

The University of Hong Kong



Final Year Project Interim Report

Blockchain-based Decentralized Platform for COVID-19 Test

Members:	Foo Zhi Qian	3035446100
	Yeo Zhi Wen	3035444516

Supervisor: Dr. S. M. Yiu

19th April 2021

Abstract

The Covid-19 pandemic is spreading across the world, causing panic in the public and harming countries' economy. As there is currently no cures and vaccines to this disease, the only way to slow down the global pandemic is to conduct mass testing to identify and quarantine the potential carriers of the Covid-19 virus. Nevertheless, there have been many concerns raised regarding the authenticity and privacy of the mass testing data that are stored in government databases. This is where blockchain technology can be leveraged to help solve the problems.

This project aims to develop a decentralized, blockchain application for Covid-19 testing. Each registered user will only be able to access their test results on the application. If a user is tested positive, a smart contract will be executed automatically to deliver the records of the user to the health department for quarantine and treatment purposes. By building a powerful solution that revolves around blockchain technology, the security, interoperability, and immutability of the mass testing data can be guaranteed. No centralized agency or third party is involved in the process so users' privacy can also be protected.

Currently, the work progress is satisfactory because we have completed the product as planned in the timeline. Vigorous testing has been conducted on our web platform and believe that it is ready to function error-free and serve the public.

Acknowledgement

First and foremost, I would like to extend my utmost gratitude to my supervisor, Dr. S.M. Yiu, for his invaluable advice throughout the project. In all humbleness and gratefulness, I am also grateful to have my groupmate, Yeo Zhi Wen, that works diligently on this meaningful project together with me. Last but not least, I would like to thank everyone that has helped me out throughout my final year project.

Table of Contents

Abstract	i
Acknowledgement	ii
List of Tables	iv
List of Figures	v
1. Introduction	1
1.1 Background	1
1.2 Motivations	1
1.3 Objectives	2
1.4 Project Requirements	3
1.4 Report Outline	4
2. Project Background	4
2.1 Ethereum and IPFS	4
2.2 Workflow of Real-Life Implementation	6
2.3 Initial User Interface Prototypes	8
3. Methodology	19
3.1 Walkthrough of User Interfaces	19
3.2 Core Functionalities of the Blockchain Web Application	29
3.3 Ethereum as the Backend Framework	29
3.4 Blockchain as the Core Technology	31
4. Testing and Limitations	33
4.1 Local Blockchain Deployment	33
4.2 Unit Functionality Testing	37
4.3 Attempt to Bypass Authentication Process	37
4.4 Limitations	37
5. Conclusion	38
References	39

List of Tables

Table 1.	The summary of main differences between blockchain platform.	8
Table 2.	The tentative timeline of the Final year Project.	9

List of Figures

Figure 1.	A QR code obtained from the user information page.	6
Figure 2.	The kiosks (Left) are located in public occasion and have SpectralLIT device (Right) in them to conduct Covid-19 speed testing on the users' saliva samples. Saliva samples are collected in the SpectraLIT device.	7
Figure 3.	Machine Learning models to classify the users' saliva samples	7
Figure 4.	View Result page on our web application	8
Figure 5.	The user login page.	9
Figure 6.	The user registration page	9
Figure 7.	The new user registration page for people who wish to sign up for Covid-19 testing. Several fields were added in the later stage. For example, public key fields to be uploaded onto the blockchain without encryption so that any user that would want to send a message can use the public key to encrypt the message before sending it. Name field is needed to be displayed on the testing result as well.	10
Figure 8.	The new supplier registration page for institutions who wish to sign up as Covid-19 testing providers. Similar to previous case, public key field is added for the encryption of message to be sent, username field to facilitate the login process, email field for notification of application approval and recovery of password.	10
Figure 9.	The landing page for a normal user. The heatmap is removed in the final product as there was no API available to retrieve real time data. Instead, a 5-day graph of number of cases in Hong Kong is added to provide information to the users.	11

Figure 10.	The user information page where the user profiles and user QR codes will be displayed.	11
Figure 11.	The view result page where the results are censored until the users have completed their payments.	12
Figure 12.	The payment page where the users will make payment in order to view their results.	12
Figure 13.	The history page where the users can check their previous testing results.	13
Figure 14.	The landing page for a Covid-19 testing supplier.	13
Figure 15.	The history page of a Covid-19 testing supplier, where the patients' results are being censored.	14
Figure 16.	The new landing page of a normal user where making appointment with health institutions to test for Covid-19 is now possible.	15
Figure 17.	The user appointment page.	14
Figure 18.	The new user history page where previous appointments can be tracked back.	16
Figure 19.	The result page where the patients who have undergone traditional Covid-19 testing can check their results directly without having to pay.	16
Figure 20.	The new landing page of a supplier where patients' appointments can be viewed.	17
Figure 21.	The incoming appointment page of a supplier.	17
Figure 22.	The appointment information page where the details of appointments are depicted.	18
Figure 23.	The input result page for a supplier who uses traditional Covid-19 testing method to enter the results of their patients.	18

Figure 24.	The user will get to choose to either register as a user or a test supplier.	19
Figure 25.	To become a new user, one has to fill in important information including Ethereum address and Public Key that they got beforehand.	20
Figure 26.	The user will get to choose to either sign in as a user or a test Supplier.	20
Figure 27.	The user will need to provide username, password, and private key for login.	21
Figure 28.	The user will be redirected to the user landing page upon login.	21
Figure 29.	The user will get to make an appointment with the health institutions which have registered in our web application for Covid-19 testing.	22
Figure 30.	The user can refer to the results in History page to retrieve their pending results.	22
Figure 31.	On clicking on any of the pending results in history page, a modal will pop out to prompt the user to pay to view his result.	23
Figure 32.	On clicking the “Pay to View” button, another modal will pop out to display the payment details and users can pay to unveil their pending results.	23
Figure 33.	On clicking the “Appointments” header on the history page, another modal will pop out to display the payment details and users can pay to unveil their pending results.	24
Figure 34.	On clicking the “User Information” on the side navigation bar of user landing page, the user will be redirected to the user information page, and this QR code can be shown to the PIC at kiosk to identify their identity and Ethereum address easily during the Covid-19 testing.	24

Figure 35.	On clicking the “Latest Test Result” button on the user landing page, users will be redirected to the latest user result page, where they will be prompted to pay to view their latest test result.	25
Figure 36.	On clicking the “Pay to View” button, another modal will pop out to display the payment details and users can pay to unveil their latest results.	25
Figure 37.	To become a new supplier, one has to fill in important information including Ethereum address, Healthcare Provider Number that they got beforehand.	26
Figure 38.	To login as a supplier, one has to provide the username, password and private key.	26
Figure 39.	Upon logging in, the supplier will be redirected to the supplier landing page.	27
Figure 40.	On clicking on the “Incoming Appointment” on the sidebar, a supplier can see all the incoming appointments.	27
Figure 41.	Clicking on any of the incoming appointments, the health institution can manually input the test result of the patient if the patient signs up for a testing kit differs from the SpectraLIT device.	28
Figure 42.	On clicking on the “History” on the sidebar of supplier landing page, a supplier can see all the incoming appointments.	28
Figure 43.	The details of a blockchain.	32
Figure 44.	The flow of creating and adding a block into the blockchain	33
Figure 45.	In this directory, run the command “truffle dev” on the command prompt. 10 Ethereum accounts will be created, and we will obtain the Ethereum addresses and Private Keys as follow for our testing purposes	34
Figure 46.	The command prompt of starting truffle environment	34

Figure 47.	As an alternative, we can download and start Ganache to achieve the same purpose while having better visualizations about the contracts deployed, gas spent and transactions that have taken place. Open up the workspace of Ganache and run truffle-config file in it as shown above	35
Figure 48.	Ganache provides detail information about each transaction, event, and log that have taken place, allowing us to debug it quickly	35
Figure 49.	The command prompt of starting our react front-end on the localhost	36
Figure 50.	The Login page where the user will be redirected to once the program starts	36

1. Introduction

In this section, the background of the project will be briefly discussed. The limitation of traditional databases will be detailed, which brings to the idea of developing a decentralized, blockchain-based application to resolve the shortcomings. The strength of the blockchain application will also be covered in Section 1.3 before we talk about the outline of the report.

1.1 Background

Currently, the new coronavirus (Covid-19) has spread to nearly every country in the world since its first emergence in Wuhan, China. At the time of writing, there are nearly 140 million confirmed cases recorded in the world, taking away the life of around 3.01 million citizens [1]. The Covid-19 pandemic has caused global social and economic disruption, including the largest global recession since the Great Depression [2]. As effective cures are still under development, the only way to stop the disease in track is to identify the carriers of the virus, who may or may not exhibit symptoms similar to flu. Therefore, mass testing for COVID-19 that aims to find people with active infection who are asymptomatic or presymptomatic to quarantine, followed by rapid finding and testing of close contacts, are crucial to interrupt the spread of the disease.

Nevertheless, the mass testing results of the public need to be handled carefully without compromising the privacy concern. Some people may be unhappy with their test results being collected and managed by a centralized agency, especially in Covid-19 testing where the information of whether a person has been tested positive is sensitive and personal. Therefore, blockchain technology, which provides a decentralized storage, security, interoperability, and immutability can be leveraged in this scenario. Our final year project will be focusing on building a blockchain-based decentralized platform for Covid-19 test to address the privacy concern issue while providing other benefits like transparency and traceability.

1.2 Motivations

Traditional database storage of the public's Covid-19 testing results has some shortcomings. Firstly, it heavily relies on a single central server, which is vulnerable to malfunctions. When the

central server goes down due to hardware failure, the entire database will be lost. Backing up thousands of millions of public's Covid-19 testing results is not favorable in terms of cost efficiency, as it will incur a lot of money to buy extra servers just to store duplicated results. Therefore, the resiliency of the database storage depends on the robustness of the cloud infrastructure, which is not desired.

Secondly, a centralized server storing the results of different people in each hospital is inefficient. The fragmentation of public health data across different hospitals implies that the government will have slow access to the data that are scattered around, and thus lack of system interoperability [3].

Thirdly, the traditional database model is susceptible to data tampering by authorized personnel. Despite the implementation of access rights control in databases, the sensitive testing results are still vulnerable to tampering by authorized persons that might want to overwrite the results for their personal interest. This is particularly dangerous as people who carry the Covid-19 virus might be falsely discharged due to the tampered, misleading result data, spreading the infectious disease to the public.

Last but not least, privacy concerns of the testing results are highly questionable when stored in a traditional database. Most of the time, people that are tested positive would not want anybody else to know, except for the government authorities to execute quarantine procedures and treatments. However, data breaching is not uncommon in traditional databases because there is no smart contract to execute instructions automatically. Result data has to pass through a lot of parties before it is transmitted to its final destination.

1.3 Objectives

Our Final Year Project strives to provide a blockchain-based decentralized platform for Covid-19 testing to solve the shortcomings of traditional data collecting methods. In a decentralized network of blockchains, each block is connected to all the blocks before and after it. This makes it tamper-resistant because in order to change a data on a block, a hacker would need to change

the block containing that record as well as those linked to it to avoid detection. As blockchains are continually updated and kept in sync across all the nodes in the network, it would require massive amounts of computing power to access every instance (or at least a 51 percent majority) of a certain blockchain and alter them all at the same time, making the cost far outweighs the benefit and rendering it highly impossible to happen [4]. Data are stored in blockchains which are publicly distributed to every node in the network, hence if a node goes down, we can always recover the blockchain from another node. Therefore, blockchain-based decentralized platforms will be able to provide data security, immutability, and tolerate single-point-failure, which resolve the problems of traditional databases in storing Covid-19 testing results.

1.4 Project Requirements

One of the basic requirements of our Final Year Project is to be completely decentralized. The intervention of human behaviors in the web application should be kept minimal. As such, databases provider like Firebase and MySQL should not be used as they can be easily modified, tampered, and hacked by people with malicious intent. Instead, decentralized storage like Ethereum blockchain and InterPlanetary File System (IPFS) are used to stored sensitive information like the results of Covid-19 to better protect the privacy and security of the users. In our final product, only the approval of the application to become a Covid-19 Testing Provider will require admin's manual approval.

Aside from that, our web application should orchestrate all the interactions among all the parties using our service. For instance, our platform should facilitate the booking of Covid-19 testing of a user with a health institution that is registered to provide such services. After undergoing the Covid-19 testing, the test result should be available to the user on the web application. User can pay using Ethereum Wallet to view their latest test results. Past appointments and history of past results are also tracked for users' conveniences. From a health service provider's perspective, ways to keep track of upcoming appointments, users' past appointments, and the manual input of user's results when automatic uploading services are not available due to different Covid-19 test kits used should be taken care of.

Moreover, privacy protection shall be given the highest priority when designing the system architecture of the web application. As privacy concern is one of the greatest motive that drives us to come out with a decentralized apps solution, user data especially the test results should be taken great care of when publishing on the blockchain. Public-key encryption should be equipped to hide sensitive information from every irrelevant party on the blockchain.

1.4 Report Outline

The remainder of the report will be presented as follows:

Section 2 will explain about the project background of this final year project, including the major frameworks used by the project, and how our application can be incorporated into real world situation. Section 3 will present the status of our project, which encompasses the internal timeline of our project, work that we have done previously, ongoing work, and scheduled future work. The difficulties and limitations that we have foreseen will also be discussed in Section 4. Lastly, a conclusion is made in Section 5 of the report.

2. Project Background

In this section, a detailed description of the project background will be outlined. Two important frameworks that enable the functioning of our application without a central authority will be discussed in Section 2.1. Furthermore, visualization of the workflow of real-world implementation of our solution will be detailed, particularly in Section 2.2. In Section 2.3, a throwback at our initial user interface prototype will be shown to better depict the differences between initial design and final system architecture.

2.1 Ethereum and IPFS

The two major groups of target audiences for our web application are normal citizens who undergo Covid-19 testing and health institutions, who provide the service of Covid-19 testing. In order to get started with our web application, users must first register an Ethereum wallet, having the Ethereum address and private key ready before hand. The Ethereum address will be used as a key to receive any incoming data and money, while the private key is used for asymmetric decryption of data on blockchains. Asymmetric cryptography is important to ensure

that data that belongs to one person can only be read by the person himself only. When a party signs up on our web application, the party can choose to sign up as either an individual user, or a healthcare institution. An individual user on our platform can choose to receive Covid-19 testing using our spectroscopy device directly or using the traditional method. While the health institution can choose to offer traditional testing method like Polymerase Chain Reaction (PCR), most of them are expected to provide the spectroscopy testing method using SpectraLIT, as it is fast and accurate. Those who employ a different test kit from the SpectraLIT device can still leverage the security that blockchains have rendered on our application, however, they will have to input the user result manually after the results are out several hour later.

The main frameworks that contribute to the decentralization of our web application are Ethereum and IPFS. Ethereum is a programmable technology that builds on Bitcoin's innovation, powering applications on blockchain that everyone can use and no one can take down [15]. In our application, Ethereum is used to store the keys of user information and testing results. Smart contracts, which are written in Solidity, are deployed to retrieve information from the users and drives the logics of automation across the application. For instance, when a user registers himself on the platform, a function will be called automatically from the smart contract to retrieve the credentials from the users and store it on the blockchain. Likewise, the appointments made with the health institutions to carry out Covid-19 testing are fetched by the smart contract as well. The smart contract basically orchestrates all the interactions among multiple participants (between those suppliers and users) in a way that is completely decentralized. While Ethereum offers a lot of benefits by executing functions and providing decentralized database without a central authority, every transaction that passes through the smart contracts costs some gas (Ether currency). This is to reward the nodes in the decentralized network of Ethereum that help to verify and validate the data added to the blockchain, maintaining the integrity of data from being tampered. Therefore, it is expensive to store a huge chunk of data on the blockchain and carry out frequent transactions to access the data.

In view of that shortcoming, relatively unimportant user data are stored on another decentralized storage called The InterPlanetary File System (IPFS). IPFS is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices [16]. The

keys of each file are cryptographic hash that point towards the content of the file. As it is free to be used, IPFS are used widely to store everything except for the testing results on our platform. One downside of IPFS is that it uses content-addressing to find a file, which means that in order to fetch what we want from the database, we either know the contents itself in advance or we remember the hash of the file, which is almost impossible in either cases. However, we can still make good use of the content-addressing feature of the IPFS, by wrapping it around the login details where user's inputs of username and passwords are hashed and compared against the IPFS nodes to find a perfect match. If the hash is able to find a match, the user has entered a correct username-password pair and we log them in. On the other hand, if the hash is unable to match an entry on the IPFS, the username or password entered by the users must have gotten wrong. In either scenario, accessing the data on IPFS and Ethereum are generally slower as they involve cumbersome and bulky blockchain operations.

Another workaround to overcome the costly operations of Ethereum and make good use of the free-of-charge nature of IPFS is to push data onto IPFS and store the resulting hash on Ethereum. In the current implementation of our web application, every user will have to provide a username, password and private key for logging in. The username and password are used for authentication purpose, whereas the private key is used to derived user's Ethereum address. Hash key for user information such as HKID, balance of Ethereum Wallet, and email address can then be fetched from the blockchains on Ethereum, using Ethereum address as the target destination and private key as the decryption key. This hash key obtained is the cryptographic hash to gain access to the user information data on IPFS.

2.2 Workflow of Real-Life Implementation

The end product of our web application can be applied in real-life situation to help screening out potential Covid-19 carriers at crowded public amenities like airports, sport centers and parks.

The workflow of the real-life implementation is as follow:

1)



Figure 1. A QR code obtained from the user information page.

A QR code containing the Ethereum address and app's username is scanned by the PIC at the kiosk in airports/residential areas.

2)

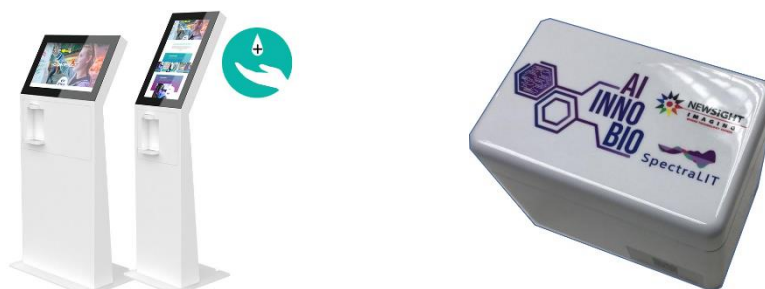


Figure 2. The kiosks (Left) are located in public occasion and have SpectraLIT device (Right) in them to conduct Covid-19 speed testing on the users' saliva samples. Saliva samples are collected in the SpectraLIT device.

3)

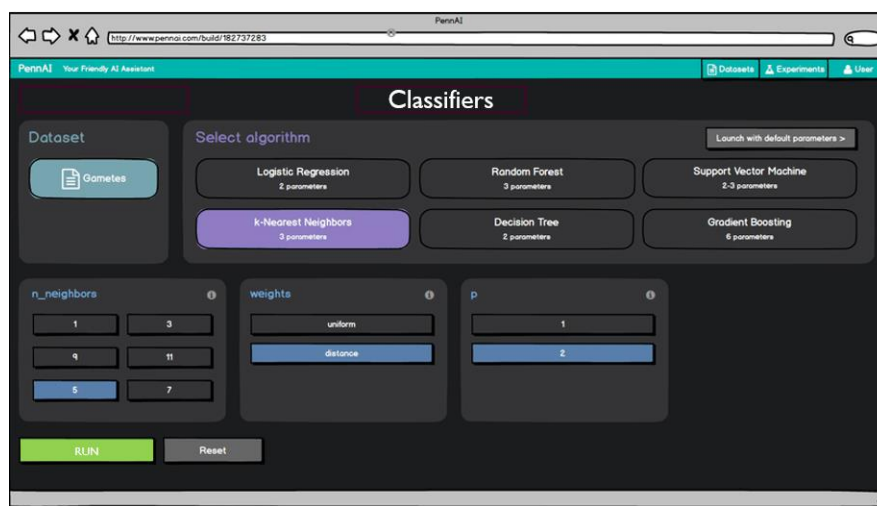


Figure 3. Machine Learning models to classify the users' saliva samples.

The signals obtained from the SpectraLIT devices are fed into machine learning models to classify the sample into either Positive/Negative. The results can be generated within seconds.

4)

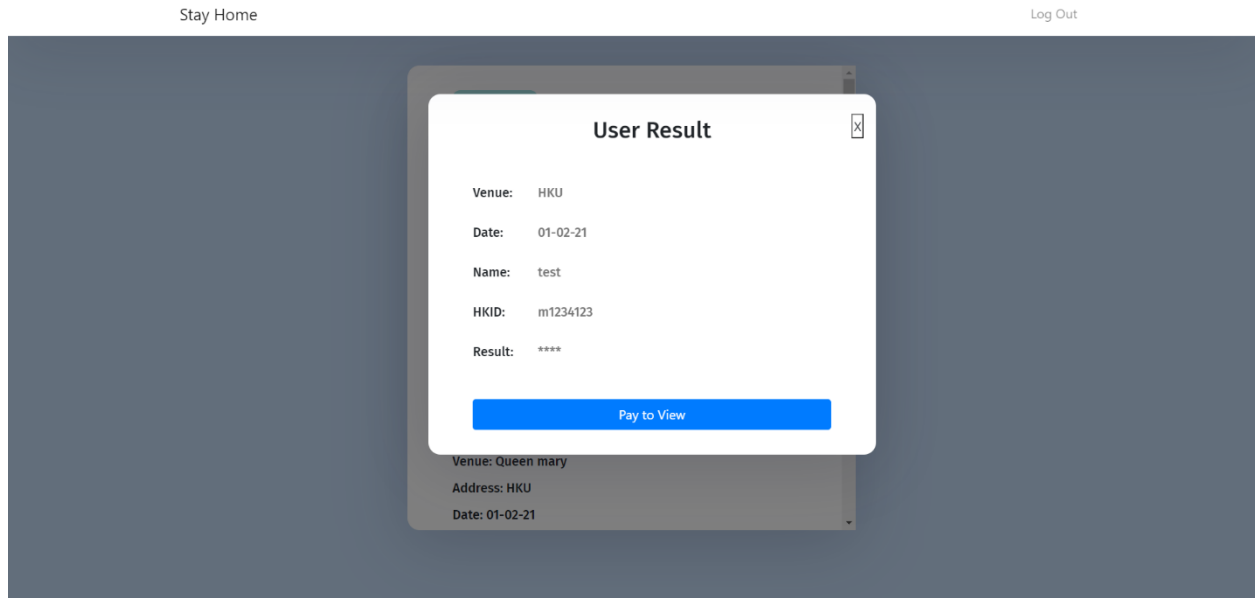
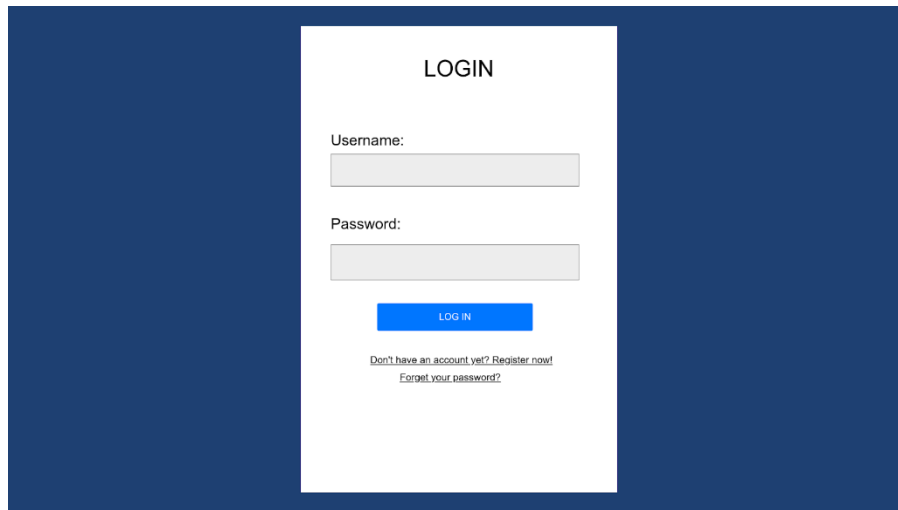


Figure 4. View Result page on our web application.

User can then check their results online as the software will automatically send the results to the user's Ethereum wallet account.

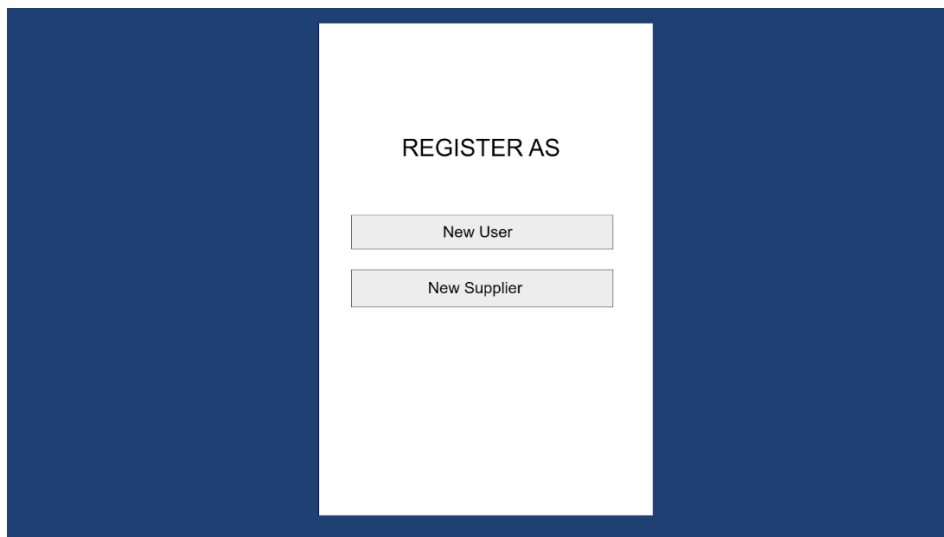
2.3 Initial User Interface Prototypes

The initial user interface prototypes are kept in this report to provide a better glance at the changes that we have made to our original product and walk you through the thought process. The preliminary user interfaces of our web application are as follow:



A user login page with a dark blue background. A white rectangular box is centered on the page. Inside the box, the word "LOGIN" is at the top. Below it are two input fields: "Username:" and "Password:". A blue "LOG IN" button is positioned below the password field. At the bottom of the box, there are two links: "Don't have an account yet? Register now!" and "Forgot your password?".

Figure 5. The user login page.



A user registration page with a dark blue background. A white rectangular box is centered on the page. Inside the box, the text "REGISTER AS" is at the top. Below it are two buttons: "New User" and "New Supplier".

Figure 6. The user registration page

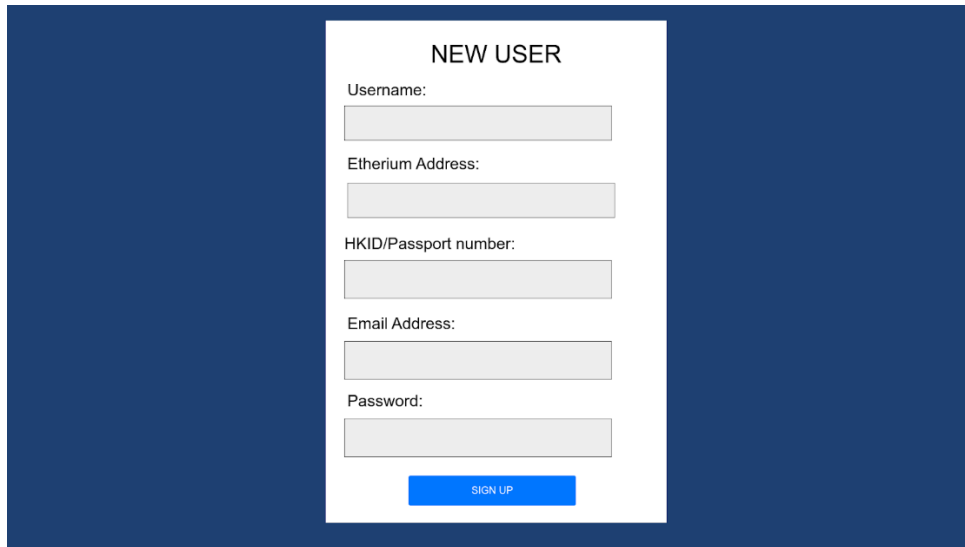
A screenshot of a web form titled "NEW USER" set against a dark blue background. The form is white and contains five input fields: "Username:", "Ethereum Address:", "HKID/Passport number:", "Email Address:", and "Password:". Each field is represented by a light gray rectangular box. Below the fields is a blue button with the text "SIGN UP" in white capital letters.

Figure 7. The new user registration page for people who wish to sign up for Covid-19 testing. Several fields were added in the later stage. For example, public key fields to be uploaded onto the blockchain without encryption so that any user that would want to send a message can use the public key to encrypt the message before sending it. Name field is needed to be displayed on the testing result as well.

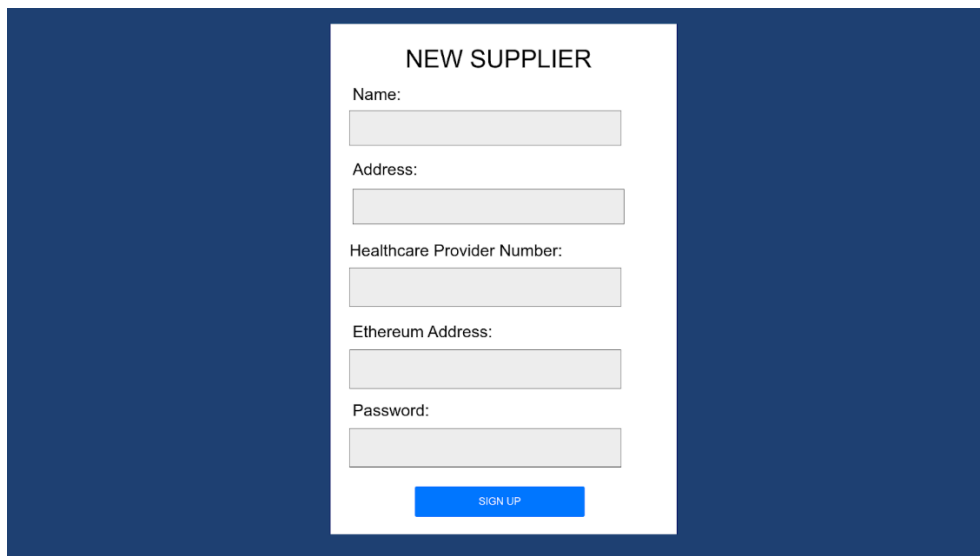
A screenshot of a web form titled "NEW SUPPLIER" set against a dark blue background. The form is white and contains five input fields: "Name:", "Address:", "Healthcare Provider Number:", "Ethereum Address:", and "Password:". Each field is represented by a light gray rectangular box. Below the fields is a blue button with the text "SIGN UP" in white capital letters.

Figure 8. The new supplier registration page for institutions who wish to sign up as Covid-19 testing providers. Similar to previous case, public key field is added for the encryption of message to be sent, username field to facilitate the login process, email field for notification of application approval and recovery of password.



Figure 9. The landing page for a normal user. The heatmap is removed in the final product as there was no API available to retrieve real time data. Instead, a 5-day graph of number of cases in Hong Kong is added to provide information to the users.



Figure 10. The user information page where the user profiles and user QR codes will be displayed.

Result
Venue : HKU
Date : 13/01/2021
Name : Foo Zhi Qian
HKID : 1234567(S)
Result : *****
Pay to view

Figure 11. The view result page where the results are censored until the users have completed their payments.

Payment
Provider name : XXXX
Provider Ethereum address: xxxxx
Provider location: xxxxx
Ether payable: xxxx
Pay

Figure 12. The payment page where the users will make payment in order to view their results.

History	
Date : xxxxx Venue : xxxxx	Result : ****
Date : xxxxx Venue : xxxxx	Result : ****
Date : xxxxx Venue : xxxxx	Result : ****
Date : xxxxx Venue : xxxxx	Result : ****
Date : xxxxx Venue : xxxxx	Result : ****

Figure 13. The history page where the users can check their previous testing results.

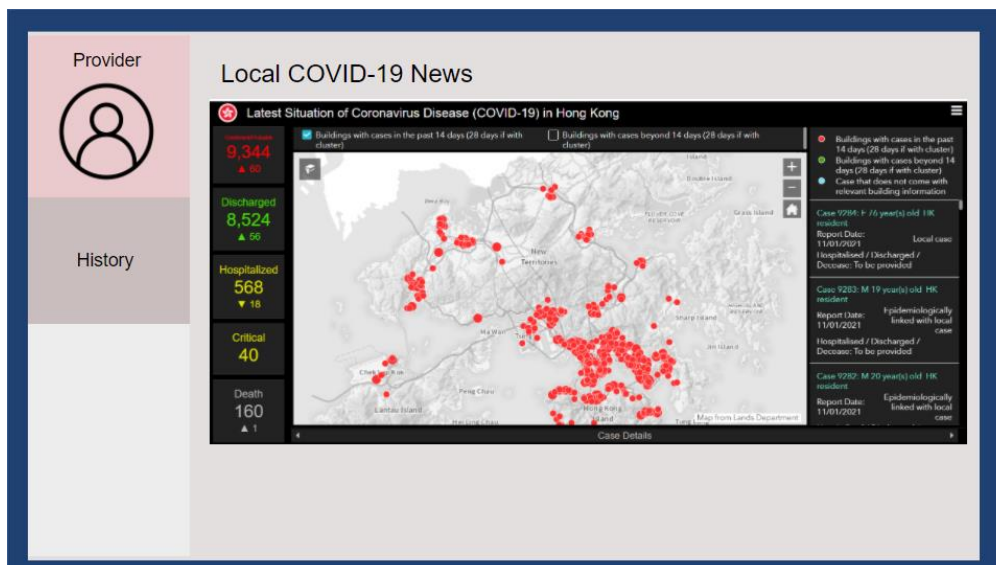


Figure 14. The landing page for a Covid-19 testing supplier. Similar arguments apply to the replacement of heatmap with daily cases graph as before.

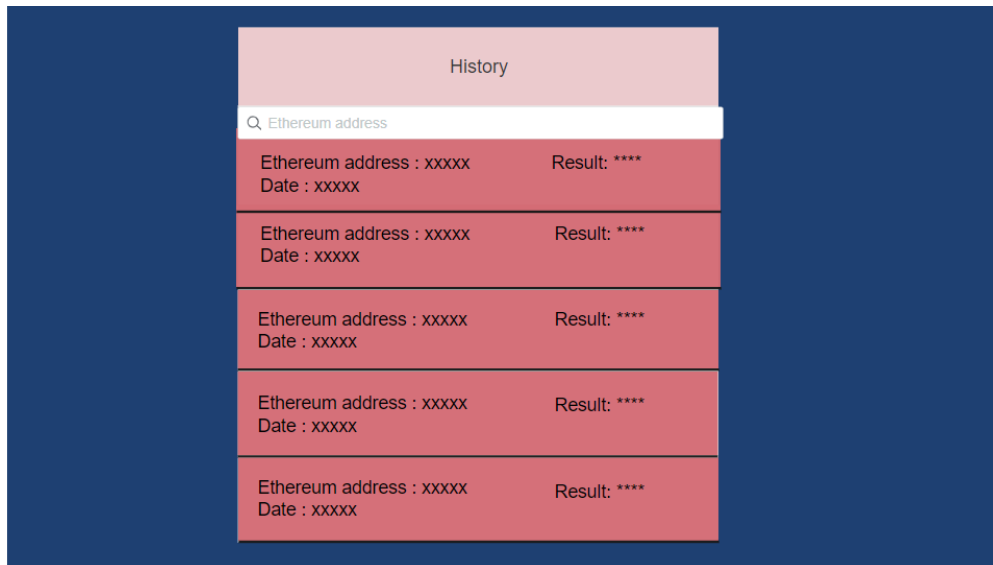


Figure 15. The history page of a Covid-19 testing supplier, where the patients' results are being censored.

The user interfaces depicted in the figures above were designed purposely to cater the institutions that use SpectraLIT devices in the Covid-19 speed testing only. After careful consideration, we had decided not to exclude other institutions who used traditional testing method like Polymerase Chain Reaction (PCR) from leveraging our blockchain web application to store their patients' results. As such, the initial system designs were revised and new functionalities were added to the user interfaces as follow:

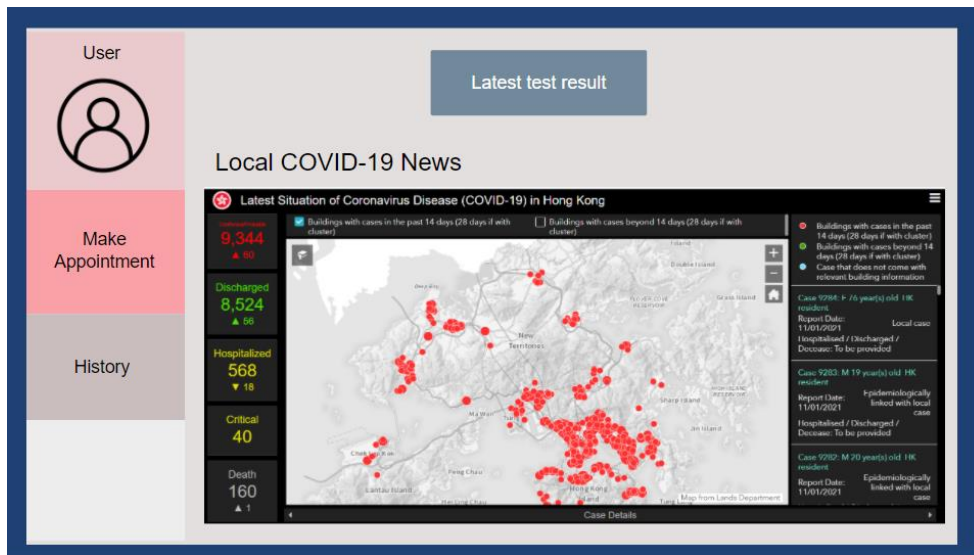


Figure 16. The new landing page of a normal user where making appointment with health institutions to test for Covid-19 is now possible.

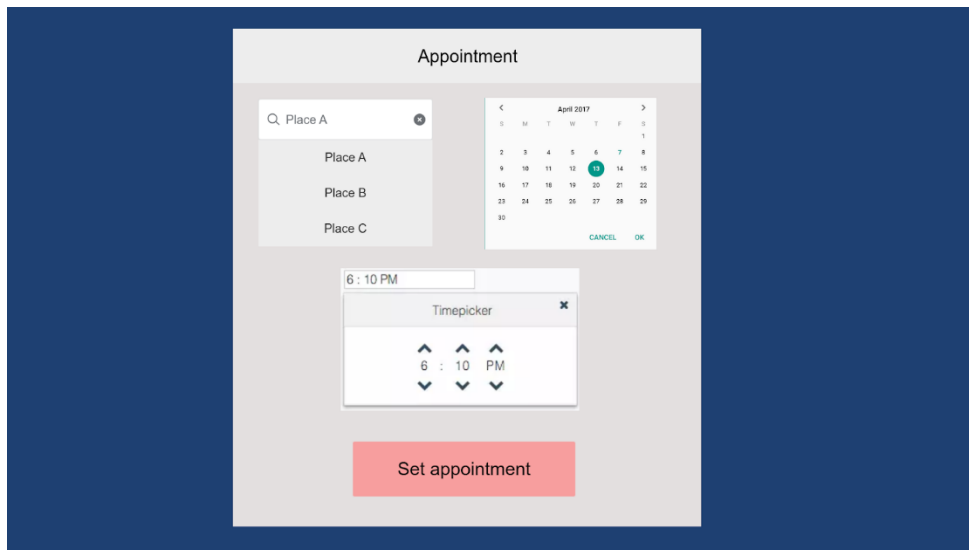


Figure 17. The user appointment page.

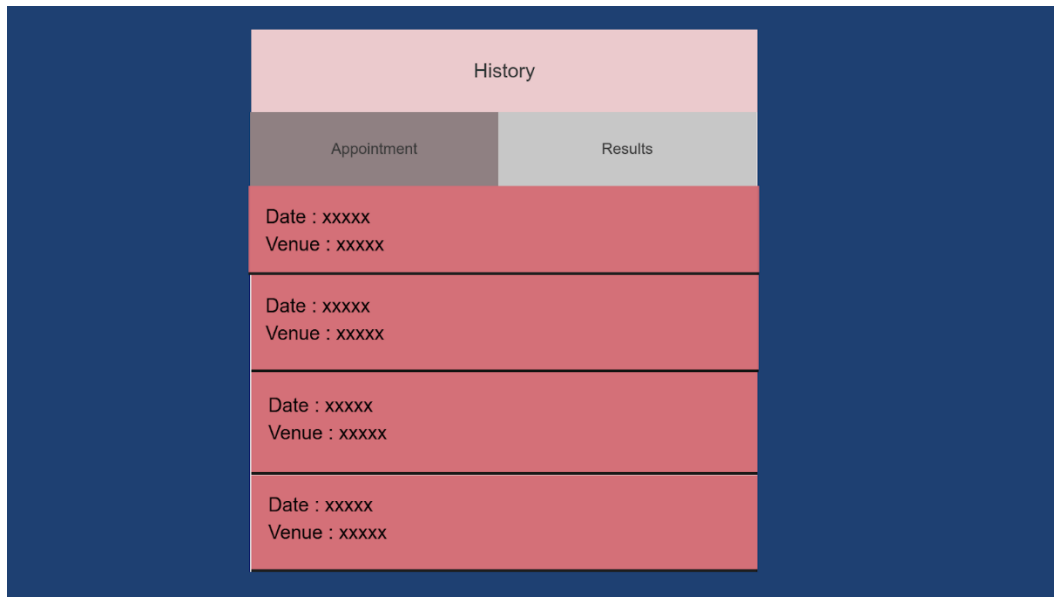


Figure 18. The new user history page where previous appointments can be tracked back.

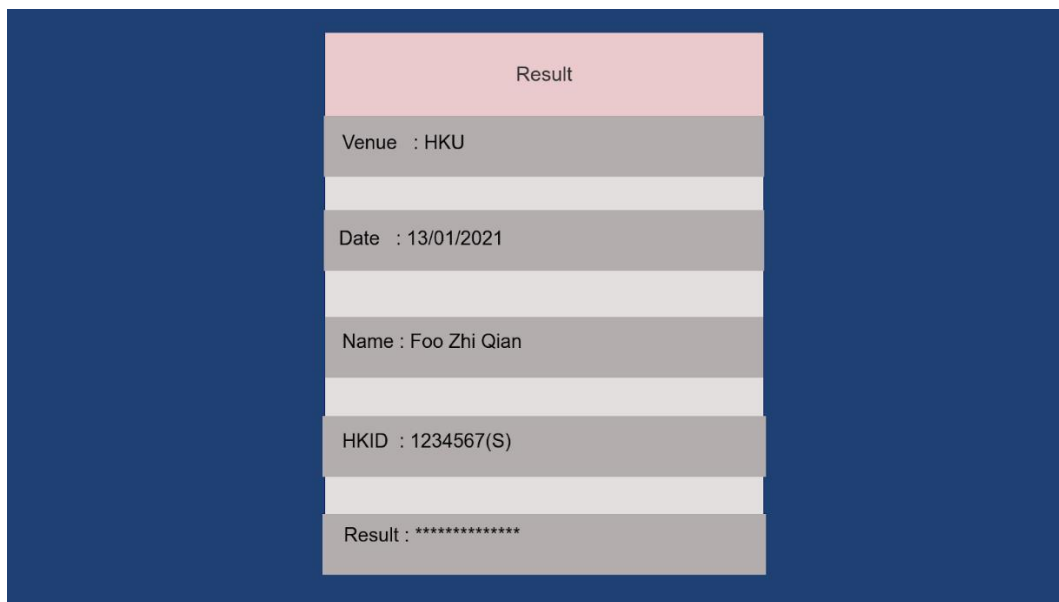


Figure 19. The result page where the patients who have undergone traditional Covid-19 testing can check their results directly without having to pay.

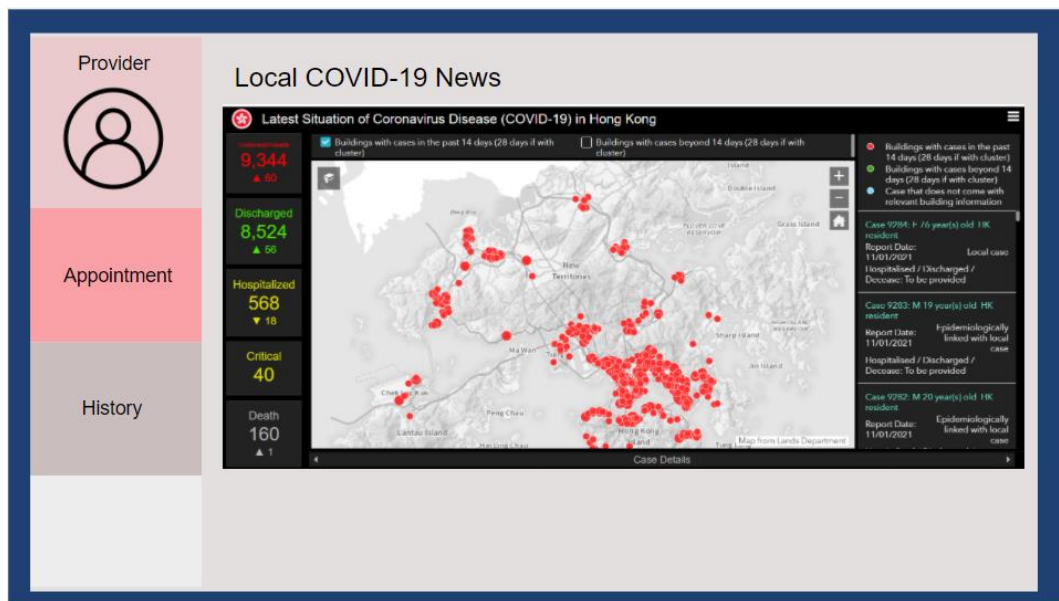
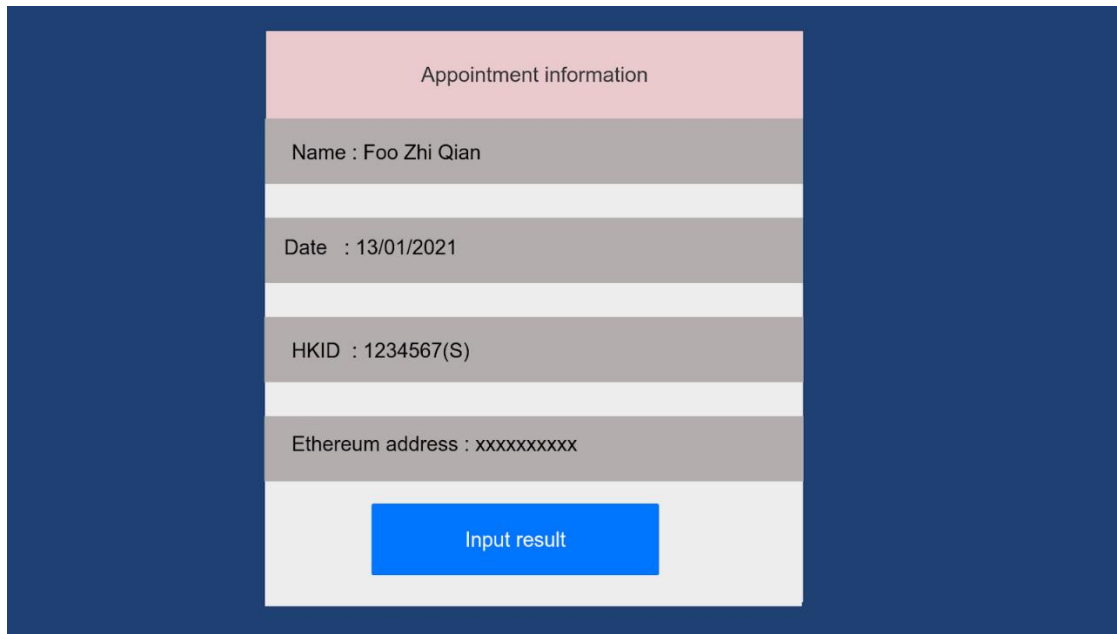


Figure 20. The new landing page of a supplier where patients' appointments can be viewed.



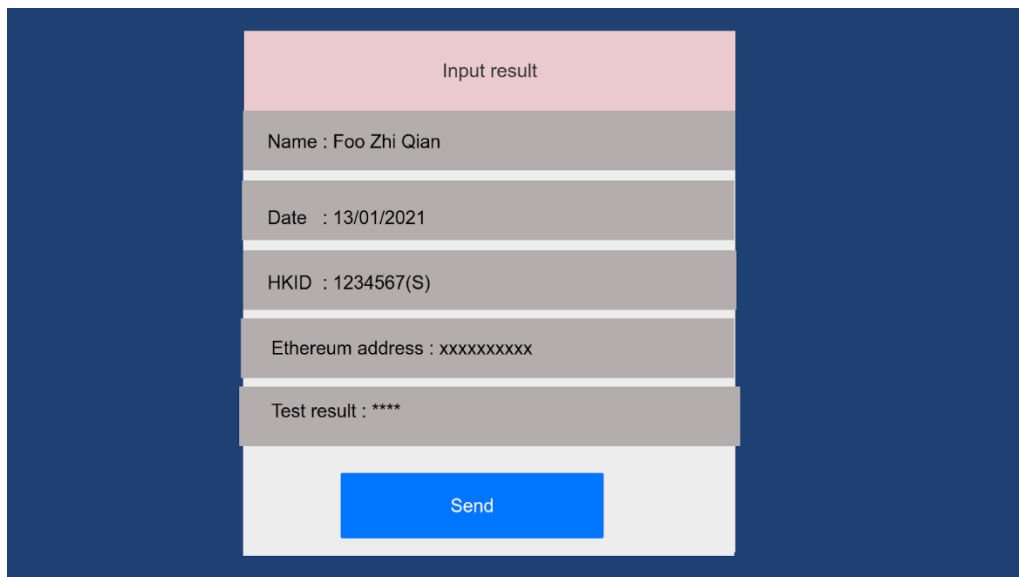
Figure 21. The incoming appointment page of a supplier.



The image shows a web interface for appointment information. It features a central form with a light gray background, set against a dark blue background. The form has a pink header bar with the text "Appointment information". Below the header, there are four horizontal bars with a light gray background, each containing a label and a value: "Name : Foo Zhi Qian", "Date : 13/01/2021", "HKID : 1234567(S)", and "Ethereum address : xxxxxxxxxx". At the bottom of the form is a blue button with the text "Input result".

Appointment information	
Name :	Foo Zhi Qian
Date :	13/01/2021
HKID :	1234567(S)
Ethereum address :	xxxxxxxxxx
<input type="button" value="Input result"/>	

Figure 22. The appointment information page where the details of appointments are depicted.



The image shows a web interface for inputting test results. It features a central form with a light gray background, set against a dark blue background. The form has a pink header bar with the text "Input result". Below the header, there are five horizontal bars with a light gray background, each containing a label and a value: "Name : Foo Zhi Qian", "Date : 13/01/2021", "HKID : 1234567(S)", "Ethereum address : xxxxxxxxxx", and "Test result : ****". At the bottom of the form is a blue button with the text "Send".

Input result	
Name :	Foo Zhi Qian
Date :	13/01/2021
HKID :	1234567(S)
Ethereum address :	xxxxxxxxxx
Test result :	****
<input type="button" value="Send"/>	

Figure 23. The input result page for a supplier who uses traditional Covid-19 testing method to enter the results of their patients.

3. Methodology

In this section, the deliverable and the core functionalities of our product will be outlined. We will first provide a quick walkthrough of the user interfaces and functional components in our platform. Ethereum, which is the framework that we have chosen to build our application, will be compared to Hyperledger, in terms of its strengths and weaknesses, and how it is more suitable to be deployed as the framework of our FYP. Blockchain technology, as the core technology that drives our decentralized web application, will also be detailed in Section 3.4.

3.1 Walkthrough of User Interfaces

We have designed our system architecture carefully and produced some amazing user interfaces to provide our users with best experience. A walkthrough of our user interfaces is as follow:

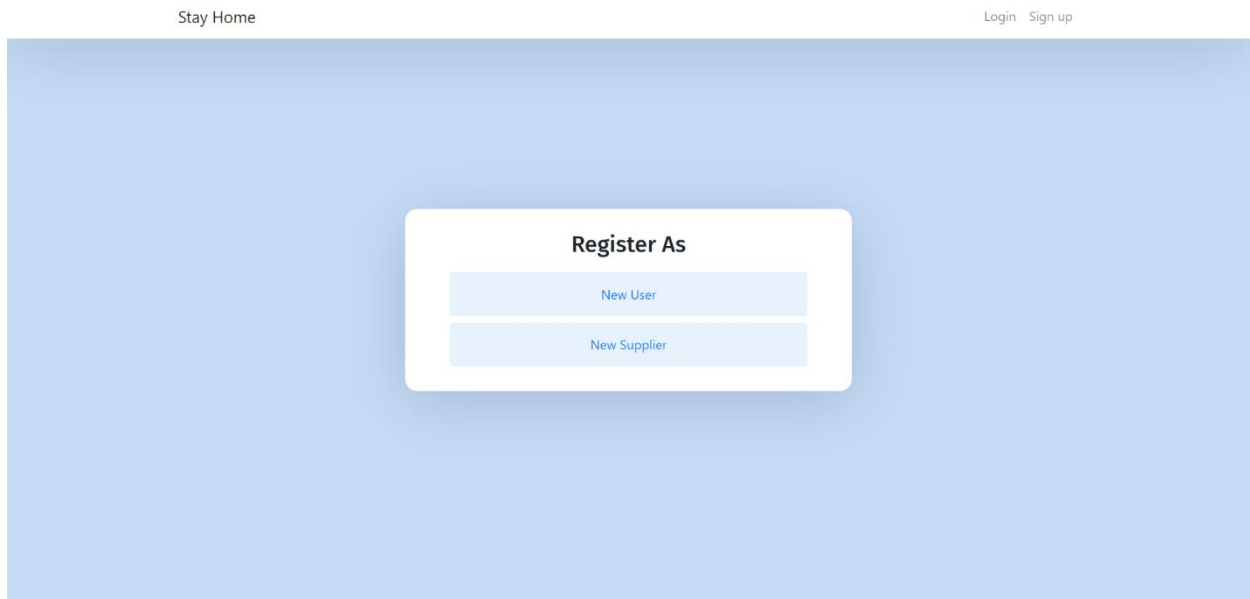


Figure 24. The user will get to choose to either register as a user or a test supplier.

The screenshot shows a web application interface with a light blue background. At the top, there is a navigation bar with 'Stay Home' on the left and 'Login Sign up' on the right. In the center, a white modal box titled 'New User' is displayed. This modal contains several input fields for registration: 'Username' (placeholder: 'Enter username'), 'Name' (placeholder: 'Enter name as per HKID'), 'Ethereum Address' (placeholder: 'Enter ethereum address'), 'HKID / Passport Number' (placeholder: 'Enter hkid or passport no.'), 'Password' (placeholder: 'Enter password'), and 'Public Key' (placeholder: 'Enter public key'). Each field is a light gray rectangle with its respective label above it.

Figure 25. To become a new user, one has to fill in important information including Ethereum address and Public Key that they got before hand.

The screenshot shows the same web application interface as Figure 25. In the center, a white modal box titled 'Sign In As' is displayed. It contains two light blue buttons stacked vertically. The top button is labeled 'User' and the bottom button is labeled 'Supplier'. Both buttons have a slight shadow and rounded corners.

Figure 26. The user will get to choose to either sign in as a user or a test supplier.

Stay Home Login Sign up

[Back](#)

Sign In As User

Username

Password

Private Key

[Submit](#)

Figure 27. The user will need to provide username, password, and private key for login.



Figure 28. The user will be redirected to the user landing page upon login.

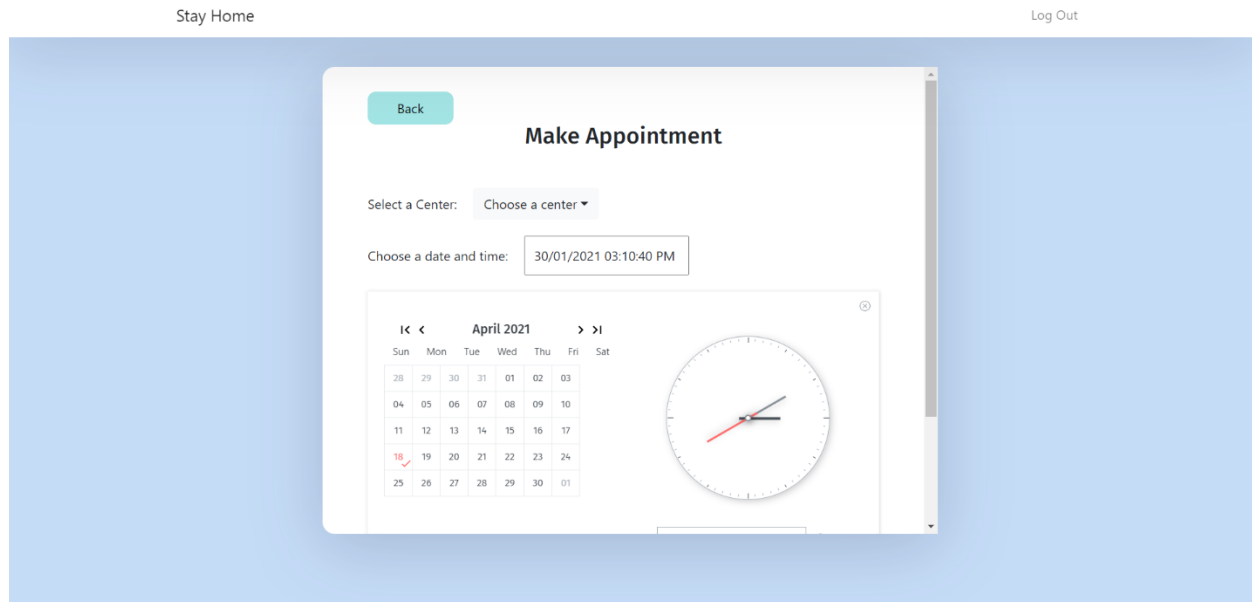


Figure 29. The user will get to make an appointment with the health institutions which have registered in our web application for Covid-19 testing.

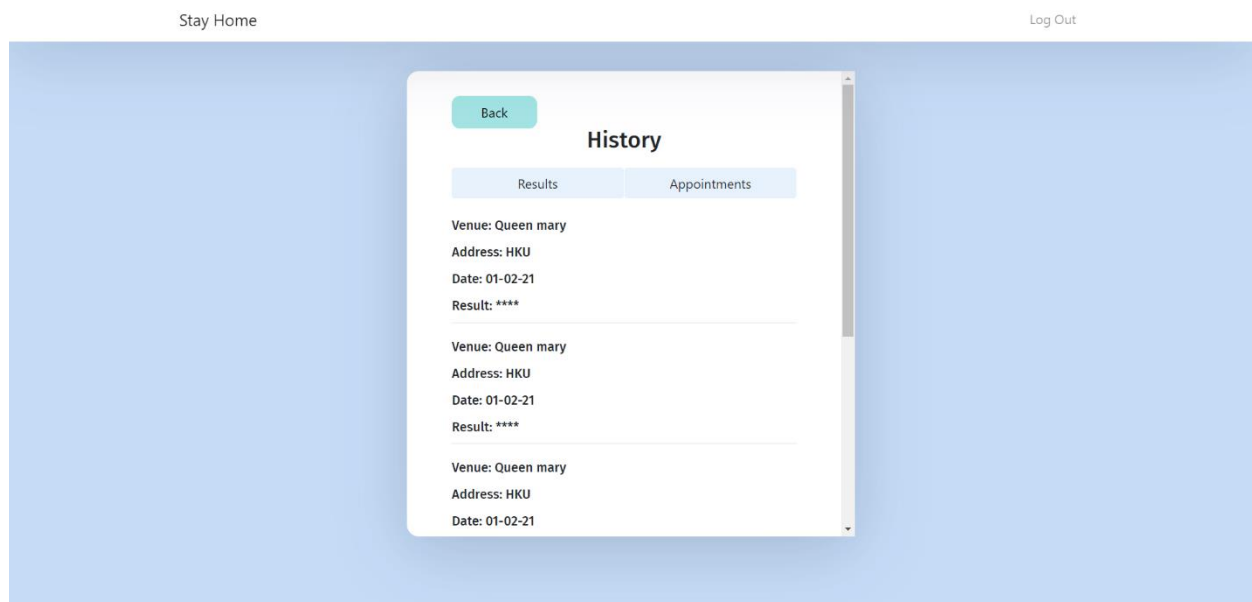


Figure 30. The user can refer to the results in History page to retrieve their pending results.

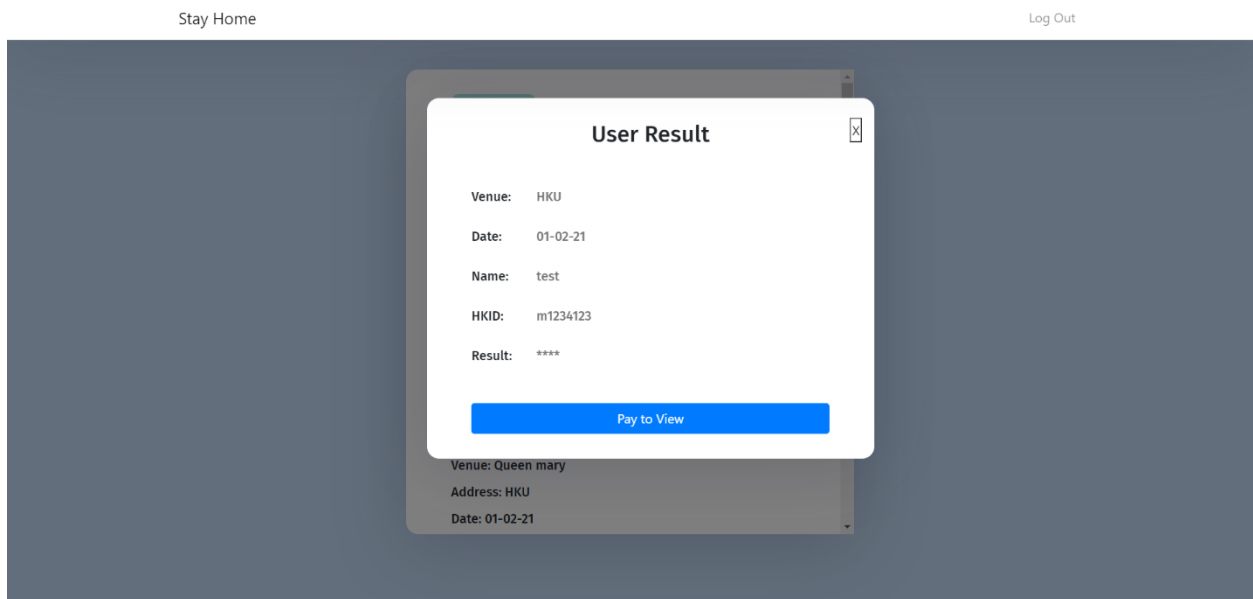


Figure 31. On clicking on any of the pending results in history page, a modal will pop out to prompt the user to pay to view his result.

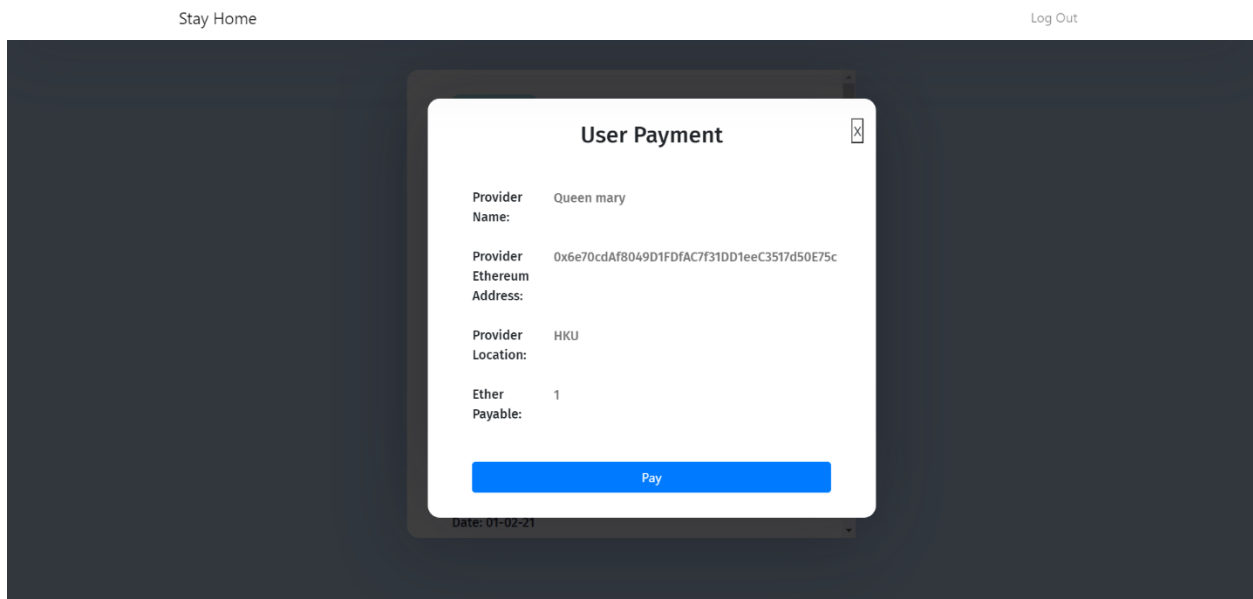


Figure 32. On clicking the “Pay to View” button, another modal will pop out to display the payment details and users can pay to unveil their pending results.

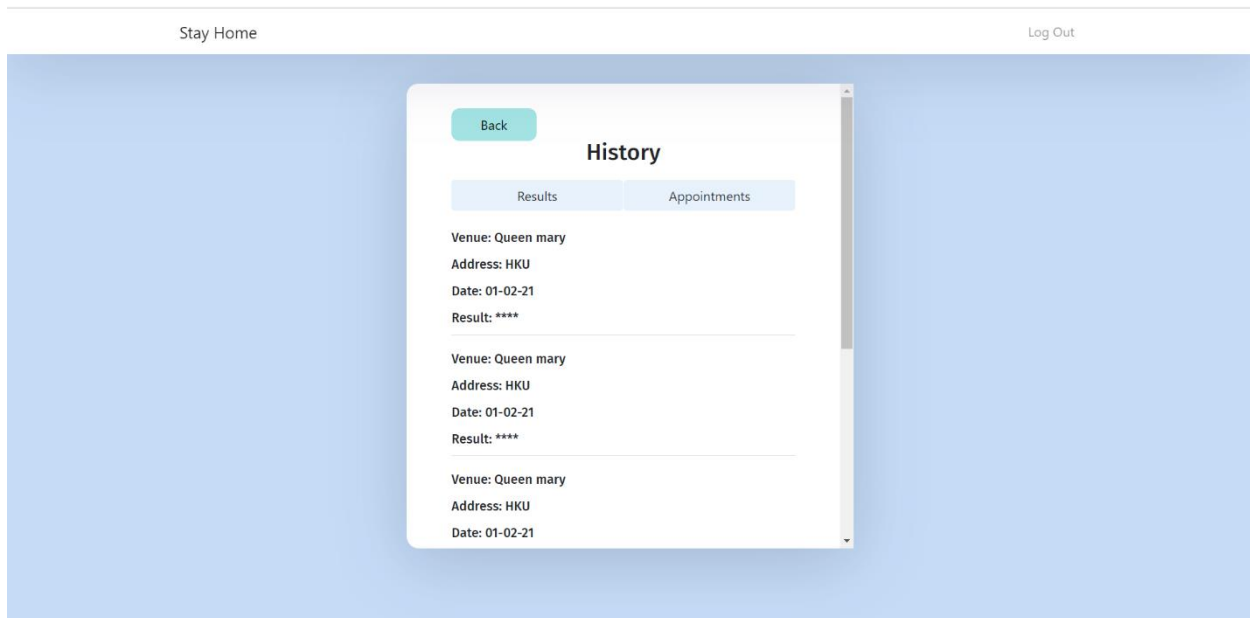


Figure 33. On clicking the “Appointments” header on the history page, another modal will pop out to display the payment details and users can pay to unveil their pending results.



Figure 34. On clicking the “User Information” on the side navigation bar of user landing page, the user will be redirected to the user information page, and this QR code can be shown to the PIC at kiosk to identify their identity and Ethereum address easily during the Covid-19 testing.

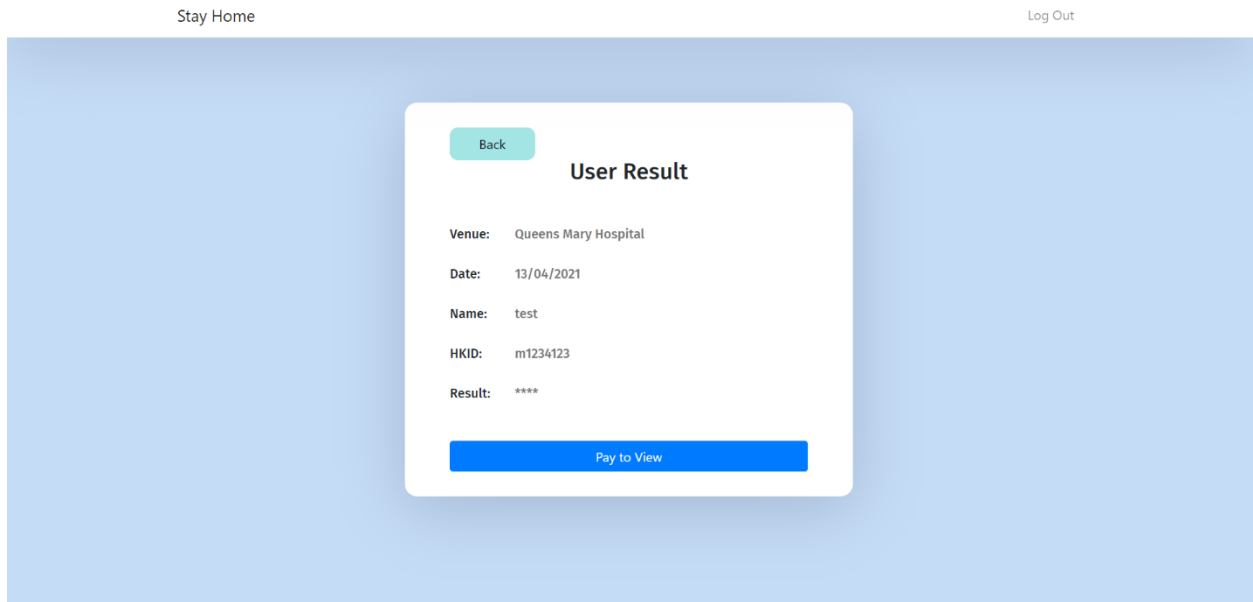


Figure 35. On clicking the “Latest Test Result” button on the user landing page, users will be redirected to the latest user result page, where they will be prompted to pay to view their latest test result.

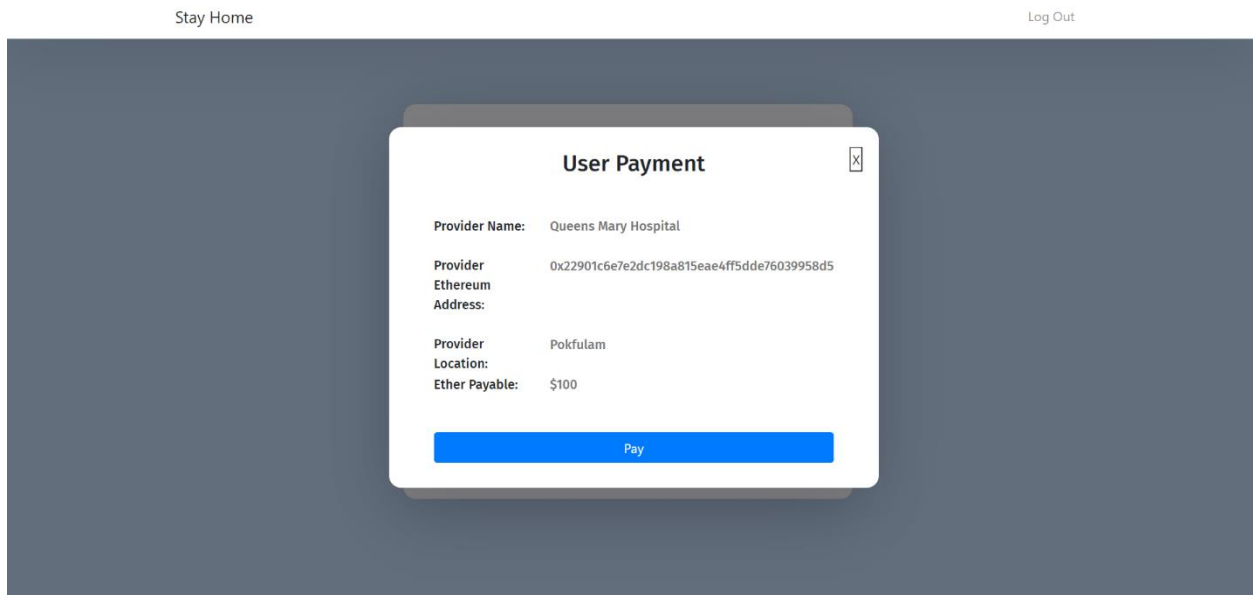


Figure 36. On clicking the “Pay to View” button, another modal will pop out to display the payment details and users can pay to unveil their latest results.

The screenshot shows a web application interface with a light blue background. At the top, there is a navigation bar with 'Stay Home' on the left and 'Login Sign up' on the right. In the center, a white modal box titled 'New Supplier' is displayed. This form contains several input fields: 'Username' (placeholder: 'Enter username'), 'Name' (placeholder: 'Enter name'), 'Ethereum Address' (placeholder: 'Enter Ethereum address'), 'Healthcare Provider Number' (placeholder: 'Enter healthcare provider no.'), 'Email' (placeholder: 'Enter email'), and 'Location' (placeholder: 'Enter location'). Each field is accompanied by a small downward arrow icon on the right side of the input box.

Figure 37. To become a new supplier, one has to fill in important information including Ethereum address, Healthcare Provider Number that they got before hand.

The screenshot shows the same web application interface. The central white modal box is now titled 'Sign In As Provider'. It features a 'Back' button in the top left corner. Below the title, there are two input fields: 'Username' (placeholder: 'Enter username') and 'Password' (placeholder: 'Enter password'). At the bottom of the form is a blue 'Submit' button.

Figure 38. To login as a supplier, one has to provide the username, password and private key.

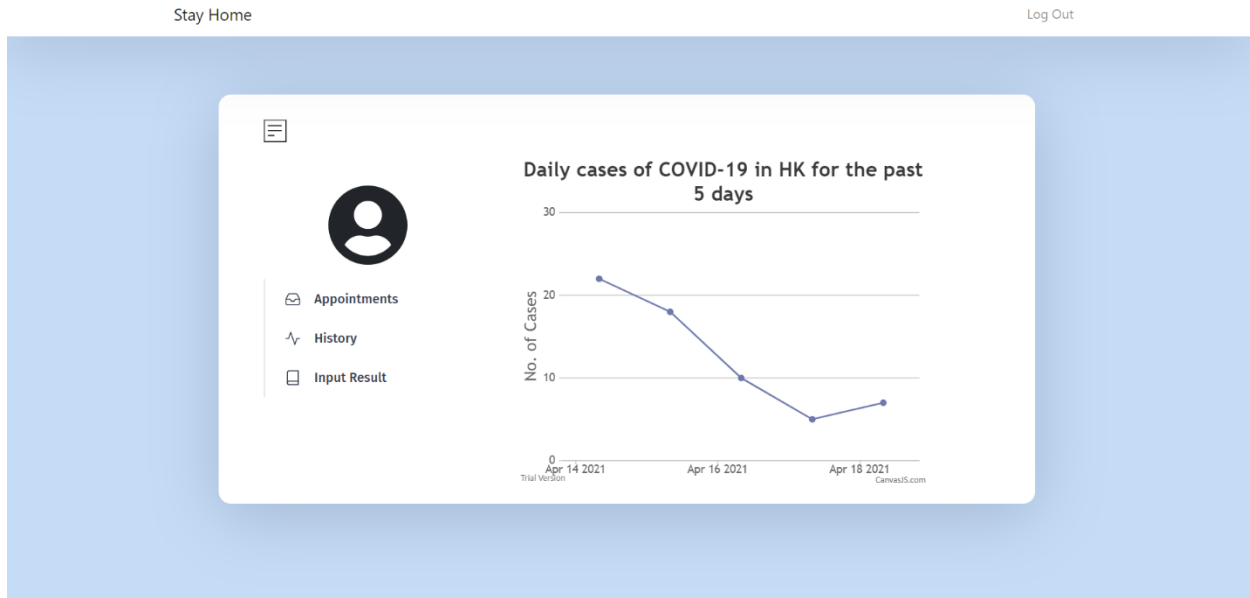


Figure 39. Upon logging in, the supplier will be redirected to the supplier landing page.

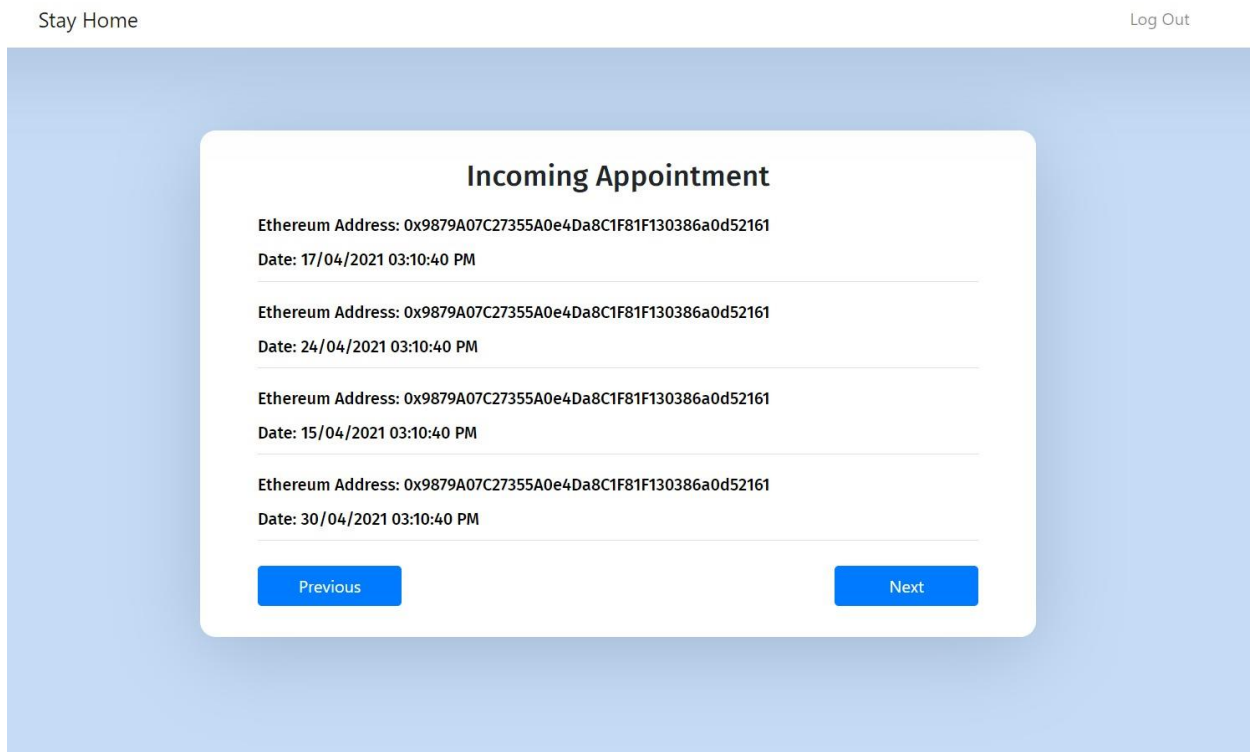


Figure 40. On clicking on the “Incoming Appointment” on the sidebar, a supplier can see all the incoming appointments.

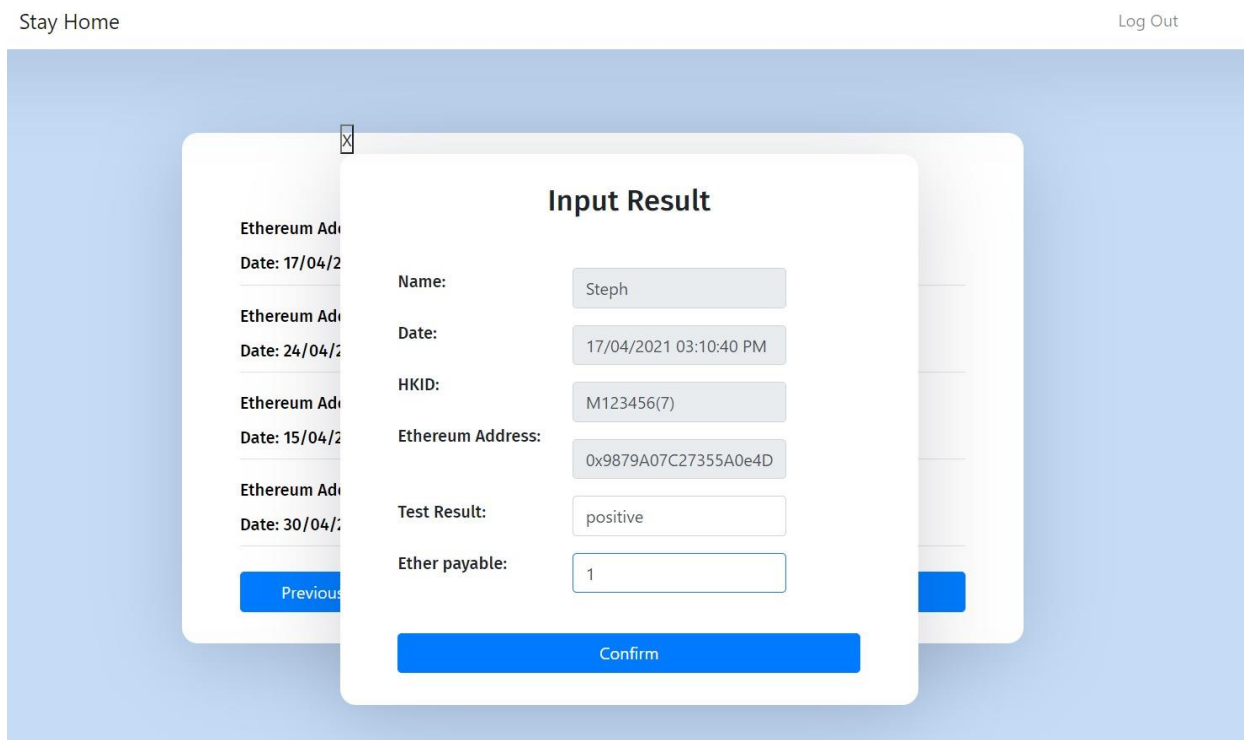


Figure 41. Clicking on any of the incoming appointments, the health institution can manually input the test result of the patient if the patient signs up for a testing kit differs from the SpectraLIT device.

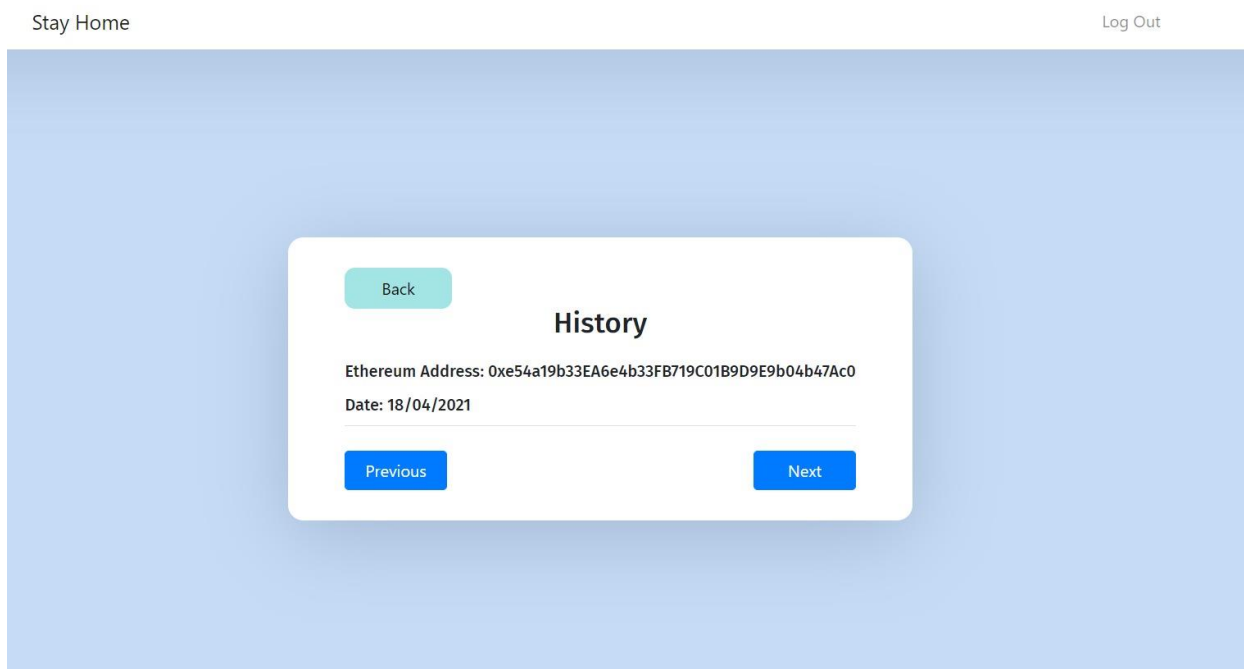


Figure 42. On clicking on the “History” on the sidebar of supplier landing page, a supplier can see all the incoming appointments.

3.2 Core Functionalities of the Blockchain Web Application

Approaching the end of this Final Year Project, we are delivering a blockchain-based decentralized web application that allows the public to register for Covid-19 testing. Each registered user must have a public key and a private key before signing up. Users will be anonymous to each other and encryption will be applied to records in the blockchain to ensure that a user can only view his or her own test result. Users can make appointment with health institutions registered on our platform to arrange for a Covid-19 test. Past results and appointments can be viewed anytime from the sidebar. No centralized agency or third party is involved in the process so users' privacy can be protected. From the healthcare suppliers perspectives, incoming appointments can be found from the sidebar. At the history page, past results and appointments with patients will be displayed.

3.3 Ethereum as the Backend Framework

Ethereum and Hyperledger are the two most prominent and developed platforms for building blockchain-based applications. The main distinction between Hyperledger and Ethereum are summarized in Table 1 below.

Features	Hyperledger	Ethereum
Purpose	Preferred platform for B2B businesses	Platform for B2C businesses and generalized applications
Confidentiality	Confidential transactions	Transparent
Mode of Peer Participation	Private and Permissioned Network	Public/Private and Permissionless Network
Consensus Mechanism	Pluggable Consensus Algorithm: No mining required	PoW Algorithm: Consensus is reached by mining
Programming Language	Chaincode written in Golang	Smart Contracts written in Solidity

Cryptocurrency	No built-in cryptocurrency	Built-in cryptocurrency called Ether
-----------------------	----------------------------	--------------------------------------

Table 1. The summary of main differences between blockchain platform [5].

After carefully reviewing the different features offered by both platform as summarized in Table 1, we have decided to use Ethereum platform for the back-end development because it is generic in purpose and supports both public and private platforms hence ideal for B2C transactions. This is well suited for our project because our target audience would be the public which means anyone can sign up for an account for Covid-19 testing. In contrast, Hyperledger is only ideal for B2B transactions since the participation of nodes is permissioned as it only approves a set of predefined members to get access to its blockchain [6]. Moreover, Ethereum is a type of programmable blockchain that supports smart contracts, a form of executable code written in Solidarity programming language. Smart contracts allow defined instructions and transactions to be carried out between different parties without the existence of a third-party central authority [7]. There are also various existing libraries with API that facilitate the development of our decentralized application. Currently, we plan to use web3.js and react.js to allow us to connect and interact with Ethereum for our front-end development. In addition to Ethereum as the blockchain database, Interplanetary File System (IPFS) will also be used to store non-blockchain information such as username, password, and email-address as storing them in a blockchain is costly due to the high transaction fee and slow access time.

One downside of using Ethereum compared to Hyperledger is that it does not provide transaction privacy as all transactions are posted to the public ledger and are visible to all participants. This is completely opposite to Hyperledger where privacy is available when transacting across different channels. With Hyperledger, we can allow something to be possible to one person and not visible to the rest. For example, if we are providing a discounted price to a customer but not the others. Hyperledger is associated with a framework which Ethereum is not, that provides the functionality of masking certain information from the others so that the other customers will not spot the discrimination. Nevertheless, confidentiality of transactions is not a huge concern when it comes to our project because all of the transactions posted on the blockchain are to be treated

equally and with the help of encryption mechanism, only a user with a key can access his or her information on the blockchain.

3.4 Blockchain as the Core Technology

Blockchain technology is a digital ledger that stores records in blocks and distributes them across several databases connected through peer-to-peer nodes, known as the “chain” [8]. As an analogy, blockchain technology can be deemed as a spreadsheet duplicated thousands of times across a network of computer, where everyone can see the data, but they cannot tamper it. Any request to add records to the blockchain will be verified by thousands or even millions of computers distributed across the network. It is a write-once, append-many technology, where an existing data on the chain can never be erased, and new data are continuously appended to the end of the chain.

As illustrated in Figure 1, every block in the blockchain is tamper resistant as it is tied to previous block through hash encryption. Every transaction in this ledger is authorized by the digital signature of the owner, which authenticates the transaction and safeguards it from tampering[8]. In order to tamper a record on the blockchain, a hacker would need to change the block containing that record as well as those linked to it to avoid detection. As blockchains are continually updated and kept in sync across all the nodes in the network, it would require massive amounts of computing power to access every instance (or at least a 51 percent majority) of a certain blockchain and alter them all at the same time, making the cost far outweighs the benefit and rendering it highly impossible to happen[4].

With blockchain technology, each page in a ledger of transactions forms a block. That block has an impact on the next block or page through cryptographic hashing. In other words, when a block is completed, it creates a unique secure code, which ties into the next page or block, creating a chain of blocks, or blockchain.

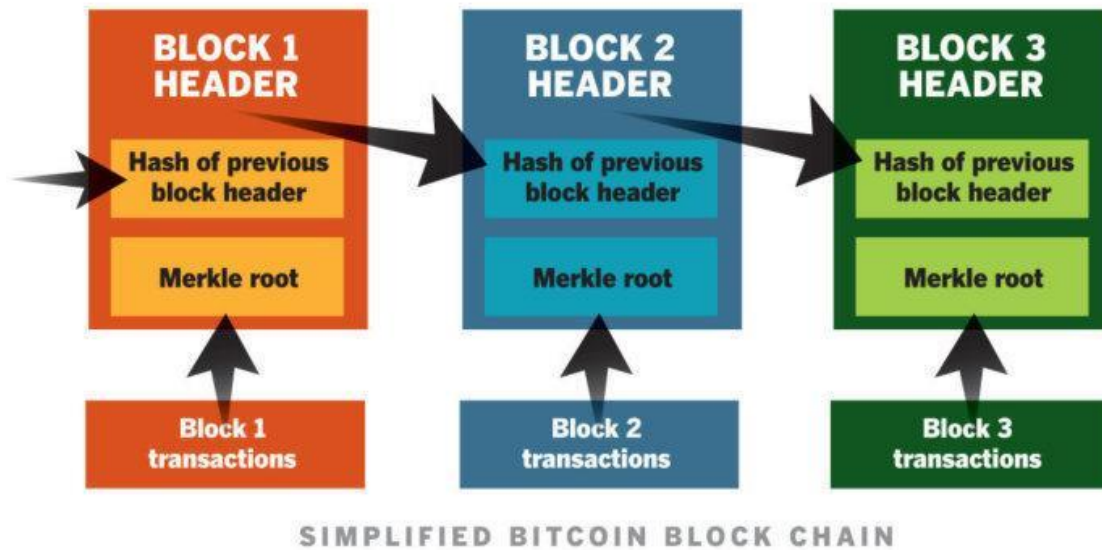


Figure 43. The details of a blockchain [9].

The concept of blockchain technology is summarized in Figure 2 below. Each digital record or transaction in the electronic ledger is called a block. When a block is completed, it creates a unique secure code that ties it to the next block. Whenever someone in the network requests to add a new block of record to the blockchain, the request will be broadcasted to every node in the network. The peer-to-peer network of nodes will validate the transaction and user's status using an algorithm and the verified block will be added to the end of the blockchain in every node.

Blockchain technology is the core technology behind our web application that makes it different from traditional database because it is completely decentralized, immutable, shared, and secured. Owing to blockchain technology, problems like inefficiency, data tampering, single-point-failure and privacy concerns that were expounded in Section 1.2 can be resolved.

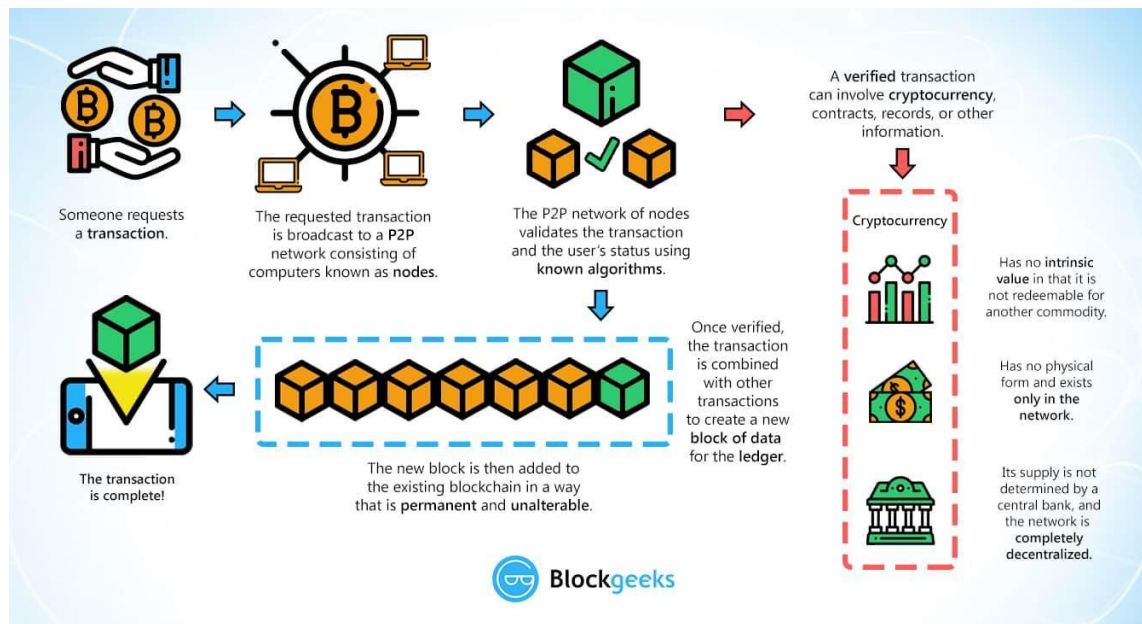


Figure 44. The flow of creating and adding a block into the blockchain [10]

4. Testing and Limitations

In this section, I will provide an overview of how we have vigorously tested our code and make sure that it works perfectly. The decentralized application is tested from three different perspectives, which are local blockchain deployment to mimic a real-world deployment, unit functionality testing to ensure that every function is performing the duties and authentication bypassing attempt to get through the security authentication.

4.1 Local Blockchain Deployment

To start testing our decentralized application on a localhost, first we need to set up truffle on our computer. Truffle (also known as Ganache), can quickly fire up a personal Ethereum blockchain locally which we can use to run tests, execute commands, and inspect state while controlling how the chain operates.

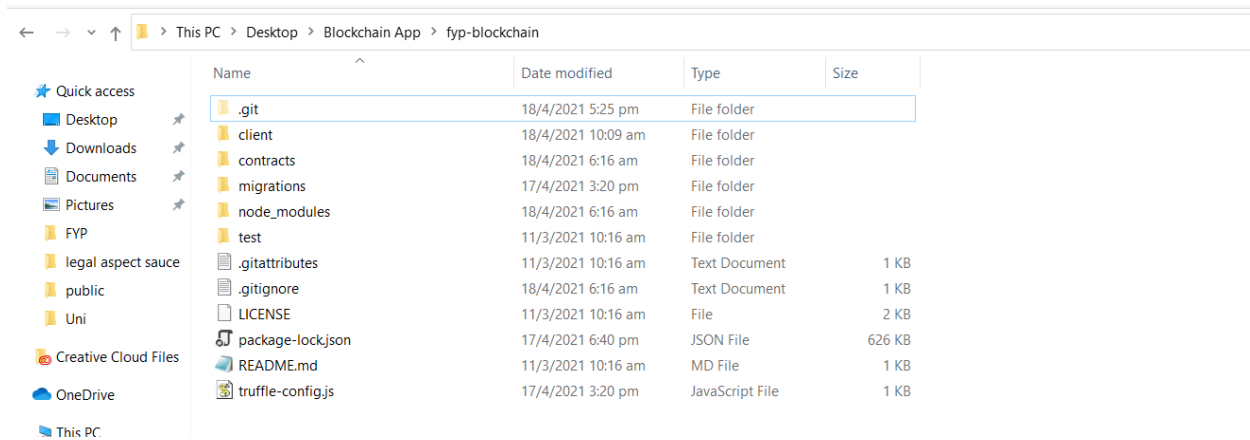


Figure 45 In this directory, run the command “truffle dev” on the command prompt. 10
Ethereum accounts will be created, and we will obtain the Ethereum addresses and Private Keys
as follow for our testing purposes

```
C:\Users\Foo\Desktop\Blockchain App\fyp-blockchain>truffle dev
Truffle Develop started at http://127.0.0.1:8545/

Accounts:
(0) 0xfa92e08e17e919f8606e7d4d5c87185294df8fd6
(1) 0xc5467cd09de83e5377f8e8a4eff06859381c6047
(2) 0x9e165a81c937705f079052166cb03f08f6175833
(3) 0x22901c6e7e2dc198a815eae4ff5dde76039958d5
(4) 0x15463ab43c2c8c80abc9f1c6a267b49c20d99f7
(5) 0x56be9077ded14a8d4e3f084e382a26c0d4caaf25
(6) 0x0ca56081112cbd4a36fd4ea695998a60c84b0b9
(7) 0x237c886206d8bfef9d7efbcad70361458fb8797
(8) 0x6855c14f5f3a60922f83525abb29681e5a70a22e
(9) 0xe7b1d0551312387d8dd37cae1999e30f3498483b

Private Keys:
(0) 17e23139f2aa4d86e2854ba9b70b104dc4c4d8ae595871ca7b64fc078bfdd001
(1) 07ee7030402d49ae48af847066871efe43b1078b0b26ff2070464a7c749f8aeb
(2) 82d474399ee41320dda30bd8ef37c3a5c75e0db14be79aa8495871ceca6872a2
(3) ab89333ebd43b89603d9141fb7604def4cd56da8a66348a9f94f9435cce88367
(4) 0dedb92301363a2b39fd1af47cda1d29df7d9e93062fd071b6c3c9a8b354783f
(5) 31c5a380198da7e4835238b702cbb868765ea99a26dee2788a47c7374db5ff1d
(6) c7be0ebb2d25fd32edfafa64b6a93c0fc86cf93a5012365b704c03e4e8f5bcf3
(7) 12f4012f89de89643dee6c837178f7f816be5c24c912d7fbc00ddfcda4b06c90
(8) a4333dfb40ae68b5275d0baa5c752fb4176e270a0dfecce6d6ba992a9344b8a
(9) 5f99c1df4678d756ecf94438806c5be75c10c64090e7e648b1c22d0ad4a045bf
```

Figure 46 The command prompt of starting truffle environment

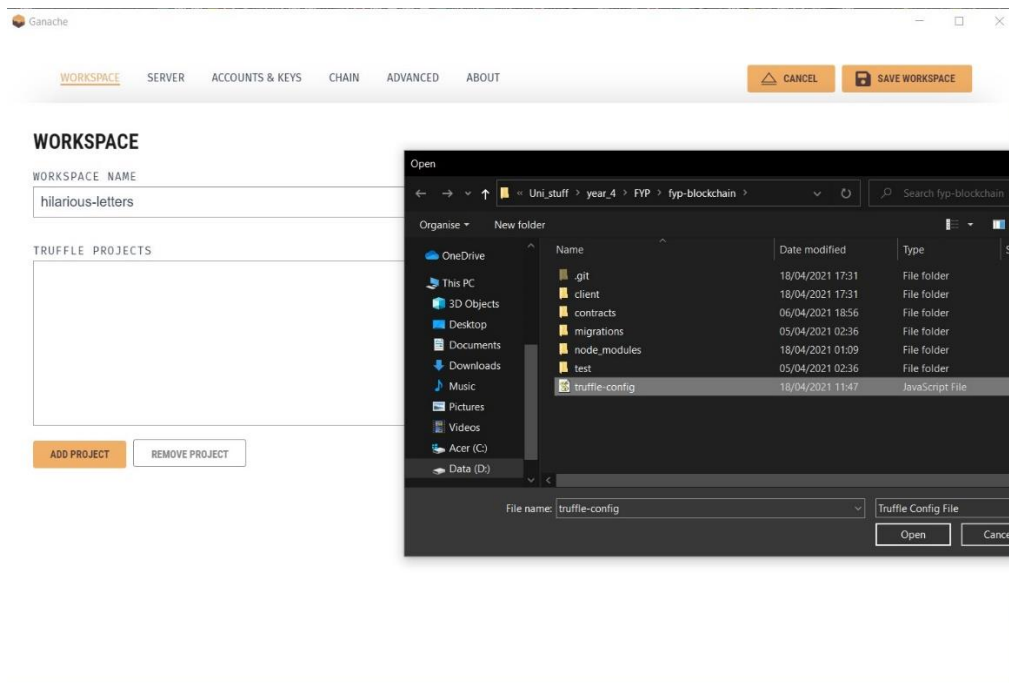


Figure 47 As an alternative, we can download and start Ganache to achieve the same purpose while having better visualizations about the contracts deployed, gas spent and transactions that have taken place. Open up the workspace of Ganache and run truffle-config file in it as shown above.

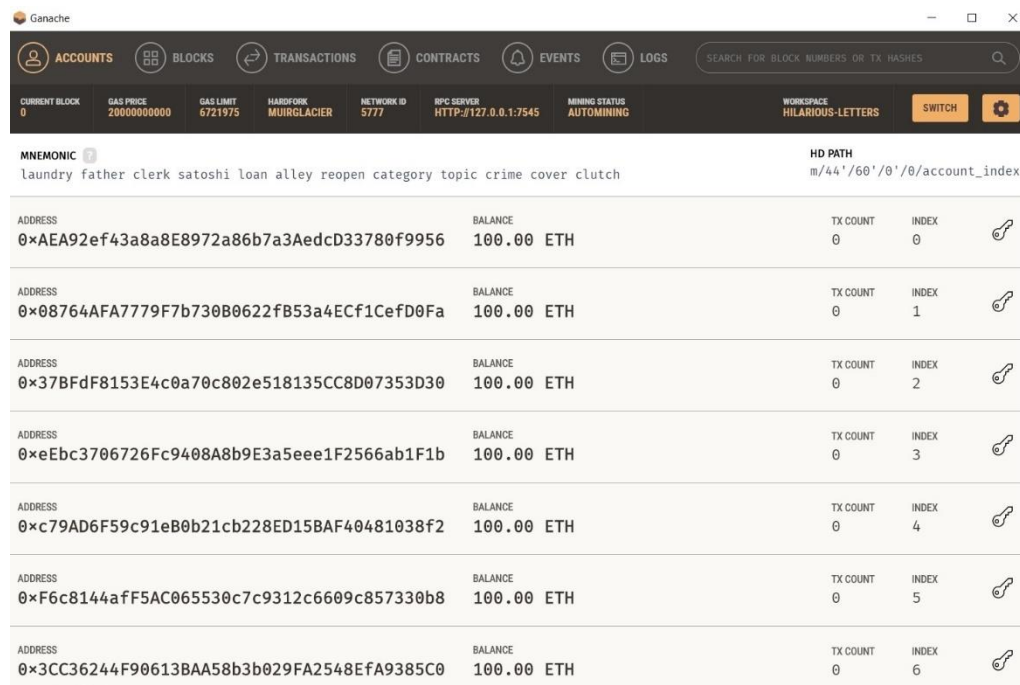


Figure 48 Ganache provides detail information about each transaction, event, and log that have taken place, allowing us to debug it quickly:

After setting up truffle/Ganache, change directories into the client folder. Run the command “npm start” on the command prompt.

```
C:\Users\Foo\Desktop\Blockchain App\fyp-blockchain\client>npm start

> client@0.1.0 start C:\Users\Foo\Desktop\Blockchain App\fyp-blockchain\client
> react-scripts start
Starting the development server...

Browserslist: caniuse-lite is outdated. Please run next command `yarn upgrade`
Compiled with warnings.

./src/App.js
  Line 1:16:  'useEffect' is defined but never used  no-unused-vars
  Line 1:27:  'useState' is defined but never used    no-unused-vars
  Line 18:8:  'getWeb3' is defined but never used     no-unused-vars
  Line 21:50: 'Link' is defined but never used        no-unused-vars

./src/page/loginPage.js
  Line 1:17:  'useState' is defined but never used  no-unused-vars
  Line 2:27:  'Router' is defined but never used    no-unused-vars
  Line 2:35:  'Switch' is defined but never used    no-unused-vars
  Line 2:43:  'Route' is defined but never used     no-unused-vars
  Line 108:26: Expected '!==' and instead saw '!='  eqeqeq
  Line 201:26: Expected '!==' and instead saw '!='  eqeqeq
```

Figure 49 The command prompt of starting our react front-end on the localhost

A window will appear which redirects you to the login page of our decentralized application on the localhost as follow:

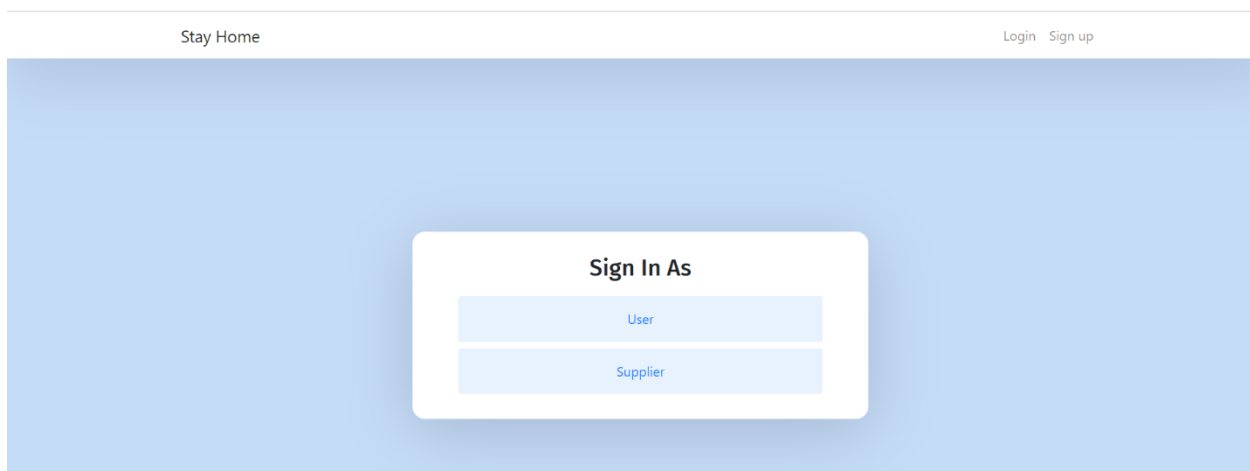


Figure 50 The Login page where the user will be redirected to once the program starts.

At this point, the first experiment is done, and our decentralized application is now hosted on the localhost.

4.2 Unit Functionality Testing

To ensure that every part of the unit functionality is performing well and interacting with each other correctly, we can run through the workflow of our web application once. A complete workflow should start from the registration of users / suppliers and end when every button and scenario has been tested out. The workflow of the interface should be simple and intuitive to avoid causing confusion to the end-users. When testing out the functionalities, we have role-played different parties on our application and mimic the interactions between them to ensure that everything is bound together correctly. For example, after we have made an appointment as a user to a hospital, we should login to the hospital's account to ensure that the appointment appear on the incoming appointments list. Likewise, after a hospital authority has entered the user's test result manually, the result should show up on the user side as well. The testing is satisfactory as we have ensured that every function is performing fine before the deployment.

4.3 Attempt to Bypass Authentication Process

In addition to functionality testing and local blockchain deployment, we have also tried bypassing the webpage by entering the correct URL without logging in. As there is a global variable remembering the session of a user login status, any unauthenticated access to our webpage's URL will redirect the user back to the login page. This is important to prevent unauthenticated access to our web application.

4.4 Limitations

One of the limitations that associate with the blockchain application is its lack of scalability. Blockchain technology relies on encryption to provide its security as well as establish consensus over a distributed network. This essentially means that, in order to "prove" that a user has permission to write to the chain, complex algorithms must be run, which in turn making it slow and cumbersome. For instance, the theoretical maximum throughput of a traditional database like MySQL is 1000 transactions/second [12]. This is significantly higher than that of Ethereum,

which can only handle approximately 20 transactions/second [13]. Therefore, the slow and unwieldy nature of blockchain is the bottleneck that hinder the proliferation of blockchain applications. Currently, blockchain companies are working on different solutions such as increasing the block size of the blockchain, providing off-chains solutions, introducing delegated consensus protocol, and sharding to provide scalable infrastructure so that blockchain application will become mainstream in the future [14]. However, before any of these solutions is proven to be effective, the throughput of our blockchain application will remain a limitation to its scalability.

5. Conclusion

In this report, blockchain technology is introduced to resolve the problems associated with traditional databases, such as privacy concerns, data tampering, lack of system interoperability, and data loss in storing Covid-19 mass testing result. At the point of writing this report, a fully functional web application has been deployed and ready to serve. This is satisfactory as we have met all the deadlines that we have set for ourselves and managed to finish everything within the time frame. Vigorous testing has been conducted on our web application and we are confident that our product will beat the expectation of users and bring them confidence in using it with no privacy concern issues anymore.

In short, blockchain technology which act as a decentralized system for recording and documenting transactions has the potential to revolutionize the world economy. With proper implementation and careful research, the technology that underpins the cryptocurrency can be extended to many other industries such as banking and healthcare.

References

- [1] “Coronavirus (COVID-19),” *Google News*. [Online]. Available: <https://news.google.com/covid19/map?hl=en-US&mid=/m/02j71&gl=US&ceid=US:en>. [Accessed: 26-Oct-2020].
- [2] “COVID-19 to Plunge Global Economy into Worst Recession since World War II,” *World Bank*. [Online]. Available: <https://www.worldbank.org/en/news/press-release/2020/06/08/covid-19-to-plunge-global-economy-into-worst-recession-since-world-war-ii>. [Accessed: 10-Oct-2020].
- [3] T. Mcghin, K.-K. R. Choo, C. Z. Liu, and D. He, “Blockchain in healthcare applications: Research challenges and opportunities,” *Journal of Network and Computer Applications*, vol. 135, pp. 62–75, 2019.
- [4] “Blockchain security: What keeps your transaction data safe?,” *Blockchain Pulse: IBM Blockchain Blog*, 18-Feb-2020. [Online]. Available: <https://www.ibm.com/blogs/blockchain/2017/12/blockchain-security-what-keeps-your-transaction-data-safe/>. [Accessed: 10-Oct-2020].
- [5] C. N. Says: A. says: and A. R. says: “Hyperledger vs Ethereum: Key Differences and Comparison,” *Edureka*, 22-May-2019. [Online]. Available: <https://www.edureka.co/blog/hyperledger-vs-ethereum/#:~:text=The most essential distinction between,leverages blockchain technology for business>. [Accessed: 26-Oct-2020].
- [6] “Hyperledger vs Ethereum,” *101 Blockchains*, 04-Apr-2019. [Online]. Available: <https://101blockchains.com/hyperledger-vs-ethereum-2/>. [Accessed: 10-Oct-2020].
- [7] “Introduction to smart contracts,” *ethereum.org*. [Online]. Available: <https://ethereum.org/en/developers/docs/smart-contracts/>. [Accessed: 10-Oct-2020].

- [8] Simplilearn, “What is Blockchain Technology and How Does It Work?,” Simplilearn.com, 13-Nov-2020. [Online]. Available: [https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology#:~:text=Blockchain technology is a structure,to as a 'digital ledger](https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology#:~:text=Blockchain technology is a structure,to as a 'digital ledger.). [Accessed: 30-Nov-2020]
- [9] L. Mearian, “What is blockchain? The complete guide,” Computerworld, 29-Jan-2019. [Online]. Available: <https://www.computerworld.com/article/3191077/what-is-blockchain-the-complete-guide.html>. [Accessed: 30-Nov-2020].
- [10] A. Rosic and Blockgeeks, “What is Blockchain Technology? A Step-by-Step Guide For Beginners,” *Blockgeeks*, 25-Nov-2020. [Online]. Available: <https://blockgeeks.com/guides/what-is-blockchain-technology/>. [Accessed: 30-Nov-2020].
- [11] Truffle Suite, “Ganache,” *Truffle Suite*. [Online]. Available: <https://www.trufflesuite.com/ganache>. [Accessed: 26-Oct-2020].
- [12] *Napkin Problem 10: MySQL transactions per second vs fsyncs per second*. [Online]. Available: [https://sirupsen.com/napkin/problem-10-mysql-transactions-per-second/#:~:text=The maximum theoretical throughput of,fsync\(2\) per second](https://sirupsen.com/napkin/problem-10-mysql-transactions-per-second/#:~:text=The maximum theoretical throughput of,fsync(2) per second.). [Accessed: 27-Oct-2020].
- [13] E. Team, “Remaining challenges of blockchain adoption and possible solutions,” *Finextra Research*, 29-Feb-2020. [Online]. Available: <https://www.finextra.com/blogposting/18496/remaining-challenges-of-blockchain-adoption-and-possible-solutions>. [Accessed: 27-Oct-2020].
- [14] Y. Sim, “Tech in Asia - Connecting Asia's startup ecosystem.” [Online]. Available: <https://www.techinasia.com/biggest-problem-blockchain-company-solving>. [Accessed: 30-Nov-2020]
- [15] “What is Ethereum?,” *ethereum.org*. [Online]. Available: <https://ethereum.org/en/what-is-ethereum/>. [Accessed: 18-Apr-2021].

- [16] “InterPlanetary File System,” *Wikipedia*, 16-Apr-2021. [Online]. Available: https://en.wikipedia.org/wiki/InterPlanetary_File_System. [Accessed: 18-Apr-2021].