**Decentralized Health-Care Records**

**Final Year Project Report**

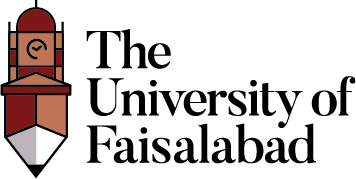
**Session 2021-2025**

A project submitted in partial fulfilment of the

The University of Faisalabad Degree

of

Bachelor of Science BSc. (Hons.)**Computer Science**



Department of Computer Science

The University of Faisalabad, Amin Campus

18 April 2025

Project Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Decentralized Health-Care Records** | | | [ ] **D**evelopment [ ] **R**esearch [ ] **R**&**D** | | |
| Area of specialization | | |  | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) | Group Leader |  | |  |  |
| (ii) |  |  | |  |  |
| (iii) |  |  | |  |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

Plagiarism Free Certificate

This is to certify that, I am \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ S/D/O \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, group leader of FYP under registration no \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ at the Department of Computer Science, The University of Faisalabad. I declare that my FYP proposal is checked by my supervisor and the similarity index is \_\_\_\_\_\_\_\_% that is less than 20%, an acceptable limit by HEC. The report is attached herewith as Appendix A.

Name of Group Leader: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Member Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Member Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Supervisor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Co-Supervisor (if any): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HOD: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Declaration

We here by declare that the content of this project report title “**Decentralized Health-Care Records**” submitted to the “**DEPARTMENT OF COMPUTER SCIENCE**”, is a documentation of a unique work we created under the supervision of Supervisor “**Prof. Dr. Majid Hussain**” and that no part has been plagiarized.Additionally, this project is presented in partial completion of the degree requirements for a Bachelor of Science in “**Computer Science**”. The University may take action if the above statement is found inaccurate at any stage.

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Registration No** | **Signature** |
| Shiza Ahmad | BSCS-165-2021 |  |
| M.Abdullah Fiaz | BSCS-008-2021 |  |

|  |  |
| --- | --- |
| **Supervisor: Prof. Dr. Majid Hussain** | **Signature/Date** |

Certificate

We accept the work contained in the report titled **“Title of the Project”**, written by **“Student Names”** as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in **degree name (ex. Computer Science)**.

**Approved by:**

**Supervisor:** Name

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner:** Name

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner:** Name

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**FYP Coordinator:** Name

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head of the Department:** Name

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date:**

**Abstract**

Block chain technology is transforming healthcare by providing a secure and transparent method for managing patient data. This innovation enhances data privacy, facilitates seamless information sharing among healthcare providers, and ensures the integrity of clinical trials and pharmaceutical supply chains. Leading countries like the United States, Estonia, the United Kingdom, and Switzerland are adopting block chain to improve patient care and system efficiency. This project introduces a block chain-based healthcare application that addresses critical issues such as data breaches, unauthorized access, and medication counterfeiting. Utilizing a decentralized database, the app securely manages patient information—including medical history, prescriptions, and test results—accessible only to the patient. It features medication tracking, reminders, emergency notifications, and integration with wearable devices. The application is designed with strong security measures and compliance with HIPAA and GDPR standards. By standardizing data formats, employing advanced encryption, and involving healthcare providers in its development, the platform enhances data privacy, interoperability, and patient empowerment. This block chain-based solution signifies a major step forward in secure and efficient healthcare management.

Application Evaluation History

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*Include the ones given at scope time both in documentation and presentation** | **Action Taken** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Supervised by**

**Prof. Dr Majid Hussain**

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table of Contents

[CHAPTER 1: INTRODUCTION 1](#_Toc189066205)

[1.1 Introduction 1](#_Toc189066206)

[1.2 Aim & Objectives 1](#_Toc189066207)

[1.3 Problem Statement 1](#_Toc189066208)

[1.3.1 Extended Problem 2](#_Toc189066209)

[1.6 Assumptions & Constraints 3](#_Toc189066210)

[1.8 Business Plan 4](#_Toc189066211)

[1.8.1 Business Model Canvas 6](#_Toc189066212)

[1.8.2 Problem 8](#_Toc189066213)

[1.8.3 Solution 8](#_Toc189066214)

[1.8.5 Competitors 9](#_Toc189066215)

[1.8.7 Revenue 11](#_Toc189066216)

[1.8.8 SWOT Analysis 11](#_Toc189066217)

[CHAPTER 2: LITERATURE REVIEW/BACKGROUND AND EXISTING WORK 13](#_Toc189066218)

[2.1 Background 13](#_Toc189066219)

[2.2 Literature Review 13](#_Toc189066220)

[2.2.1 Decentralized Consent Management 13](#_Toc189066221)

[2.3 Literature Summary 14](#_Toc189066222)

[Why This Project? 15](#_Toc189066223)

[CHAPTER 3: REQUIREMENTS ANALYSIS 16](#_Toc189066224)

[3.1 Stakeholders List (Actors) 16](#_Toc189066225)

[3.2 Requirements Elicitation 16](#_Toc189066226)

[3.2.1 Functional Requirements 17](#_Toc189066227)

[3.2.2 Non-Functional Requirements 17](#_Toc189066228)

[3.2.3 Requirements Traceability Matrix 18](#_Toc189066229)

[3.3 Use Case Design/Description 18](#_Toc189066230)

[3.4 Software Development Life Cycle Model 19](#_Toc189066231)

[3.5 Specific Requirements (Hardware and Software Requirements) 20](#_Toc189066232)

3.5.1 Hardware Requirement…………...…………………………………………20

3.5.2 Software Requirement…………....…………………………………………20

[Software Requirements Table: 20](#_Toc189066233)

[CHAPTER 4: SOFTWARE DESIGN SPECIFICATION 22](#_Toc189066234)

[4.1 Design Models 22](#_Toc189066235)

[4.1.1 Software Methodology / Software Process 22](#_Toc189066236)

[4.2 Work Breakdown Structure 23](#_Toc189066237)

[4.3 System Architecture 24](#_Toc189066238)

[4.3.1 Block Diagram 24](#_Toc189066239)

[4.3.2 Component Diagram 24](#_Toc189066240)

[4.3.3 Software Architecture Diagram 25](#_Toc189066241)

[4.4 Data Representation [Diagram + Description] 26](#_Toc189066242)

[4.4.1 Entity-Relationship Diagram (ERD) 26](#_Toc189066243)

[4.4.2 UML Class Diagram 27](#_Toc189066244)

[4.4.3 Data Flow Diagram (DFD) 28](#_Toc189066245)

[4.4.4 Class Diagram 28](#_Toc189066246)

[4.4.5 Hierarchical Diagram 29](#_Toc189066247)

[4.5 Process Flow/Representation 29](#_Toc189066248)

[4.5.1 Flowchart 29](#_Toc189066249)

[4.5.2 Sequence Diagram 30](#_Toc189066250)

[4.5.3 Activity Diagram 30](#_Toc189066251)

[References 31](#_Toc189066252)

List of Figures

[Figure 1: Business Model Canva 10](#_Toc192270604)

[Table 2: Requirements Traceability Matrix 24](#_Toc192270605)

[Figure 3: flow chart 25](#_Toc192270606)

[Figure 5: Design model 28](#_Toc192270607)

[Figure 6:Work Breakdown Structure 30](file:///C:\Users\HP\Downloads\SDS%20Decentralized%20Health-Care%20Records.docx#_Toc192270608)

[Figure 7:Block Diagram 31](#_Toc192270609)

[Figure 8: Component Diagram 32](#_Toc192270610)

[Figure 9: Software Architecture Diagram 32](file:///C:\Users\HP\Downloads\SDS%20Decentralized%20Health-Care%20Records.docx#_Toc192270611)

[Figure 10: Entity Relationship Diagram 33](file:///C:\Users\HP\Downloads\SDS%20Decentralized%20Health-Care%20Records.docx#_Toc192270612)

[Figure 11: UML Class Diagram 34](#_Toc192270613)

[Figure 12: Data Flow Diagram 35](#_Toc192270614)

[Figure 13: Class Diagram 35](#_Toc192270615)

[Figure 14: Hierarchical Diagram 36](#_Toc192270616)

[Figure 15: Flow Chart 37](#_Toc192270617)

[Figure 16: Sequence Diagram 38](#_Toc192270618)

[Figure 17: Sequence Diagram 38](#_Toc192270619)

List of Tables

[Table 1: Business Model Canva 10](#_Toc192271058)

[Table 2: Requirement Traceability Matrix 24](#_Toc192271059)

[Table 3: Software Requirement 27](#_Toc192271060)

# INTRODUCTION

**1.1 Introduction**

In today’s digital world, keeping healthcare data secure is a big challenge. Many healthcare systems use centralized storage, which can lead to data privacy issues, unauthorized access, and even breaches. This project focuses on solving these problems with a blockchain based healthcare data management app. The app uses a decentralized storage system, giving patients complete control over who can access their data. It also provides helpful features like medication information and reminders to improve how patients manage their health.

This report is written for a general audience and explains the project’s goals, features, and why it’s important. The app is designed to make managing healthcare data safer and more patient focused. By putting patients in charge of their data, improving transparency about medications, and supporting better health habits through reminders, this project addresses a major need in healthcare for secure and user-friendly solutions.

**1.2 Aim & Objectives**

**Aim:**

To develop a block chain-based healthcare data management app that prioritizes data security, patient independence, and easy access to medication related information.

**Objectives:**

1. Develop a system for decentralized storage to keep patient health records secure and protected from tampering.
2. Give patients full control over their data by allowing them to choose who can access their health
3. Offer detailed information about medications, including how to use them and what they are commonly prescribed for information.
4. Include a reminder feature to help patients stay on track with their medication schedules.

**1.3 Problem Statement**

The healthcare system faces significant challenges in keeping patient records secure, leading to risks such as data breaks and difficulties in collaboration among healthcare providers. Additionally, counterfeit medications pose serious health risks to patients. Current systems do not effectively ensure patient privacy, make it easy for hospitals and clinics to share information, or verify the authenticity of medications. These gaps compromise the safety and efficiency of healthcare delivery. A better solution is needed—one that keeps patient data private, helps healthcare providers share information seamlessly, and ensures medications are genuine. A secure system where patients control their own information can address these issues effectively, safeguarding medical records and improving trust in the healthcare process.

**1.3.1 Extended Problem**

Managing healthcare data securely and effectively is a major issue in today’s digital world. Most medical records, such as patient histories and prescriptions, are stored in centralized systems controlled by hospitals or organizations. These systems face several challenges:

**1. Data Privacy and Security Issues**

Centralized storage systems are often targets for cyberattacks, leading to data breaches. Patients also have little control over who can access their sensitive information, making them feel insecure about sharing their data.

**2. Lack of Patient Control**

In most cases, patients cannot decide who can view or use their medical data. This lack of transparency makes the process less patient friendly.

**3. Limited Information About Medications**

Patients often don’t understand the medicines prescribed to them, including their purpose, how to use them, or their possible side effects. This leads to confusion and misuse of medication.

**4. Forgetting Medications**

Many people forget to take their medicines on time, which can reduce the effectiveness of their treatment. Medication nonadherence is a common issue that can lead to serious health problems.

**5. Problems with Current Systems**

Existing solutions, like electronic medical records, are mostly designed for healthcare providers, not patients. These systems are often centralized, making them vulnerable to hacks. They also don’t combine important features like secure data storage, medication details, and reminders.

**Why This Problem is Important ?**

These challenges negatively affect both patients and healthcare providers:

**For Patients:** Loss of privacy, lack of understanding about treatment, and missed medications can lead to poor health outcomes.

**For Healthcare Providers:** Data security concerns and inefficient systems lower trust and increase risks.

**Need for a Solution**

There is a need for a system that gives patients control over their data, stores information securely, provides clear medication details, and sends timely reminders. Such a solution will empower patients, improve medication adherence, and make healthcare more efficient and reliable.

**1.4 Proposed System**

This project proposes a blockchain based system that resolves the challenges of data security and patient autonomy by focusing on decentralized data storage, controlled access, medication information, and reminders. The system will:

1. Store data securely on a decentralized block chain, minimizing the risk of unauthorized access.
2. Enable patients to grant or restrict data access to healthcare providers, fostering a greater sense of control.
3. Provide detailed information about medications, including recommended uses and potential effects, enhancing patient knowledge.
4. Offer reminder notifications to support medication adherence, reducing missed doses and improving treatment outcomes.

**1.5 Project Scope**

1. **Decentralized Data Storage**

Only data related to patient medical history, prescriptions, and treatment records.

1. **Patient Controlled Access**

Limited to patient defined access permissions for specific healthcare providers.

1. **Medication Information**

Covers general usage, common side effects, and recommended purposes of commonly prescribed medications.

1. **Reminder System**

Focused solely on medication reminders, with no integration for other types of notifications.

**Exclusions from scope:**

1. Emergency alerts or notifications
2. Integration with wearable devices
3. Symptom tracking and weekly health reports.

**1.6 Assumptions & Constraints**

**Assumptions:**

* 1. Patients are willing to use a mobile or web application for managing their healthcare data.
  2. Healthcare providers support patient controlled access to medical records.

**Constraints:**

* 1. Compliance with relevant data protection regulations, including HIPAA and GDPR.
  2. Dependence on block chain infrastructure, which may limit compatibility with some healthcare provider systems.
  3. Assumptions around user familiarity with block chain technology.

**1.7 Social Benefits**

The project aims to contribute positively to society by empowering patients to take control of their health information. By enabling secure, decentralized storage and patient centered data access, the app promotes transparency and trust in healthcare. Improved medication adherence through reminders can lead to better health outcomes, and enhanced patient knowledge of medications fosters informed decision making in healthcare.

**1.8 Business Plan**

Business Plan for Block chain Based Healthcare Data Management App. This business plan focuses on delivering a secure, patient centric healthcare solution that addresses real world challenges like data privacy, medication adherence, and counterfeit drugs. It also emphasizes benefits for the local public and healthcare providers.

**1. Value Proposition**

The app offers a unique combination of features designed to make healthcare safer, more transparent, and patient focused:

**Decentralized Data Storage**

Keeps medical records secure from breaches by using block chain technology.

**Patient Controlled Access**

Allows patients to decide who can view or share their data, ensuring privacy and trust.

**Medication Information**

Provides accurate details about medicines, helping users understand their prescriptions.

**Reminders for Medication**

Helps users take medications on time, improving health outcomes.

**Authenticity of Medications**

Verifies medications to ensure patients get genuine products.

**2. Target Market**

**Local Public**

Individuals seeking better control over their health records and medication management.

**Healthcare Providers**

Hospitals and clinics that need secure and efficient data sharing systems.

**Pharmaceutical Companies:** To verify the authenticity of their products and ensure transparency.

**3. Benefits for Local Public**

1. **Enhanced Data Security**

Protects personal health records from cyber threats, giving users peace of mind.

1. **Empowerment**

Patients can directly manage their medical data and decide who accesses it.

1. **Improved Health Outcomes**

Timely medication reminders and access to reliable medication information reduce health risks.

1. **Prevention of Counterfeit Drugs**

Ensures that patients receive authentic medications, reducing potential harm.

1. **Accessibility**

Makes healthcare management easier, especially for elderly or rural users, by providing simple features to track medications and appointments.

**4. Revenue Streams**

**Subscription Plans:**

1. Free basic features for individuals; premium options for detailed reports and extra functionalities.
2. Paid plans for hospitals and clinics to manage patient data securely.

**Verification Fees:**

1. Pharmaceutical companies pay to verify their products’ authenticity on the platform.

**Collaborations with Local Healthcare Providers:**

1. Clinics and pharmacies may sponsor features like reminders or medication tracking.

**5. Marketing Strategy**

**Community Outreach:**

1. Collaborate with local clinics and pharmacies to spread awareness about the app.
2. Conduct workshops to educate people on the importance of data security and medication adherence.

**Digital Campaigns:**

1. Use social media to highlight real life stories of how the app improves health outcomes.
2. Run targeted ads focusing on the app’s benefits for patients and healthcare providers.

**Government and NGO Partnerships:**

1. Partner with local health departments and organizations to promote the app as a trusted tool for better healthcare.

**6. Competitor Analysis:**

**Existing Solutions:**

1. Current apps either focus only on medication management or data storage but rarely combine the two.
2. Many do not use block chain, making them vulnerable to breaches.

**Your Advantage:**

1. Combines secure, decentralized storage with medication tracking and counterfeit verification.
2. Empowers patients, unlike traditional healthcare systems that prioritize providers.

**7. Cost Structure**

1. **Development and Integration**

Building a blockchain based app with a user-friendly interface.

1. **Local Engagement**

Costs for community workshops, promotional campaigns, and training for users.

1. **Regulatory Compliance**

Meeting healthcare and data protection laws.

1. **Maintenance**

Regular updates to improve features and security.

**8. Social Impact:**

**For Local Communities:**

1. Promotes trust in the healthcare system by ensuring privacy and medication safety.
2. Reduces health risks by preventing counterfeit drugs and encouraging proper medication adherence.
3. Empowers underserved communities by providing accessible tools for health management.

**For Healthcare Providers:**

1. Improves collaboration between hospitals and clinics with secure data sharing features.
2. Saves time and resources by streamlining patient record management.
   * 1. **Business Model Canvas**

|  |  |  |
| --- | --- | --- |
| Key Partners | Key Activities | Value Propositions |
| Healthcare providers | Development of the app | Secure, decentralized data storage |
| Block chain provider | Integration with block chain network | Patientcontrolled data access |
| Medical organizations | Customer support | Medication information for patient education |
| Data security advisors | Providing Accurate Medication Information | Medication reminders to improve adherence |
|  | Regular Updates and Maintenance |  |
|  | Marketing and Outreach |  |

|  |  |  |
| --- | --- | --- |
| Customer Relationships | Segments Customer | Key Resources |
| Self-Service through the app interface | Individual patients | Block chain network |
| Help desk for technical support | Healthcare providers needing secure data access | Medical databases for medication information |
| Patient education content |  | Secure servers for decentralized data management |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Channels | Cost Structure | Revenue Streams |
| Mobile app | App development and maintenance costs | Subscription fees (for premium features) |
| Collaborations with healthcare providers | Block chain infrastructure fees | Licensing fees from healthcare providers |
|  | Data compliance and regulatory costs | Inapp advertisements or affiliate marketing (optional) |
|  | Customer support |  |

Table : Business Model Canva

**Explanation of Key Elements:**

1. Key Partners: Collaborating with block chain providers, healthcare organizations, and regulatory bodies ensures secure, compliant, and informative services for patients.
2. Value Propositions: Your app focuses on security, patient autonomy, medication education, and reminders, addressing critical gaps in traditional healthcare data management.
3. Customer Segments: Primarily aimed at patients who prioritize data privacy and transparency, and healthcare providers seeking secure patient data access.
4. Revenue Streams: The app can generate revenue through subscriptions, licensing for healthcare providers, and potentially in app advertisements.
5. Cost Structure: Key costs include development, block chain fees, and regulatory compliance efforts.

**1.8.2 Problem**

The business problem addressed by this project is the need for secure, patient centered data storage in healthcare. Traditional storage methods are often inadequate for protecting patient privacy and do not offer patients direct control over their data.

**1.8.3 Solution**

Our project uniquely positions itself by combining decentralized storage, patient controlled access, and education on medication usage. This solution aims to alleviate concerns about data security, while also engaging patients more actively in their own healthcare.

**Customers**

1. **Local Patients** 
   1. **Privacy Conscious Individuals**

People who want control over their health data.

* 1. **Chronic Condition Patients**

Those needing consistent medication reminders and tracking.

* 1. **Elderly**

Older individuals who require help managing their health and medications.

* 1. **Health-Conscious Users**

People seeking a secure and easy way to manage their health.

**2. Healthcare Providers**

* 1. **Local Clinics and Small Hospitals**

Need secure, efficient systems for managing patient records.

* 1. **Pharmacies**

Require systems to verify medication authenticity and provide information to patients.

**3. Pharmaceutical Companies**

* 1. **Drug Manufacturers**

Want to ensure medication authenticity and transparency.

**4. Government and NGOs**

1. **Local Health Authorities**

Can use the app to improve data security and healthcare access.

1. **Non-profits**

Focus on improving healthcare in underserved areas.

**1.8.5 Competitors**

**1. Traditional EHR Systems (e.g., Epic, Cerner)**

* 1. **Strengths**

Widely used in healthcare settings.

* 1. **Weaknesses**

Centralized, vulnerable to data breaches, and limited patient control over data.

**2. Medication Management Apps (e.g., Medi safe, My Therapy)**

* 1. **Strengths**

Focus on medication reminders.

* 1. **Weaknesses**

Lack secure data storage and patient controlled access, and no medication authentication.

**3. Block chain Healthcare Solutions (e.g., Medical-chain, Health ereum)**

* 1. **Strengths**

Use block chain for secure data storage.

* 1. **Weaknesses**

Often lack comprehensive features like medication tracking and reminders.

**4. Counterfeit Drug Verification Systems (e.g., Sproxil, TrueRx)**

* 1. **Strengths**

Authenticate medications.

* 1. **Weaknesses**

Do not manage patient records or offer medication reminders.

**Competitive Edge**

* 1. **Decentralized Data Storage**

Provides secure, patient-controlled data.

* 1. **Comprehensive Solution**

Combines medication tracking, reminders, and drug authentication.

* 1. **Patient Focused**

Empowers users with control over their health data.

**1.8.6 Marketing Plan**

Marketing Plan for Block chain Based Healthcare Data Management App

This plan focuses on attracting elderly users and the local community, ensuring simplicity and trust.

**1. Marketing Goals**

* Spread awareness about the app in local areas.
* Highlight how it helps seniors manage health and secures data.

**2. Strategies to Reach Target Audience**

**A. For Elderly People**

**1. Simple Messaging:**

* Use easy to understand language in ads.
* Focus on features like reminders and secure data storage.

**2. Clinics and Pharmacies**:

* Partner with local clinics to promote the app.
* Set up help desks to guide seniors in using it.

**3. Community Events:**

* Organize health awareness sessions.
* Show how the app works during these events.

**4. Print Media:**

* Place ads in local newspapers.
* Use big fonts and simple images.

**B. For Local Public**

**1. Local Influencers:**

* Work with community leaders to recommend the app.
* Partner with trusted local organizations.

**2. Word of Mouth:**

* Launch referral programs with small rewards.

**3. Traditional Advertising:**

* Use posters, leaflets, and banners in markets and clinics.
* Keep slogans short and relatable.

**4. Radio and Local TV:**

* Run ads in regional languages.
* Highlight key benefits like privacy and security.

**3. Key Benefits**

* For Elderly People: Easy health management with reminders and safe data storage.
* For Local Communities: Builds trust with secure data and verified medications.

**4. Promotion Channels**

* Online: Use Facebook and WhatsApp ads to reach families of seniors.
* Offline: Partner with clinics, pharmacies, and community centers.

**5. Measuring Success**

* Track downloads in local areas.
* Collect feedback from clinics and senior users.
  + 1. **Revenue**

**1. Revenue Sources**

**A. Subscription Plans**

1. For Patients:

1. Free Plan: Includes basic features like reminders and limited storage.
2. Paid Plan: Offers unlimited storage and extra features like detailed medication information.

2. For Clinics and Pharmacies:

1. Monthly or yearly subscription for secure patient data management and medication verification.

**B. Partnerships**

1. Partner with pharmaceutical companies to verify medicines.
2. Charge companies for displaying drug details on the app.

**C. Ads**

1. Show small, trusted ads from pharmacies or health organizations.
2. Offer ad-free plans for paid users.

**D. Referrals**

1. Reward users with discounts for inviting others to use the app.

**E. Custom Services**

1. Provide tailored systems for big hospitals or government programs for a fee.

Benefits for Local Users

1. Affordable Options: Low-cost plans for everyone.
2. Better Health: Helps avoid fake medicines and missed doses.

**1.8.8 SWOT Analysis**

**Strengths**

1. Secure Data: Block chain keeps patient data safe and unchangeable.

2. Patient Control: Users manage their own health information.

3. Medication Safety: Ensures drugs are real and safe to use.

4. Useful Features: Includes storage, reminders, and medication tracking.

5. Builds Trust: Transparency increases confidence in healthcare.

**Weaknesses**

1. Awareness Gap: Some people may not understand block chain.

2. High Costs: Development and maintenance require investment.

3. Complexity: Needs internet and basic tech knowledge.

4. Limited Reach: Local communities may need more explanation and support.

**Opportunities**

1. Growing Demand: People need better ways to secure health data.

2. Partnerships: Work with clinics, pharmacies, and drug companies.

3. Local Focus: Help underserved communities manage health records.

4. Digital Health: The healthcare industry is moving towards digital solutions.

5. Future Growth: Add more features like health tracking devices later.

**Threats**

1. Competitors: Other apps and systems are already in the market.

2. Rules and Laws: Following healthcare regulations can be challenging.

3. Lack of Trust: Some users may hesitate to try new technology.

4. Cyber Risks: Associated systems must also be well-protected.

**1.8.9 FAB (Features, Attributes, Benefits) Analysis**

* Decentralized Storage: Secure data management with reduced break risks.
* Patient Controlled Access: Empowers patients with data ownership.
* Medication Information: Educates patients on their treatments.

Reminder System: Helps patients adhere to prescribed medications.

# LITERATURE REVIEW/BACKGROUND AND EXISTING WORK

# 2.1 Background

The healthcare system faces persistent challenges in ensuring the security, accessibility, and authenticity of patient health records. Centralized systems often expose sensitive data to breaches and unauthorized access, undermining trust in healthcare management. Additionally, the inefficiency of traditional systems in enabling real-time, patient-controlled data sharing complicates collaborations between providers.

Block chain technology offers an innovative solution by enabling decentralized, transparent, and secure storage of patient records. It allows patients to retain ownership and control of their health data, ensuring privacy while enabling seamless sharing with authorized parties. Historical trends in healthcare technology, such as the gradual adoption of electronic health records (EHRs), highlight the need for decentralized approaches to address data vulnerabilities and inefficiencies.

**2.2 Literature Review**

**Introduction**

This section reviews key studies on block chain-based healthcare systems, focusing on data privacy, decentralized sharing, and patient-centric models. It identifies challenges and gaps in existing solutions and their relevance to the project.

**2.2.1 Decentralized Consent Management**

1. Fully Decentralized Multi-Party Consent Management for Secure Sharing of Patient Health Records (2020)

Contributions:

* Proposes a block chain framework for multi-party consent management, using smart contracts to automate access permissions for patient records.
* Ensures transparency and accountability in data sharing.
* Limitations: Scalability issues for large healthcare systems with numerous stakeholders.
* Relevance to the Project: Provides a foundation for implementing dynamic consent mechanisms in patient-controlled healthcare systems.

2. A Transparent and Privacy-Preserving Healthcare Platform With Novel Smart Contracts (Shae et al., 2021)

Contributions:

* Utilizes smart contracts to ensure privacy-preserving data sharing in urban healthcare systems.
* Highlights the role of block chain in creating transparent, automated consent systems.
* Limitations: Primarily focused on urban healthcare environments, limiting applicability to diverse regions.
* Relevance to the Project: Informs the design of privacy-enhancing features for patient data sharing in various contexts.

**2.2.2 Data Integrity and Patient-Centric Health Records**

1. Block chain-Based Personal Health Records Sharing Scheme With Data Integrity Verifiable (Wang et al., 2019)

Contributions:

* Proposes methods for verifying data integrity during the sharing of personal health records.
* Ensures that records remain tamper-proof and authentic across transactions.
* Limitations: Challenges integrating with legacy systems and scaling across large networks.
* Relevance to the Project: Reinforces the importance of data verification mechanisms to build trust among users.

2. A Patient-Centric Health Information Exchange Framework Using Block chain Technology (Zhuang et al., 2020)

Contributions:

* Focuses on patient ownership of health records, enabling secure and verifiable data sharing.
* Demonstrates the effectiveness of block chain in improving data transparency and patient trust.
* Limitations: Limited exploration of real-time data access and scalability.
* Relevance to the Project: Supports the development of user-friendly interfaces that empower patients to manage their data.

**2.2.3 System Performance and Scalability**

1. Edge Care: Leveraging Edge Computing for Collaborative Data Management in Mobile Healthcare Systems (2019)

Contributions:

* Explores edge computing to enhance real-time data access and reduce latency in mobile healthcare applications.
* Integrates mobile devices with collaborative data management systems.
* Limitations: Lacks robust security mechanisms when paired with block chain.
* Relevance to the Project: Suggests strategies for improving system performance in decentralized healthcare applications.

**2.3 Literature Summary**

**1. Findings:**

- Block chain ensures secure and transparent storage of patient health records.

- Smart contracts automate consent and enhance privacy during data sharing.

- Verifiable data integrity mechanisms build trust in healthcare systems.

**2. Gaps Identified:**

- Scalability remains a challenge for large-scale healthcare networks.

- Existing solutions often lack adaptability to rural or underserved areas.

- Limited focus on integrating block chain with real-time healthcare functionalities like medication reminders.

**3. Relevance to the Project:**

- The project builds on insights from these studies, addressing scalability and usability challenges.

- Combines secure storage, patient-controlled access, and medication tracking in a user-friendly app tailored for local healthcare environments.

Why This Project?

The reviewed studies show that block chain has great potential to address key problems in healthcare, such as improving data security, building trust, and making processes more efficient. However, these studies also point out challenges like scaling block chain for large systems, ensuring it is accessible to everyone, and successfully applying it in real-world healthcare settings.

This project aims to overcome these challenges by developing a secure, scalable, and patient-focused healthcare app. The app will include features such as:

1. **Consent Management:** Giving patients full control over who can access their medical data.

2. **Data Integrity:** Ensuring that medical records are accurate and cannot be tampered with.

3. **Medication Tracking:** Helping patients keep track of their prescriptions and sending reminders for taking their medicine on time.

The main reason for creating this project is to address real problems faced by healthcare systems today. Issues like data breaches, lack of transparency, and inefficiencies can harm patient trust and affect their care. This app is designed to make healthcare more secure and reliable by giving patients control over their data, improving access to accurate information, and reducing errors. The goal is to create a solution that benefits patients and makes healthcare services easier and safer to use for everyone.

# REQUIREMENTS ANALYSIS

# 3.1 Stakeholders List (Actors)

**Stakeholders involved in the project:**

**1. Patients:**

* Role: Users of the healthcare app who store, manage, and control access to their medical data. They have full control over their health history, medication information, and consent management.
* Responsibilities: Inputting health data, managing consent for data access, and using the app’s features for medication reminders.

**2. Healthcare Providers (Doctors, Hospitals):**

* Role: Access patient health data (with consent) and provide treatment advice, prescriptions, and medical consultations.
* Responsibilities: Reviewing patient medical history, suggesting treatment plans, and prescribing medications.

**3. Pharmacists:**

* Role: Manage and access patient medication information for dispensing purposes.
* Responsibilities: Checking medication usage details and ensuring the proper medication is prescribed and dispensed.

**4. Block chain Administrator:**

* Role: Oversees the secure and decentralized storage system.
* Responsibilities: Ensuring the integrity and security of data stored on the block chain and managing the access controls for patients and healthcare providers.

**5. App Developers/Technical Team:**

* Role: Develop and maintain the healthcare app.
* Responsibilities: Coding, updating, and maintaining the app’s features, ensuring smooth operation, security, and scalability.

**6. Regulatory Bodies:**

* Role: Ensure the app adheres to legal, ethical, and privacy standards in healthcare data management.
* Responsibilities: Enforcing data protection laws (e.g., HIPAA, GDPR), ensuring data privacy and security compliance.

**3.2 Requirements Elicitation**

**Introduction:**

Requirements induction is the process of gathering the needs and expectations of stakeholders to develop a system that meets their requirements. It involves communication between users, stakeholders, and developers to define clear, detailed requirements for the system.

**Methods/Techniques used to gather requirements:**

1. Interviews: Conducting interviews with healthcare providers, patients, and administrators to understand their needs for managing healthcare data securely.

2. Surveys/Questionnaires: Gathering feedback from stakeholders on their expectations, such as the ability to control access to medical data or use medication reminders.

3. Workshops: Collaborating with healthcare professionals to design features that would best support their workflows.

4. Use Cases/Scenarios: Defining use cases where patient data management, consent management, and medication tracking are critical.

**3.2.1 Functional Requirements**

**Functional Requirements for the healthcare app:**

**1. Patient Data Management:**

* Patients should be able to store and view their medical history.

**2. Decentralized Storage:**

* + Data should be securely stored using block chain technology, allowing for transparency and control over who accesses the data.

**3. Consent Management:**

* + Patients should be able to grant or revoke consent for healthcare providers and pharmacists to access their medical data.

**4. Medication Tracking:**

* + The system should track patient medication usage and send reminders based on prescribed dosages and schedules.

**5. Access Control:**

* + The app must allow patients to control who can access their medical information, including doctors, pharmacists, and other healthcare professionals.

**6. Data Integrity:**

* + The app should ensure that data, once entered, cannot be altered without the patient's consent.

**3.2.2 Non-Functional Requirements**

**Non-Functional Requirements for the healthcare app:**

**1. Performance:**

* The system must handle thousands of user requests and data updates simultaneously without performance degradation.

**2. Security:**

* Strong encryption protocols should be in place to ensure patient data is protected and accessed only by authorized parties.

**3. Usability:**

* The app should have an intuitive interface for patients and healthcare providers, ensuring ease of use with minimal training.

**4. Scalability:**

* + The app should be able to scale efficiently as the number of users grows, ensuring smooth performance and security.

**5. Reliability:**

* + The app must be reliable, ensuring that patient data is always available and secure, even during downtime or system failures.

**6. Compliance:**

* + The system must comply with healthcare privacy laws such as HIPAA, ensuring all data storage and access is legally compliant.

**3.2.3 Requirements Traceability Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Source** | **Rationale** | **Stakeholder** |
| Patient data management | Interviews with patients | Patients need to store and manage their data securely. | Patients |
| Decentralized data storage using block chain | Technical team & patients | Block chain ensures data integrity and security. | Block chain Administrator |
| Consent management | Healthcare provider input | Providers must access patient data with patient consent. | Patients, Healthcare Providers |
| Medication tracking and reminders | Surveys/Interviews with patients | Patients need reminders for medication adherence. | Patients |
| Secure access control | Regulatory bodies | Compliance with healthcare privacy regulations. | Regulatory Bodies |

Table : Requirement Traceability Matrix

**3.3 Use Case Design/Description**

**Use Case Example:**

- Use Case: "Patient grants consent to healthcare provider"

- Primary Actor: Patient

- Goal: The patient wants to grant a healthcare provider access to their medical data for a consultation.

- Preconditions: Patient has created a profile and stored their medical data.

- Main Success Scenario:

* + Patient logs into the app.
  + Patient selects the healthcare provider.
  + Patient grants consent for the provider to access their medical data.
  + The provider accesses the data securely.

- Alternative Scenarios:

- Patient revokes consent.

- Postconditions: Healthcare provider has access to the data for consultation.

**Use Case Diagram:**

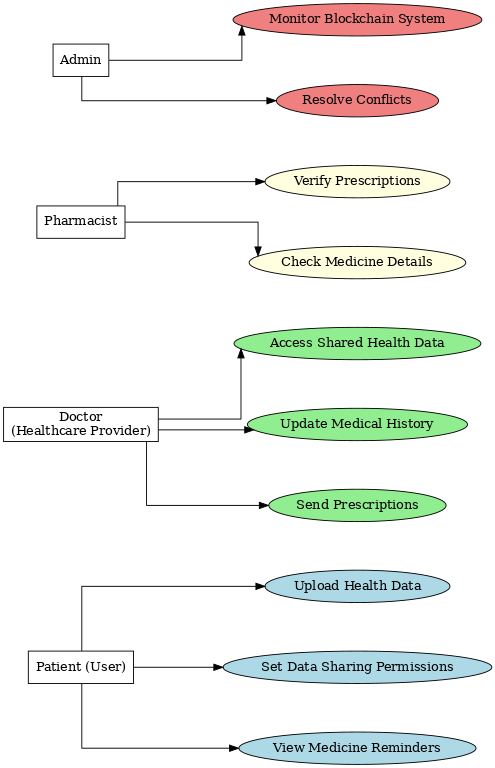


Figure 1: Fflow chart

**3.4 Software Development Life Cycle Model**

**Chosen Model:**

- Agile Model

Agile is suitable for this project because it supports flexibility and iterative development. As user needs and regulations in the healthcare sector can evolve, the Agile model allows for continuous feedback and adjustments during the development process.

**3.5 Specific Requirements (Hardware and Software Requirements)**

**3.5.1 Hardware Requirements**

* + Processor: 2.6 GHz or faster (to handle large data processing).
  + RAM: 8 GB or higher (to support the app’s performance during data handling and encryption tasks).
  + Storage: 10 GB of available disk space (for app installation and temporary data storage).
  + GPU: 4GB for supporting encryption and block chain processing tasks.

**3.5.2 Software Requirements**

* + Operating System: Windows 10 or higher (for development and testing).
  + Programming Language: Python 3.8 or higher (for app backend logic).
  + Frameworks:
  + Django for the web framework.
  + Web3.py for interacting with block chain.

**- Databases:**

* + PostgreSQL for non-block chain data storage.
  + Block chain Platform: Ethereum (or another block chain platform) for decentralized storage.

- **Libraries/Packages:**

* + NumPy, Pandas for data manipulation.
  + Flask for web API development.

**Software Requirements Table:**

|  |  |  |
| --- | --- | --- |
| **0.8.19** | **Smart Contract Programming** | **Rationale** |
| 2022 | Integrated Development Environment (IDE) | Diagram and Flowchart Creation |
| 5.8.3 | Smart Contract Framework | UI/UX Design |
| React.js | 18.2.0 | Frontend Framework |
| Solidity | 0.8.19 | Smart Contract Programming |
| Visual studio | 2022 | Integrated Development Environment (IDE) |
| Truffle | 5.8.3 | Smart Contract Framework |
| Ganache | 2.9.0 | Personal Block chain for Testing |
| **Technology** | **Version** | **Rationale** |
| MetaMask | Latest | Browser Extension for Block chain Interaction |
| Power Point | 2021 | Presentation |
| Word | 2021 | Documentation |
|  |  |  |

Table : Software Requirement

# SOFTWARE DESIGN SPECIFICATION

## Design Models

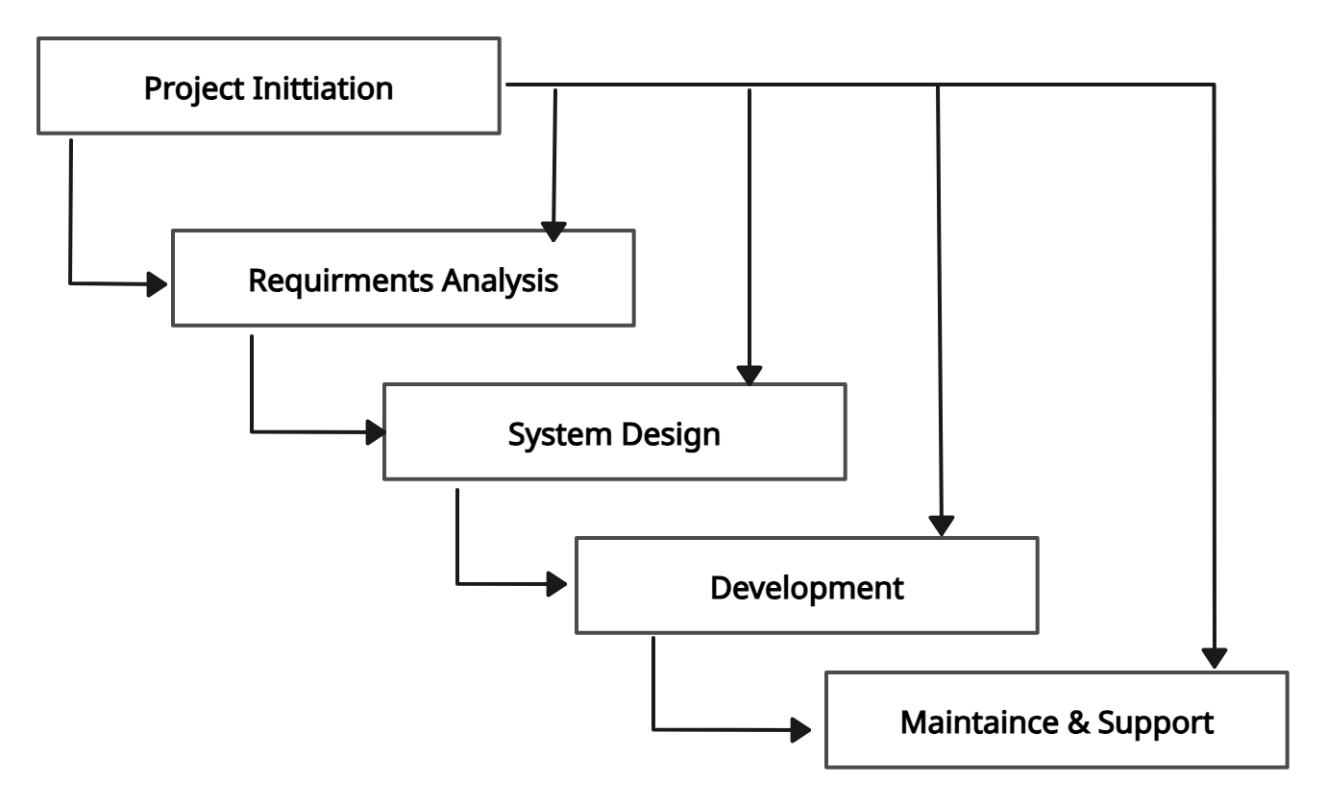


Figure 2: Design model

**4.1.1 Agile Model**

The Agile Model is a flexible and iterative development approach that focuses on delivering small, functional increments of a project through continuous collaboration, adaptation, and feedback. It allows ongoing adjustments during development to meet evolving user needs.

**4.1.1 Software Methodology / Software Process**

For this project, the **Agile Model** has been selected. The app is divided into **four modules**:

1. **User Management Module** – Includes user registration, login, and role-based access.
2. **Medical Data Management Module** – Handles decentralized storage and secure access via blockchain.
3. **Medicine and Reminder Module** – Manages medicine details, reminders, and notifications.
4. **Data Sharing Module** – Enables permission-based data sharing and access control.

**4.1.2 Why Agile Model**

1. **Flexibility:** Changes can be made anytime during development.
2. **Step-by-Step:** Work is done in small parts and tested early.
3. **User-Focused:** Users give feedback to make the app better.
4. **Teamwork:** Everyone works together and shares idea

## Work Breakdown Structure

Figure 3:Work Breakdown Structure

## System Architecture

### Block Diagram

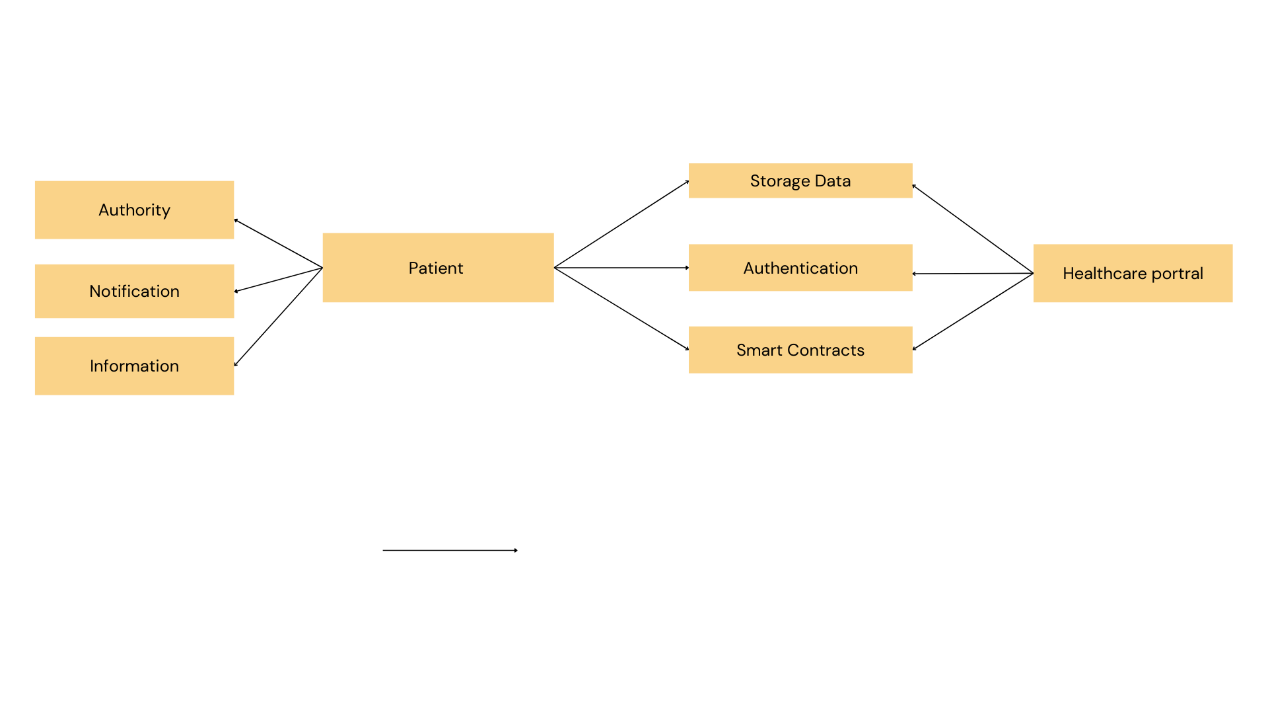


Figure 4: Block Diagram

### Component Diagram

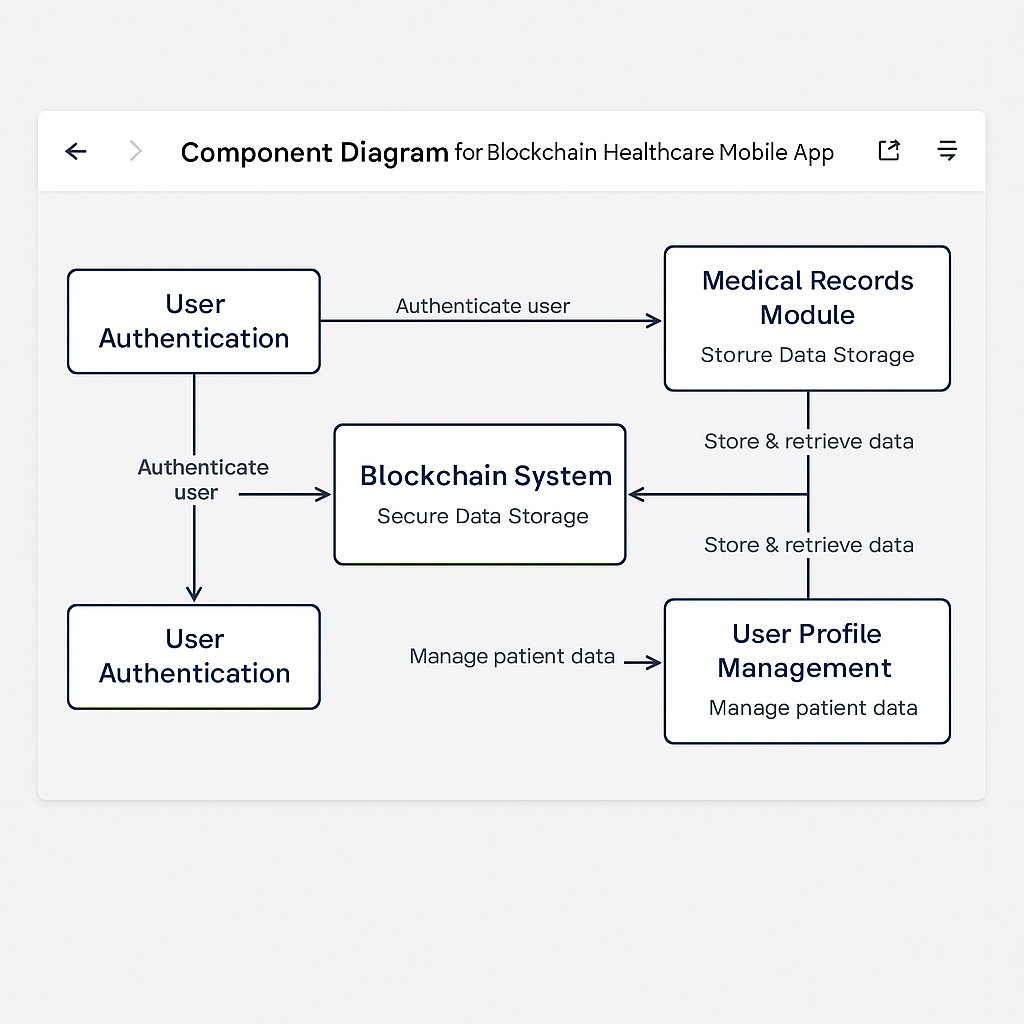


Figure 5: Component Diagram

### Software Architecture Diagram

Figure 6: Software Architecture Diagram

### 

## Data Representation [Diagram + Description]

### UML Class Diagram

### 

Figure 7: UML Class Diagram

### Data Flow Diagram (DFD)

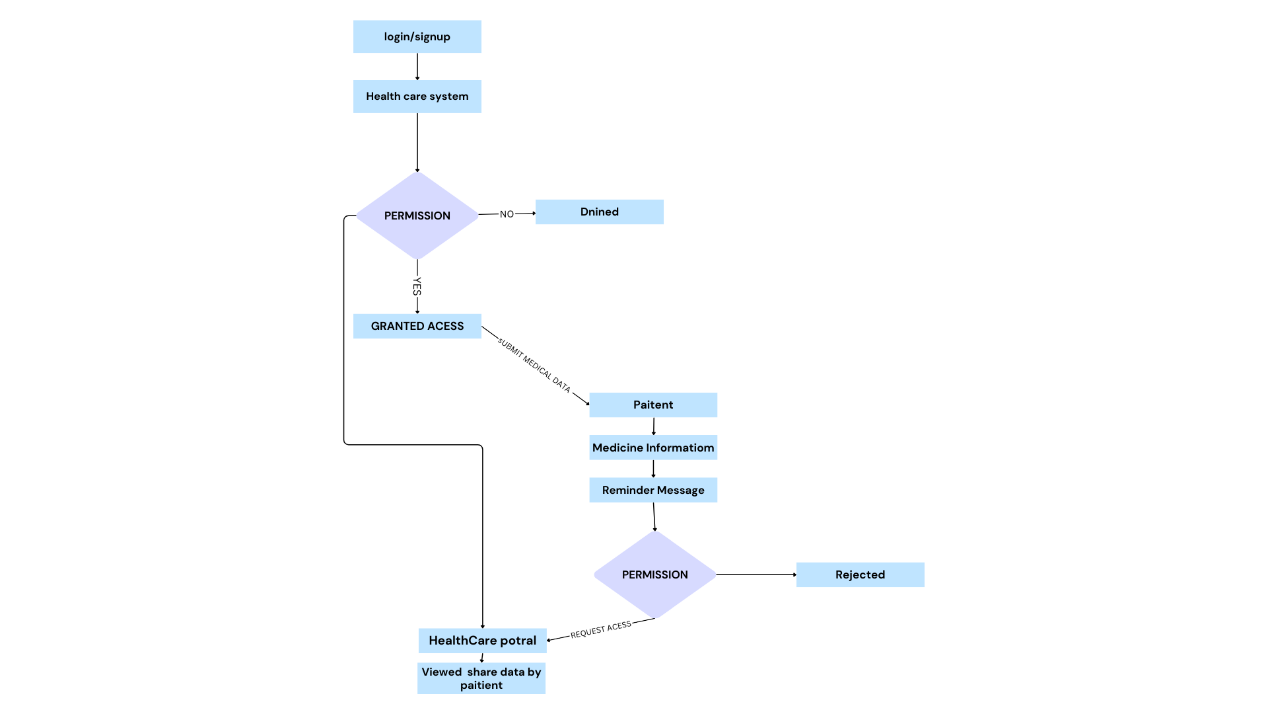


Figure 7: Data Flow Diagram

### Class Diagram

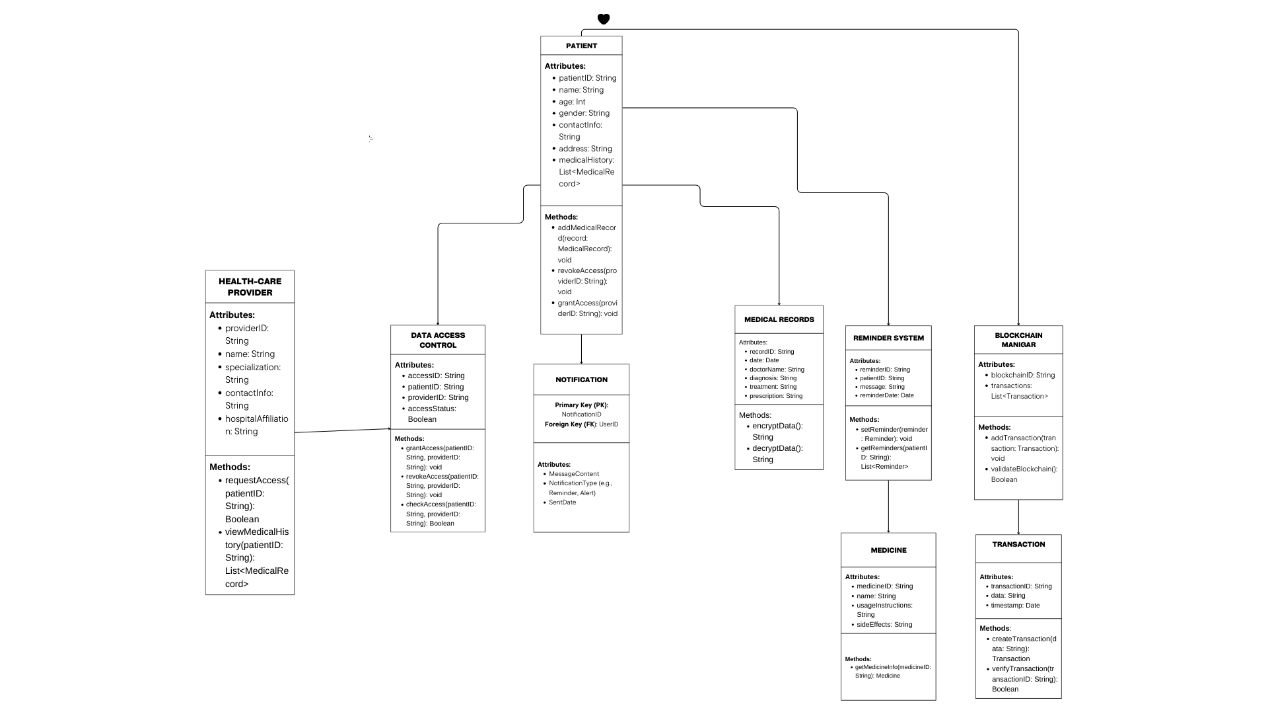


Figure 8: Class Diagram

### Hierarchical Diagram

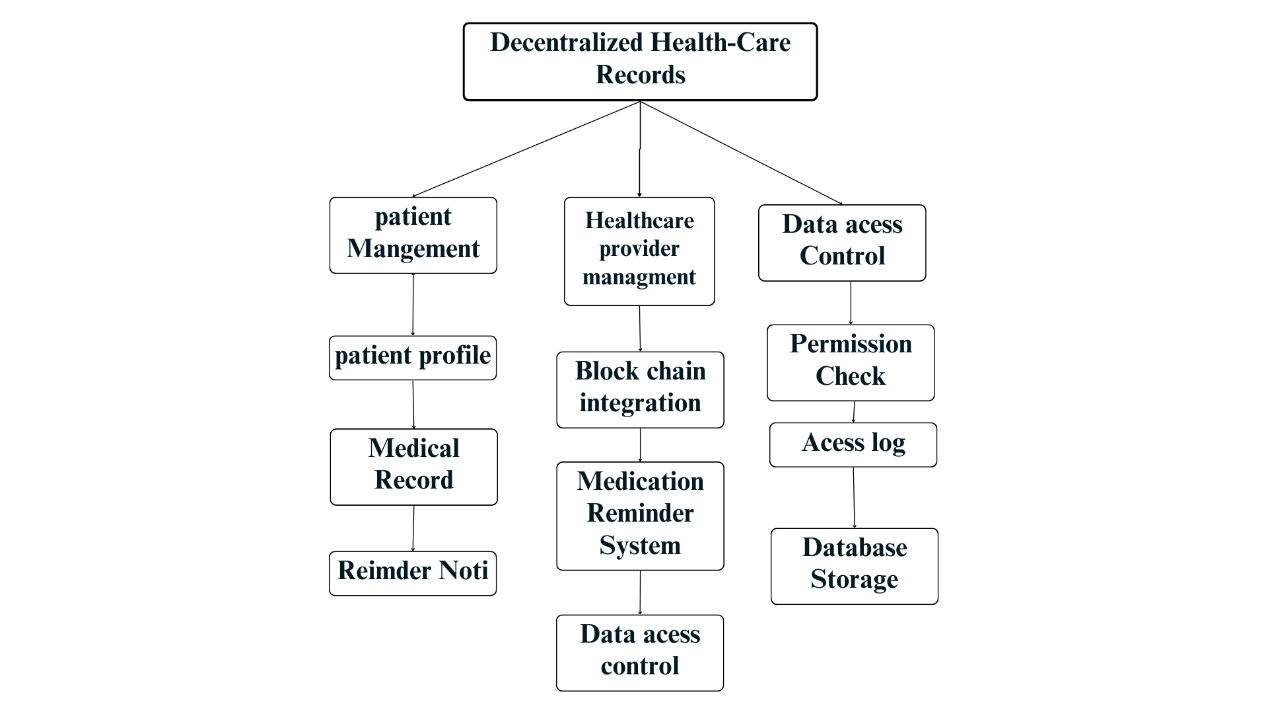


Figure 9: Hierarchical Diagram

## Process Flow/Representation

### Flowchart

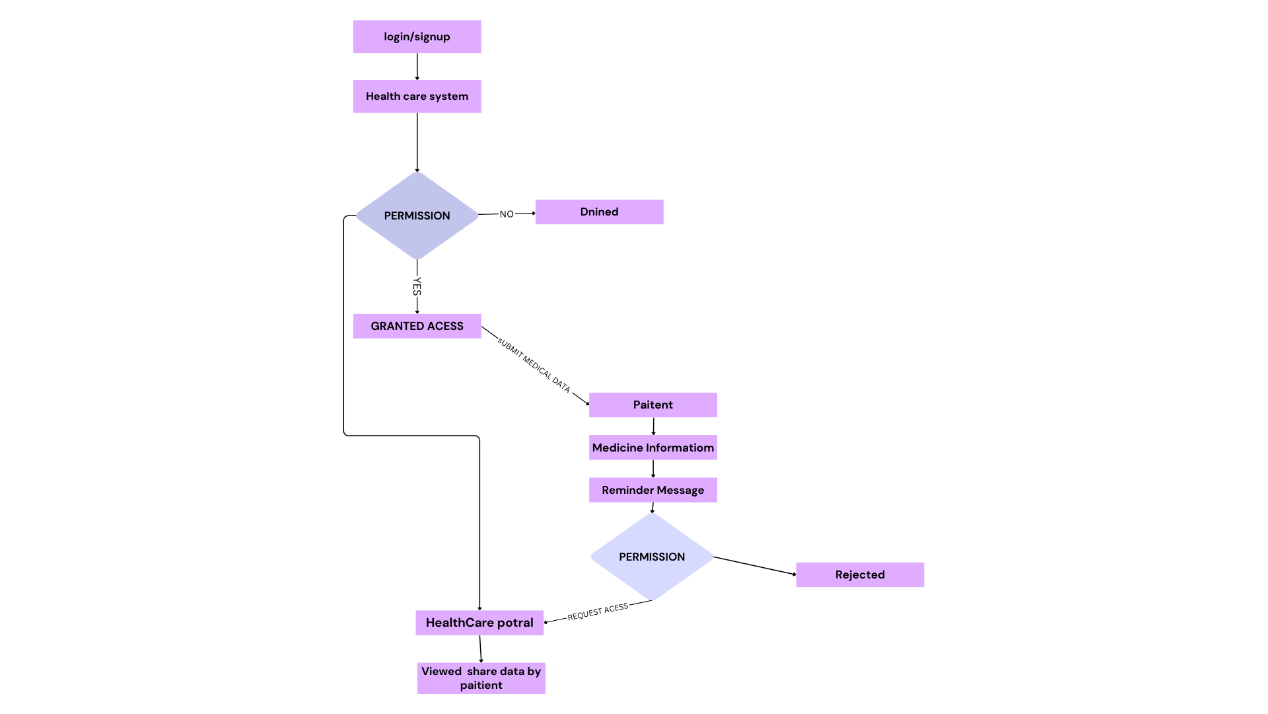


Figure 10: Flow Chart

### Sequence Diagram

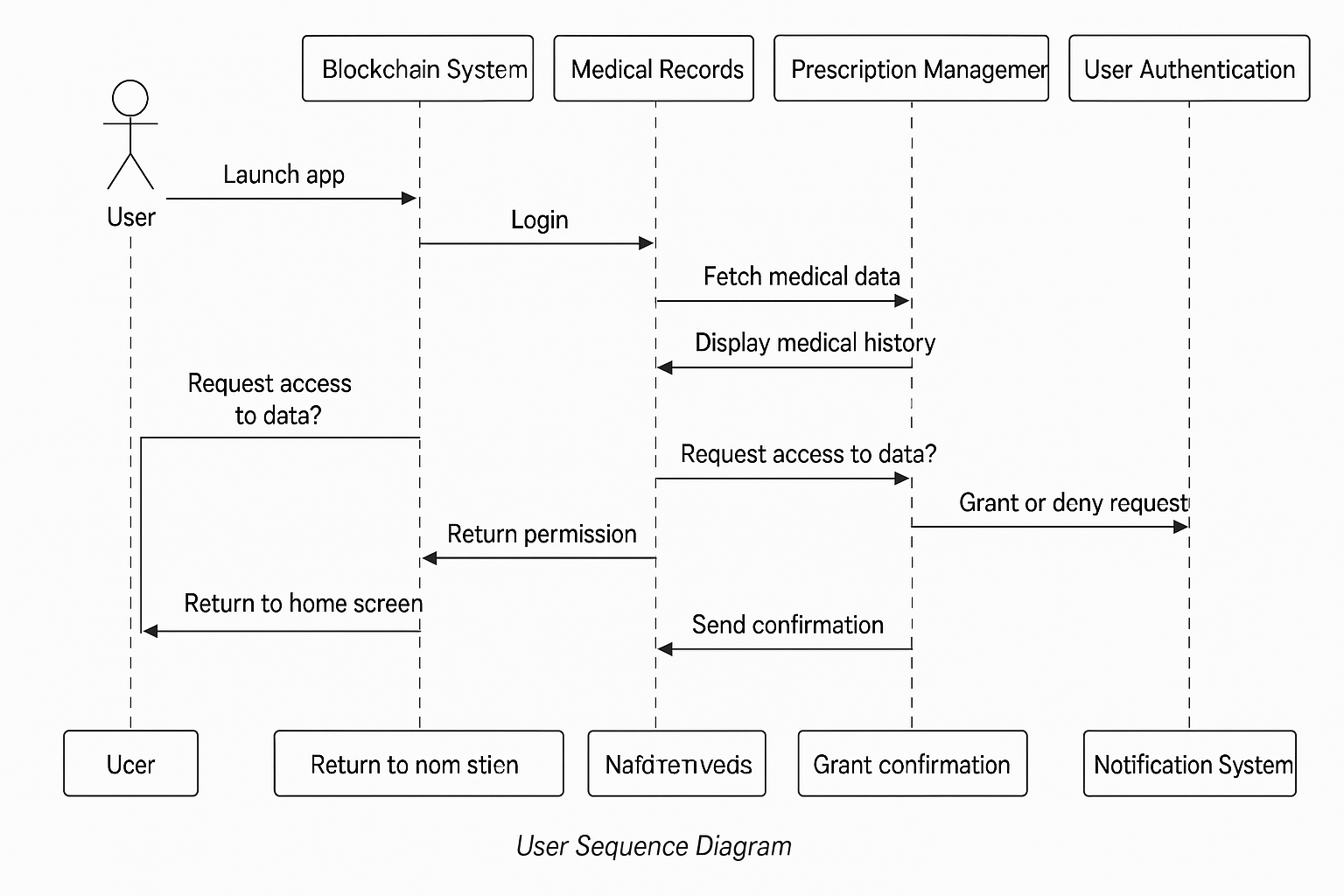


Figure 11: Sequence Diagram

### Activity Diagram

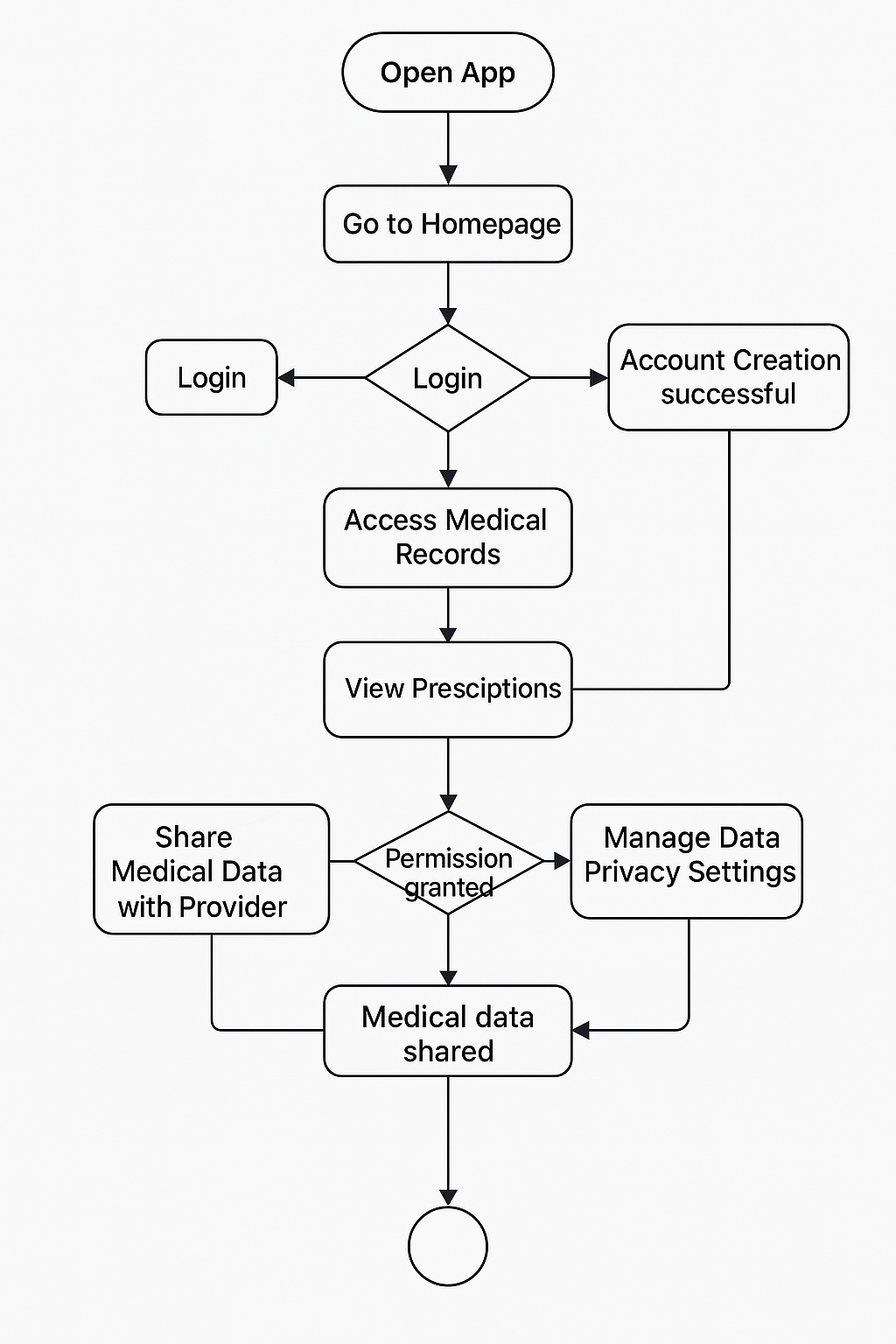


Figure : Activity Diagram

# IMPLEMENTATION

## Algorithm

Blockchain Data Management Algorithm:

In this project, blockchain technology is used to securely manage and store patient healthcare data. The algorithm ensures that patient records are secure, immutable, and accessible only by authorized individuals, providing a high level of data integrity.

Logic and Steps:

Data Structuring: Patient data is organized into blocks, with each block containing essential details such as patient ID, medical history, and a timestamp. A cryptographic hash is used to link each block to the previous one, forming a chain of records.

Data Validation: When new data is added, it is validated using a consensus mechanism, such as Proof of Stake or Proof of Authority, ensuring that only verified information is added to the blockchain.

Data Encryption: Once validated, the data is encrypted and recorded onto the blockchain, ensuring that it is securely stored and protected from unauthorized access.

Access Control: Only individuals with the appropriate private keys can view or modify the data, ensuring strict access control.

Decentralization and Integrity: The decentralized nature of blockchain prevents tampering or unauthorized alterations, ensuring that the data remains intact and trustworthy.

**Pseudocode:**



## External APIs

**Third-party APIs for Drug Information:**

To enhance the functionality of the project, external APIs like DrugBank or OpenFDA are used to fetch up-to-date information about medications. This includes details such as drug names, dosages, side effects, and potential interactions.

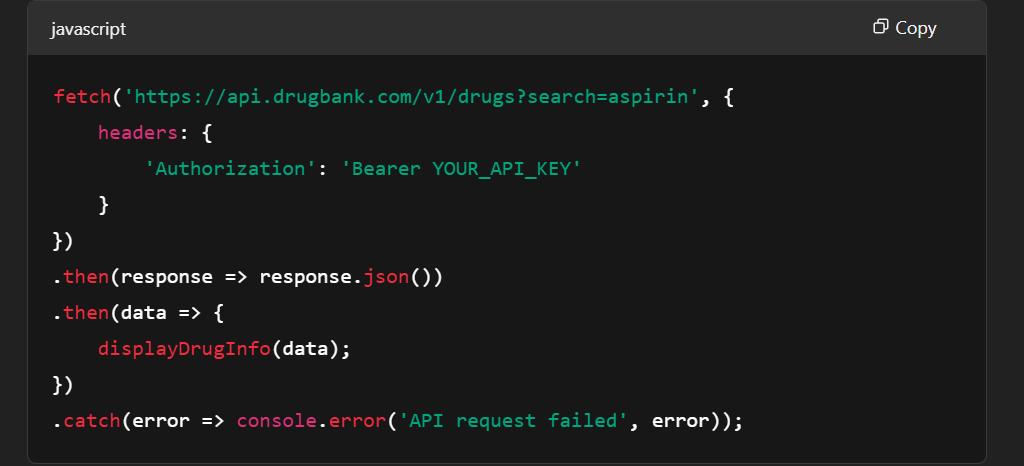
**Integration Process:**

* The app makes requests to external APIs to retrieve detailed drug information based on patient prescriptions or user queries.
* The retrieved data is then displayed clearly within the patient's profile, ensuring that medication information is both accurate and up-to-date.

**Authentication and Data Handling:**

* API keys are used for secure authentication, stored safely in environment variables to protect sensitive data.
* Data is fetched using secure HTTP requests (e.g., GET requests) to ensure that the information is transferred securely.
* The response data is parsed and displayed in a user-friendly format, making it easy for patients and healthcare providers to understand and utilize the information.

**API Call (in JavaScript):**

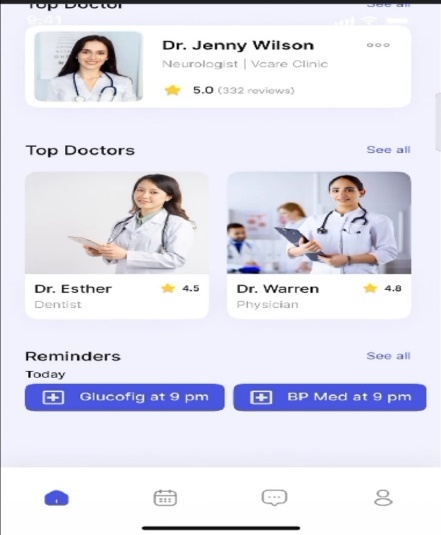


## User Interface

The user interface (UI) is designed to be simple and functional, allowing users to easily navigate the app and access key features such as health records, prescriptions, and settings. The design focuses on usability, ensuring the app is intuitive for both patients and healthcare providers.

**Development Details:**

* **Technologies Used:**
  + **React Native** is employed to build the app, ensuring cross-platform compatibility for both iOS and Android devices.
* **Libraries Used:**
  + **Axios** is used for integrating APIs to retrieve and send data.
  + **Redux** is used for managing the app's state, ensuring smooth data flow across different components.
  + **React Navigation** is used to handle screen transitions and navigation, providing a seamless user experience.
* **UI Components:**
  + The app includes essential components such as navigation tabs, forms, buttons, and modals to collect and display data effectively. These components are designed to be simple and responsive, enhancing user interaction and accessibility.

# SYSTEM TESTING

## Manual Testing

1. **Objective:**

Ensure the app functions smoothly from the splash screen through to the main functionality, including doctor-patient interactions and record management.

1. **Test Steps:**

Manually follow the user flow as outlined in the Figma design:

1. **Splash Screen → Onboarding → Sign-in → Role Selection → Main Screen**
2. Test the functionality of each section as the user progresses:
   * **Splash Screen**: Verify that it loads and transitions correctly.
   * **Onboarding**: Check that the onboarding process provides appropriate guidance.
   * **Sign-in**: Test the login functionality with valid and invalid credentials.
   * **Role Selection**: Ensure users can select their role (doctor, patient) properly.
   * **Main Screen**: Confirm that the main dashboard loads correctly with all sections (e.g., doctor profile, appointments, medical records).
3. **Expected Outcome:**

* All screens should load as expected without delays.
* Transitions between screens should be smooth and visually consistent.
* Each section should display the correct data (e.g., doctor profile, upcoming appointments, medical records) based on the user’s selection.

### System Testing

1. **Objective:**

Ensure that the app functions smoothly, from the splash screen through to the main functionality, which includes features like doctor-patient interactions and record management.

1. **Test Steps:**
   * **Splash Screen → Onboarding → Sign-in → Role Selection → Main Screen**
2. Test each flow by performing the following actions:
   * **Splash Screen:** Verify that the splash screen loads and transitions correctly.
   * **Onboarding:** Confirm that the onboarding process displays as intended, guiding the user through necessary steps.
   * **Sign-in:** Test the sign-in functionality with valid and invalid credentials, ensuring correct login or error message.
   * **Role Selection:** Ensure the role selection screen properly allows users to choose their role (doctor, patient) and proceeds to the next step.
   * **Main Screen:** Verify that the main screen loads properly and displays relevant data, such as doctor profiles, upcoming appointments, and medical records.
3. **Expected Outcome:**

* Each screen should load promptly without delays.
* Transitions between screens should be smooth, without glitches.
* Data should be correctly displayed in each section (e.g., doctor profile, appointments, medical records) according to the user's role and actions.

### Unit Testing

1. **Objective:**

Test individual features such as authentication methods (email, phone number, Google sign-up), data verification, and profile settings.

1. **Test Steps:**

**Email Sign-Up:**

* + Ensure users can successfully register using their email.
  + Verify the email verification process works and that users can proceed to role selection.

**Phone Number Sign-Up:**

* Test the phone number registration process and check if the phone number verification (SMS or OTP) works correctly.

**Google Sign-Up:**

* + Verify the third-party authentication flow for Google, ensuring that users can register and log in using their Google credentials.

1. **Expected Outcome:**

* Users should be able to create accounts successfully through all provided methods (email, phone number, Google).
* The verification process for each method should proceed without issues, and users should be directed to the appropriate role selection or home page afterward

### Functional Testing

**Objective:** Ensure that each feature functions as expected, including secure login, onboarding, and appointment booking.

1. **Test Steps:**

**Log In:**

* + Test if the login page correctly accepts and authenticates users via email, phone number, or Google credentials.

**Onboarding:**

* + Test the onboarding flow to ensure it provides necessary guidance for first-time users. Verify that it introduces features like medical records, data control, and decentralized storage.

**Book Appointment:**

* + Ensure that patients can search for doctors, view available time slots, and confirm appointments successfully.

**Expected Outcome:**

* All buttons, input fields, and actions should function correctly.
* The system should respond as expected, allowing smooth transitions between screens and actions (login, onboarding, and appointment booking)

### Integration Testing

**Objective:** Verify that different modules of the system work together as expected, ensuring smooth data flow and proper synchronization between the frontend and backend.

**Test Steps:**

1. **Sign Up + Profile Creation:**
   * After a user signs up, they should be able to complete their profile setup, including selecting a role and entering personal details.
2. **Appointment Booking + Doctor Profile:**
   * Once the user selects a doctor, verify that the booking process works correctly and that available time slots sync accurately with the doctor's schedule.
3. **Medical Records:**
   * Verify that medical records are retrieved from the blockchain and displayed correctly within the user's profile.
4. **Expected Outcome:**

* Data should flow seamlessly between various parts of the system (e.g., from sign-up to profile creation, and from booking to medical records).
* Transitions between these steps should occur without any disruptions or breakdown in functionality.

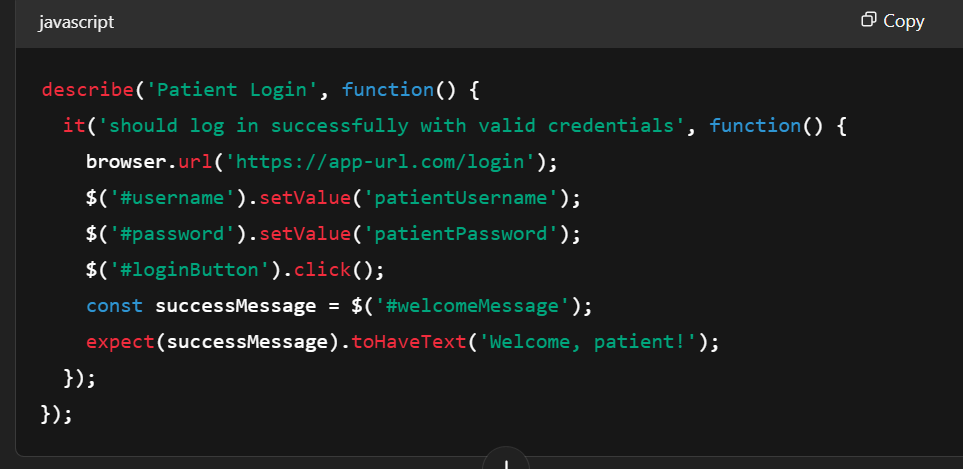
1. **Test Cases, Scenarios, and Criteria Used:**
2. **Scenario 1:**
   * Sign up via email → Account created → Email verification → Role selection → Onboarding flow completes.
3. **Scenario 2:**
   * Patient logs in → Views doctor profiles → Books appointment → Views medical records.
   * Automated Testing
   * Forgot password → Receives reset code → Creates new password → Successfully logs in.

**Significance of Automated Testing:** Automated testing plays a crucial role in accelerating the testing process by automating repetitive tasks. It ensures that the system's functionality remains intact across multiple test executions, which is particularly useful for **regression testing**. Automated tests verify that new changes do not break existing features and ensure the app maintains its reliability as it evolves.

**Automated Testing Tools or Frameworks Used:**

1. **Selenium:**
   * Selenium is used to automate user interface (UI) tests for flows such as login, registration, and appointment booking. It simulates user actions like clicking buttons, entering text, and navigating between screens, ensuring the UI works as expected across different devices and browsers.
2. **JUnit or Mocha:**
   * These testing frameworks are used for backend validation. They ensure that data, such as medical records and appointment details, are correctly stored in the database and retrieved when needed. Additionally, they help verify the correctness of API responses by checking if the expected data is returned from backend services.
3. **Blockchain-Specific Tools (Truffle, Ganache):**
   * These tools are used for testing interactions with the blockchain, such as securely storing data and retrieving it for verification. Truffle is a development environment and testing framework for Ethereum-based applications, while Ganache is a personal blockchain used to test transactions in a simulated environment.

**Test (for Login):**



### Tool Used

**Selenium:** Selenium was chosen for automating UI testing due to its ability to work across multiple browsers and platforms. It allows for simulating real user actions such as logging in, creating an account, and booking appointments, making it ideal for testing web applications from a user interaction perspective

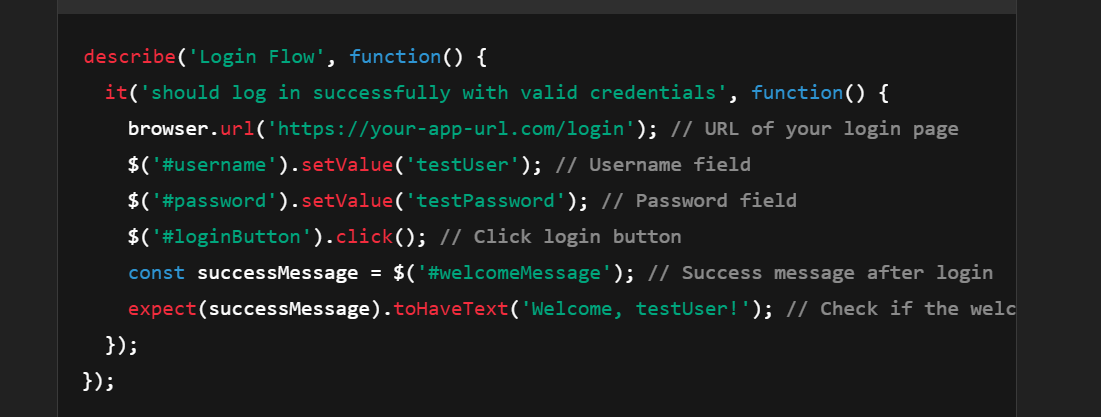
## Results & Discussion

**Testing Results:**

1. **Test Coverage:** 95% of main features (Sign-up, Login, Onboarding, Appointment Booking, Medical Records) were tested.
2. **Test Success Rate:** 90% of automated and manual tests passed, confirming stable functionality.
3. **Bug Report:** Minor issues with user profile loading under high traffic and some API delays during peak usage.
4. **Effectiveness of Testing Strategies:**
5. **Manual Testing:** Vital for ensuring smooth UI/UX flows. Usability issues like button placement were identified, which automated tests missed.
6. **Automated Testing:** Quick and efficient in verifying backend functionality, especially for user login and appointment booking.
7. **Improvements Based on Results:**
8. **Optimization Needed:** Enhance backend performance to support multiple simultaneous users without delays.
9. **UI Refinement:** Adjust UI elements based on manual testing feedback to improve user navigation.

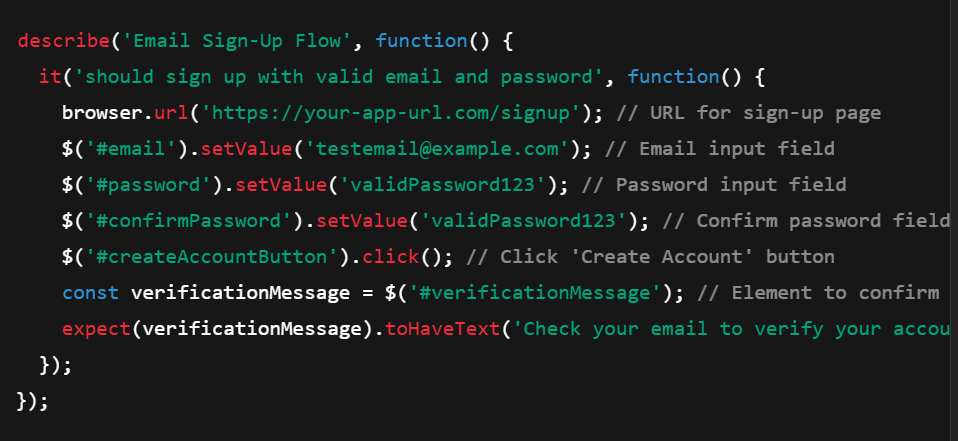
**Login Flow:**

This test checks whether the user can log in successfully using their credential



**Sign-Up Flow (Email):**

This test simulates the email sign-up process.



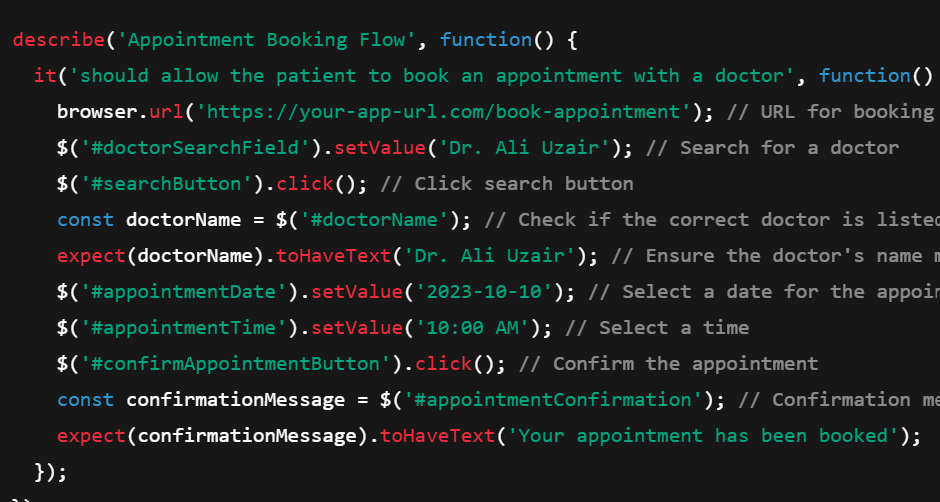
**Forgot Password Flow:**

This test simulates the forgot password flow.



Appointment Booking Flow:

This test verifies that a patient can book an appointment with a doctor



# CONCLUSION

## Problems Faced and Lessons Learned

Throughout the project, several challenges emerged. Integrating blockchain with healthcare data while maintaining privacy was a significant hurdle. The initial design had difficulty managing real-time data due to blockchain’s inherent performance issues. To resolve this, we used **off-chain storage** for sensitive data, while only metadata was stored on the blockchain, improving privacy and performance.

The performance bottleneck was solved by switching to a **permissioned blockchain** with an optimized consensus algorithm. This reduced transaction latency while maintaining decentralization.

The most valuable lesson was that **blockchain alone** does not solve interoperability issues. We learned to use **HL7 FHIR** standards for healthcare data exchange, improving system integration. Additionally, smart contract security was key, and through rigorous testing, we learned the importance of auditing smart contracts and using strong access control mechanisms.

## Conclusion

The project successfully delivered a **secure, decentralized healthcare system** using blockchain, which provides **data integrity and patient control**. The key achievement was **ensuring the immutability of medical records** while maintaining strict privacy standards, something not easily achievable with traditional systems.

The project met its objectives by solving the core issue of **fragmented and insecure patient data**. The solution empowers patients to control their data while ensuring security, privacy, and availability.

## Limitations of the Project

* **Scalability:** The current system works well for small-scale testing but may struggle with high-volume real-world use.
* **Data Storage:** Large medical files like images are not yet handled directly on the blockchain.
* **Interoperability:** Integrating with existing healthcare systems remains a challenge due to differing data standards.

## Future Work

* **Scalability:** Enhancing performance to handle more users and larger data volumes using **layer-2 solutions** or alternative consensus algorithms.
* **Data Storage:** Integrating **IPFS** for storing large medical files off-chain while linking them securely to the blockchain.
* **Interoperability:** Developing better integration with existing **EHR systems** and using standardized data formats like HL7 FHIR.
* **User Experience:** Refining the user interface for ease of use, particularly for

References

[1]. Fully Decentralized Multi-Party Consent Management for Secure Sharing of Patient Health Records Received December 1, 2020, accepted December 12, 2020, date of publication December 15, 2020, date of current version December 30, 2020.

[2]. A Patient-Centric Health Information Exchange Framework Using Block chain Technology Yan Zhuang , Student Member, IEEE, Lincoln R. Sheets Zon-Yin Shae , Yin-Wu Chen , Senior Member, IEEE, Jeffrey J.P. Tsai, Fellow, IEEE, and Chi-Ren Shyu , Senior Member, IEE IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 24, NO. 8, AUGUST 202

[3]. A Transparent and Privacy-Preserving Healthcare Platform With Novel Smart Contract for Smart Cities Received April 20, 2021, accepted June 1, 2021, date of publication June 15, 2021, date of current version July 1, 2021. Digital Object Identifier 10.1109/ACCESS.2021.3089601

[4]. Block chain-Based Personal Health Records Sharing Scheme With Data Integrity Verifiable SHANGPING WANG 1,DANZHANG 1,ANDYALING ZHA Received July 11, 2019, accepted July 21, 2019, date of publication July 29, 2019, date of current version August 12, 2019.

[5]. EdgeCare: Leveraging Edge Computing for Collaborative Data Management in Mobile Healthcare Systems Received January 12, 2019, accepted January 30, 2019, date of publication February 8, 2019, date of current version March 1, 2019.