www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 13 Issue 05 May 2024, Page No. 26147-26157
ISSN: 2319-7242 DOI: 10.18535/ijecs/v13i05.4818

Android-Based Electronic Patient Record Implementation Using Software Engineering Principles

Ftoon Kedwan, Anjana Padmanabhan

College Of Computer And Cyber Sciences, University Of Prince Mugrin, Medina, Saudi Arabia,
Department Of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada,.

Abstract
An Android-based electronic patient record application is developed. The goal is to emphasize the importance of automating medical and clinical procedures, especially when data collection and management is involved. A secondary goal is to manifest the efficiency of developing an application using software engineering principles. This application shall replace a paper-based patient record, allowing the medical practitioners to create electronic records of patients. Furthermore, it can be a tool to help health care workers in maintaining patients' information with higher efficiency and security.

Keywords: Software Engineering, Electronic Patient Record, Android Application

Introduction

Software engineering is a computing discipline that involves all stages of a software development from the beginning to the end [1]. Every software that is designed and developed is unique and requires a certain type of software process model to back it up. These processes are tried and tested ways to effectively move software from requirements stage to the shelf. The project will be focused on developing an electronic patient record application on an Android-based development platform. This development will be from the beginning to the end using one of the prescribed software models, thereby demonstrating the importance of having a software model to back up the development.

Health care is an indispensable aspect of human existence [2]. It has seen so many improvements over the years and now information technology in health care is inevitable [3]. This project will aim to do a small part in helping the doctors keep track of the visiting patients without having to maintain a physical paper record.

The application requirements will be obtained from a client at a medical institute. The requirements will be then broken down into several software engineering stages [4, 5, 6] of a preferred model.

Scope

This project is for a mobile application that can be installed in any android phone and used with ease. The application will allow registration and login options, once a new user is registered, the doctor can login and use the system. The doctor can add new patient information and keep a track of their visits, medical conditions, and the medications each time they see a doctor. It will also assist the doctor in keeping track of all the patients they have treated in a day or at any given day.

The project's scope is on serving one hospital records that are maintained by the associated staff members, with a limit in the number of users that can be supported at a given time. The application will also use a local database for demonstration. However, the project can be expanded further to a scale where multiple hospitals and health care entities can use this application to store the patient information with very particular

permissions defined for each user. The scaled-up project would deliver a unified patient records system that many governments and health ministries are now leaning towards. In regard to the implemented database, it can be further expanded to a de-centralized storage such that the data will be more secure and accessible.

Implementation And Methodology

Software Development Lifecycle

The project uses the classic waterfall model as its software process guide. This is basic and traditional, and used when the requirements are decided and finalized by the stakeholders beforehand. This model progresses from a stage to a stage in such a way that one step follows the other after getting signed off by all the stakeholders. This model has been chosen for this project due to the simplicity as well as due to the nature of the requirements.

- *Requirements:* In this stage, the demands from the customer are gathered and are analyzed for the feasibility. This is a fairly elaborate process that is spread over various stages such as requirements elicitation, specification, and validation.
- *Design:* This is the phase where the finalized requirements are translated into paper to give it a proper structure for making the implementation stage easier. The software, hardware, technological and human resource requirements are weighed andfinalized.
- *Development:* The design document is now translated into a real time application. Agroup of developers take up parts of the program and develop the code into the final intended product.
- *Testing:* This phase is used to test the various functionalities of the software to checkif the application is performing as expected. The testing is done in phases such as the unit test (individual components are tested) and system and integration test (the system as a whole and the integration between them is tested).
- *Deployment:* In this stage, the final software is validated and deployed in production for the user to experience in the market/customer environment.
- *Maintenance:* This is the longest and final stage of the lifecycle where a maintenance team constantly looks out for any failures and support required by the customers.

Requirements Specification

Requirements specification is the first and foremost step in the waterfall model. This step aims to understand the user's needs and check for their feasibility. The requirements are usually gathered from the main project stakeholders.

The project's stakeholders, anyone directly or indirectly impacted by the application, expand from the user to the developer with multiple intermediate internal and external factors. A health care application such as this project usually have many stakeholders such as:

1. *Doctors and Staff:* Most Doctors and Clinical staff appreciate the usage of the patient record system since it offers a variety of facilities such as adding, viewing, and updating patient details with ease.
2. *Patients:* Patients are the major stakeholders when it comes to this system. The records of patients are entered by the doctors and hence are trusted sources to learn about any condition and its progression in treatment.
3. *Family and caregivers:* Closely related with the patient's care, families and caregivers are also notable stakeholders of this system as they depend on the accuracy of the entries in the records for the treatments to be the most effective.
4. *Clinicians:* People who work one-to-one with the patients to ensure that the concerned person receives the most accurate and effective treatment based on the doctor's entries.
5. *Health care facilities:* The hospitals and other related health care facilities highly depend on the efficacy of the healthcare professionals as well as the health of the patients. It is paramount for these facilities to be able to track the patients so that any space for improvement can be considered.
6. *Researchers:* The researchers may use the patient record system to understand the course and progress of a certain condition in order to develop an effective clinical trial system or even assist the course of designing a vaccine.

7. *Payers and Purchasers:* The payers and purchasers including the health insurers may use the patient record system in order to arrive at a better conclusion on the support system as well as acting as a means to reduce the potential medical errors.
8. *Employers:* Any potential employer can be benefited from such a system from the fact that these systems can be used as a way to track the professionals and their standards.
9. *Additional Stakeholders:* Any other potential stakeholders such as a third-party vendor or even the developer who designs the system and who gets benefitted by thesystem.

After requirements specification and validation is done, it is finalized to be moved to the next stage of the lifecycle.

User Story:

Title: Home page and Doctor's login
When a user accesses the system, they should be able to choose between two options, Doctor, and Staff. As a Doctor, they should be able to login as a registered user and access a dashboard of items. This page will allow them to choose between different functionalities such as adding a new patient, update an existing patient record, and search for patient information using the Patient ID or the date of visit. The patient ID is returned by the system when the new entry is added. **Acceptance criteria** Given login credentials, the employee database is checked for the correctness and then lets the doctor access the home page. In case of invalid login, the user is redirected to the login page.

Title: Staff login	**Priority: High**
As a Clinical staff, they should be able to register into the system and then login. Afterlogging in, the staff should be able to search the patient details with the patient ID or date of visit to view the records. **Acceptance Criteria** The staff credentials are entered in the Register/login page after they enter into the Staff portal. These credentials are checked with the database and if the values are valid, they can view the details, else, the user is redirected to the login page.	

The declared user story can be used as a tool to understand what exactly the customer wants and also to decide on the various functional and non-functional requirements wanted by the customer.

Use Case Model:

The use case model is a simple diagram that explains about the Primary users and the features that are available to them.

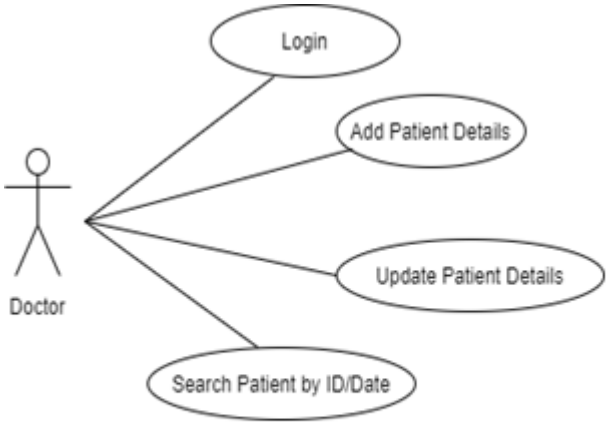

Figure 1 The Doctor Use Case Model

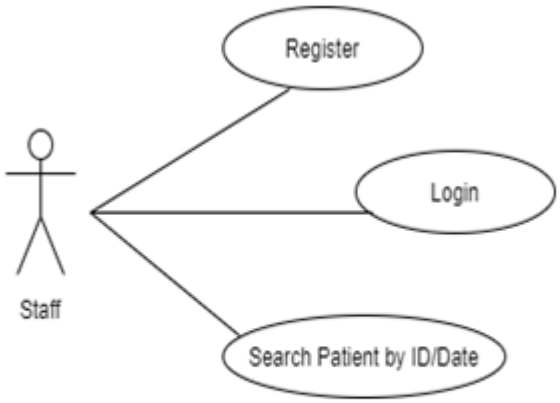

Figure 1 The Staff Use Case Model

From the user stories in Figures 1 and 2, it is understood that the customer wants a separate page for both Doctor and Staff. Thus, we have a separate use case diagram for the functionalities, Doctor, and the staff. This can be used to define the various functional and non-functional requirements clearly for development.

Functional Requirements:

The tangible functions of a system that have been demanded by the customers come under the functional requirements.

- Doctors and Clinical Staff should be able to register/login into the system through separate portals using ID number and password.
- Doctors should be able to add new patient details each time they have a visitor, and the system should display a unique patient ID at the end of entry.
- Doctors should be able to view the patients' records through patient ID or the date of visit, and update them if required.
- Any registered Clinical Staff should be able to search and view patient details using the patient ID or date of visit.

- Non-Functional Requirements:
- The resultant performance and characteristics of the functional requirements are called the non-functional requirements. For this project, the non-functional requirements are Security [7], Performance and Capacity, and Maintainability as explained hereafter.
- Security
  - The login information should be kept safe and private.
  - The information collected from the patient is extremely confidential and this confidentiality should be maintained during the exporting and storing them in the application as well.
  - The read and write rights of each user (Doctor or Clinical Staff) should be distinguished clearly.
- Performance and Capacity
  - User gets an acknowledgement at the end of each activity within one second
  - The system should be able to support 100 users at a time.
- Maintainability
  - The system should offer a backup for the patient records in case of any failures.
  - The system should be able to keep a track of any errors and log them.
- Hardware and Software Requirements:
- *Hardware requirements for Developer (minimum):* Windows/Mac PC, 4 GB RAM, Intel i3 Processor.
- *Software Requirements for Developer:* Windows 7(minimum), Android Studio, and JavaJDK.
- *Hardware used for testing:* Android Mobile – Nexus 5x

Architecture

The model chosen for this project is the ***Layered Architecture model***. It depicts the distinction between the various levels of software such as the *Presentation, Application, Domain, and Infrastructure* layers. The project involves multiple transactions between the User Interface and the Database thus making this model appropriate for this application. There are four layers in this architecture with separate depiction outlined in Figure 3.

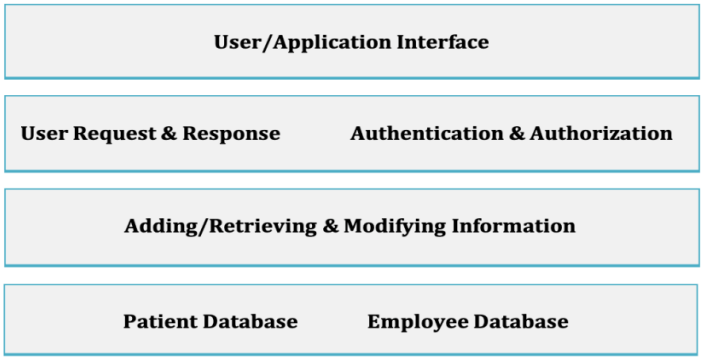

Figure 1 Layered Architecture of the Android Patient Record

The first layer consists of the User Interface (UI) with all of the components in the home screen. The second and third layers are interfaces between the UI and the database. These interfaces are in charge of multiple actions, such as adding new patient details, editing existing data, and searching for a patient in the database, as well as authenticating the existing users by cross checking with the Employee database and add new users when registered. Here, all the doctors will be allowed to login even without registering while the staff will not be able to. The innermost layer consists of the twodatabases, both patient and Employee databases. This layer is the most secure portion of the architecture. The Employee database will constantly be updated every time a doctor joinsfor work, and the Patient database is updated by the doctor after every patient visit.

Design

The software design process is done using the system modeling tool. System modeling is a study of the use of different diagrams to represent various aspects of the software. The diagrams are designed in such a way that any stakeholder with or without extensive technological knowledge is able to understand the view of the

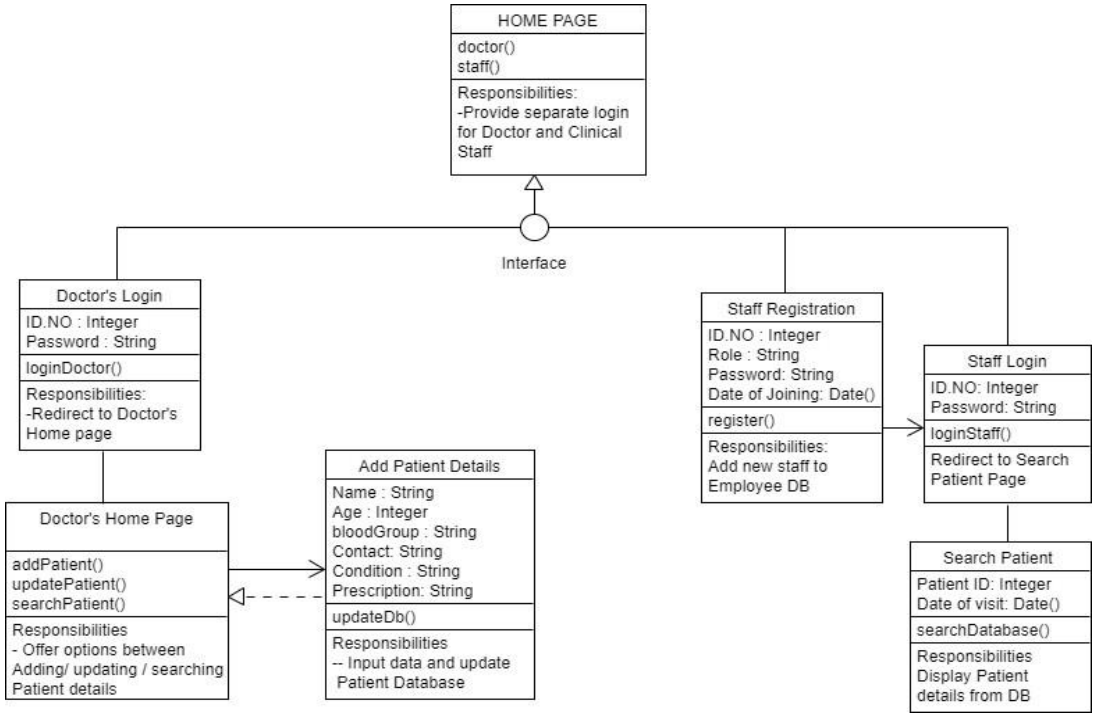

Figure 2 Class Diagram of Android Patient Record

developer. There are multiple types of diagrams available to represent this process.

In this project, the most important diagrams, such as the class, sequence, and activity diagrams are considered for explaining the application. Each of these diagrams represents the application from a different perspective, thus becoming the best way to explain the construction of the system.

Class Diagram:
The class diagram is used to represent the object-oriented system model [8]. It gives an idea about the classes, attributes, and the association between each of these components. This program uses ***java*** as the primary language of development in android IDE and all the objects and attributes are given in Java terms. The object class will represent the classes, attributes, and purpose of each process that are involved in the application.

The class diagram in figure 4 is a representation of the application in a detailed manner from a developer's perspective. The diagram cascades from HOME PAGE as it is first displayed on the application. The home page then splits into separate Doctor and Staff login pages that have specific functions and permissions for the users. Each of these objects performs a certain task (takes help from other objects, if required), which is then mentioned in the responsibilities section. The Object name and the data type of the attribute are given for reference.

Sequence Diagram:
Sequence diagram is a type of interaction diagram that gives a detailed diagram displaying the dynamic aspects of the application. The layers of the application, their interaction or communication between the user interface and the database can be described clearly using this diagram. The sequence for both Doctor and Clinical staff is given in separate diagramsand as separate entities.

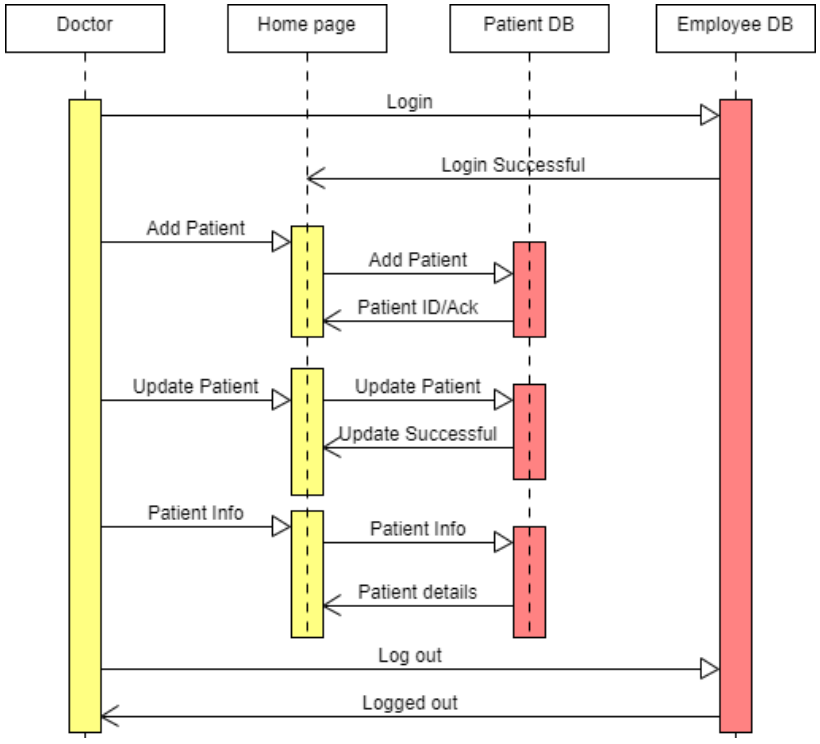

Figure 3 Sequence Diagram of the Doctor's page

The Doctor's page sequence diagram in Figure 5 has four entities for easy understanding. The actor here is the Doctor, the home page (Doctor's Login), the Patient Database (PDB), and finally the Employee Database (EDB). The patient database will hold the patient records that are stored by the doctor and the employee database will have the hospital's employee details for authorization purposes. The sequence describes five actions that will be performed by the user.

- *Login*: When the doctor logs in using their ID number and Password, the credentials are sent to the EDB

to be validated. Once they are approved, an acknowledgement of the successful login is sent to the user. The doctor will then be able to choose the action to be performed from the dashboard.

- *Add Patient*: If they choose the "add patient" option, then he/she will be asked to enter the details of the patient and upon saving, the details will be sent to the PDB and gets saved there. Every time the doctor saves a new Patient, the system will return a patient ID number which will be the index number of a new entry in the database.
- *Update Patient*: The doctor can also update an existing record by entering the Patient ID and editing the record. Once the record is saved again, the record on the PDB gets updated and an acknowledgement of the successful update is sent back.
- *Search for Patient Information*: The doctor will also be able to search for a particular patient using the ID. They can also view the list of patients visited on a certain date along with their details. The command will fetch the details from the DB and display it in the UI.
- *Logout*: Once the work is done, the doctor can logout and the session in the database will be closed. They will be then taken back to the login page.

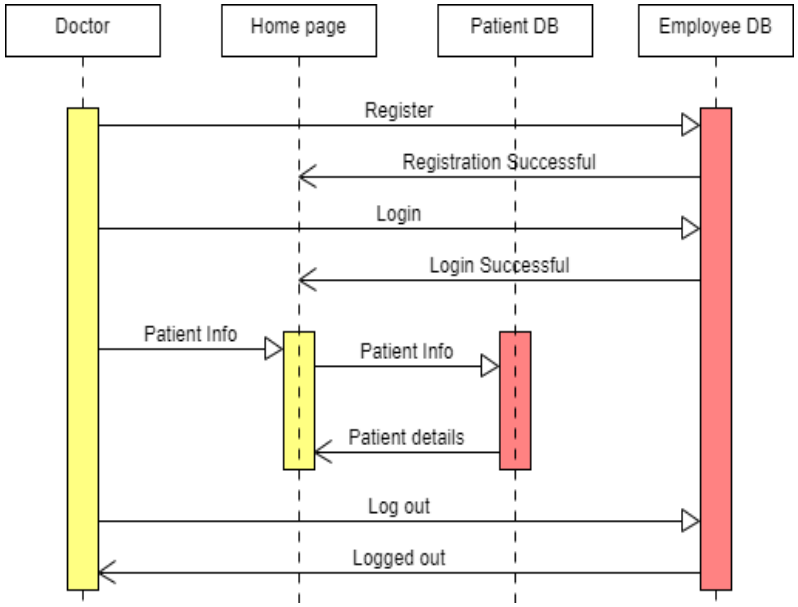

Figure 4 Sequence Diagram for Clinical Staff page

As outlined in Figure 6, the Clinical staff will have one extra option which is the registration of a new user. Unlike doctors, any clinical staff should register into the system before using. The EDB is checked for their identity and then they can use the ID number to login into the system. The Clinical staff is given permission to only view the patient details, which can be looked up by the patient ID or the date. The details are fetched from the PDB and displayed. Once the staff is done with their work, they can log out and the session will end.

Activity Diagram

An activity diagram is used to describe how the events are coordinated and how it moves from one stage to another. Here, the two user experiences are displayed in separate activity diagrams.

This doctor's login activity diagram in Figure 7 describes the organization and overall activity performed at each stage in a simple way. Here, after the login is done, the adding and editing of patients' details can be interrelated and are given together followed by the search patient details. The activities are merged at the end to lead to the log out session.

The staff login activity diagram, Figure 8, is much simpler as there is a limited access to the staff. Here, the home page will take the staff to a register/login page. The clinical staff can only view the patient details that the doctor has added and then they can log out.

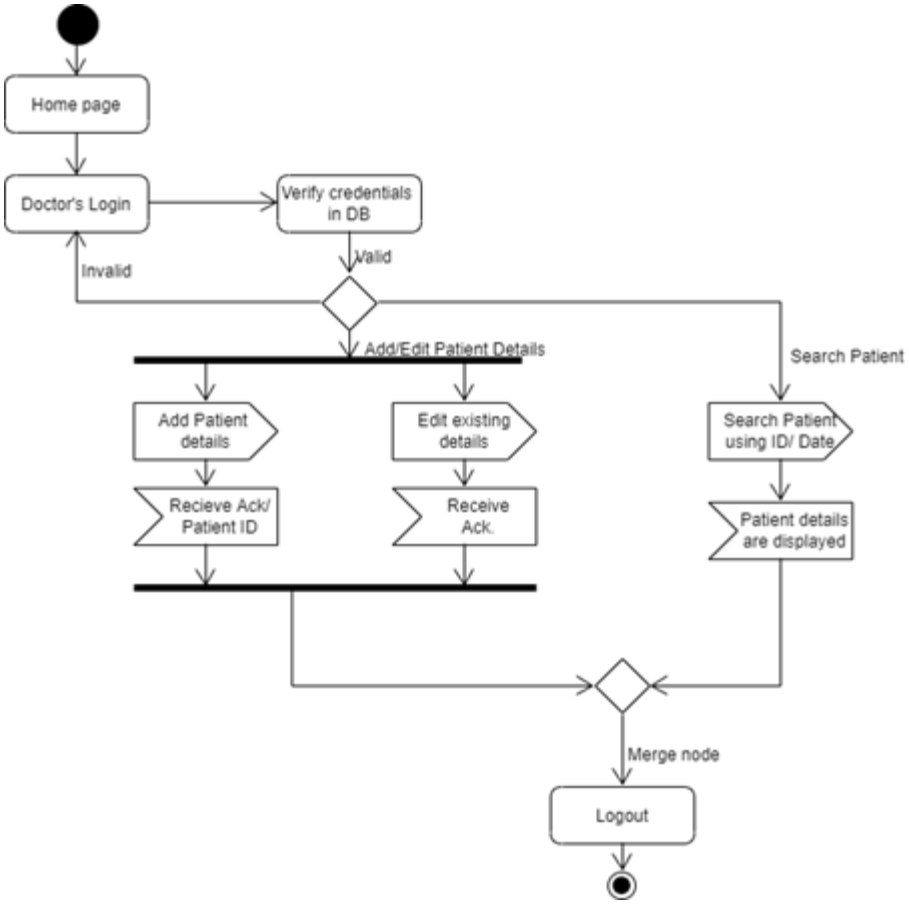

Figure 6 Doctor's Login Activity diagram

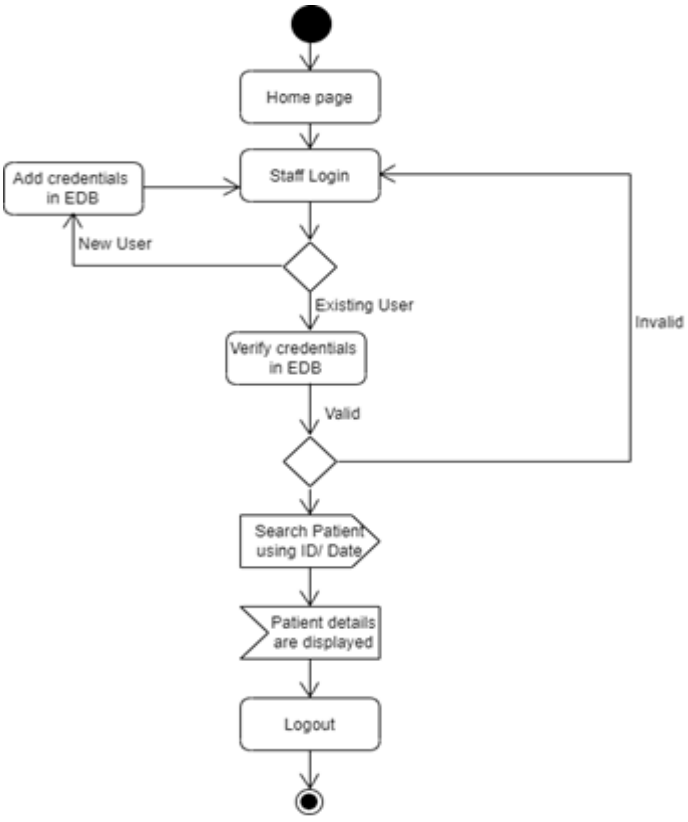

Figure 7 Staff Login Activity Diagram

Implementation

The UI of the application at each step is designed and demonstrated with the helpof a few screenshots. Fake hospital name, staff and doctors' details are used for the sake of this demonstration.

Figure 5 Staff Login Activity Diagram

Screen 1 Doctor and Staff Login Home screen

Screen 2 Doctor's Login

Screen 3 Doctor's Dashboard

Screen 4 Add Patient Details Form for Doctors

Screen 5 Patient Added Acknowledgement and ID

Screen 6 Edit Records Page

Screen 7 Search Patient Page

Screen 8 Staff Registration Page

Screen 9 Staff Login Page

Screen 10 Search result using Patient ID

Screen 11 Result while using Date

Testing

Testing is a step that is as important as developing a software. The testing is done in stages usually, that is Unit, Integration, and System testing.

Table 1 Unit and Integration Testing Template

TEST TITLE		**PRIORITY**		**TEST DATE**	
Android Patient Record		High		10/01/2024	
TEST DESCRIPTION					
Unit and Integration testing					
TEST DESCRIPTION		**CONDITIONS**			
The unit test will analyze the proper functionality of each of the modules in the application using a set of predefined test cases. The integration testing will analyze the communication between pages.		The test will implement a correct as well as wrong scenario to test the results of the modules			
SN	**STEP DESCRIPTION**	**TEST DATE**	**EXPECTED RESULTS**	**ACTUAL RESULTS**	**PASS /FAIL**
1	Home page to Doctor's login page	10/01/2024	On clicking "Doctor" in home page, doctor's login is displayed	The Page moves to Doctor's Login	Pass
2	Doctor/Staff Login	10/01/2024	On entering the correct login, the page should move to the dashboard On entering wrong credentials, it should redirect to login page	The correct login takes the page to Dashboard Incorrect login redirects to Login page	Pass
3	Staff Registration	10/01/2024	When new details are entered and registered, the EDB should be updated, and the page is redirected to login	The Database gets updated properly and the page is redirected	Pass
4	Add Patient details	10/01/2024	When new patient details are entered andsaved, the PDB gets updated	The update is successful	Pass
5	Search and Edit details	10/01/2024	When a patient detailis searched using patient ID or date, thedetails should be displayed	The correct details are fetched from PDB	Pass
6	Doctor / Staff session logout	10/01/2024	When the user clicks the log out, the sessionwill end, and the page should return to login page	The logout button redirects the page to login	Pass

The test cases and conditions in Table 1 are a set of basic steps to test the function and integrity of the application. The listed functions will basically achieve a little of unit as well as the integration testing to see if the individual as well as the component communication is proper. If the system passes all these conditions that are given above, then it can be considered production ready.

Deployment

The application has proved to be user friendly, through end user testing, and can be easily navigated. After the testing is done, the application is deployed at the user environment. The deployment is through application stores like Google Play or the Apple Store which can then be downloaded into any android mobile device and used anywhere anytime.

Maintenance

The application post deployment will be under the responsibility of the developing party. Therefore, the application is constantly observed for any runtime problems that may arise and are provided patches and support.

Conclusion

This project was a way of exploring the various stages and challenges of the software engineering processes that an application goes through before the deployment. The Android application developed as a result will strive to help the health care community who puts their life online every day in performing their best with ease.

References

1. Stol, K. J., Goedicke, M., & Jacobson, I. (2016). Introduction to the special section—General Theories of Software Engineering: New advances and implications for research. Information and Software Technology, 70, 176-180.
2. He, X., Cai, L., Huang, S., Ma, X., & Zhou, X. (2021). The design of electronic medical records for patients of continuous care. Journal of infection and public health, 14(1), 117-122.
3. Oluwagbemi, O., Oluwagbemi, F., & Ughamadu, C. (2016). Android Mobile Informatics Application for some Hereditary Diseases and Disorders (AMAHD): A complementary framework for medical practitioners and patients. Informatics in Medicine Unlocked, 2, 38-69.
4. Ainieh, K., Muslemani, A., & Kedwan, F. (2024). Software Engineering Implementation Model on a Tool Rental System. International Journal of Engineering and Computer Science, 13(04), 26068–26114. https://doi.org/10.18535/ijecs/v13i04.4808
5. Kedwan, F. (2024). A Software Engineering Approach on Developing a Real Time Radar Target Generator for Airborne Targets. Applied Science and Engineering Journal for Advanced Research, 3(2), 1–34.
6. Kedwan, F. (2023). Traffic Light Timer Adjustment System Using Threshold Edge Detection for Image Processing: A Software Engineering Approach. International Journal of Current Research in Multidisciplinary (IJCRM). ISSN: 2456-0979.
7. Hussain, M., Al-Haiqi, A., Zaidan, A. A., Zaidan, B. B., Kiah, M., Iqbal, S., ... & Abdulnabi, M. (2018). A security framework for mHealth apps on Android platform. Computers & Security, 75, 191-217.
8. Kedwan, F. H., & Sharma, C. (2019). Model-Driven Software Development Platforms Reviews. International Journal of Computer Applications, ISBN: 973-93-80900-26-6, Volume 9– No. 2019919165.