Rutgers University

Software Engineering

Group 5
Rizwan Chowdhury, Nathaniel Arussy, Dang Khoa Dinh, Smeet Kathiria, Eric Rivera, Hersh Shrivastava, Suva Shahria, Khalid akash

Contribution Breakdown

Based off if assignments were completed in a timely manner when signed up for. Not completing parts that members signed up for had a negative influence on grading. The percentages below are what each member contributed to out of 100%. Please allocate the proportionate amount of points for each member. Note: Khalid Akash is in this group even though on Sakai, he is signed up for Group 4 (there was a switch)

Name	Contribution	Percentage
Nathaniel Arussy	Report 1: Editor, reviewed requirements, submitted comments, additional pictures for UI, prioritization analysis	8.32%
Rizwan Chowdhury	Report 1: -Wrote User Interface Requirements, - Created pictures for UI(data entry form, graphical visualization, and non-graphical visualization). - Edited Problem Statement a little and added UI related statements. Report 2: - Tracibility Matrix - UI mock-up UC-3	15.46%
Dang Khoa Dinh	Report 1: Problem statement Report 2:Use case diagrams Add some details to stakeholders Report 3: Mathematical model Attribute definition Project size estimation Plan of work	13.179%
Smeet Kathiria	Report 1: Glossary terms	15.46%

	Addition to UI Requirements,Use Cases. Report 2- Actors and goals, Addition to Use cases. Report 3: Addition to Concept Definitions, Traceability Matrix(Part 3), enhanced plan of work flowchart.	
Eric Rivera	R1P2: Stakeholders 5-9, update 3&4 R1P3: Domain Analysis Concept Definitions, Association Definitions, Domain Model Diagram	10.89%
Suva Shahria	Report 1: Enumerated Nonfunctional Requirements Report 2: Use Case Casual Description , Fully Dressed Cases, User Effort Estimation, Project Management	15.46%
Hersh Shrivastava	Report Part 2: Stakeholders 1- 2, Report Part 3: 5b System op contract.	5.75%
Khalid Akash	Report 1: Enumerated Functional System Requirements, Customer Problem Statement, Peer Reviewing, Report Organization/Cleanup Report 2: System Sequence Diagrams, Partial Fully Dressed Use Cases, Partial User Interface Specification Report 3: Mathematical Models	15.46%

	Project Management	

1. Customer Problem Statement

1.a Problem Statement

World health in perspective: chronic, noncommunicable diseases (NCDs), like cardiovascular disease or obesity or diabetes, are steadily increasing around the world. There are 422 million people with diabetes in 2014. This means roughly 1 in every 16 people on the planet has this disease. The WHO estimates that diabetes was the seventh leading causes of death in 2016. Diabetes can be treated and its consequences avoided or delayed with diet, physical activity, and checking blood glucose regularly. The majority of the burden is shouldered by low, and middle income countries. Population growth and aging are the largest contributors. In the US, the number of people over 65 is expected to triple in 2030, while there is a shortage of healthcare professionals. Furthermore, the market for biosensor is expected to surpass 29 billion dollars in 2024. Combating non-communicable disease has been outlined in a number of reports by WHO and US government which included reducing deaths from NCDs by 25 percent by 2025.

World Bank study estimates NCDs will cost the global economy about $35 trillion from 2005 to 2030. Data-driven diagnostic and analysis are the way of the future. There are 3 reasons for that: increased older population and rising awareness among people, shifting interest toward home diagnosis, and increased adoption of personalized and technological products.

Our goal shall be to offer a meaningful service to our client by anticipating health problems and enabling the customers to live a healthier life. Now, users can see their health status on demand. The product will use the large portion of population data and data visualization to put the information into perspective. The product fits well with the trend of personalized medicine and early diagnosis. We will design our product with innovation in mind. So, the system can be adjusted to customer's preferences, medical studies, and advances in biosensor.

The users will need to input medical (blood pressure, heart rate) or exercise data to our web-based service and then receive the result. We may make available means to upload data from wearable devices. We are working hard to make the service up to date with diverse features to give an in-depth look into the current state of population health. We hope that the product will be an evolution by bringing data analysis to health-conscious users and health experts alike. For health professionals, it can be a powerful tool for research, analysis, fact checking because we put the data into perspective. Currently, our service focuses on physiological parameter. Users, however, can manually input their dietary data should they wish. We will allow the blockchain to persist all user data and attempt to filter out any malicious requests to alter the overall descriptors of the population.

The product will strive to effectively communicate the data and analysis to customers through visual and non-visual means. We will provide visual and non-visual data to help customers compare their own data to others as well as see the bigger picture. The system will contain UI to show graphs that map the aggregate data with respect to each health parameter(or a set of parameters) for groups of people. The graphs will always pin-point the location of the users' data so that customers can see where they are relative to other users in a population group. For data that cannot be represented visually there will also be non-visual and text-based UI to show users their data and analysis of that data.The visual data presentation will be a key part in this product. Visual aids like graph will bridge the gap between everyday customers and complex domain technicalities. If the data is mostly numbers than user may have problems coming to proper conclusions. It would also make the results boring and stale, which would cause loss of attention/interest from the customers. Visual graphs can retain attention and effectively communicate the data and analysis. User will be able to see their place amongst population groups regarding different parameters or conditions and be able to better analyze their own circumstances. They will see connections between different health parameters as well. This would ultimately lead to better decision making on the customer's end.

One things our app will focus on is gathering data on diabetes and physical parameters tied to the condition. Users will be able to enter data on BMI, Insulin Levels, Glucose Levels, and even Diabetes Pedigree Function. We should be able to provide users with information regarding their parameters compared to others in order to help them make decisions regarding their lives.

Our product uses the cutting edge technology of blockchain. The essence of blockchain is decentralized. Since, there is no central server to hack into, a number of users with strong computing power can act as super node to process the data. With a blockchain, anyone with network access can readily access a growing list of user data persisted permanently as long as the blockchain network stays up. However, the cost of performing computation and storing data in any decentralized blockchain comes with its own costs.

Blockchain data can be persisted by having participating nodes mining (submitting proof of work) and offering to perform computations. Replicating all transactions that have been made and constantly competing with other nodes to submit a block is expensive. Estimates show that storing 1 gb of data in a decentralized blockchain like Etherium costs $186,700 while a typical database costs $00.02 for an equivalent amount. Storage is only part of the costs as the blockchain we would like to utilize is Ethereum; a blockchain that can also perform arbitrary computation on nodes which brings additional costs. Along with this, blockchains are not optimized for mass data retrieval such as other SQL or NoSQL databases.

Ultimately, we don't believe blockchain to be a one stop solution for all purposes; rather, it is a powerful, customizable (in the context of Etherium), reliable and persistent solution for public data. Using blockchain technology and a group of microservices (centralized servers and databases) to augment the blockchain, we would like to create a tool to allow customers to truly gauge their health in the context of their peer population. We will ensure that public data is always persisted, not only for our use, but for the use of any other services that would like to use it.

1.b Glossary Terms

Smart Contracts- A smart contract is an immutable computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. Smart contracts can be made by uploading Ethereum Virtual Machine code to the Blockchain and can be invoked by any outside party via transactions.

Storage- Place where all the contract state variables reside. Every contract has its own storage and it is persistent between function calls.

Memory - This is used to hold temporary values. It is erased between (external) function calls.

Ethereum Virtual Machine - The Ethereum Virtual Machine (EVM) is a Turing complete virtual machine that allows anyone to execute arbitrary EVM Byte Code. Every Ethereum node runs on the EVM to maintain consensus across the blockchain.

Truffle Framework - Truffle is a development environment, a testing framework and a crypto asset pipeline in one for development of smart contracts in Solidity programming language.

Gas(Ethereum) is a unit that measures the amount of computational effort that it will take to execute certain operations.

Blockchain - A blockchain is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

Block - A block records some or all of the most recent transactions that have not yet entered at any prior blocks. In other words, it is a group of transaction at a particular time.

Block time- The block time is the time needed to generate the next block in the chain. It is essentially the time the blockchain miners need to find a solution to the block hash.

Hash - A hash is a function that converts an input of letters and numbers into an encrypted output of a fixed length.

Keccak256 computes the Ethereum-SHA-3 (Keccak-256) hash of the arguments passed into the function. This is how String comparison is done in Solidity. Input values are hashed to a uniform 32 bytes, and the resulting hash code of the strings are compared. It can also be used for pseudo-random number generation.

Solidity is known as a contract-based, high-level programming language. This platform has similar syntax to the scripting language of JavaScript. Solidity as a programming language is made to enhance the Ethereum Virtual Machine. Solidity is statically typed scripting language which does the process of verifying and enforcing the constraints at compile-time as opposed to run-time.

Nonce - A nonce is an abbreviation for "number only used once," which is a number added to an encrypted block in a blockchain that, when rehashed, meets the difficulty level restrictions.

The nonce is the number that blockchain miners are solving for. When the solution is found, the blockchain miners are offered cryptocurrency in exchange.

Ethereum network- A collection or a group consisting of several different nodes running an Ethereum client. Otherwise referred to as a supernode.

Node - A node is a participating computing system on a blockchain network. A powerful machine that is able to run blockchain software. The role of a node is to support the network by maintaining a copy of a blockchain and, in some cases, to process transactions.

2. System Requirements

2.a Enumerated Functional Requirements

Enumeration	Priority	Description
REQ-1	10	The system shall keep immutable data submitted by participants in a population and persist it forever.
REQ-2	9	The system shall keep user-data anonymous to anyone other than the user.
REQ-3	5	The system shall allow the user to use their own data as payment for descriptors of the entire population in general for the given data-type.
REQ-4	2	The system shall parse the data time-series data on behalf of the user and present it.
REQ-5	5	The system shall allow the user to upload user-data from any device with a web browser available.
REQ-6	2	The system shall allow the user to request for updated data. The system should periodically update data presented to the user by-itself without manual intervention.
REQ-7	1	The system shall allow participants to retrieve only their data via a popular transport format (CSV, JSON).
REQ-8	4	The system shall parse the data and present it in a useful visual format for the user to see where they stand in the population and detect trends (different forms of graphs).
REQ-9	8	The system shall provide ways to store identity information

		and persist it permanently.
REQ-10	6	The system shall provide ways to authenticate a given user for its population descriptors.
REQ-11	2	The system shall provide a service to provide third parties with authentication information given a known protocol (OAuth 2.0)
REQ-12	1	The system shall provide a way for simple UI for users to log in to this and other services (similar to how Google and Facebook allow OAuth logins)
REQ-13	7	The system shall allow any third party to access public data on the blockchain.
REQ-14	5	The system shall process the blockchain data and give useful statistical information to the user in addition to data visualization.
REQ-15	3	The system should try to rate the user's standing within the population positively or negatively and give users web links to more information on how to improve such factors.

2.b Enumerated Non-functional Requirements

Enumeration	Priority	Description
REQ-16	3	The system will log out users on the web client after prolonged inactivity
REQ-17	2	The amount of transactions by system required in a certain period of time (hours)
REQ-18	6	The amount of data the system needs to store must be malleable. The amount of storage needs to scale with new users.
REQ-19	7	A user can't log into another user's account through malicious activity.
REQ-20	4	User experience is enhanced by quick response from the system, gathering and uploading data to the database is fast, ui runs smoothly.

REQ-21	2	Should have an easy and intuitive UI for existing and new users.
REQ-22	3	User should be notified of failed login attempts.
REQ-23	5	Users can access the web client through all popular web browsers on computers and smartphones.

2.c User Interface Requirements

UI requirements	Priority	Description
REQ-24	10	**Graphical Visualization of Aggregate Population Data**. Users require the ability to view their standings within populations. Line graphs are a good means by which to provide that insight. Users will be able to view graphs of aggregate data, and view their position within graphs. Elements include: - Line Graph - Search Bar(for particular parameters) - Display Bar: displays values being hovered over in graph.
REQ-25	8	**Non-graphical data display**. Some data or data summaries, such as means or medians, are better displayed as text. Users will need ui to view data they wish to see in an organized manner. Data will be sorted according to parameters and UI will contain selection bar from which to select particular parameter to view. Elements include: - Search Bar(to search parameter of choice) - Drop-Down Menu(if user wishes to browse) - Text-Fields(to display particular data)
REQ-26	7	The system will contain **DATA ENTRY FORMS** to obtain personal population parameters from users. Will contain various input methods to obtain required data based on type of question. User's weight will require text entry while whether or not a

		user has a certain condition will require a selection.. Input methods include: - Text Entry Fields(to enter user data like weight or height) - Radio Buttons (for single choice question) - Multi-Select Buttons
REQ-27	4	**User Login Page.** Users will require input means to enter their login information in order to access services and create an account with the system. Elements include: - Text input field for username - Text input field for password(characters will be hidden) - Login Button
REQ-28	4	**UAuth page.** Users may need to login to other services provided by other parties, they can do this with blockchain id. To log into other sites they will require easy UI to obtain permission and login. Elements : - Button(will asks if user approves of login) - Button(if user doesn't approve) (Assumption here is that user's credibility is already confirmed so only approval is necessary, and no other info)
REQ-29	7	**User Log-Out Button** - Showing log-out button on certain screens once the user is logged into the system. - On user interaction with logged out button, system should go back to the login page.

Figure 2.1 - Data Entry Forms

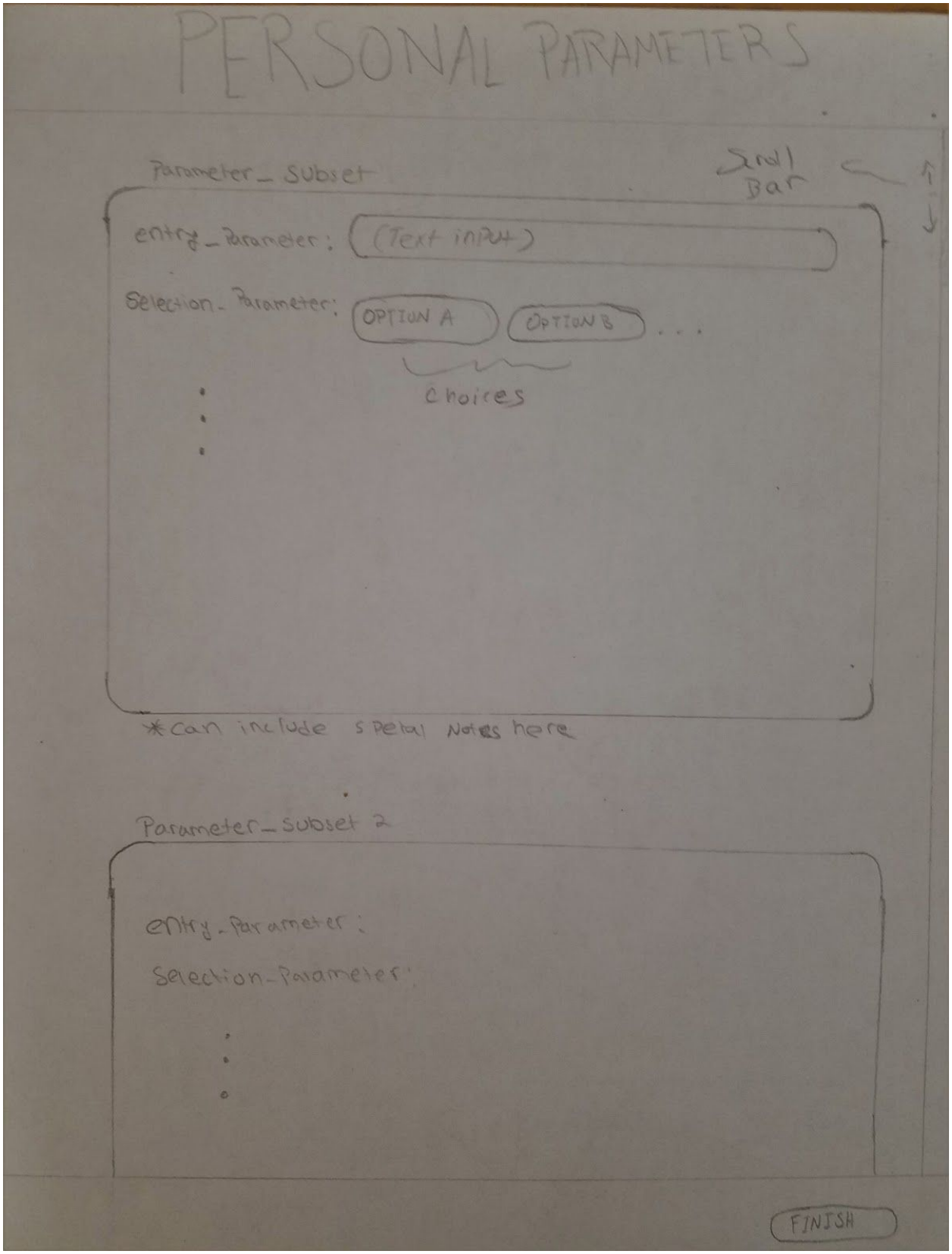


Figure 2.2 - Non-Visual Data Presentation

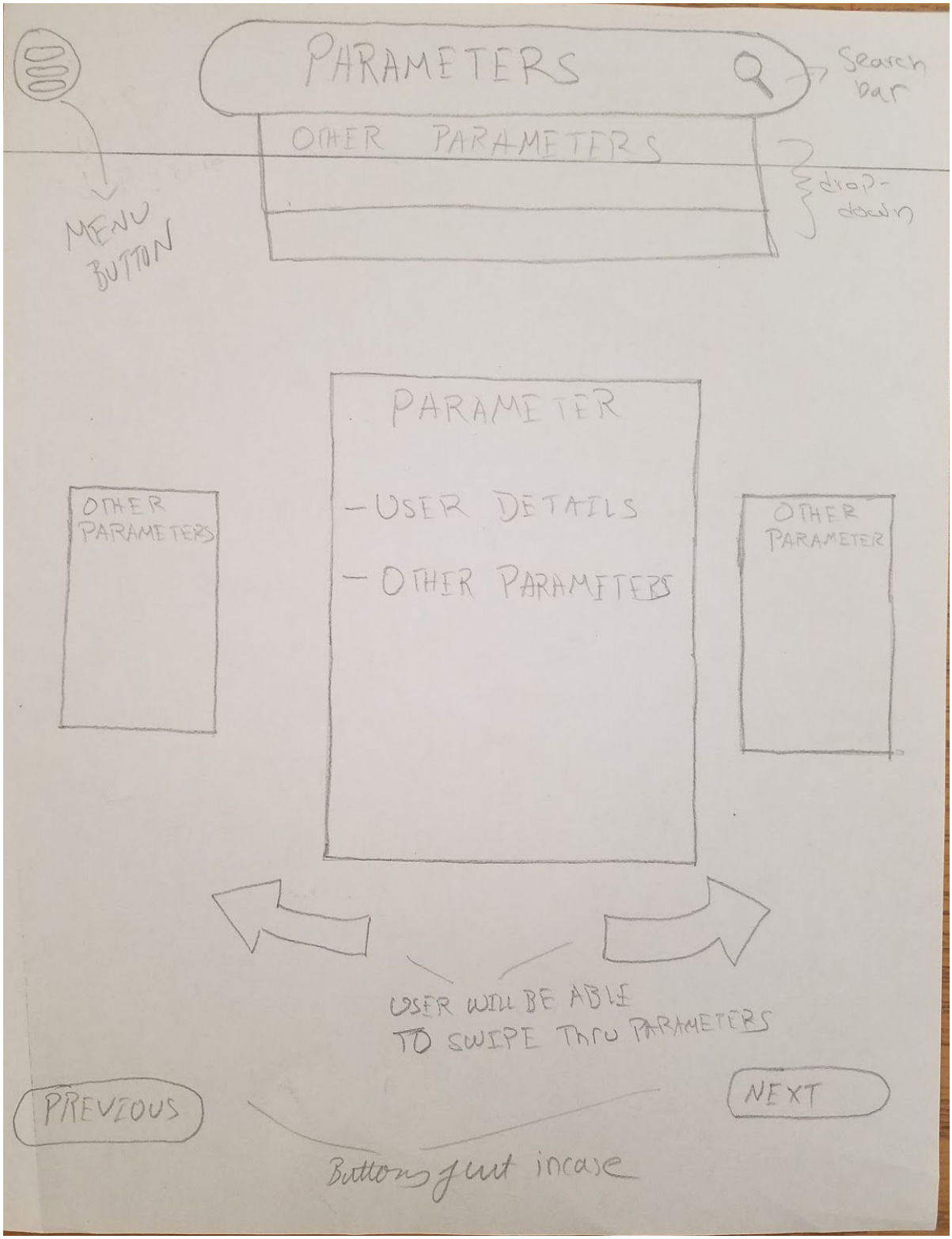


Figure 2.3 - Visual Data Presentation

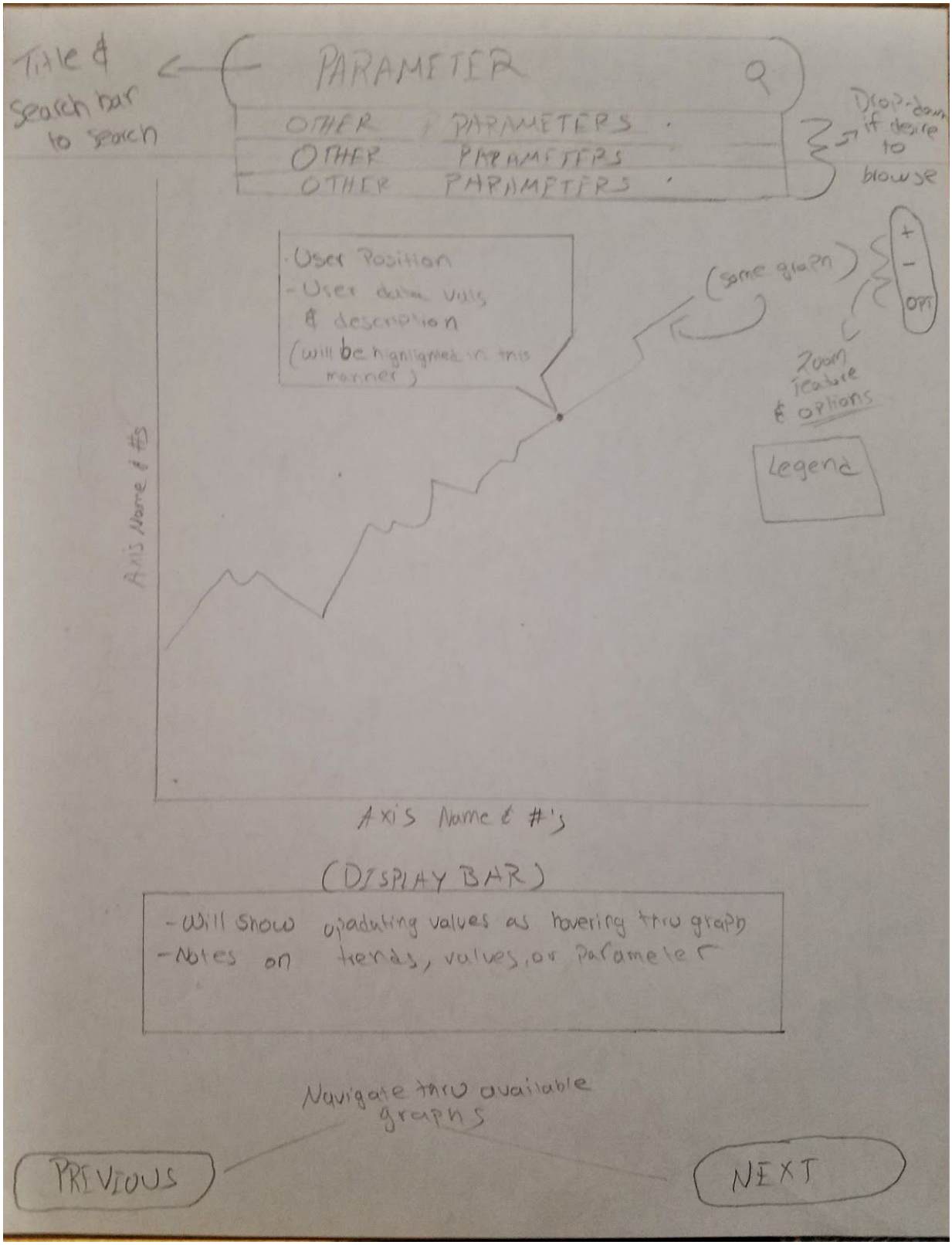


Figure 2.4 - User-Login Page

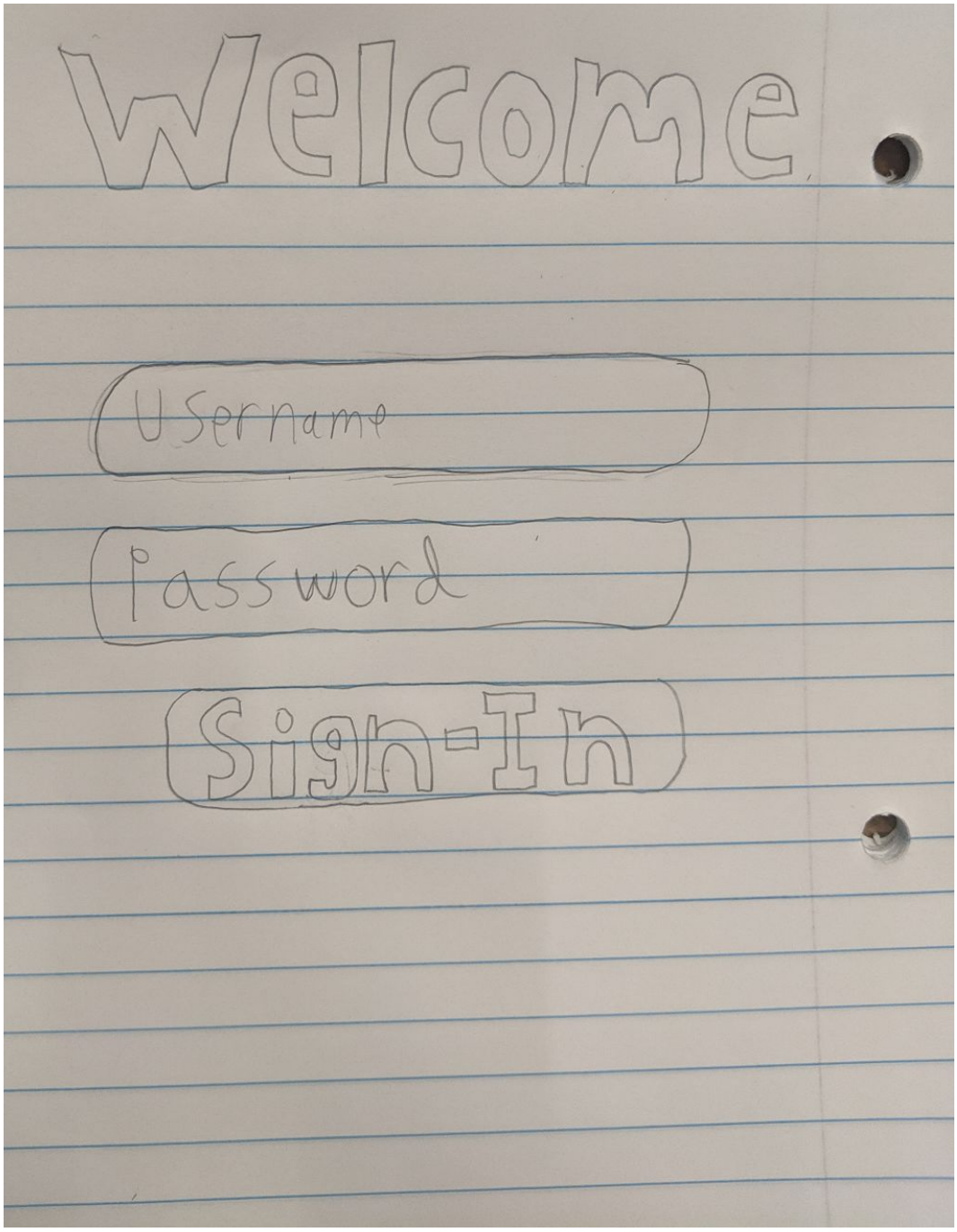
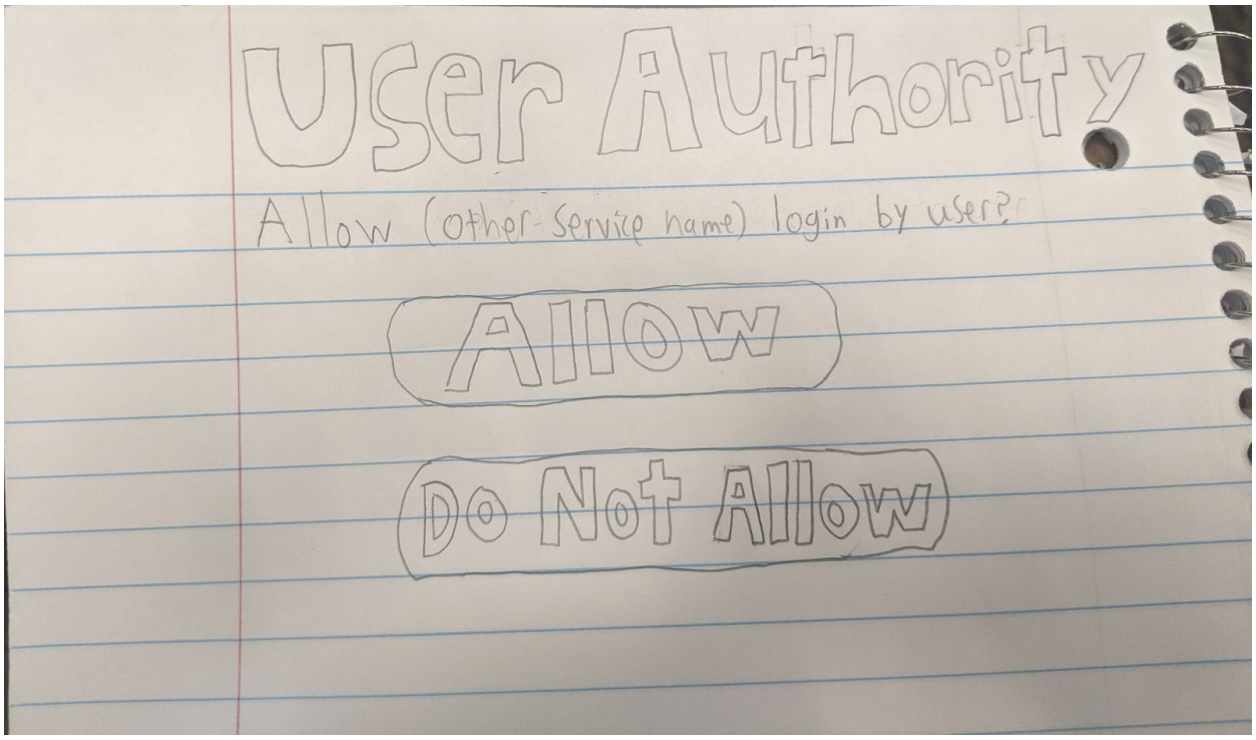


Figure 2.5 - OAuth Page



3. Functional Requirements Specification

3.a Stakeholders

I. Health App User - Has an interest in the system, because the user wants to monitor their own health status.
II. Blockchain - Is a ledger that records transactions, and, in this project, health data. The cryptography of the blockchain has to keep this data secure.
III. The medical service provider – Access to mass data analytics on secure data
IV. Third party services and products - Any party who integrates this service into their product for improved functionality and service
V. Researchers – Can view their data through the lens of our mass data to put their research in context
VI. Caregivers and healthcare facilities– Use the system to provide a fuller diagnosis and information relative to others
VII. Health Patients – Benefit from their caretaker giving contextualized diagnoses and information relative to others
VIII. Health Insurance Companies – Can use the data to shape their plans and services
IX. Government Agencies – Measure general public's health to inform policy

3.b Actors and Goals

Actors	**Goals**
User(Initiating)	To be able to access the webpage and be able to login into the system.
User(Initiating)	To be able to access information of their own profile and health data.
User(Initiating)	To be able to see and compare their current health standings relative to other users on a graph that maps users health data.
User(Initiating)	To be able to input their health information.
Non-User(Initiating)	Able to create an account with the system.

Site Administrator(Initiating)	Perform administrative work for the website, manage database,smart contracts,server.
SmartContracts (Participating)	computes statistics and health data based on user input.

3.c Use Cases

i. Casual Description

Use Case	Name	Description	Requirements
UC-1	login	The ability for one to access their account after authorization.	REQ-27,REQ-22
UC-2	Auth	The system shall provide a service to provide third parties with authentication information given a known protocol so that users can log in to other websites such as google and facebook.	REQ28,REQ-27,REQ -12, REQ-11
UC-3	InputData	Allows the user to input their health related information into the system	REQ3, REQ5,REQ-26, REQ-9,
UC-4	ReceiveDataForUser	The user will be able to see all their data at any given time.	REQ6, REQ7,REQ25
UC-5	CompareData	The system will compare data from the user to the population through various visual means. E.g. graphs, tables,	REQ-25, REQ-24, REQ-15, REQ-14, REQ-8, REQ-4

		etcetera	
UC-6	PublicAccess	Allow third party User to view publicly accessible data.	REQ-23, REQ-13
UC-7	DisplayVisualAnalytics	The system shall display all data in a graphical/visual format.	REQ4, REQ8, REQ-24, REQ-25, REQ-21
UC-8	LogoutUser	The system logs out the user after prolonged inactivity or if the user requests to be logged off,	REQ29, REQ-16
UC- 9	Register(Account Creation)	Allows non-user to create an account with the system to use the services.	REQ-27
UC-10	Data Administration	Allows site administrators to manage user data stored on blockchain like data request,persisting,output and write efficient smart contracts.	REQ-14,REQ-1 REQ-4
UC-11	User Notification	Notifies user on incorrect data input such as user name and password	REQ-22

ii. Use Case Diagram

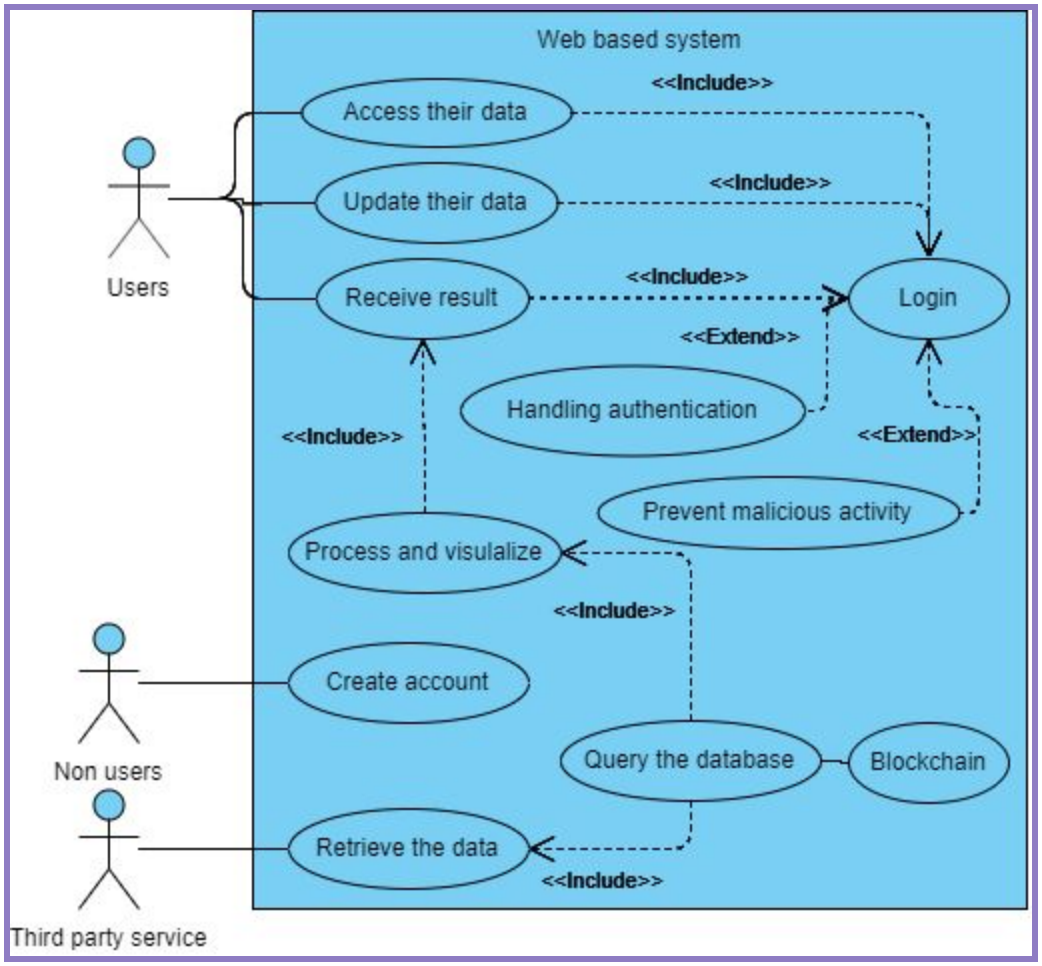


Figure 3c

iii. Traceability Matrix

Reqs	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9
REQ1	10	X		X						
REQ2	9	X								
REQ3	5			X						
REQ4	2					X		X		
REQ5	5	X		X						
REQ6	2				X					
REQ7	1				X					

REQ8	4					X		X		
REQ9	8	X		X						
REQ10	6									
REQ11	2		X							
REQ12	1		X							
REQ13	7						X			
REQ14	5					X				
REQ15	3					X				
REQ16	3								X	
REQ17	2									
REQ18	6			X						X
REQ19	7	X								
REQ20	4	X	X	X	X	X	X	X	X	X
REQ21	2	X		X	X	X				X
REQ22	3	X								
REQ23	5						X			
REQ24	10					X		X		
REQ25	8				X	X		X		
REQ26	7			X						
REQ27	4	X	X							X
REQ28	4		X							
REQ29	7								X	
MaxPW		10	4	10	8	10	7	10	7	6
TotalPW		52	15	47	17	38	16	28	14	16

iv. Fully-Dressed Description

Use Case: UC-3	InputData
Requirements	REQ3, REQ5,REQ-26, REQ-9, REQ-5, REQ-3
Initiating Actor:	User
Actor's Goals:	To access the system and put in health information, log in information.
Preconditions:	The user has access to the internet to interact with the system. The system asks for and has ways to fill out all the necessary information
Postconditions:	The information will be stored in the system to be used by the blockchain. The user can view their information.

Flow of Events:

1. <- System asks for identification in the form of a login
2. -> User supplies the right identification information.
3. <- The system prompts for information to be filled out.
4. ->User fills in the necessary information they want to fill out.

Extensions

1a User chooses to create an account because they're a new user.
-> User click on the create new account button
<- System takes the user to the page to create new account

1b Login failed
<- System notifies user of failed attempt
<- System allows user to try to login again.

Use Case: UC-5	CompareData
Requirements	REQ-25, REQ-24, REQ-15, REQ-14, REQ-8, REQ-4
Initiating Actor:	User
Actor's Goals:	To parse through the block chain data and give back useful information comparing the user's data to the population data.
Preconditions:	The user is logged in to the system.

Postconditions:	The system will display the users data vs the populations data to the user in various visual medium; only the *type*

Flow of Events:

1. <- The user asks system to show population data
2. <- The system parses and analyzes the population's data.
3. <- The system creates useful visual mediums to display
4. <- The system makes comparisons and analyzes the user's data vs the population's data.
5. <-The system display the user's standing compared to the population
6. <-The system provides web links based off their standings to further improve the user's health

Use Case: UC-4	ReceiveDataForUser
Requirements	REQ6, REQ7,REQ25
Initiating Actor	User
Actor's Goals	User will initiate an action to receive his or her personal data from the blockchain and be presented with visual analytics of his/her data.
Preconditions:	The user is logged in to the system
Postconditions:	The user will receive processed data from the blockchain and will have graphical representations of his or her personal data.

Flow of Events:

1. <- The user asks to show user data
2. <- The frontend asks a proxy service to gather data from the blockchain using a user identifier
3. <- Blockchain smart contract is invoked to retrieve paginated data
4. -> Data is returned to proxy service to process and format for different visualization data charts.
5. -> Data is returned to the frontend to be presented to the user

Use Case: UC-1	Login
Requirements	REQ 22, REQ 27
Initiating Actor	User
Actor's Goals	User will access his or her account.
Preconditions:	The user has been authorized to access the account entered.
Postconditions:	The user will now be able to modify and view his or her data.

Flow of Events:

1. -> System prompts user for the username and password.
2. <- User enters username and password.
3. -> System looks through database to find the account.
4. -> If account exists, system authorizes the user.

Alternate Flow of Events

5. -> System prompts user for the username and password.
6. <- User enters username and password.
7. -> System looks through database to find the account.
8. -> If account does not exist, system prompts for the correct information.
9. <- User enters information again.

3.d System Sequence Diagrams

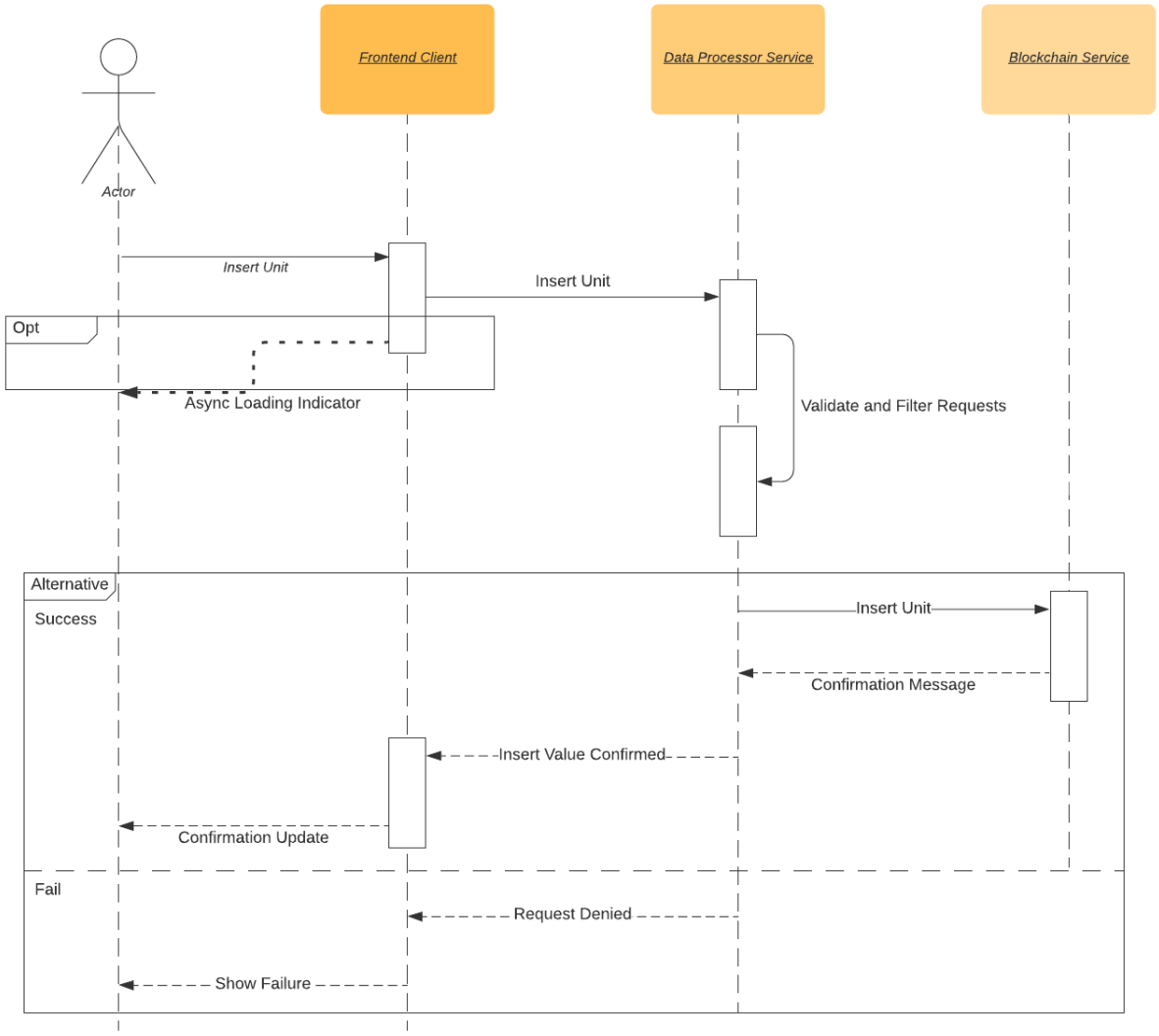


Figure 3.c.1 UC-3

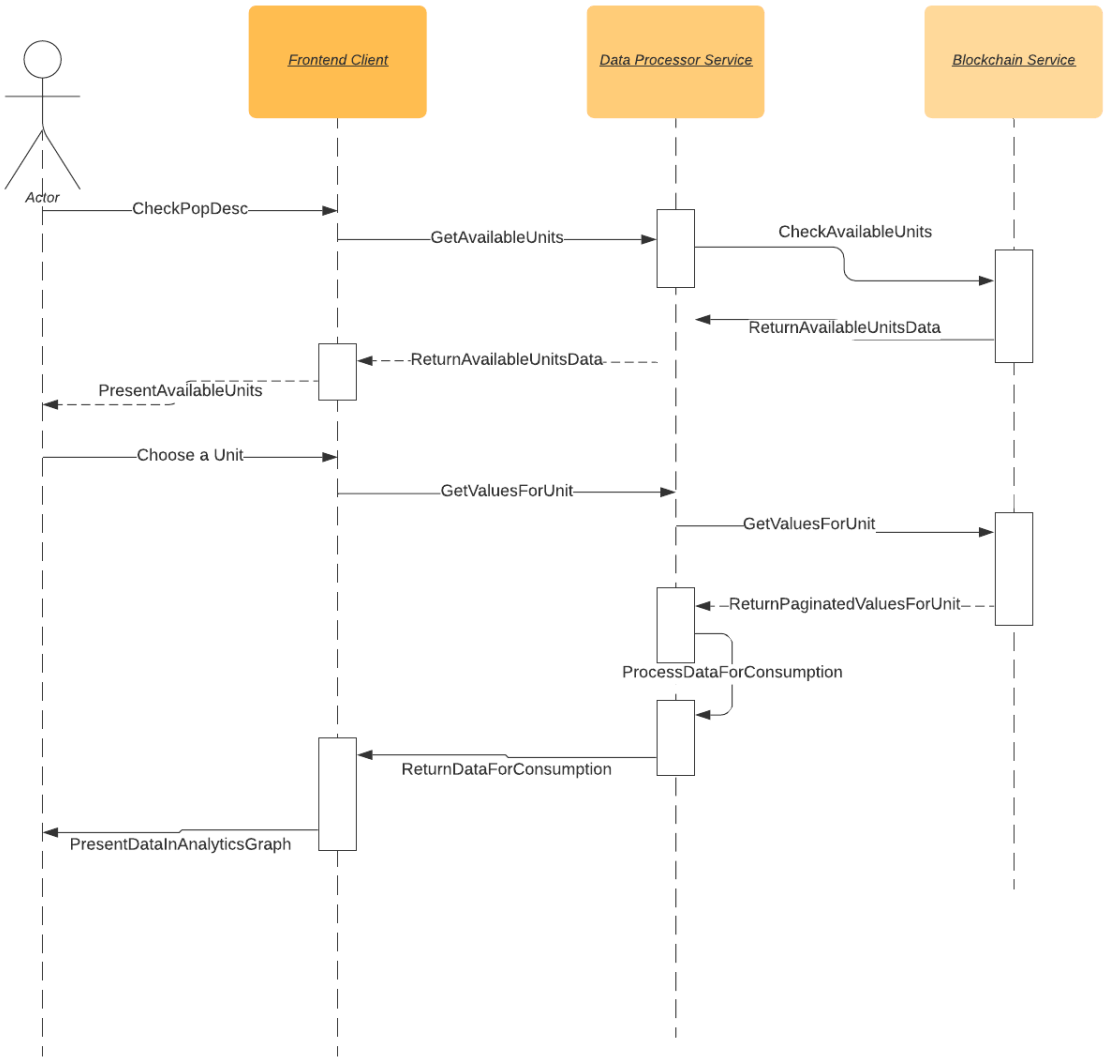


Figure 3.c.2 - UC-5

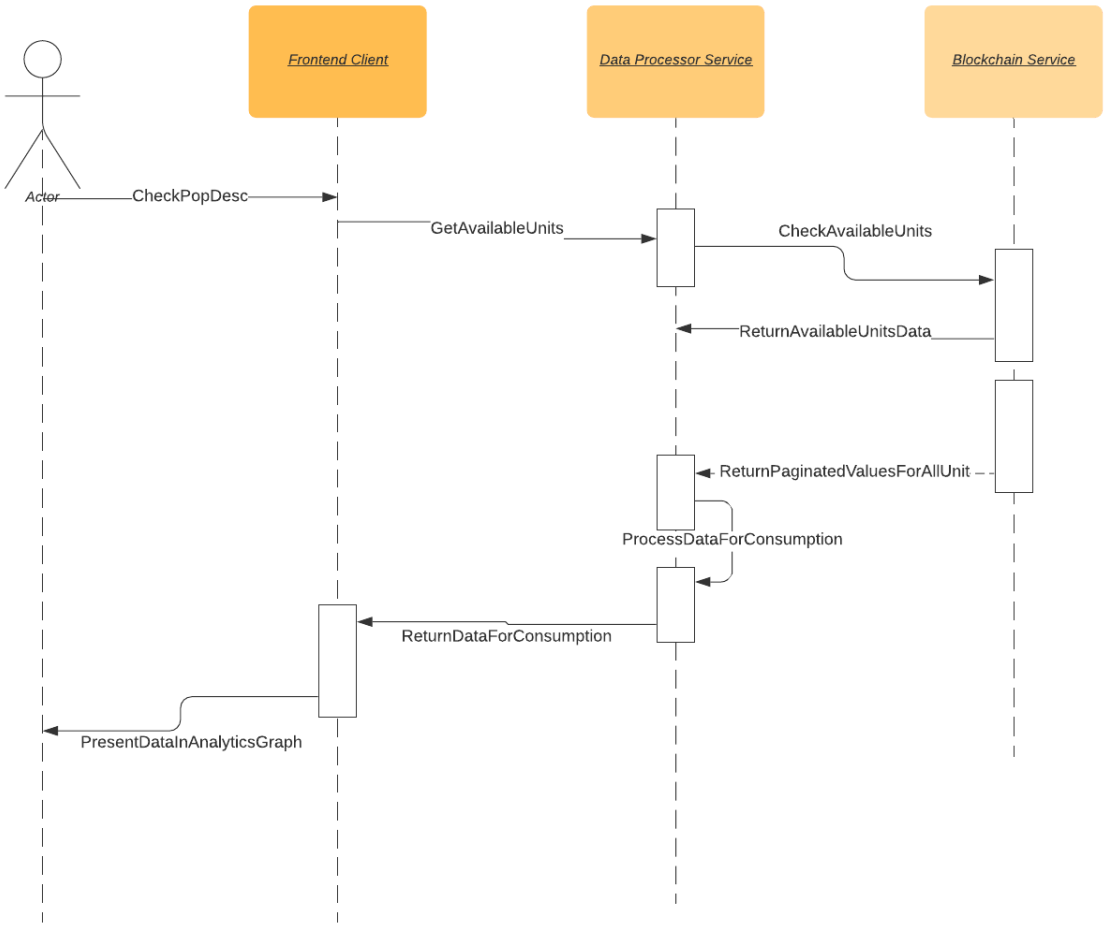


Figure 3.c.3 UC-4

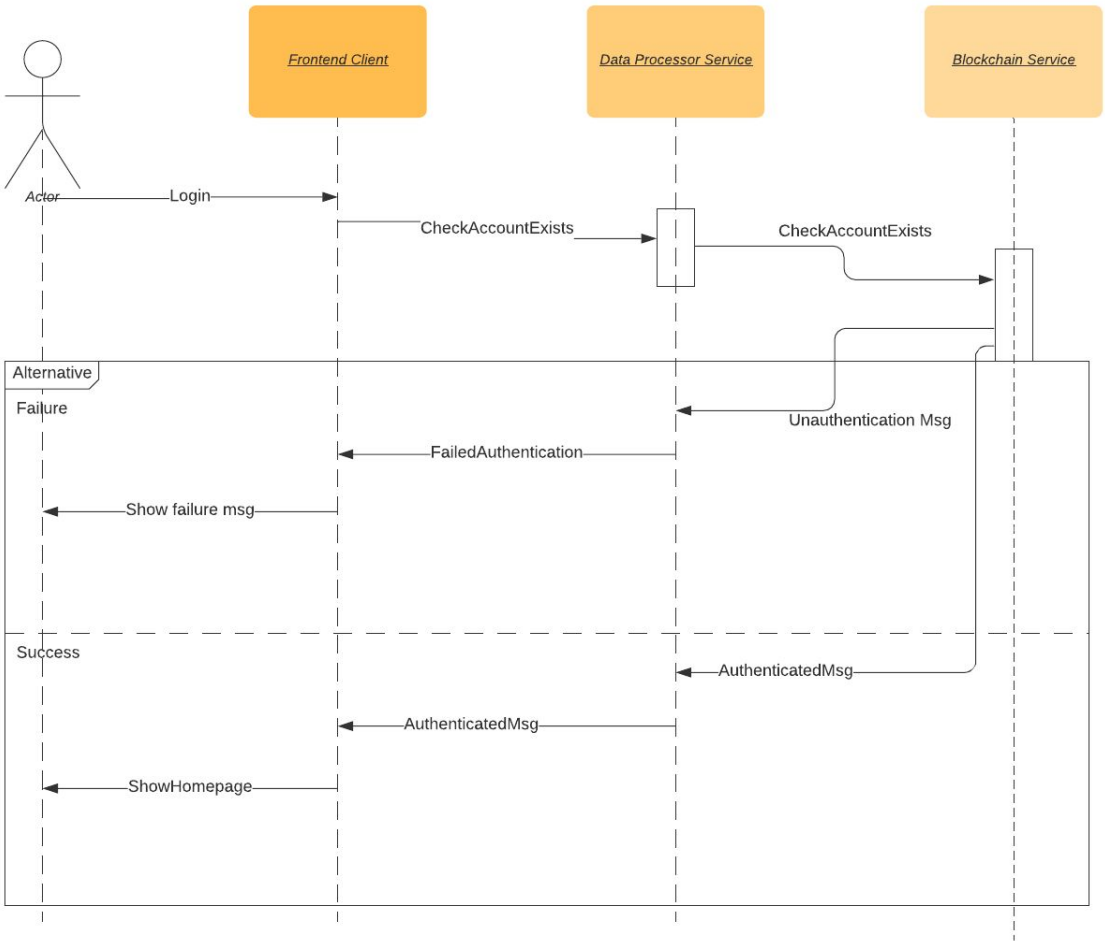
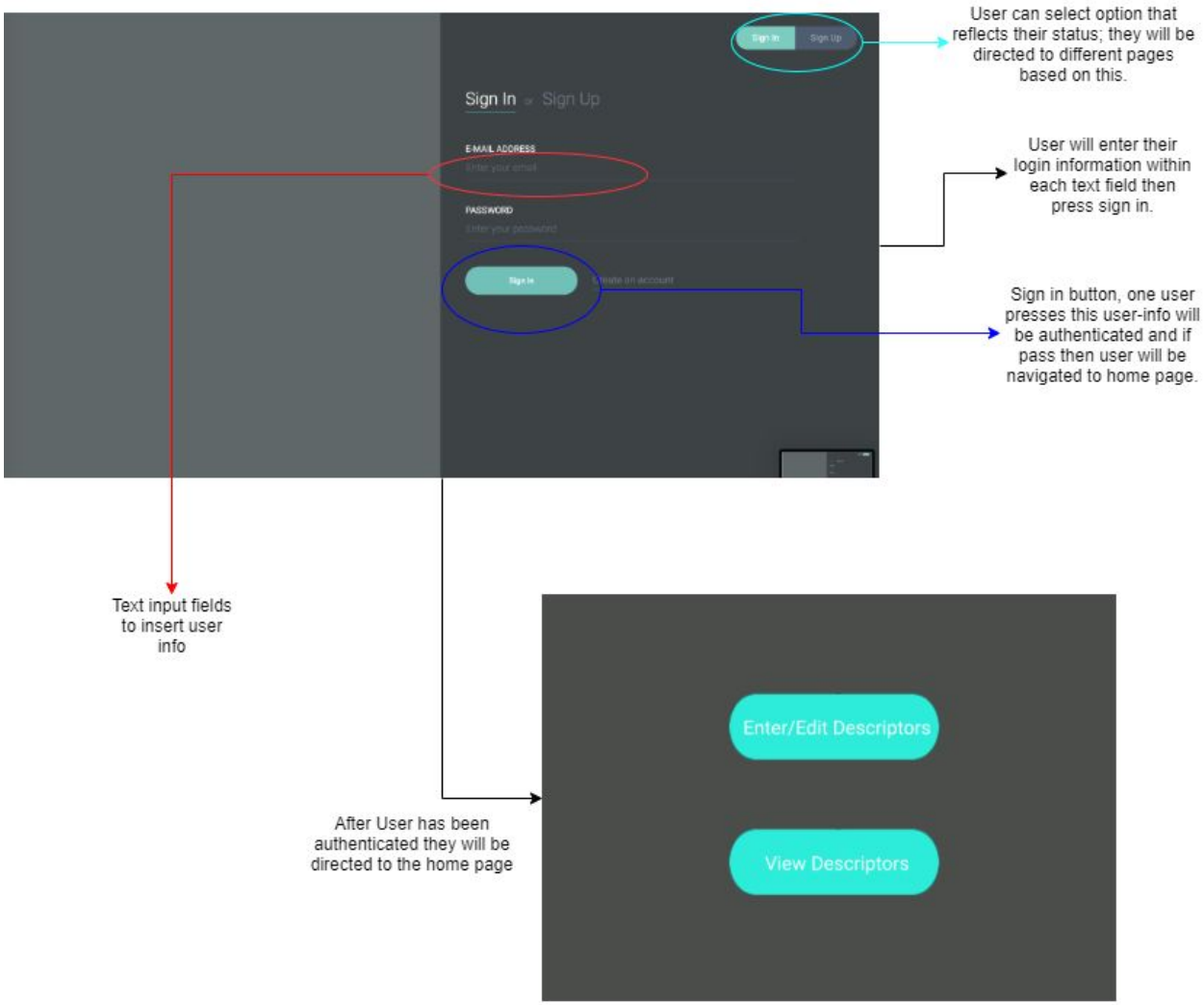


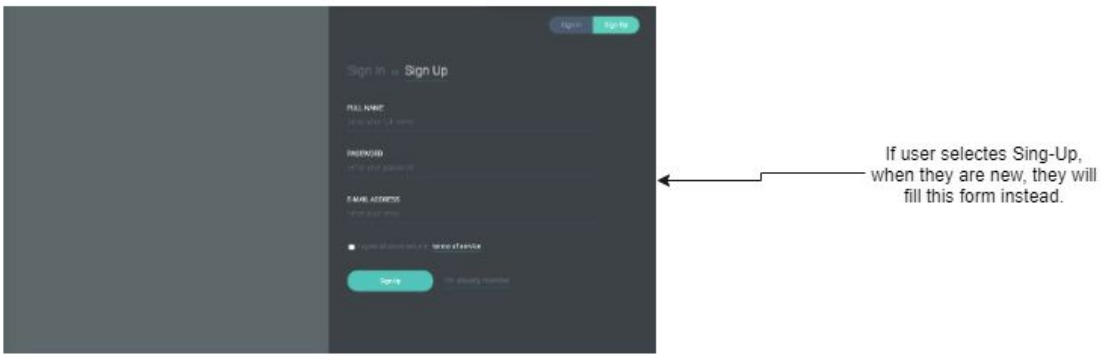
Figure 3.c.4 UC-1

4 User Interface Specification

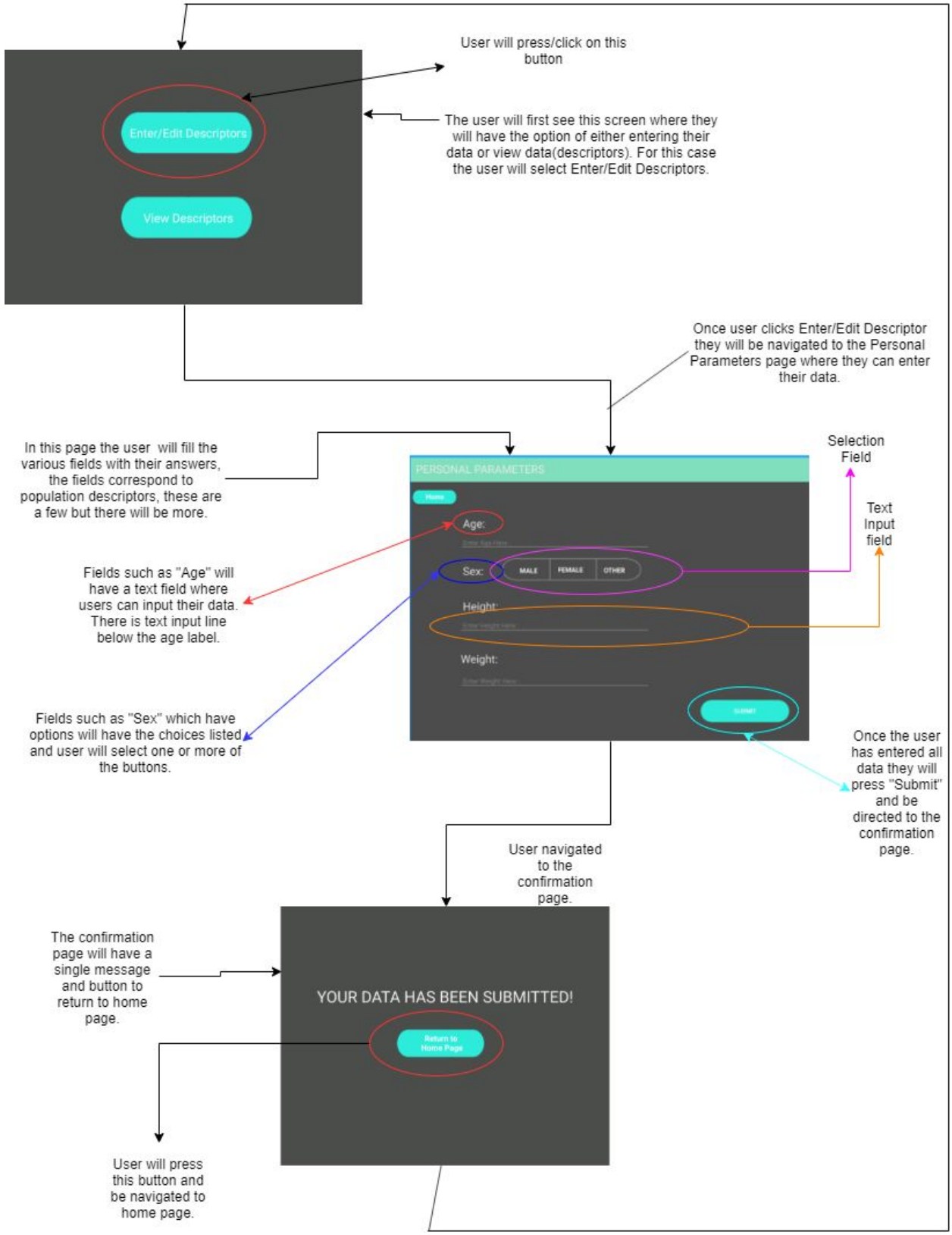
4.a Preliminary Design

UC-1 Login:

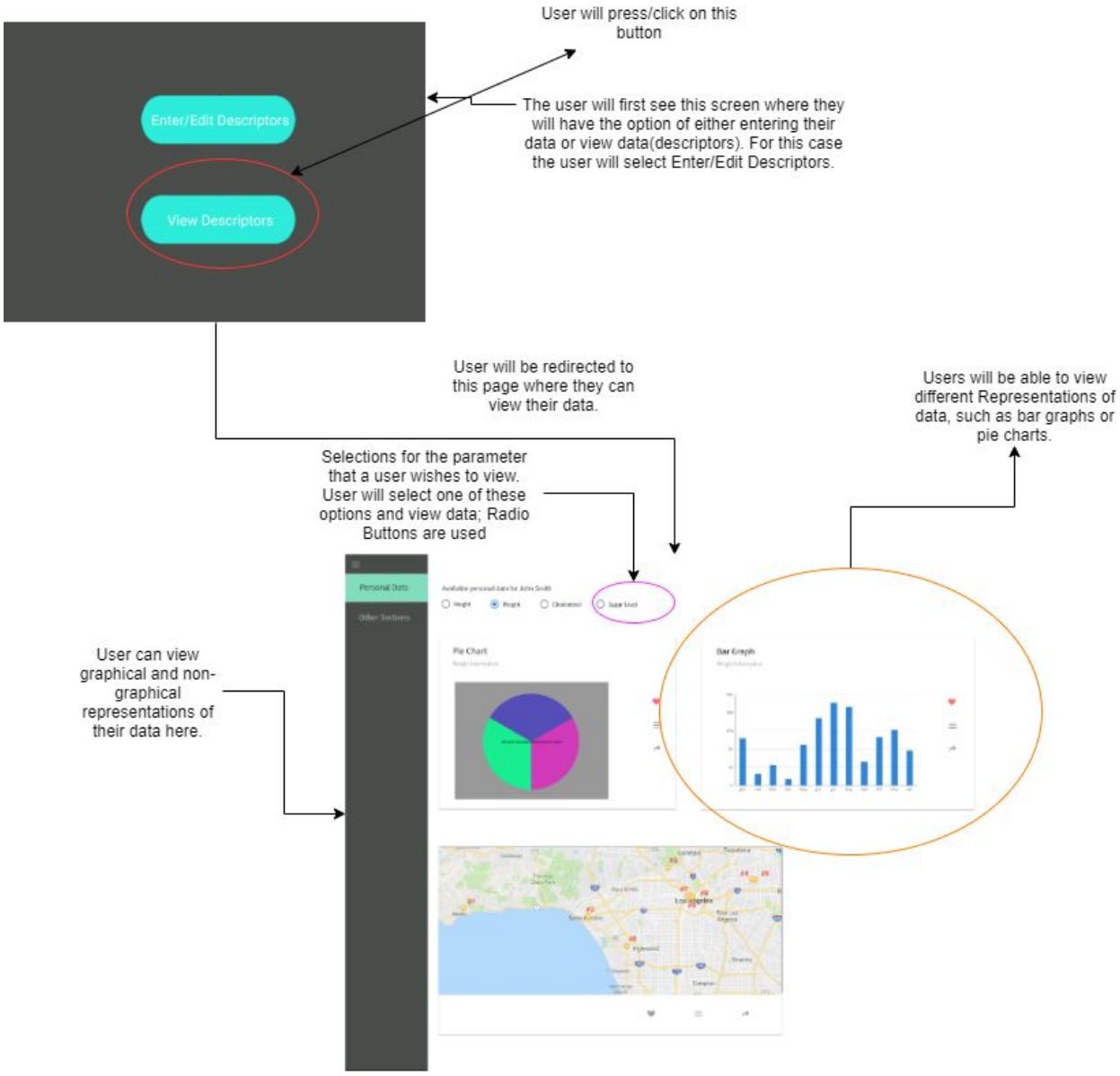

Sign In
Sign Up
E-MAIL ADDRESS
PASSWORD
User can select option that reflects their status; they will be directed to different pages based on this.
User will enter their login information within each text field then press sign in.
Sign in button, one user presses this user-info will be authenticated and if pass then user will be navigated to home page.
Text input fields to insert user info
After User has been authenticated they will be directed to the home page
Enter/Edit Descriptors
View Descriptors


Sign In
Sign Up
If user selectes Sing-Up, when they are new, they will fill this form instead.

UC-3: User Data Input:



UC-4 - RecieveDataForUser



4.b User Effort Estimation

UC-1 Login

1. Navigation
   A: Click Sign in at top.
   B: Click Sign in.
2. Data Entry
   A: Click on enter your email
   B: Type in your email. This depends on the user. Roughly 6-20 keystrokes
   C: Click Enter your password.
   D: Type in your password. This depends on the user. Roughly 6-20 keystrokes

Create an Account.

1. Navigation
   A: Click Sign up at the top to create an account.
   B: Click Sign up.

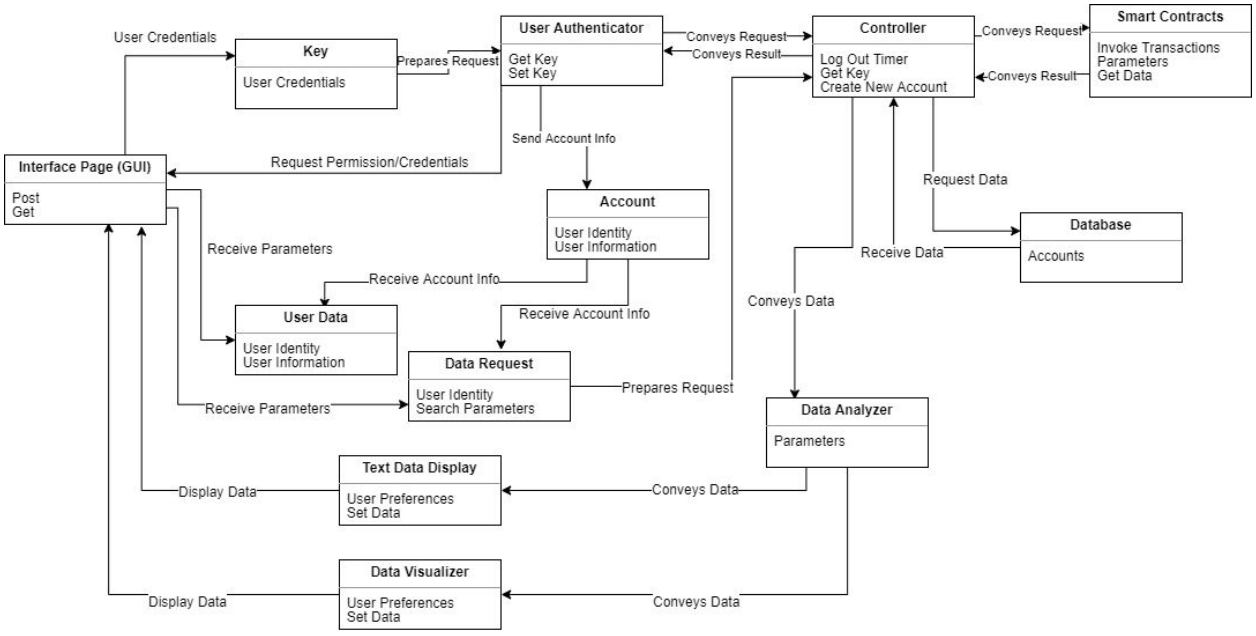
2. Data Entry
   A: Click Enter your full name.
   B: Type in your name. This depends on the user. Roughly 6-20 keystrokes
   C: Click Enter your password.
   D: Type in your password. This depends on the user. Roughly 6-20 keystrokes
   E: Click Enter your email.
   F: Type in your email. This depends on the user. Roughly 6-20 keystrokes
   G: Click on white box next to I agree to all statements.

UC-4 ReceiveDataForUser

1. Click on Personal Data tab on left side
2. Click on available units (Everything else will autofill)

5. Domain Analysis

5.a Domain Model



Domain Model Diagram

i. Concept definitions

Responsibility Description	Type	Concept Name
Website with React pages for account log in, account creation, and viewing outcomes and results in UI made from Scalable vector graphics.	K	User Interface (GUI)
Form specifying the parameters for data retrieval from the blockchain, as well as parameters for desired data analysis.	K	Data Request
Data input form where the user enters their health information	K	User Data

Render non-graphical data and summaries in an organized way for user-requested data	D	Text Data Display
Create data visualizations for the user-requested data	D	Data Visualizer
Establishes a connection to the Ethereum Blockchain. Accepts data requests and user data, and returns the raw data	D	Smart Contract
Analyses the raw data for the requested measurements	D	Data Analyzer
Container for user's authentication data (individual and third-party users)	K	Key
Verifies that a user with appropriate credentials exists. If not, inform the user and proceeds accordingly. Obtain permission for third-party login.	D	User Authenticator
Coordinate actions of system concepts and user requests. Responsible for data retrieval and transfer to concepts. Refresh data periodically and log out users after prolonged time.	D	Controller
Holds account information of a specific user and provides complete flexibility in managing users own data.	K	Account
Stores account data, user data and collaborates in all activities related to data visualization, administration and storage.	K,D	Database

ii. Association definitions

Concept Pair	Association Description	Association Name
User-Interface (UI) <-> Key	User enters their login information or new user information on the UI	User Credentials

Key <-> Authenticator	The authenticator takes the user's information and prepares verification request, which is sent to the controller.	Prepares Request
Authenticator <-> Controller	(1) Controller receives verification requests, used to invoke the appropriate smart contract. (2) Controller informs authenticator of successful login	(1) Conveys Request (2) Conveys result
Controller <-> Data Request	Controller receives a request for data. It prepares a formal data request	Prepares Request
Controller <-> User Data	Controller receives user data. It prepares a formal data upload request	Prepares Request
Controller <-> Smart Contracts	(1) Controller generates a request to invoke the appropriate smart contract for data retrieval (2) Controller receives raw data from the blockchain	(1) Generates Request (2) Receive Data
Controller <-> Data Analyzer	Controller passes raw data to the data analyzer	Conveys Data
Analyzer <-> Data Visualizer	Analyzer passed processed data to be visualized	Conveys Data
Authenticator <-> User-Interface	(1) Authenticator requests permission for third party log in (2) Authenticator requests valid user credentials	(1) Request Permission (2) Request Credentials

Analyzer <-> Text Data Display	Analyzer passed processed text data to be displayed	Conveys Data
Data Request <-> User-Interface	User enters parameters for data request, which is then contained in a form	Receive Parameters
User Data <-> User-Interface	User enters request for personal data, given the parameters (e.g. user id)	Receive Parameters
Text Data Display <-> User-Interface	Display the non-graphical data in an organized way	Display Data
Data Visualizer <-> User-Interface	Display graphical data	Display Data
Controller <-> Database	(1) Controller generates a data retrieval/upload request (2) Controller receives data from the database	(1) Request Data (2) Receive Data
Authenticator <-> Account	Once authenticated, user account data is stored in Account concept (from database)	Send Account Info
Account <-> Data Request	Data request receives account information to be passed along with the request	Receive Account Info
Account <-> User Data	User Data form receives account information to be passed along with the user data	Receive Account Info

iii. Attribute definitions

Concept	Attributes	Attribute Description
Data Visualizer	Set data	This is setter for the system where the data is converted into visual form to fetch to GUI
	Set user's preference	This allows the data visualizer to be customized to each user's taste
Smart Contracts (Ethereum network)	Invoke a transaction	The server will the blockchain to add a new node in the common chain which typical occur when there is a new user or old user update their data
	Add parameters	This adds flexibility to the system as the administrator adjust the system based on the user's preference
	Get Data	An entry or interface between the server and the blockchain to extract the data for the user
Authenticators	Get key	This is for the getter and setter for the front and back to authenticate and distinguish between the user and third party
	Set key	
Controller (server)	Create new account	This shall allow the new users to register to the service
	Log out timer	Used to trigger automatic logout after idle time
	Get key	Together with the Authenticator forms the login mechanism
GUI (or web page)	Post	This is the primary way that the front interacts with the server and the Ethereum network
	Get	

Key	Credentials	User's identification information, such as user ID and password
User Data	User Identity	User's identification information
	User Information	User's health information to be uploaded
Data Request	User Identity	User's identification information
	Search Parameters	Parameters indicate the wanted data/measurements
Text Data Display	User Preferences	Customization parameters for results
	Set Data	This is setter for the system where the data is converted into organized text form to fetch to GUI
Data Analyzer	Parameters	Parameters indicate the wanted data/measurements
Database	Accounts	Record of existing accounts
Account	User Identity	User's identification information
	User Information	User's health information

Iv: Traceability Matrix

Traceability Matrix

	Use Cases									
Domain Concepts	Login	Logout	Display	Compare Data	Authorization	Register	Public Access	Input	Data Administration	Notify
Controller	X	X	X	X	X	X	X	X	X	X
GUI	X	X	X		X	X	X	X		X
Key	X				X	X			X	
Smart Contract				X					X	
Authenticator	X		X							X
User Data				X				X	X	
Data Request				X				X	X	
Text Data Display			X	X					X	
Account	X	X			X	X		X		
Data Analyzer				X					X	
Database	X	X	X	X	X	X	X	X	X	X
Data Visualization			X	X					X	

5.b System Operation Contracts

Name:	Login
Responsibilities:	To have the user access his or her account.
Use Case:	1
Exception:	Password is wrong, or user does not have an account.
Precondition:	User has an account. System prompts user for both username and password.
Postcondition:	User is now logged in and able to use and modify the account.

Name:	InputData
Responsibilities:	To have the user put in health information.
Use Case:	UC-3
Exception:	User does not have an account. User does

	not put in their correct information.
Precondition:	The user has access to the internet to interact with the system. The system asks for and has ways to fill out all the necessary information
Postcondition:	The information will be stored in the system to be used by the blockchain. The user can view their information.

Name:	ReceiveDataForUser
Responsibilities:	System will visually display to the user his or her data, when the user prompts the system for data.
Use Case:	4
Exception:	None
Precondition:	User has logged in/been authorized.
Postcondition:	User’s health data will be displayed graphical representations.

Name:	CompareData
Responsibilities:	System will retrieve health data from the blockchain and compare this to that of the user.
Use Case:	5
Exception:	None?
Precondition:	User has logged in.
Postcondition:	The app/system will display the user’s health data compared to the overall population’s health data.

5.c Mathematical Model

A large amount of population data will be collected on our blockchain of public data. Our job will be to present this data to the user via basic statistical methods.
For example, we will present different variables of data over a period of time via line graphs and show changes in slope over periods of time. For personal data, we can show change of physical characteristics over a period of time. The following graphs are samples given by the open source Apache ECharts library.

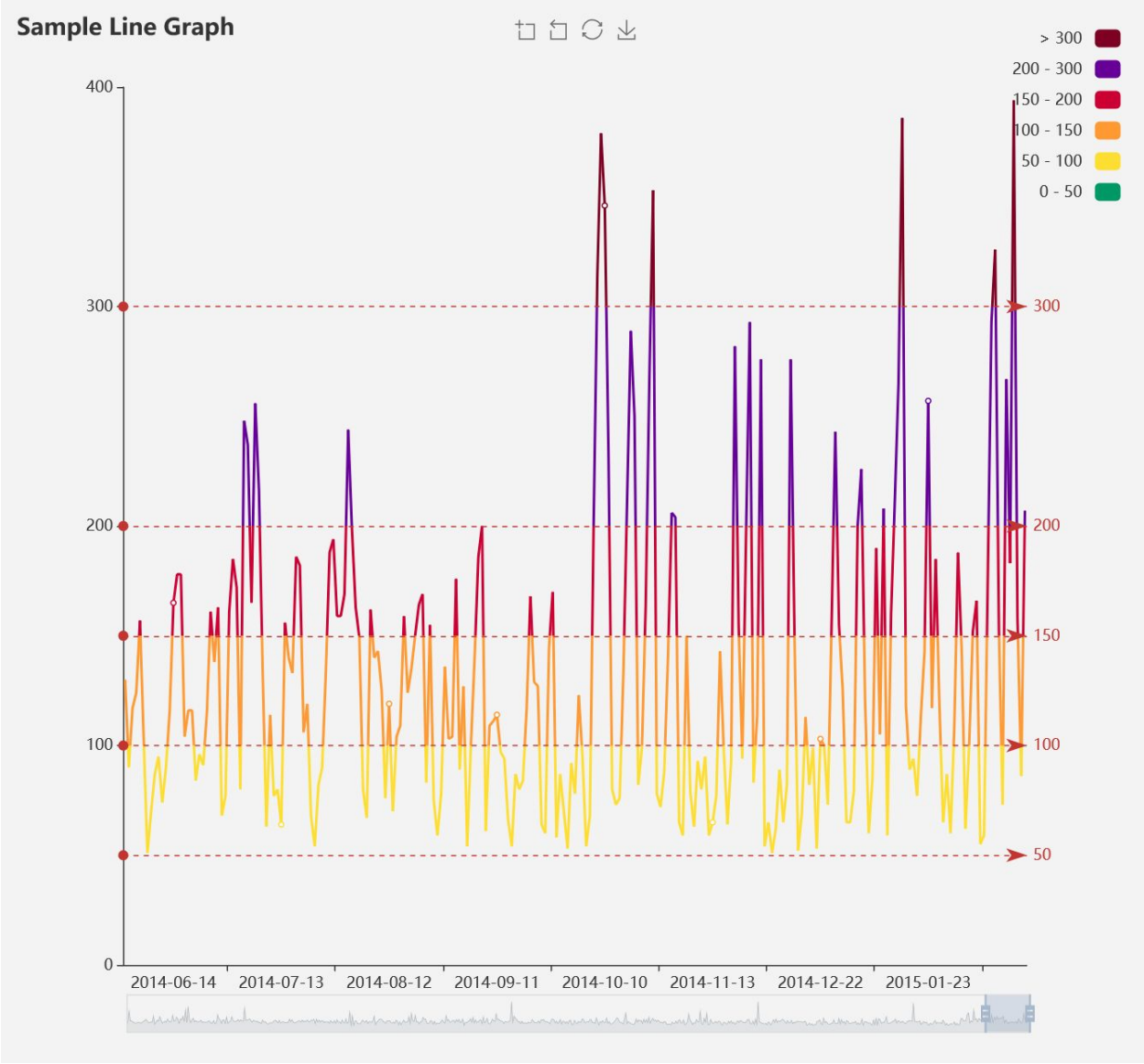


Figure 5.c.1

We can also present bar graphs whilst displaying standard deviations, averages, modes, and medians on those graphs.

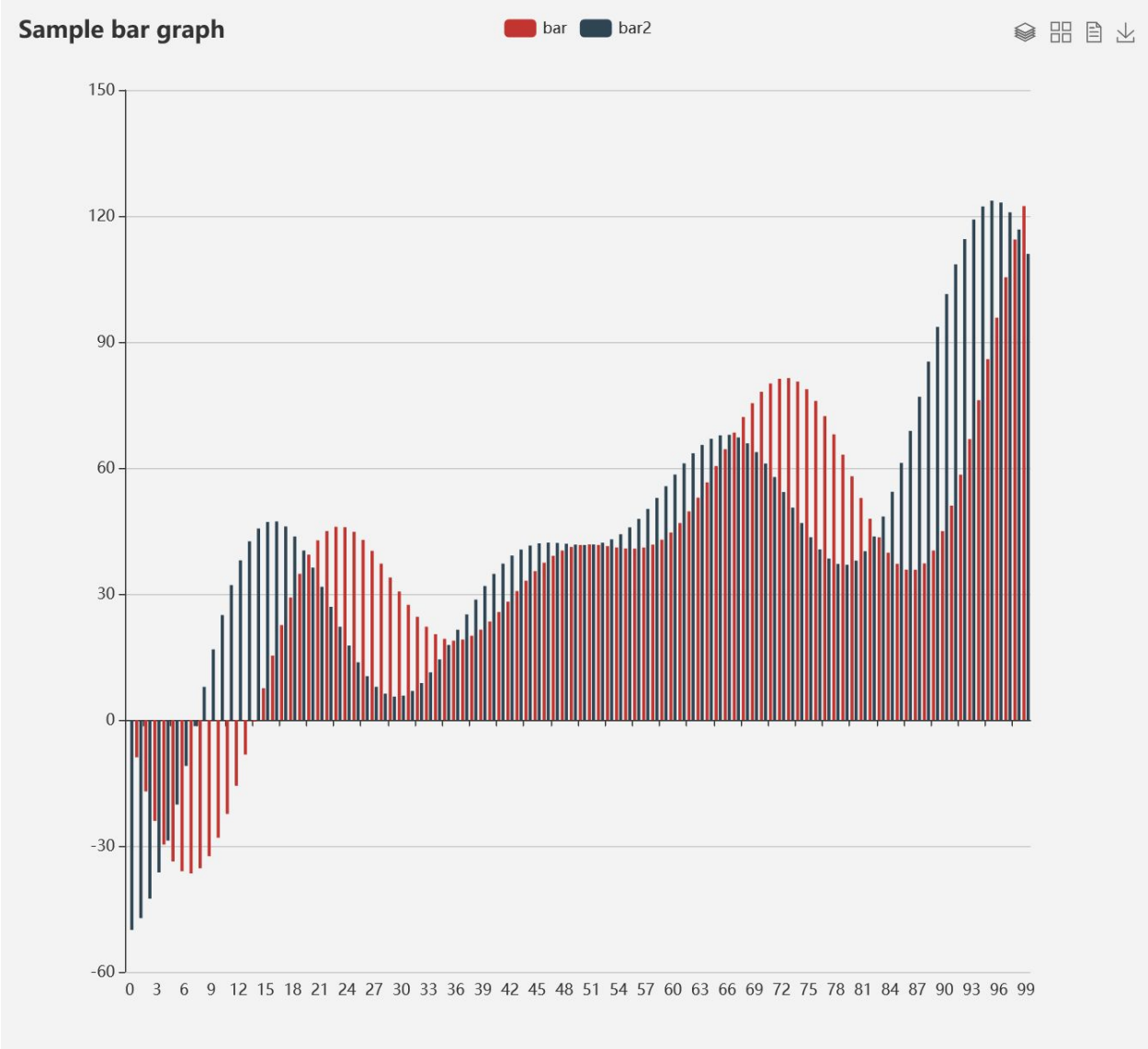


Figure 5.c.2

We may also choose to record the location of the user to show frequency of a certain characteristic in a geographic location.

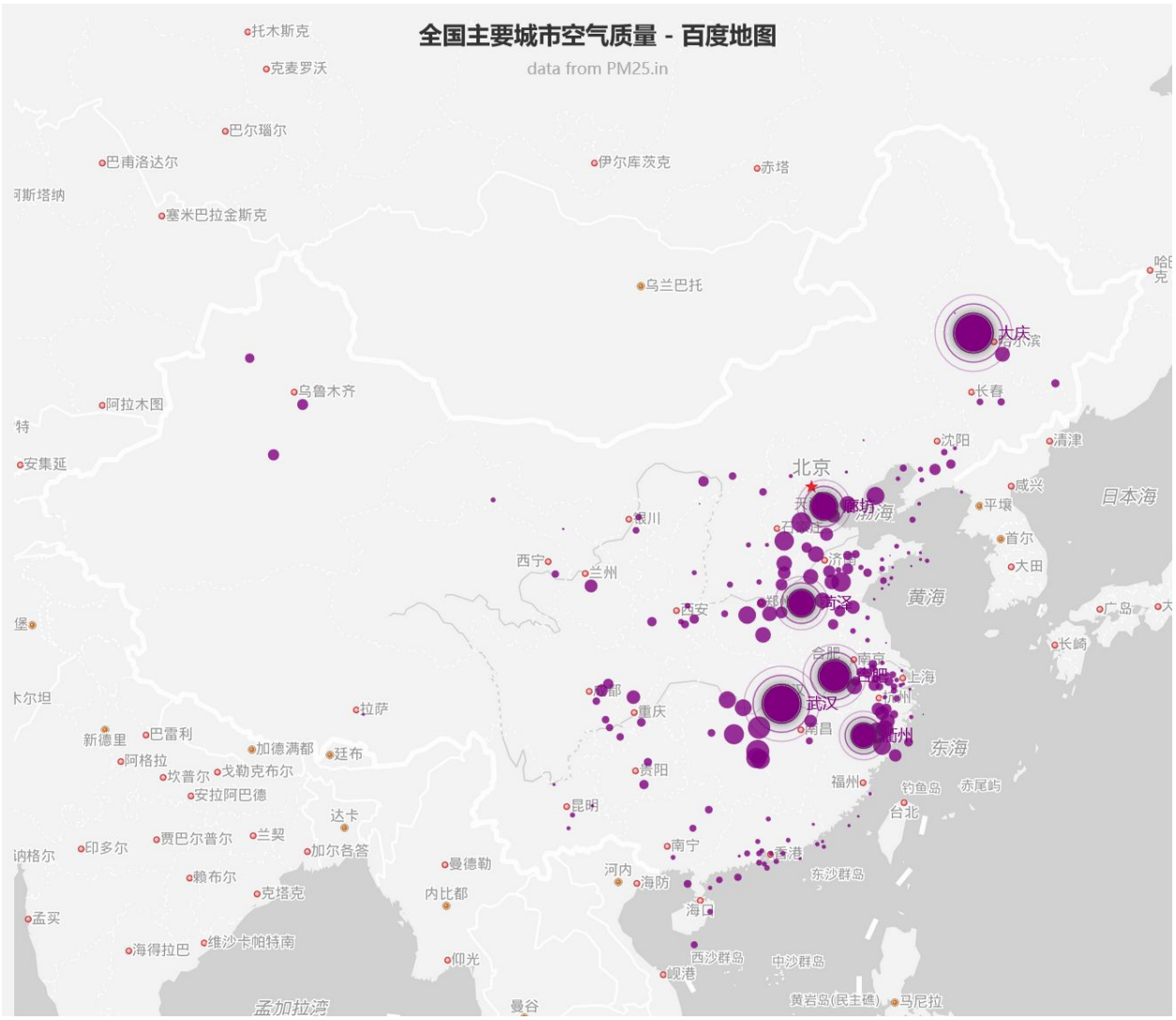


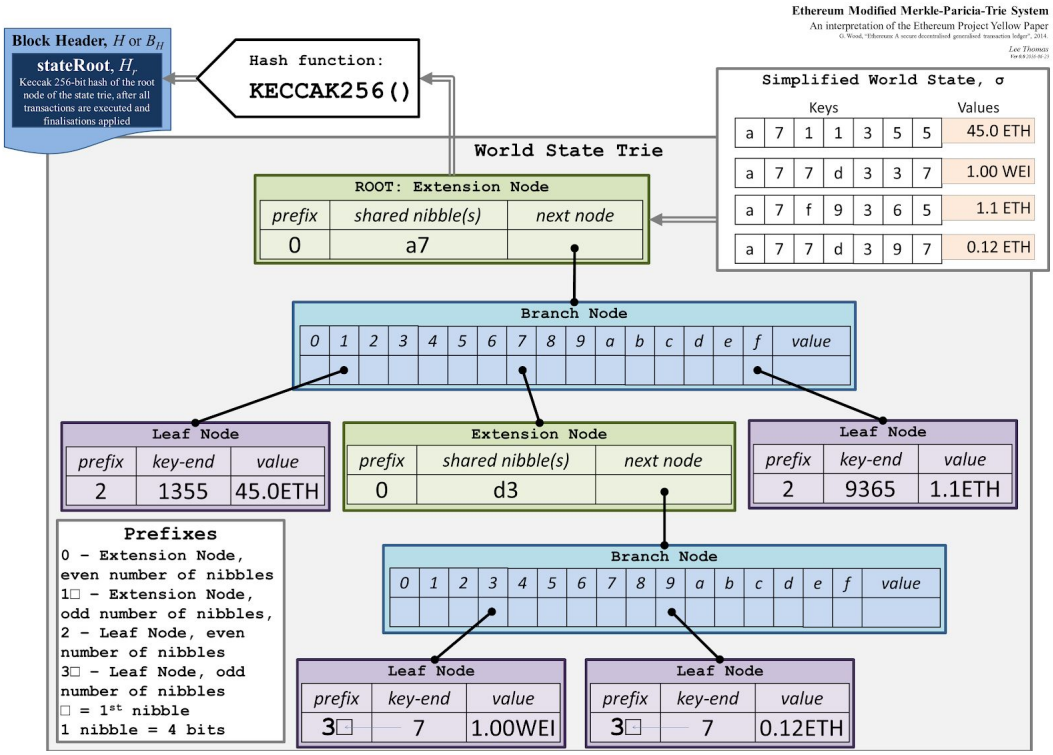
Figure 5.c.3

To support user data privacy, we will implement a cryptographic encryption algorithm to all data. For now, we are planning on using the asymmetric algorithm, RSA, to secure user data. As a private key, we can use the same private key that is used for access to the Blockchain. Thus only users that inputted the data can read from that data. For global data that is visible to all users, we will store it in plain text in the blockchain.

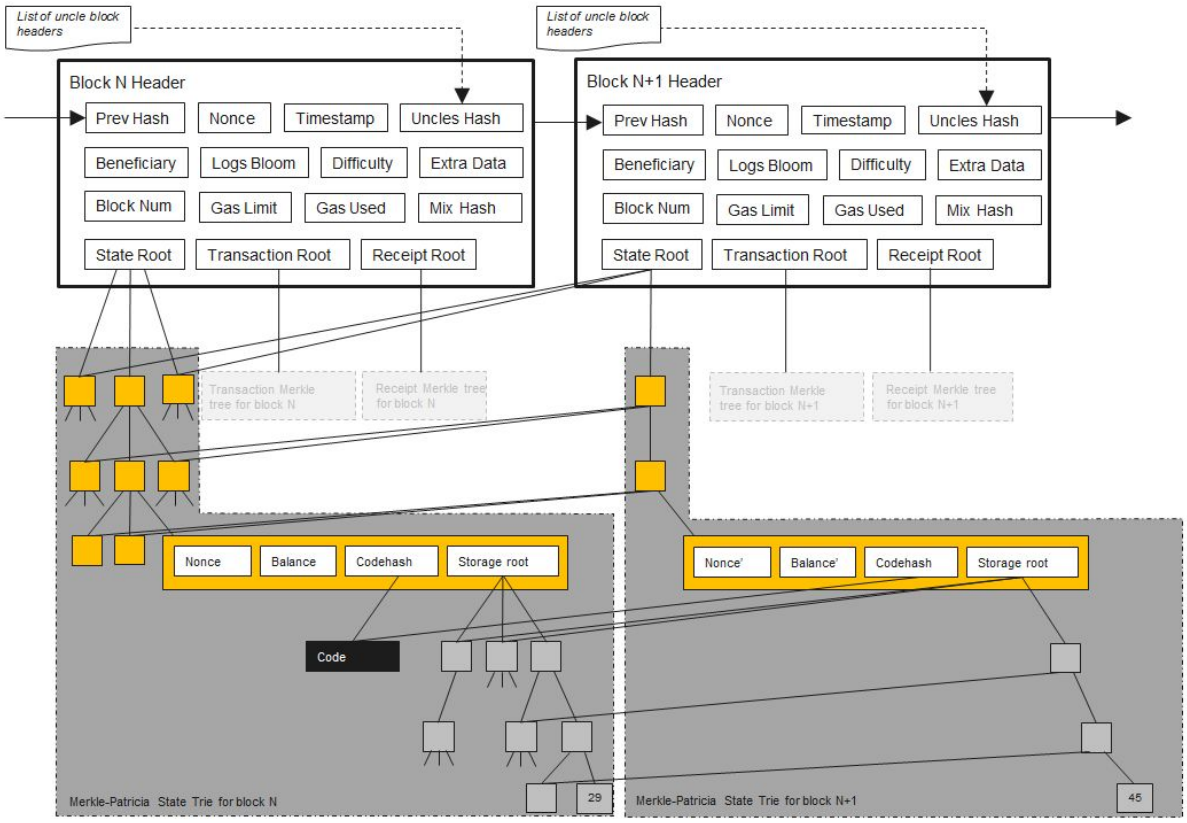
The server and blockchain interaction is dependent on the implementation of Ethereum network which is defined in the Ethereum yellow paper. Ethereum is a project which attempts to build the generalised technology; technology on which all transaction-based state machine concepts may be built. The execution model is specified through a formal model of a virtual state machine, known as Ethereum Virtual Machine. The machine has a simple stack-based architecture. The word size of the system is 256-bit.

Furthermore, Ethereum implementation relies on Merker trees . Although it is definitely theoretically possible to make a blockchain without Merkle trees, simply by creating giant block headers that directly contain every transaction, doing so poses large scalability challenges that arguably puts the ability to trustlessly use blockchains out of the reach of all but the most powerful computers in the long term. Thanks to Merkle trees, it is possible to build Ethereum nodes that run on all computers and laptops large and small, smart phones, and even internet of things.

The data structure and implementation of ethereum



Merkel tree in Ethereum

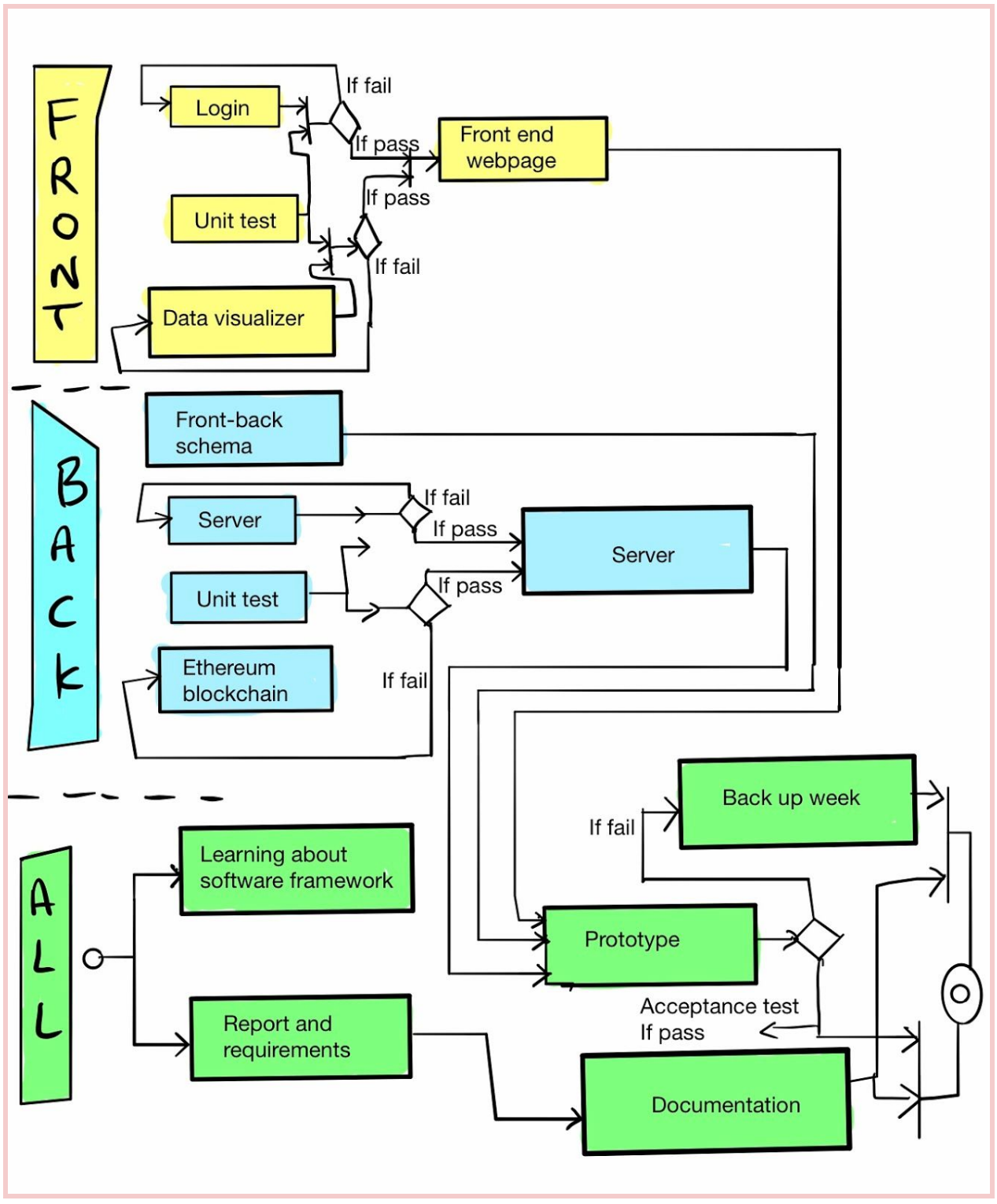


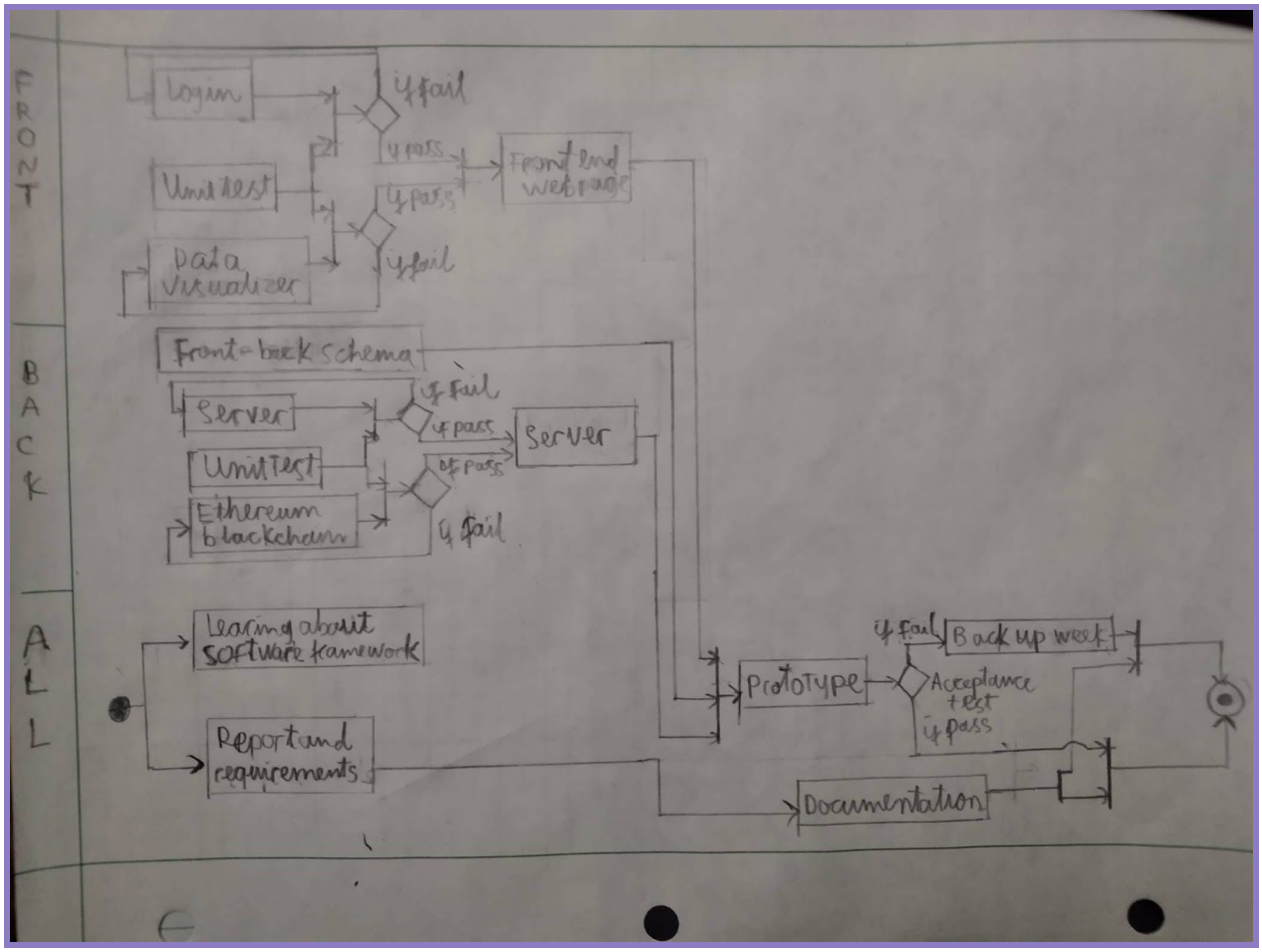
6. Project size estimation

Task	Time
Report, business requirement and other high-level descriptions	3 weeks
Implementing the login page (with React)	2 weeks
Build a data visualization system (with D3 library)	3 weeks
Implement the blockchain service (with Solidity and Truffle)	3 weeks
Implement a server (with Express)	2-4 weeks

Interfacing between the frontend and backend (with Graphql, Express and React)	1 week
Comprehensive testing and Acceptance test	1-2 weeks
Designing	3-5 weeks
Learning about the implementation in software domain	4-5 weeks
Unit testing(done concurrently with the implementation)	1-5 weeks
Backup week	1 week

7. Plan of Work





8. References

Ethereum. “Ethereum/Wiki.” *GitHub*, github.com/ethereum/wiki/wiki/White-Paper.

Truffle Suite. “Truffle: Overview: Documentation.” *Truffle Suite*, www.trufflesuite.com/docs/truffle/overview.

“web3.Js - Ethereum JavaScript API.” *web3.Js - Ethereum JavaScript API - web3.Js 1.0.0 Documentation*, web3js.readthedocs.io/en/v1.2.0/.

“The U.S. Government and Global Non-Communicable Disease Efforts”, www.kff.org/global-health-policy/fact-sheet/the-u-s-government-and-global-non-communicable-diseases/.

WHO“Global report on diabetes”, www.who.int/diabetes/global-report.

“UML Reference”: https://www.uml-diagrams.org/class-reference.html