists of several pages to support the needs of users in using the healthy post application. The page is as follows:

1. Login page, used for authentication and authorization from the Healthy Pos Admin.
2. Dashboard, the main page of the application that displays the features menu of the Pos Sehat Application. It consists of a log out button, View Patient button to display a list of patients, View Medicines button to view a list of medicines owned, and the Add Patient button used to add patients.
3. View Patient, this page will display all patient data that is owned which displays patient information such as Name, Age, Weight, Address, Blood Pressure then when the patient data is clicked it will be directed to Patient Details which displays patient data but with additional features, namely drug input and will be displayed in the patient's drug history.
4. See Drugs. This page will display all drug data that is owned and also displays drug information such as name, type, stock, expired. At the bottom of this page there is a "plus" button which will be directed to the Add Drug page. Here you will enter data such as name, type, quantity and expiration of the drug you want to input.
5. Add Patient, this page will display several textbox lines that are used to input data from the new patient to be inputted, the data requested includes Name, Age, Weight, Address and Blood Pressure.

RESEARCH METHODS

The research method used to design and develop this application is Waterfall, one of the Software Development Life Cycle (SDLC) methods. This method involves several stages, including analysis, design, implementation, testing and maintenance.

1. Analysis: This stage involves gathering information on the application's requirements. This includes understanding the work process of healthcare workstations, patient data requirements and drug inventory management.
2. Design: Having gathered the requirements, the next process is to design the system according to the needs that have been identified. This includes the design of application interfaces, databases and other processes.
3. Implementation: At this stage, the system design will be implemented in a usable application. This Android application will be created using the Kotlin programming language and Android Studio, while Node.js will be used to create the API. Firestore will be used as the database, and Google Cloud Platform (Cloud Run) will be used to deploy the API.
4. Testing: Once the application has been developed, a series of tests will be carried out to ensure that it functions correctly and meets the requirements identified during the analysis stage.

Design

At this stage the author will make the necessary designs and designs in building the Android-based Admin Health Post application. The design process carried out is as follows:

1. Creating a Navigation Structure
2. Creating Login Pages
3. Making Dashboards
4. Design menu Patient

The following is a wireframe from the Pos Sehat Admin application which will display arrows as the path between pages to other pages.

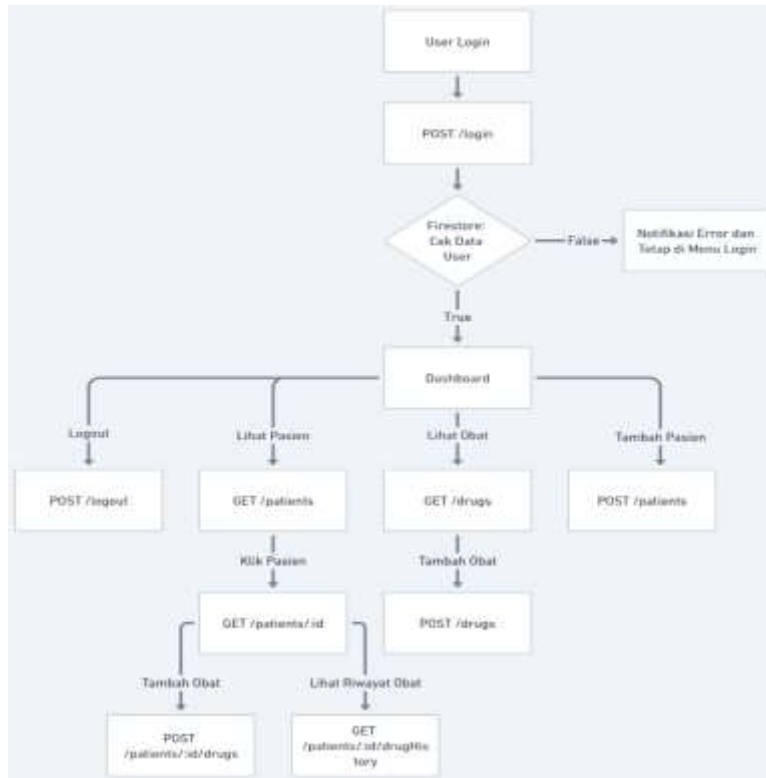


Figure 1. Admin Health Post Wireframe

The following is a Flowchart of the Pos Sehat Admin application which will display the flow and functions of each API endpoint used based on the features described in the wireframe above.

Navigation Structure

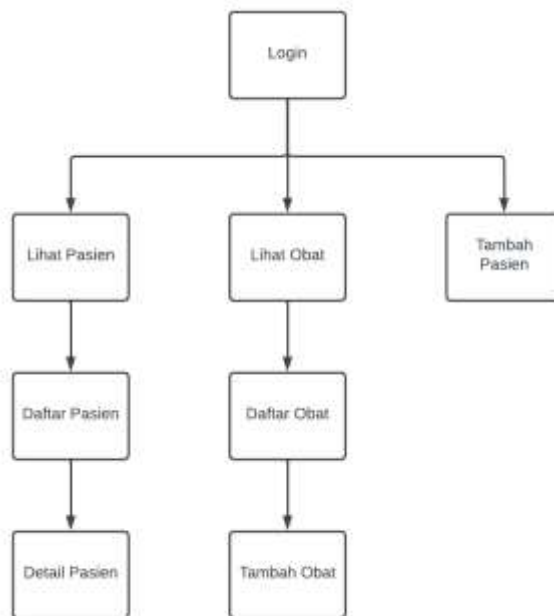


Figure 2. Navigation Structure

The navigation structure is a plan that contains the flow of the application. The navigation structure in this application uses a hierarchical navigation structure, this navigation structure can be seen in Figure.

Login Page

The login page is the first point of user interaction in the "Pos Sehat Admin" application. This page has an important role to authenticate and secure access to applications. In this sub-chapter, we will discuss the login page design process. The Login page is designed with two text inputs, namely "Email" and "Password", as well as a "Login" button. Interface design must reflect simplicity and focus on the main function, namely user authentication.

1. "Email" input: This input is designed to enter an admin email. This feature should handle real-time email validity checks, ensuring that the format the user enters matches the common email format. In addition, the application must also provide clear feedback if the email format is invalid.
2. Input "Password": This input is designed to enter the password of the admin account. For security reasons, the characters entered must be masked or hidden. Additionally, options to show and hide passwords can be added to improve user experience.
3. "Login" button: This button is responsible for sending information that has been entered into the system for the authentication process. If the data entered is correct and matches the data in the system, the user will be directed to the application Dashboard.

At this stage of the analysis, the login process flow is also an important part that must be considered. If the login is successful, the user will be immediately directed to the Dashboard. However, if an error occurs, for example the email or password entered does not match, the application must provide a clear error message to the user. In addition, another consideration that must be considered is how to handle users who have forgotten their password, by providing a "Forgot Password" feature that will help them reset their password via a registered email. All of these elements need to be carefully designed and implemented to ensure a good user experience and application system security. Furthermore, this design will be implemented and tested to ensure that all functions work properly and according to user expectations.



Figure 3. Login Page

Dashboards

The dashboard is a crucial component in the application that provides an overview of the available functions and features. Dashboards must be designed with care to ensure easy and intuitive navigation for the user. At this stage, what is being done is to design the Dashboard appearance of the "Pos Sehat Admin" application which has been planned and analyzed for its needs. It covers the visual design, user interaction and workflow for every main element in the dashboard. The design of this application involves several key elements, including the menu "View Patients", "View Medicines", "Add Patient", and the "Log Out" button.

1. The "View Patients" menu is a feature where the admin can see a list of all registered patients. The interface design for this menu must pay attention to how information is presented in a clear and easy to read manner, as well as how the admin can perform interactions such as searching for certain patients, sorting lists, and navigating patient details.
2. The "View Drugs" menu provides a list of all drugs in the database. The user interface must be designed in such a way as to make it easier for admins to search for specific drugs, sort by category, and view drug details.

3. The "Add Patient" feature allows admins to register new patients into the system. The user interface design here should focus on ease of use and efficiency, bearing in mind that this is a task that an admin might often do. Registration forms must be clear and easy to fill out and must provide appropriate feedback to ensure data accuracy.

The "Log Out" button provides a function for the admin to log out of the account safely. Although simple, it is important to ensure that these buttons are clearly and intuitively placed in the interface design.



Figure 4. Dashboard

Design Menu Patient

The "Design Menu Patient" view is designed with the main focus to make it easier for admins to view and manage patient data. This sub-chapter will explain further about how this display is designed and implemented in the "Pos Sehat Admin" application. On the "View Patient" page, we use a list structure to display patient data. This structure was chosen because of its efficiency and ease in displaying large and continuous data. Each item on this list will display important information about the patient such as name and age.

1. Patient List: This list is designed to display patient data retrieved from the API. This data will be updated in real-time every time there is a change on the server. Using the API allows patient data to always be up-to-date and consistent across all devices that use this application.
2. Text Name and Age: Each item in the list displays some important details about the patient, including name and age. This text is designed with an easy-to-read font and a large enough size for user comfort. Color and lighting are also considered to ensure text is clearly legible in various lighting conditions.

This design also prepares for each text view so that it can be clicked and will direct the user to patient details according to the patient's name that is clicked. Overall, the "View Patient" view design is designed to make it easier for admins to view and manage patient data efficiently and effectively. The simple and intuitive design is intended to make this process more comfortable and reduce possible errors.

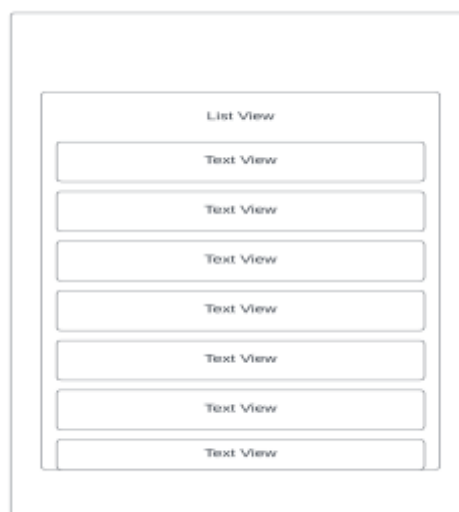


Figure 5. Design Menu Patient

CONCLUSION

Through the process of developing and testing applications, android applications developed using Kotlin and Android Studio have run smoothly and have succeeded in creating applications according to initial requirements specifications. This application is able to display and manage patient biodata, store and display drug quantity data, and provide drug recommendations based on the type of disease and drug stock. The API located on the Google Cloud Platform works efficiently in managing data communication between applications and the Firestore database. The application has been successfully tested on various types of smartphones and managed to run smoothly on all devices. Therefore, this application has fulfilled its original purpose, which is to help manage health posts more efficiently and effectively and improve the quality of services for patients.

Suggestion

In the context of further development, this application can be extended by adding additional features such as scheduling and personnel management. This application can also be developed to support multi-user or larger scale usage. Additional features such as notifications when drug stocks are low, analysis of disease trends based on patient historical data, search features for patients and drugs as well as improvements to data backup and restore features, and strengthening data security can be considered. In addition, conducting trials and surveys to users periodically can also help in getting feedback and improving the quality of the application. Thus, this application can continue to be developed and adapted to user needs and the latest technology, as well as being a better tool to help manage healthy posts.

REFERENCES

- [1] Chinnathambi, K, "Kotlin for Android App Development", Packt Publishing Ltd, 2020.
- [2] Crockford, D, "JavaScript: The Good Parts", O'Reilly Media, Inc, 2018.
- [3] Kotlioglo, V, "Node. js Design Patterns", Packt Publishing Ltd, 2018.
- [4] Leitão, J, "Practical Android: 14 Complete Projects on Advanced Techniques and Approaches", Apress, 2019.
- [5] Pressman, R. S, "Software Engineering: A Practitioner's Approach", McGraw-Hill Education, 2014.
- [6] Ray, R. D, "Android Studio 4.0 Development Essentials - Kotlin Edition: Developing Android Apps Using Android Studio 4.0 and Kotlin", Media Payloads, 2020.
- [7] Android Developers, "Developer Guides", Retrieved from <https://developer.android.com/guide/index.html> on 13 June 2023.
- [8] Google Cloud, "Cloud Run Documentation", Retrieved from <https://cloud.google.com/run/docs> on June 13, 2023.
- [9] Google Firebase, "Firestore Documentation", Retrieved from <https://firebase.google.com/docs/firestore> on June 13, 2023.
- [10] Node.js Foundation, "Node. js Documentation", Retrieved from <https://nodejs.org/en/docs/> on 13 June 2023.