

MASTER OF INFORMATION AND COMMUNICATION STUDIES
Capstone Project



**UNIVERSITY OF THE PHILIPPINES
OPEN UNIVERSITY**

MASTER OF INFORMATION AND COMMUNICATION STUDIES

MARK JOHNDY COPRADA

**IMPLEMENTING A MEDICATION TRACKING SYSTEM USING BLOCKCHAIN
TECHNOLOGIES**

Thesis Adviser:

RIA MAE BORROMEIO
Faculty of Information and Communication Studies

05 January 2023

Permission of the classification of this academic work access is subject to the provisions of applicable laws, the provisions of the UP IPR policy and any contractual obligations:

Invention (I)	<input type="checkbox"/> Yes	or	<input checked="" type="checkbox"/> No
Publication (P)	<input type="checkbox"/> Yes	or	<input checked="" type="checkbox"/> No
Confidential (C)	<input type="checkbox"/> Yes	or	<input checked="" type="checkbox"/> No
Free (F)	<input checked="" type="checkbox"/> Yes	or	<input type="checkbox"/> No

Student's signature:

Thesis adviser signature:

University Permission Page

IMPLEMENTING A MEDICATION TRACKING SYSTEM USING BLOCKCHAIN TECHNOLOGIES

“I hereby grant the University of the Philippines a non-exclusive, worldwide, royalty-free license to reproduce, publish and publicly distribute copies of this Academic Work in whatever form subject to the provisions of applicable laws, the provisions of the UP IPR policy and any contractual obligations, as well as more specific permission marking on the Title Page.”

“I specifically allow the University to:

Specifically, I grant the following rights to the University:

- a. Upload a copy of the work in the theses database of the college/school/institute/department and in any other databases available on the public internet*
- b. Publish the work in the college/school/institute/department journal, both in print and electronic or digital format and online; and*
- c. Give open access to the work, thus allowing “fair use” of the work in accordance with the provision of the Intellectual Property Code of the Philippines (Republic Act No. 8293), especially for teaching, scholarly and research purposes.*

Mark Johndy Coprada
January 05, 2023

Acceptance Page:

This paper prepared by **MARK JOHNDY COPRADA** with the title:
“IMPLEMENTING A MEDICATION TRACKING SYSTEM USING BLOCKCHAIN TECHNOLOGIES” is hereby accepted by the Faculty of Information and Communication Studies, U.P. Open University, in partial fulfillment of the requirements for the degree Course.

RIA MAE BORRAMEO
Adviser

24 January 2023

(Date)

ROBERTO FIGUEROA JR.
Program Chair

18 July 2023

(Date)

DIEGO SILANG MARANAN
Dean
Faculty of Information and Communication Studies

ABSTRACT

This project aims to address the issue of self-medicating with antibiotics in the healthcare sector. A web application was developed to improve medication therapy compliance and reduce the incidence of antibiotic resistance by enabling the dispatching, reporting, and monitoring of medications using blockchain technologies. The system architecture comprises three layers: a blockchain network powered by Hyperledger Fabric, an API gateway built using Golang and the go-zero code generator, and a user interface developed using React with Progressive Web App (PWA) features for enhanced user experience. Integration testing, unit testing and coverage testing were performed, and Google Lighthouse audits yielded positive results. The use of blockchain technologies in the project provides added benefits of security, reliability, availability, and transparency but also increases the complexity and resource requirements of the system.

TABLE OF CONTENTS

Abstract	iv
Table of Contents	v
Introduction	1
Review of Existing Alternatives	3
Project Details	4
A. Overview	4
B. Theoretical Framework	6
C. Technologies Used	7
D. System Design	9
E. Implementation	10
F. Resulting System	11
Project Assessment	19
A. Usability Testing	19
B. Unit Testing	19
C. Integration Testing and Bug Fixes	20
D. Google Lighthouse Audits	20
Discussions	23
Conclusion	24
Future Work	25
References	27
Appendices	28

Dedicated to:

My parents, who are always supportive of my decisions

Chapter I

INTRODUCTION

Studies [1, 3, 10] have shown that Filipinos tend to self-medicate whenever they feel under the weather. Some medications, primarily antibiotics, should not be taken without a physician's order. To add, it is alarming that Filipinos are more interested in a "silver bullet" drug instead of vaccines in response to the recent COVID-19 pandemic [2, 6].

Because of these behaviors, we are accelerating the rate at which bacteria may become antibiotic-resistant.

As someone who has attained a BS degree in nursing, I was interested in how I can contribute to improving the Philippine healthcare system. There was an initiative to digitize prescriptions from the government roughly four years ago [5], but there have been no updates.

The main objective of this project is to decrease the rate of occurrence of antibiotic resistance by improving the compliance rate of Filipinos when it comes to antibiotic therapy.

The project will be able to provide real-time data to pharmaceutical companies and possibly to the Department of Health (DOH) Antimicrobial Resistance Surveillance Program (ARSP). If successful, this project will open the doors to data transparency within the healthcare sector and pave the way to other government sectors where total transparency is essential.

The scope of this project includes the development of the following core components of the system: (1) the blockchain network, (2) API gateway, and (3) web application. The project will gather data on patients' medication intake. The web application will be visualizing the data.

The electronic Drug Safety System (eDSS) was first announced in 2018 and was expected to roll out by 2020 [5]. While the system sounded promising, there hasn't been any recent update if eDSS was successfully implemented or not, or if it was a system meant to be used internally. In contrast, the eDSS mainly focuses on digitizing prescriptions, whereas our project is about tracking patient compliance with any medication therapy.

One reason why Filipinos self-medicate or discontinue medication therapy (e.g. antibiotic therapy) is to save money [3] and aggravated by the fact that pharmacies selling antibiotics without prescriptions exist [4]. If a system that can provide evidence that antibiotics are widely misused can be created, then the general public can be alarmed by the risks associated with this behavior.

Chapter II

REVIEW OF EXISTING ALTERNATIVES

Two systems are related to this project: (1) mClinica's electronic Drug Safety System (eDSS) for digitizing prescriptions [5], and (2) Keyhole Software's whitepaper on tracking lab results using blockchain technologies [7].

eDSS is a system that modernizes pharmacies' logbook prescriptions and converts paper prescriptions into electronic data. This system can synergistically work with our system by synchronizing the medication records, instead of us populating the whole medication database. Furthermore, only the transactions (dispatched medications) are publicized by our system without publicizing any personal information.

This project was mostly inspired by a system developed by Keyhole Software that uses blockchain to track lab results. While related, I will be tackling a different problem — tracking patient compliance with medication therapy.

Even though there might have been undocumented attempts to solve our problem, what we're mostly interested in is utilizing blockchain technologies and their associated features to create a transparent and trustworthy system.

Chapter III

PROJECT DETAILS

A. Overview

The system architecture had three main routes for users to access: *Collector*, *Dispatcher*, and *Inspector*. These user-facing routes had specific functionalities for each user group.

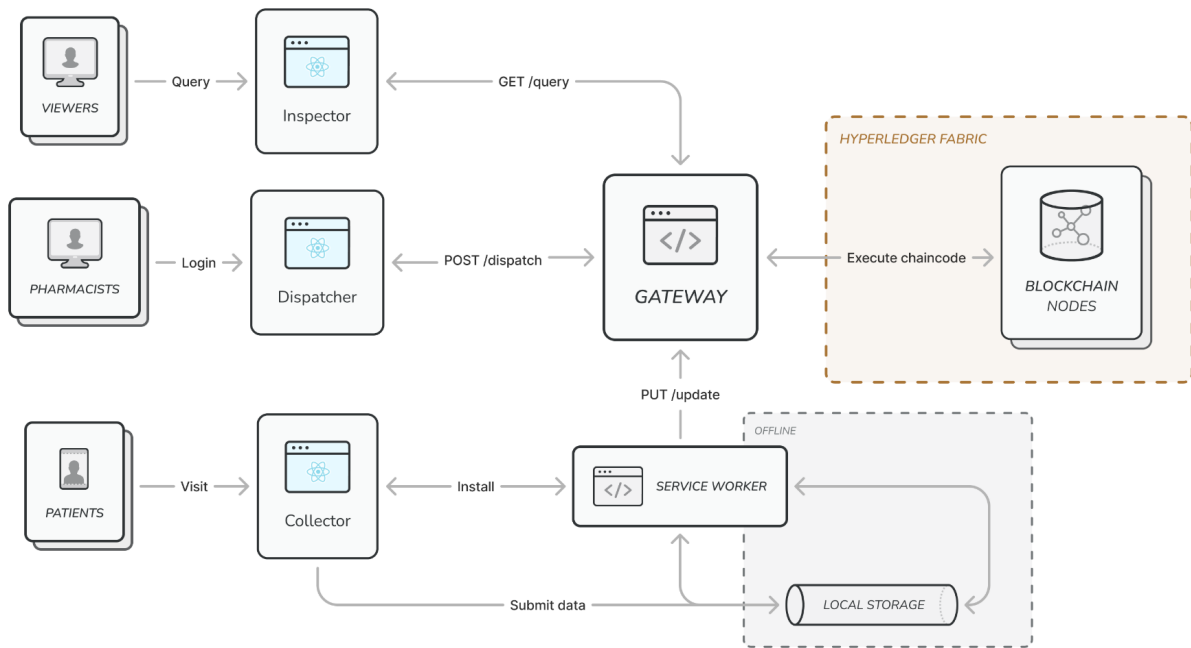


Fig. 1. System Architecture Design

The *Collector* route's purpose was to collect data from users who were receiving medication therapy. This route was also PWA-enabled. Making it PWA-enabled allowed users to install it as a standalone mobile application and use it offline, which made it a seamless experience.

The *Dispatcher* route was a dashboard for pharmacists to enhance the user experience for medication dispatch use. For every medication dispatch, the system generated a QR code. This QR code can then be printed by the pharmacists to give

to patients. In addition, the latest medication dispatches reflected on the top page of the dashboard in real time.

The *Inspector* route enabled querying medication compliance rates and visualized them as a basic bar chart. This route was publicly accessible, and anyone with network connectivity allowed viewing all data.

All routes communicated with the blockchain network via the API gateway. The API gateway and blockchain network's mode of communication was by using *Smart Contracts* – predefined rules that function as the business model within the blockchain network.

The system's target users were Patients, Pharmacists, and the general public for the Collector route, Dispatcher route, and Inspector route, respectively.

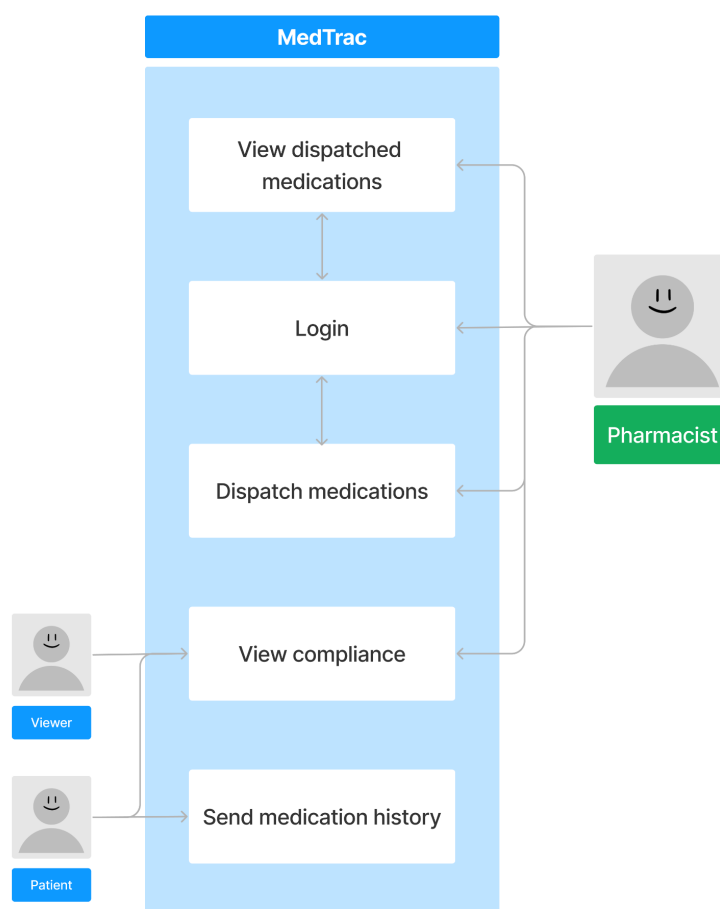


Fig. 2. Use Case Diagram

B. Theoretical Framework

The primary design principles used in the blockchain component of the project were decentralized and distributed computing, where the blockchain nodes were decentralized, and the blockchain ledger itself was distributed among the network.

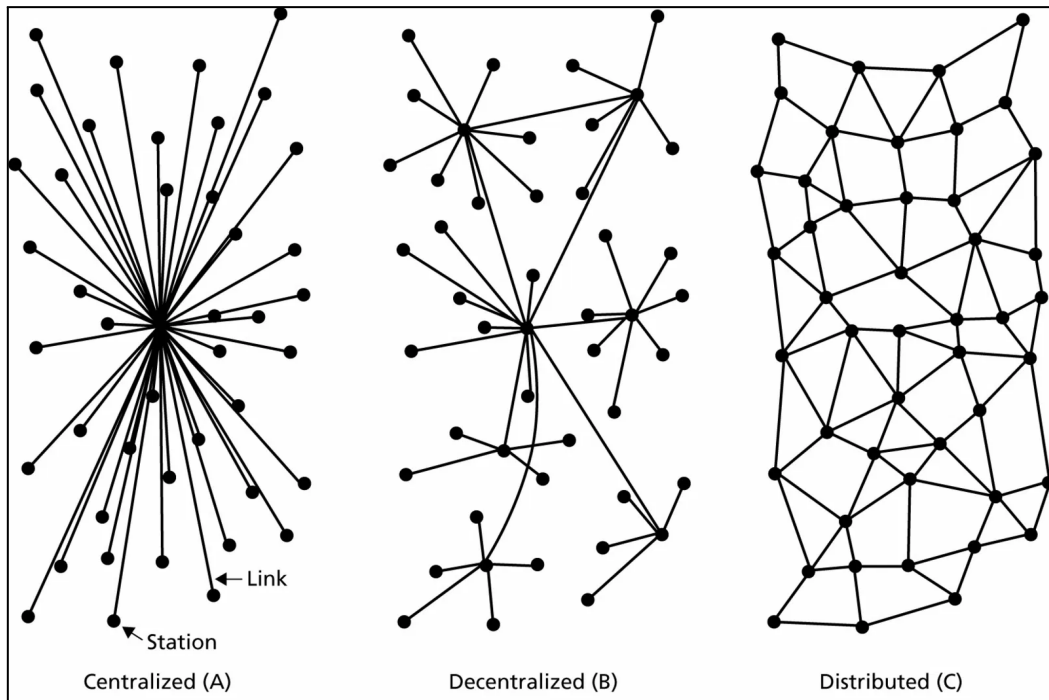


Fig. 3. P. Baran (1964), Centralized, Decentralized and Distributed networks [8]

The web application used a client-server architecture using Representational State Transfer (REST) to allow clients to execute processes on the blockchain network. Furthermore, the Application Programming Interface (API) endpoints were designed to reach Level 3 of the Richardson Maturity Model [9] as much as possible.

I have implemented the blockchain network as a decentralized and distributed system, where ideally, there shall be more than one node that operates.

In this way, every time a new block is created, all the nodes will verify and validate the transaction to confirm that the new block is legitimate. The blockchain ledger was distributed across all nodes to increase the availability and reliability of the system. When a node goes offline, the whole system will still function normally.

One benefit of following the REST architecture is that it is platform-agnostic. Any developer can use any programming language to implement the API as the request and response go through the standard HTTP web protocol. Another is that any representation appropriate for each API resource can be chosen, be it JSON, XML, WEBP, plain text, or any other data format available.

C. Technologies Used

a. Backend

Hyperledger Fabric was selected as the blockchain framework. Documentations were thorough; sample implementations were available online, making this project more manageable to grasp in terms of understanding the foundations of blockchain technologies.

Because *Hyperledger Fabric* is in Golang, our API gateway was also written in Golang. Implementing the chain code and API gateway was seamless when both services were in the same programming language. With the help of *go-zero*, a code generator for making API endpoints, the development of the API gateway has been sped up two-fold.

b. Frontend

Before starting the development, mockups were created with *Figma*. Using a tool like *Figma* helped us envision how I wanted the final web application to look while enhancing the user interface and user experience. Simple monochrome colors were used while making the mockups to minimize decision fatigue. Preparing mockups before writing front-end code was crucial for speedy development without having to think if "this looks good." Otherwise, more time would have been spent just experimenting with designs on-the-fly.

React using *Typescript* was the main library used for front-end development. *Typescript* made the web application more robust with static typing and seamless *VSCode* support. Other libraries that supported were: *react-router* for page routing, *recharts* for visualizing data, *immer* for immutable state management, *rambda* for a minimal functional programming library, and *dayjs* for managing date and time formats.

For bundling the whole front-end web application, *Vite* was used. *Vite* was selected because it has a plugin that enables PWA in our web application. With just a few tweaks on the configuration file and service workers, all features of a PWA were enabled.

c. Operations

Using Hyperledger Fabric as the blockchain framework was operations-heavy. I had to focus more on the network architecture than on implementing the chaincode. For this project, the least number of nodes required for achieving the minimum viable product were used.

Hyperledger Fabric had available *Docker* images, which I could have used. Unfortunately, these images weren't updated to support the *Apple M2* chipset at the time of writing. Hence, the binaries were built from source code to proceed with the development.

In the Hyperledger Fabric sample codebase, a test network using *Rancher* and *k3s* was available. I utilized the same samples, but made modifications inside the shell scripts to support the *Apple M2* chipset.

Hosting the web application was not feasible as all nodes in the blockchain network were *Kubernetes pods* and would have been expensive to host and

maintain. I opted to use *ngrok* instead to make a private tunnel, where it was accessible by anyone with the link.

All source code written was pushed in a Github repository, publicly-accessible by anyone.

D. System Design

a. System Features

As a Progressive Web Application (PWA), the interface between patients and the system had the following features enabled mainly on the Collector route:

1. *Installable* – on any device with the help of web manifests
2. *Reliable* – service workers provided functional user experiences even under unstable network conditions
3. *Capable* – web applications should be able to do anything native applications can

b. Database Design

Below is the entity-relationship diagram for the data inside the blockchain network. Every dispatch can have one history and many medications.

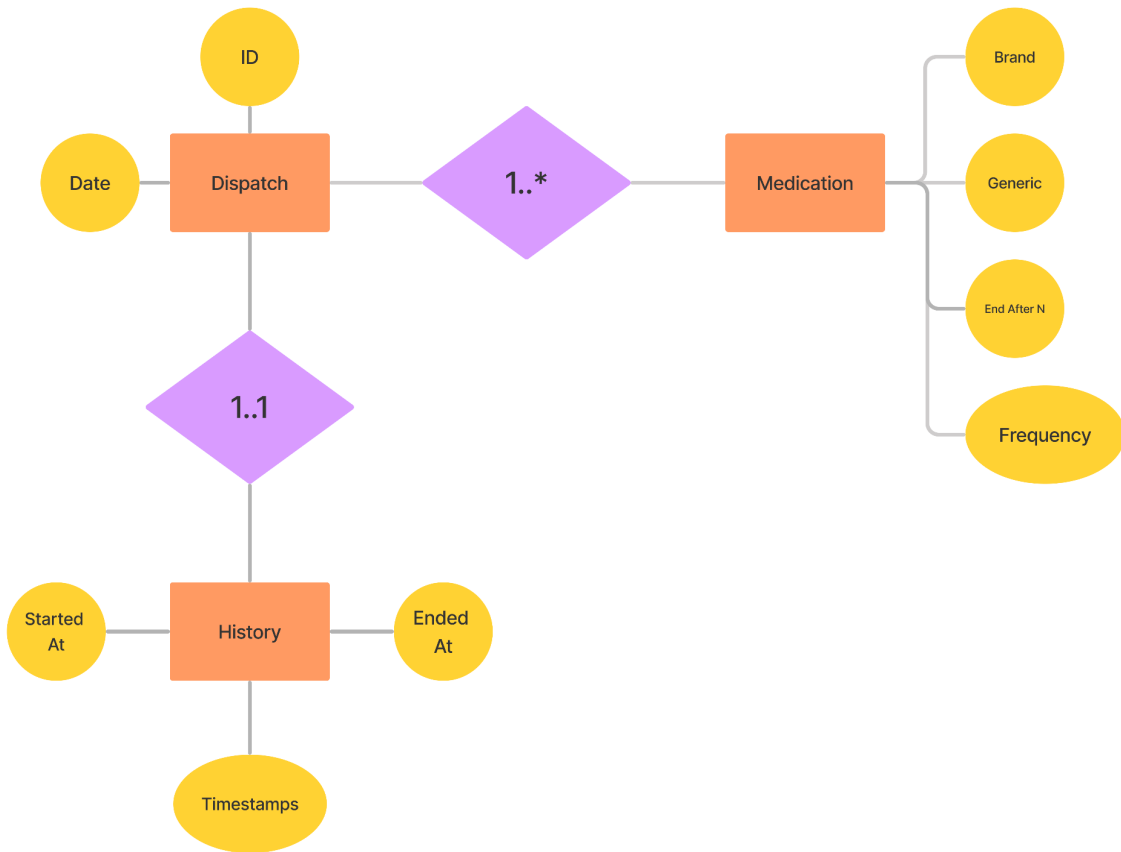


Fig. 4. Entity-Relationship Diagram

E. Implementation

The project was implemented from early August 2022 to mid-December 2022. Below is the Gantt chart representing the implementation timeline.

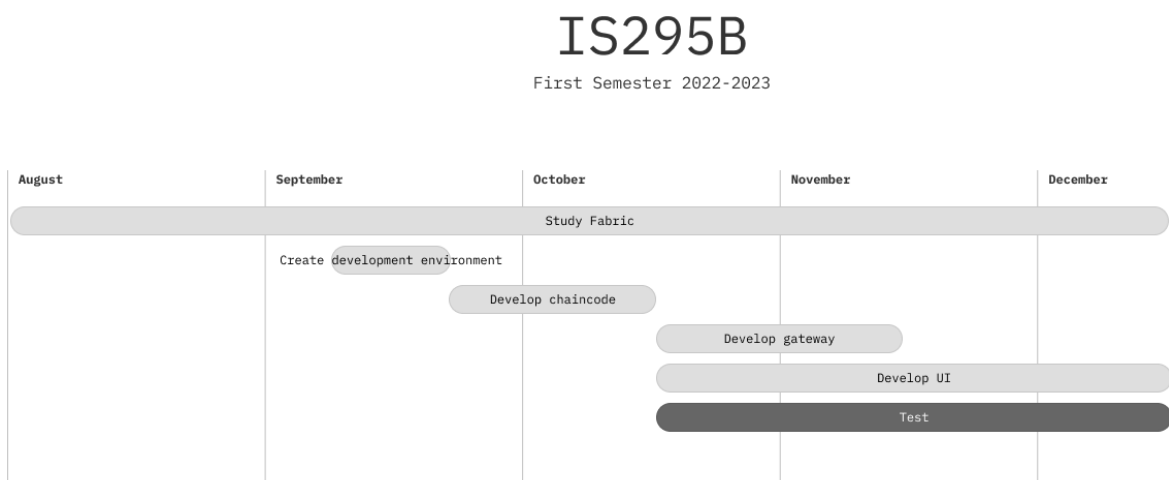
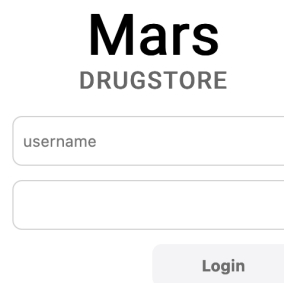


Fig. 5. Project Implementation Timeline

F. Resulting System

In order to access the Dispatcher route, the user must pass through the faux login mechanism. Credentials can be found inside the source code.



The image shows a login page for 'Mars DRUGSTORE'. At the top, the word 'Mars' is written in a large, bold, black font, with 'DRUGSTORE' in a smaller, all-caps, black font directly below it. Below the text are two input fields: the first is labeled 'username' and the second is empty. Below the input fields is a light gray button with the text 'Login' in a small, black font.

Fig. 6. Pharmacy Login Page

Once the user has successfully logged in, a dashboard shall be displayed where the recent dispatch feed is queried every 2 seconds. In the button display area, “Dispatch” is the only functional button. “Inspect” and “History” are currently not implemented as there was no plan to implement those pages during the mockup design phase.

Mark C.
RPh

Jan 13, 2023 19:11:25 Fri

LOGOUT

Mars

DRUGSTORE

DISPATCH

**INSPECT
(NON-FUNCTIONAL)**

**HISTORY
(NON-FUNCTIONAL)**

Dispatch Feed

2022-12-22 12:40:06	amoxicillin	53bb80f0-81aa-11ed-aaa9-de5572c4a535
2022-12-22 11:34:37	clindamycin	2dad1242-81a1-11ed-aaa9-de5572c4a535
2022-12-22 11:14:10	ciprofloxacin	5240be5e-819e-11ed-aaa9-de5572c4a535
2022-12-20 01:04:50	amoxicillin	dde8fbe0-7fb6-11ed-b103-de5572c4a535
2022-12-17 20:24:39	metronidazole	6538534a-7dfd-11ed-ab15-de5572c4a535
2022-12-17 20:23:44	azithromycin	448c4232-7dfd-11ed-ab15-de5572c4a535

Fig. 7. Pharmacy Dashboard (Dispatcher Route)

Clicking a dispatch ID link will redirect the user to the dispatch summary page, where the QR code can be printed by the user. Clicking the Logout button will end the session and redirect back to the login page.

Mark C.
RPh

Jan 13, 2023 19:11:41 Fri

LOGOUT

Dispatch

Generic

Brand

Frequency in hours

Count

Dispatch

Fig. 8. Dispatch Request Form

Clicking on the “Dispatch” button from the dashboard will redirect the user to the dispatch request form, medication information such as generic name, brand name, frequency, and total count will be asked to be input. All fields in this form are required, and input validation is enabled.

Once the user clicks on the “Dispatch” button inside the form, a new dispatch will be created.

Mark C.
RPh

Jan 13, 2023 19:12:08 Fri


LOGOUT

Mars
DRUGSTORE

Dispatch Information

Mars Drugstore

Please scan QR code below and log your medication intake.



53bb80f0-81aa-11ed-aaa9-de5572c4a535
2022-12-22 12:40:06

PRINT

Fig. 9. Dispatch Summary

The patient may scan the generated QR code to be able to send medication therapy reports through the Collector route. Each dispatch is equivalent to one medication therapy session.

19:17



Hi there!



It's been

a few seconds

since your last input.



clindamycin

cleocin

OK

Fig. 10. Collector Route, installed as an iOS application

Once the user selects the medication and clicks the “OK” button, the Collector route will display the time relative to the latest intake. If the next estimated medication intake time has been missed, a message reminding the user to input will be displayed.

The data will be saved in the device’s local storage and will attempt a sync every 5 seconds. This behavior makes the app function even without internet connectivity.

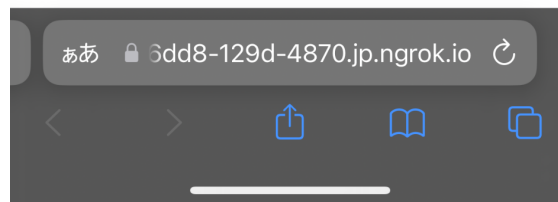
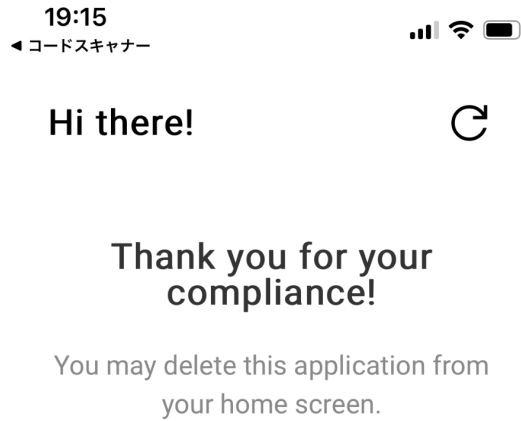


Fig. 11. Collector Route, accessed from iOS Safari browser

If a dispatch is accessed and the medication therapy has completed, the Collector route will display a message that the app can now be deleted from the home screen. If local data has not been completely synced, a warning message will be displayed prompting the user to connect to the internet before deleting the app.

The rest of the use cases from the Figma mockups can be found in the repository's test snapshots for the Collector route.

RxWatch

Generic

Start

End

Fig. 12. Watcher Route query form

For viewing medication compliance rates, the Watcher route is publicly accessible. This form has input validation enabled and it allows users to query medications within specific date ranges.

RxWatch

Compliance rates for medicines: amoxicillin, metronidazole from 2022-12-01 to 2022-12-31

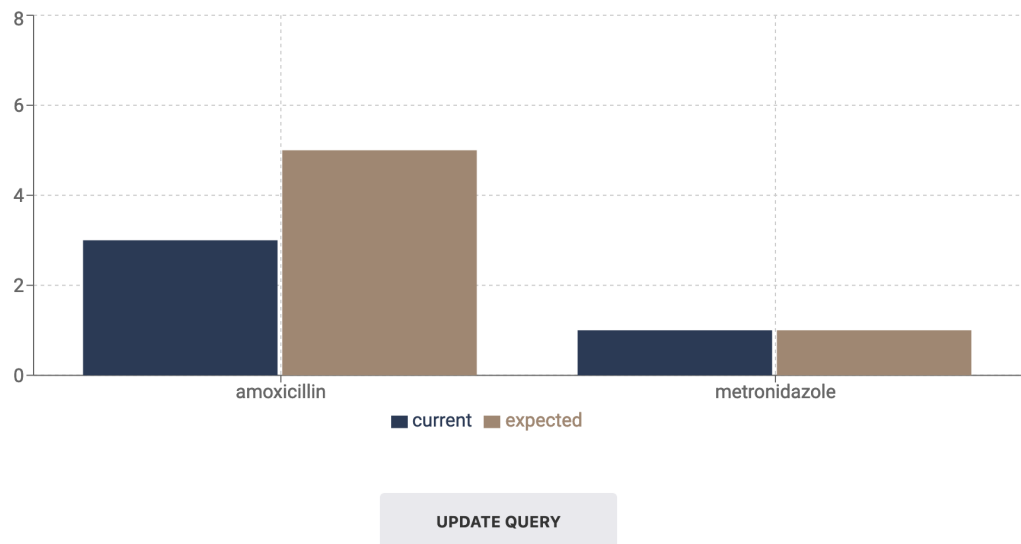


Fig. 13. Watcher Route query results, visualized

Medication compliance rates are visualized as a basic bar chart. “Current” refers to completed medication therapy dispatches, and “Expected” refers to the medication therapy dispatches that are expected to be completed within the queried date range.

Given the screenshot above, the compliance rate for *amoxicillin* from 2022-12-01 to 2022-12-31 is 3 out of 5 or 60%.

Chapter IV

PROJECT ASSESSMENT

A. Usability Testing

There were plans to do a simple survey for usability testing and improving the user interface, but have failed to do so. Therefore, the effectiveness of the system cannot be guaranteed.

B. Unit Testing

Unit tests for the chain code were written using the built-in Golang testing library. Each function exposed to the API gateway was tested to check success and error cases. Tests were isolated using mocks and stubs for Hyperledger Fabric functions. The test specifications for the chain code functions are inside the Github repository. All test cases passed, as shown below.

```
mark ~/Documents/workspace/academe/fabric-medtrack/cc/chaincode [ut] $ go test -v
=== RUN   TestInitLedger
--- PASS: TestInitLedger (0.00s)
=== RUN   TestCreateAsset
--- PASS: TestCreateAsset (0.00s)
=== RUN   TestReadAsset
--- PASS: TestReadAsset (0.00s)
=== RUN   TestAddHistory
--- PASS: TestAddHistory (0.00s)
=== RUN   TestGetAllAssets
--- PASS: TestGetAllAssets (0.00s)
=== RUN   TestGetNAssets
--- PASS: TestGetNAssets (0.00s)
PASS
ok      cc/chaincode  0.777s
```

Fig. 14. Chaincode Unit Test Results

Unit tests for the front-end application were written, together with coverage tests. Each component's behavior was tested, including all views with their corresponding states. All test specifications are inside the Github repository and coverage tests included: statement, branch, function, and line coverage. Results were acceptable.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	89.96	71.42	83.58	91.5	
assets	100	100	100	100	
refresh.svg	100	100	100	100	
components/ListItem	100	100	100	100	
ListItem.tsx	100	100	100	100	
components/Logo	100	100	100	100	
Logo.tsx	100	100	100	100	
outlets/Dashboard	100	100	100	100	
Dashboard.tsx	100	100	100	100	
outlets/Dispatch	100	50	100	100	
Dispatch.tsx	100	50	100	100	56-103
outlets/Inspect	100	83.33	100	100	
Inspect.tsx	100	83.33	100	100	25
outlets/Query	100	50	100	100	
Query.tsx	100	50	100	100	27-62
outlets/Visualize	100	100	100	100	
Visualize.tsx	100	100	100	100	
store	50	0	50	54.54	
collector.ts	50	0	50	54.54	17-33
views/Collector	87.05	75.47	77.27	89.02	
Collector.tsx	87.05	75.47	77.27	89.02	65, 96, 127-134, 142, 149, 226, 278, 290
views/Dispatcher	94.11	50	83.33	94.11	
Dispatcher.tsx	94.11	50	83.33	94.11	15
views/Login	90.32	77.77	80	93.1	
Login.tsx	90.32	77.77	80	93.1	58-59
views/Watcher	100	100	100	100	
Watcher.tsx	100	100	100	100	

Test Suites: 16 passed, 16 total
 Tests: 22 passed, 22 total
 Snapshots: 19 passed, 19 total
 Time: 5.944 s
 Ran all test suites related to changed files.

Watch Usage: Press w to show more.

Fig. 15. User Interface Unit Test Results and Coverage

C. Integration Testing and Bug Fixing

Implementation of the API gateway, UI, and tests were done in parallel. There were bugs on the way, but were fixed during implementation. End-to-end tests were initially planned as part of integration testing but had abandoned the idea due to a lack of time. Instead, I have compromised by performing dogfooding.

D. Google Lighthouse Audits

Google Lighthouse audits yielded favorable results in all routes due to enabling PWA features in our web application. The Collector route had the heaviest front-end logic; performance results were behind the rest of the views, and further code optimization was needed to improve the results.

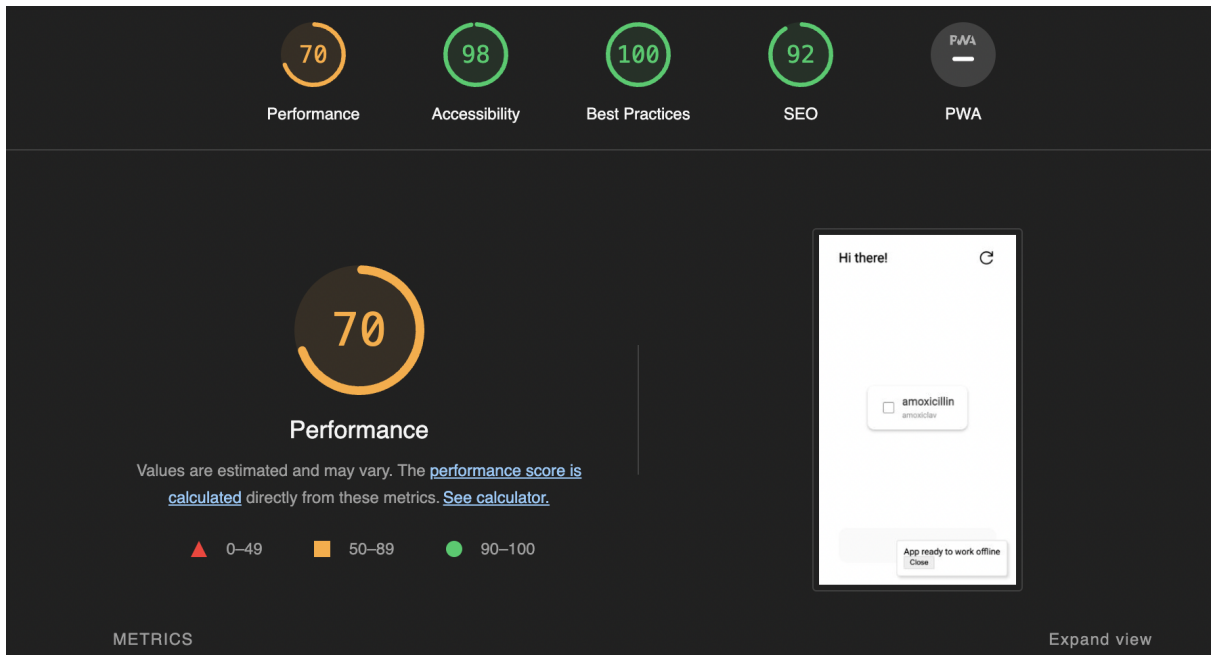


Fig. 16. Google Lighthouse Audit Results: Collector Route

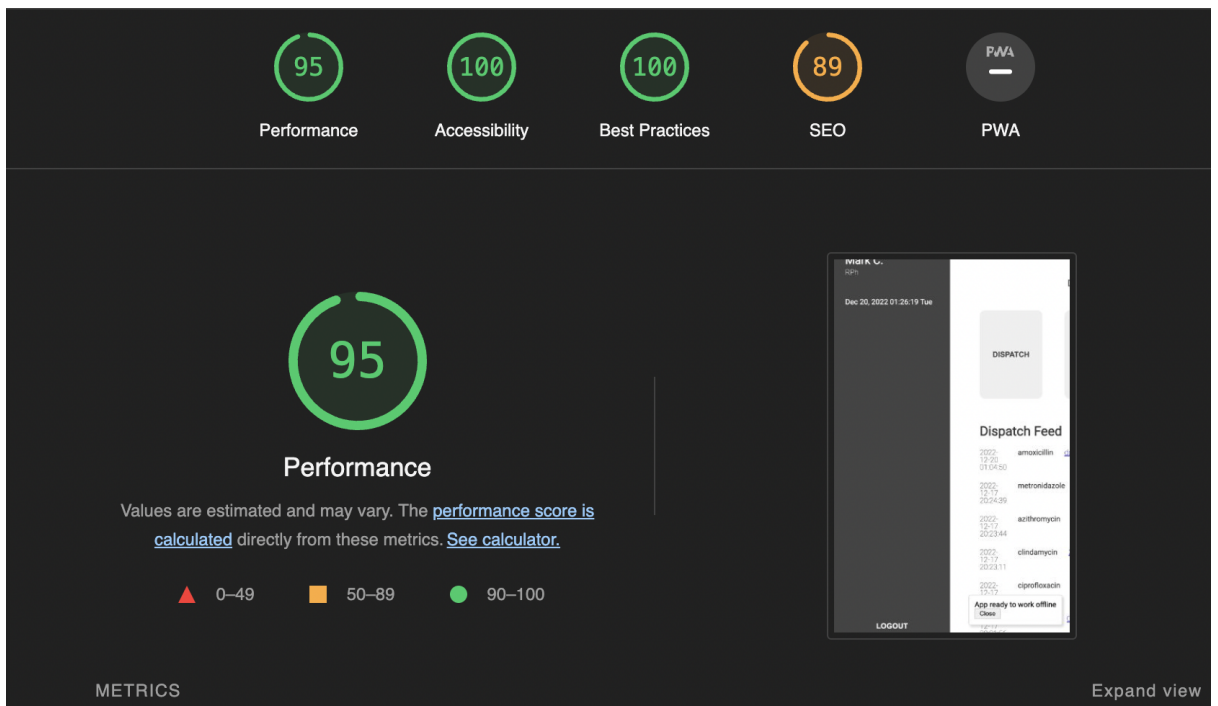


Fig. 17. Google Lighthouse Audit Results: Dispatcher Route

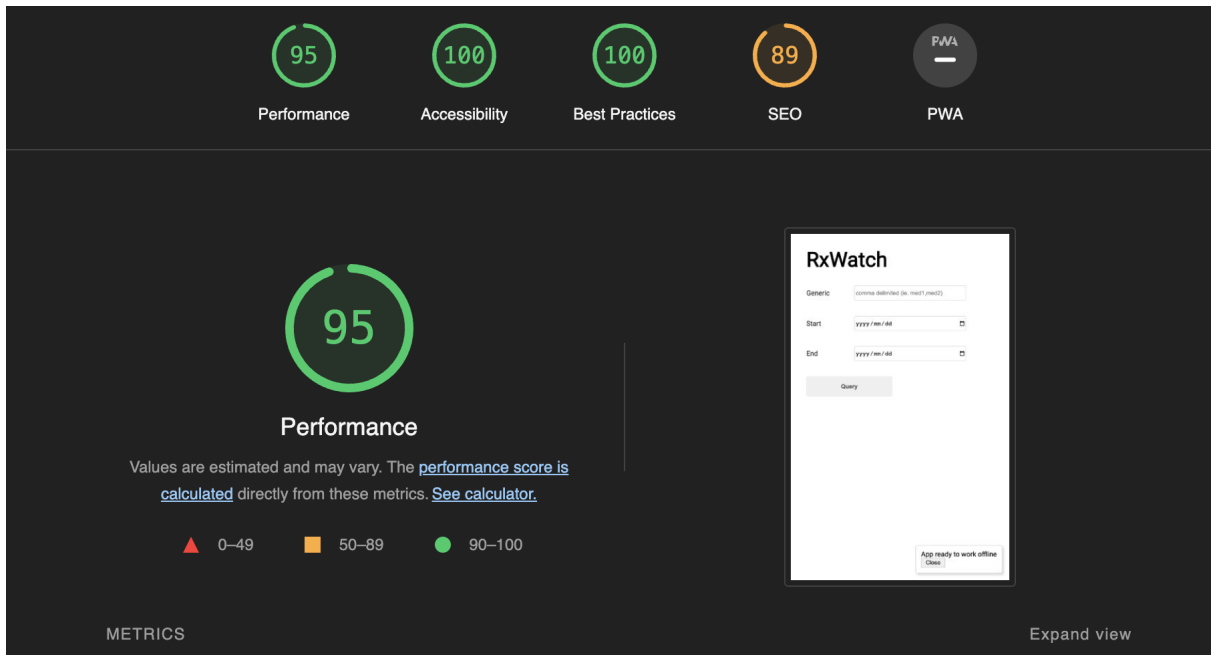


Fig. 18. Google Lighthouse Audit Results: Inspector Route

Chapter V

DISCUSSIONS

Blockchain technologies are mostly known for digital currencies, but the concept of blockchain can pave new ways of achieving a secure, reliable, non-fraudulent, and transparent system. Without exposing user information, this project was able to track medications of public interest, such as antibiotics, making all transactions private and transparent.

Doing this project alone using Hyperledger Fabric was difficult. Before diving into blockchain technologies and Hyperledger Fabric, one must know how to lay out secure network nodes, design infrastructures, create database schemas, and know what problems blockchain technologies try to solve. I initially tried to make a complete production setup per the documentation. However, it was not feasible for the project's timeline and budget, let alone host all nodes in the cloud. Therefore, I compensated by making Kubernetes pods instead. Ultimately, a proof-of-concept was created, demonstrating a system that leverages all features of blockchain technologies.

Hyperledger Fabric has more functionalities to offer that this project did not utilize. Private data and multiple chains are some, to name a few. However, if blockchain technologies can be utilized to create a digitized government, aiming to become a high-trust society is attainable. Blockchain-enabled digital elections are the first to come to mind.

As of this writing, a few interested individuals are already looking into our repository because the codebase is public. After this project, I shall maintain the source codes as a side hobby.

Chapter VI

CONCLUSION

This project solved the problem by making a digitized system that pushes patients to complete medication therapy using their own devices. However, the complexity increased by introducing technology with a steep learning curve and overhead, which led to a long implementation phase due to the need to study the framework used. Running a simple environment with all the necessary nodes took time.

I could have done the same with traditional relational database systems, but transparency, reliability, availability, and security were the main non-functional requirements of this project. Doing so with relational database systems would eventually lead into making a system similar to blockchain technologies or making compromises in the project's requirements. Regardless, learning blockchain technologies was worth it but might have been too much for this project.

Hyperledger Fabric is very heavy in terms of system administration and network operations. It is not advisable to utilize this framework singlehandedly unless experienced in recent DevOps processes. Even the documentation for Hyperledger Fabric itself has target audiences for every concept.

Chapter VII

FUTURE WORK

There were challenges encountered while implementing this project. Compromises were made due to time and resource constraints. In this section, I shall enumerate ideas and why these can be considered for future work.

Within the blockchain network, investigating concepts on private data and multiple chains are advised. These concepts are well-documented on the Hyperledger Fabric website. If it is feasible to implement private data in this project, then nodes with specific authorizations may have read access to patient information. On the topic of multiple chains: the project only utilized a single chain for dispatching and reading medications. Because a faux login mechanism was temporarily used, having another chain to handle logins and pharmacy credentials will secure the pharmacist-facing web application against attacks.

The web application was not tested for usability. Gradually improving user experience is essential for web development to keep users engaged, and it is achievable by gathering user input and updating the user interface.

For operations, making this project production-ready is recommended. Only local Kubernetes pods were utilized. These pods were hosted on a single machine where the user interface was tunneled using ngrok . Carefully allocating resources is crucial if on a tight budget. Also, all user interfaces can be hosted separately, having one peer node for each user interface.

Finally, forming a team to make this project robust is not necessary but highly advised. Anything that involves Hyperledger Fabric or blockchain technologies in general should not be underestimated unless experienced. Just because something is trendy does not mean we should jump into the bandwagon. If a project can be

implemented using simple technologies, then there is no need to complicate it any further.

REFERENCES

- [1] N. Saito, "Frequent Use of Antibiotics among a Low-Economic Status Population in Manila, the Philippines: A Prospective Assessment Using a Urine Antibiotic Bioassay," *The American Journal of Tropical Medicine and Hygiene*, vol. 98, issue 5, pp. 1512-1519, March 2018. [Online serial]. Available: <https://www.ajtmh.org/view/journals/tpmd/98/5/article-p1512.xml>
- [2] B. Cahilas-Magkilat, "Pinoys more interested in Ivermectin than vaccine registrations – research," *Manila Bulletin*, April 11, 2021. <https://mb.com.ph/2021/04/11/pinoys-more-interested-in-ivermectin-than-vaccine-registrations-research/>
- [3] M. T. Montemayor, "Antibiotics abuse: Saving money at the expense of one's health," *The Philippines News Agency (PNA)*. October 3, 2019. [Online]. Available: <https://www.pna.gov.ph/articles/1082194>
- [4] S. Crisostomo, "FDA: Report drugstores selling antibiotics without prescription," *The Philippine Star*, February 7, 2013. [Online]. Available: <https://www.philstar.com/metro/2013/02/07/905742/fda-report-drugstores-selling-antibiotics-without-prescription>
- [5] L. Esmael, "FDA e-logbook app expected to modernize public health," *The Manila Times*, August 29, 2017. [Online]. Available: <https://www.manilatimes.net/2017/08/29/business/fda-e-logbook-app-expected-modernize-public-health/347406>
- [6] E. Quismorio, "Pro-science consumer group president asks crucial question to Ivermectin backers," *Manila Bulletin*, April 25, 2021. [Online]. Available: <https://mb.com.ph/2021/04/25/title-92/>
- [7] Keyhole Software, "Tracking Lab Results Better With Blockchain Technology," April 2020. [Online]. Available: <https://keyholesoftware.com/wp-content/uploads/Tracking-Lab-Results-Better-with-Blockchain-Keyhole-Software-White-Paper.pdf>
- [8] P. Baran, "On Distributed Communications," *RAND Corporation*, Memorandum RM-3420-PR, August 1964. [Online]. Available: http://www.rand.org/content/dam/rand/pubs/research_memoranda/2006/RM3420.pdf. [Accessed May 29, 2017].
- [9] M. Fowler, "Richardson Maturity Model," March 18, 2010. [Online]. Available: <https://martinfowler.com/articles/richardsonMaturityModel.html>
- [10] Barber, D.A. *et al.* (2017) "Prevalence and correlates of antibiotic sharing in the Philippines: Antibiotic misconceptions and community-level access to non-medical sources of antibiotics," *Tropical Medicine & International Health*, 22(5), pp. 567–575. Available at: <https://doi.org/10.1111/tmi.12854>.

APPENDICES

The project's source code is available via the link below. Each subdirectory has "readme" files documenting steps to run the applications, including troubleshooting guidelines. The whole codebase is tailored to run on machines using an Apple M2 chipset, so there might be a possibility that this codebase might not run outside the processor's architecture.

<https://github.com/paradoc/fabric-medtrack>

The Figma mockups are publicly available in:

<https://www.figma.com/file/05shuQCk9Z9AMOAM4fdQeG/IS295?t=6WP2CKIBXohAbEe8-1>