

SOFTWARE ENGINEERING I

REPORT - 1

I. CUSTOMER PROBLEM STATEMENT

A. PROBLEM STATEMENT

The *world* is beginning to realize the importance of health in a common man's daily life and the influence it has on his activities down the line. While work and its related aspects are a critical part of our lives, health plays an even bigger role than most people imagine. In order to perform everyday activities efficiently, we need to have a good and healthy body. The scope of "exercise" is expanding and encroaching into other activities, like dance (zumba, aerobics and so on). Some kinds of exercise, like jogging, hiking, swimming, trekking and physical fitness are becoming hugely popular around the globe.

However, expensive health check-ups and inadequate facilities remain an obstacle to most people who wish to know the condition of their body and the ways to improve on it. The health condition of a person can be easily obtained by checking the vital signs of him. Crucial inferences can be drawn by analyzing the blood pressure stats, blood sugar levels, sleeping patterns and so on. Most people are aware of the consequences but always relied on medical help, which led them to incur monetary expenses.

We propose a digital ledger of sorts for the various users on the internet. This ledger contains data entered by the user as well as data streamed from the users devices like AppleWatch, Fitbit etc. This data contains many important metrics such as the following:

- Age
- Location
- Weight, height and BMI (Body Mass Index)
- Blood pressure
- Heart rate
- Sleep patterns
- Cholesterol levels

Our aim is to create a fully operational user-interface that would be friendly to use and understand and you can enter these details onto the website and the ledger will be directly populated with these details. You can also give the application

permission to access data, like your heart rate, blood pressure, blood sugar etc. from your AppleWatch, Fitbit etc, which will be collected at regular intervals from your device and registered into the ledger.

One of the critical needs of such an application is that any user should know how healthy or unhealthy he is as compared to others. He must be able to select different parameters (like sleep patterns, blood sugar levels, cholesterols etc.) and based on those parameters compare his data to the average/ general health of the other users. He can also select data for comparison from a specific location because geographically the health conditions considered normal can vary and also from a particular window of time, since people's health can vary in a location from time to time (eg. before and after the holiday season).

Also this application provides for the user to make his data available to or kept private from other users when those other users would be performing similar data analysis operations for their own purposes. However the application should stress on the fact that all users must try to make the data available to other users because that would be helpful to others in figuring out their health requirements and that their data would be anonymously shared.

Now, people would be naturally apprehensive to allow their personal details to be shared with this application, due to the concern of getting hacked. To address this concern, we are using the **Blockchain** technology, which provides several advantages over a regular database:

- The data in each block in the blockchain will be encrypted to create its hash which is like the block's fingerprint (a unique hash for each block). The data also contains the *previous* blocks hash in it, so if any attacker wishes to alter or change any blocks data, then he would have had to change the hashes of all the blocks following that block.
- Each node (i.e user) will be part of this network and will have an entire copy of the blockchain with them. Any change in the blockchain will be an addition of a block to the existing to the blockchain and the hash generated by the new block will be verified by all the nodes in the network. Only then can the block be added to the blockchain and this increases the security of the data.
- Since all the nodes have a copy of the entire blockchain we do not have to worry about maintaining a centralized server (a "honey-pot" if you may) which may be prone to hacking attacks or maybe at risk of losing all the

data if it shuts down or encounters a hardware issue. Any node can leave the network and on re-entry it can be updated with the latest version of the blockchain from the other active nodes.

Now, you may be wondering why would I want to go through all this trouble? Well, because we are implementing a blockchain in this solution, all the personal details of the users on the network are securely stored in the blockchain. This solution does not have a centralized authority, like a server or cloud, that stores all the data; the data is shared among the various users in the network and can be compared among them.

Since the blockchain is nearly immutable, the existing blocks cannot be changed. All the information stored in the system will be the latest and most reliable information, that will be regulated at the point of entry to the system. It will be like a community helping out its members by sharing their health parameters. The information will be much more recent and relevant as compared to the similar data found in government databases.

Also, the government-provided statistics may be incomplete or, at times, outdated. Since the user manually enters his/her details into the blockchain, with the help of user interface, the data is up to date.

The interactive application must also provide for the users to communicate with each other using their usernames through a forum which will be included in the application. The application will also be able to provide you daily links to health-related, diet-related articles and videos.

So, with all these facts and details, we can say with some aplomb that our proposed approach is highly innovative.

B. GLOSSARY OF TERMS

-
1. ***BMI (Body Mass Index)*** : BMI of a person is the weight of the person (in kgs.) divided by the person's height squared (in meters)
 2. ***Sleep Patterns*** : Classifying people in different categories (like Healthy larks, sleep savvy seniors etc.) based on the average hours of sleep at night.

3. **Cholesterol level** : Record Low density lipoprotein (LDL) cholesterol along with the total cholesterol level so that they can control their dietary habits.
4. **Blood Sugar** : Compare the blood sugar of a person with the blood sugar chart so that they can self manage any diseases like diabetes. Can intake values both fasting and 1-2 hours after eating.
5. **Blood Pressure** : Compare the blood pressure input from the user and compare with the normal upper and lower bounds to judge whether it is normal or not
6. **Ledger** : A book or collection for recording the personal details of the users on the network.
7. **Blockchain** : A collection of blocks interconnected with each other; each block contains the details that are given by each user of the network.
8. **Hash** : Also called as the *hashtag function*, this is a function that is used to encrypt the blocks so that attackers cannot access the information in the blocks.
9. **Node** : Each user on the network is referred to as a node.

II. SYSTEM REQUIREMENTS

A. ENUMERATED FUNCTIONAL REQUIREMENTS

<u>Identifier</u>	<u>Priority Weight</u>	<u>Requirements</u>
REQ -1	10	System should have secure login
REQ -2	8	System should allow to update the parameters
REQ -3	7	System should integrate with health device
REQ -4	10	System should store data in secured manner
REQ -5	9	System should compare data with authentic data
REQ -6	7	System should restrict outlier data
REQ -7	5	System should notify the user if a new parameter is introduced

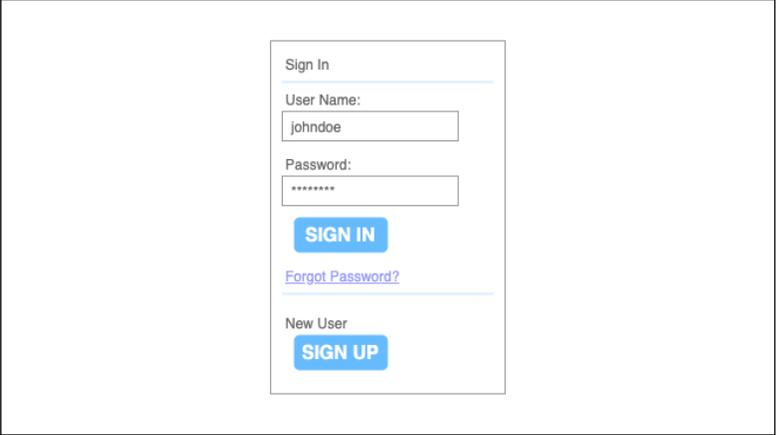
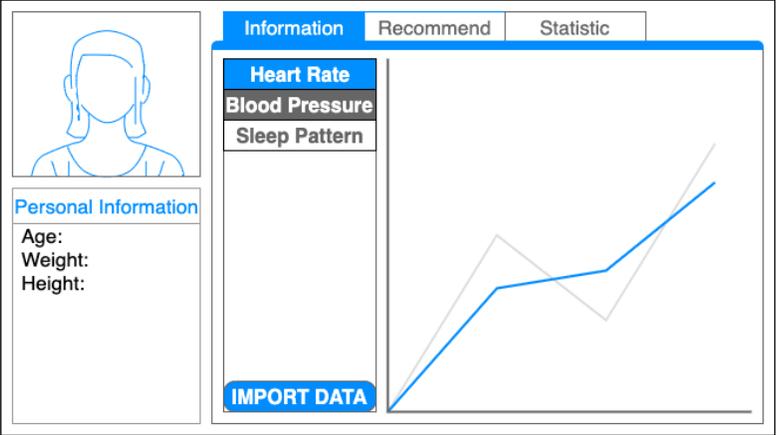
REQ -8	6	System should recommend health measures analyzing my data
REQ -9	4	System should allow users to give feedback and provide support as well
REQ -10	6	System should generate historical report for users to view/access
REQ -11	10	System should allow user to unregister and delete data upon unregistering
REQ -12	9	System should ask the user for consent to share data for statistical comparison
REQ -13	10	System should allow user to view/compare average statistical data of other users
REQ-14	3	System should allow user to communicate with other users in the system

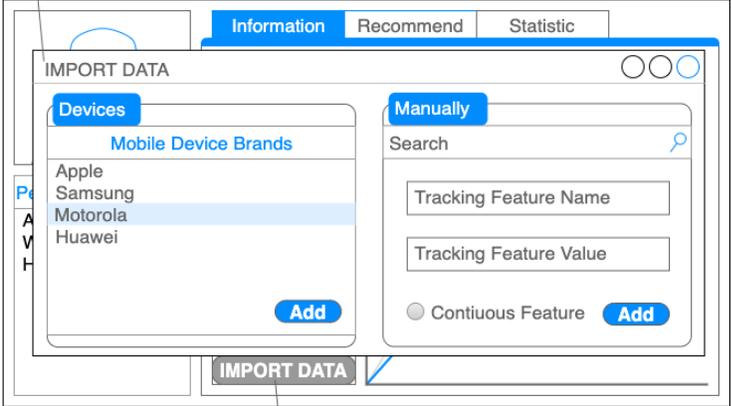
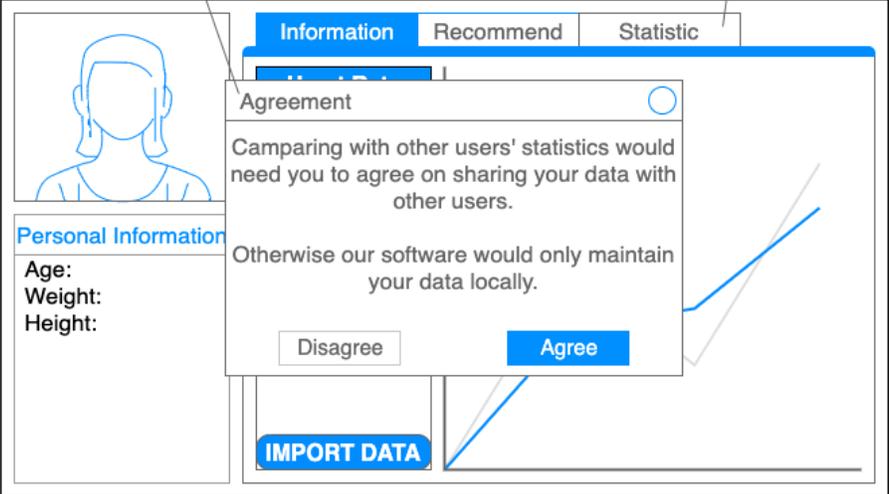
B. ENUMERATED NON-FUNCTIONAL REQUIREMENTS

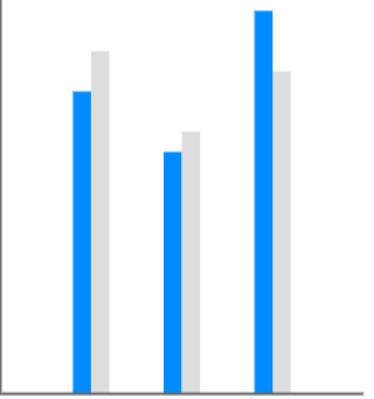
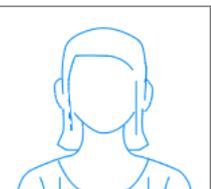
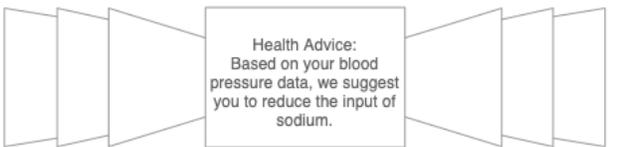
<u>Identifier</u>	<u>Priority Weight</u>	<u>Requirements</u>
REQ -15	8	As a system, size and generality of the data must be defined
REQ -16	10	As a system, all the user data must be encrypted
REQ -17	7	As a system, communication between system actors must be secured
REQ -18	5	As a system, application/portal must be supported in different browsers of both web and mobile platforms
REQ -19	2	System should be scalable and load balanced

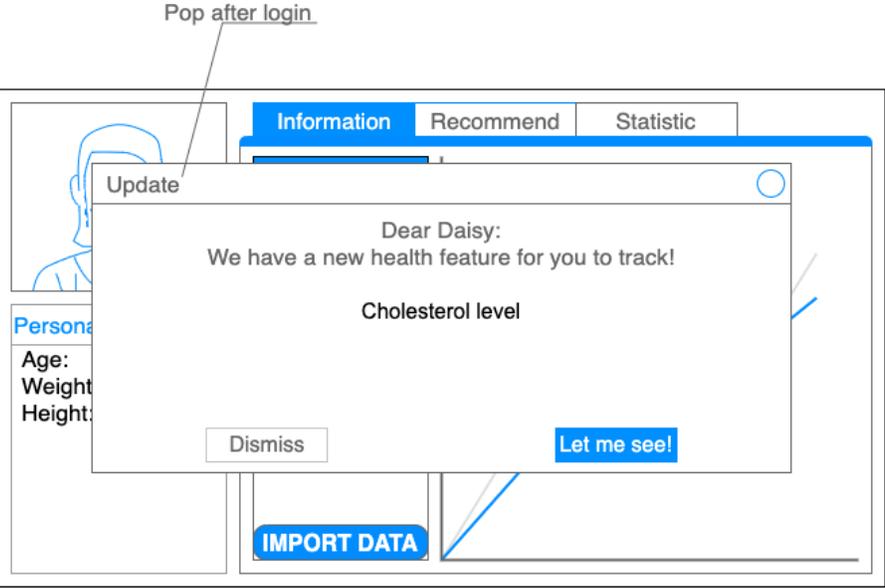
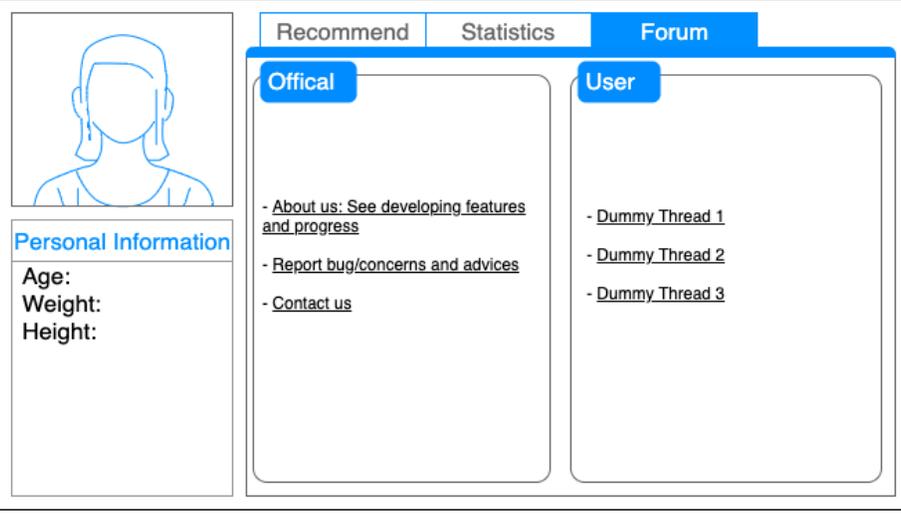
REQ -20	4	As a system, data of the entire system must be archived periodically
REQ -21	6	As a system, backups of the data must be taken periodically
REQ -22	7	As a system, appropriate business continuity policies and disaster recovery strategy must be implemented
REQ -23	3	As a system, system maintenance should be done regularly in order to keep systems up-to-date
REQ -24	8	As a system, user requests/issues should be supported and addressed
REQ -25	8	As a system, deployment strategy should be implemented along with roll back strategy in case of deployment failure
REQ-26	10	As a system,data across all the nodes in the system must be synchronized
REQ-27	4	As a system, logging and monitoring of the system and application must be in place
REQ-28	9	User manual and Architecture Diagram along with proper Documentation of the system must be provided

C. USER INTERFACE REQUIREMENTS

<u>Identifier</u>	<u>Priority Weight</u>	<u>Requirements</u>
REQ-29	10	<p>GUI must have a landing page(register and log in).</p> 
REQ-30a,b	10	<p>GUI must have a main page of users' personal information. GUI must have a page for user to view his historical data.</p> 
REQ-31	10	<p>GUI must have a page to enter users' health data.</p>

		<p>2. Pop up</p>  <p>1. Click</p>
REQ-32	10	<p>GUI must have consent option for sharing data for comparison.</p> <p>2. Pop up</p>  <p>1. Click</p>
REQ-33	8	<p>GUI must have a page for statistical data for comparison.</p>

		<div data-bbox="560 210 1445 703">  <p>Personal Information Age: Weight: Height:</p> </div> <div data-bbox="803 210 1437 693"> <p>Information Recommend Statistic</p> <p>Resting Heart Rate</p> <p>BMI</p> <p>Sleep Pattern</p>  </div>
REQ-34	4	<p>GUI must have recommendation page on users' health data.</p> <div data-bbox="560 1365 1445 1858">  <p>Personal Information Age: Weight: Height:</p> </div> <div data-bbox="803 1365 1437 1848"> <p>Information Recommend Statistic</p>  <p>Health Advice: Based on your blood pressure data, we suggest you to reduce the input of sodium.</p> </div>

REQ-35	4	<p>GUI must have a page to notify users the new update.</p>  <p>Pop after login</p>
REQ-36a b,c	2	<p>GUI must have a forum page for users' communication. GUI must have support page for any issues. GUI must have a page about us for our progress on development.</p> 

III. FUNCTIONAL REQUIREMENT SPECIFICATIONS

a. Actors and Goals

Actor	Actor's Goal	Use Case Name
Visitor/User	To login to the web portal/application to enter health data, view statistics, get health recommendations, view comparison reports, etc.	Login (UC-1)
Visitor/User	To register in the system/application via web portal	Register (UC-2)
Visitor/User	To add data in the system/application via web portal	Add Data (UC-3)
Visitor/User	To view the historical health data for analysis	Historical Data Presentation (UC-5)
Visitor/User	To compare health data with other registered and active users	Comparison Data Presentation (UC-6)
Visitor/User	To get health recommendations based on the data entered in the system	Health Recommendations (UC-7)
Visitor/User	To communicate with other registered and active users and also with system administrator/support team	Communication (UC-8)
Visitor/User	To have its data validated before adding data in the system	Data Validation (UC-4)
System Admin	To resolve issues that users might face while user performs any operations or has any other issues	Communication (UC-8)
Database/Repository	To store the user information, login details, data, recommendations, historical and comparison data	UC1, UC2, UC3, UC4, UC5, UC6, UC7, UC8

b. Use Cases

i. Casual Description

UC#1 Logging In

The user can enter the portal/application by entering his credentials used for registration. He can then perform various operations, like entering his data into the application, viewing the statistics, getting health recommendations, viewing the comparison reports and so on.

UC#2 New Registration

The user can register into the blockchain-based network to be a part of the community. Then he or she can start with the service provided by us and all the other users. After the user finished registration, he or she can log in to the whole system and do all the operations listed in UC#1. Also, he can unregister from our system, and his data will be cleared from the blockchain-based network.

UC#3 User Data Addition

The user, once registered into the system, will be able to integrate his or her health device with the system and securely store the health data without manual intervention. If needed the user can also manually update the parameters of their health data

UC#4 Validation of Data

The data validator will check if the data added by the user into the web portal/application is within the appropriate range. It will allow the user to proceed with adding or updating his data if it is valid; if any outliers exist in the data entered, an error message will pop up prompting the user to re-enter his valid data.

UC#5 Historical Data Representation

Users can request our system to generate a historical data representation of their health information. The graphical view of their cumulative health data will allow users to better perceive the changes in their health parameters. Users can monitor and take preventive measures with this information.

UC#6 Comparison of Data Report

Users can request our system to generate the histograms, pie charts and scatter plots to show his position in the whole population depends on his or her requested parameter. For example, the user could see how the other users are distributed on age in a pie chart, or where is his or her position to the whole users in a histogram. The UI would also allow the user to customize multiple parameters in the comparison. Last but not least, before a user joining the pool of user comparison, he or she must agree to share his or her data to the whole network as a base of comparison. The user can choose to exit the pool with his data removed from the cloud whenever he or she wants.

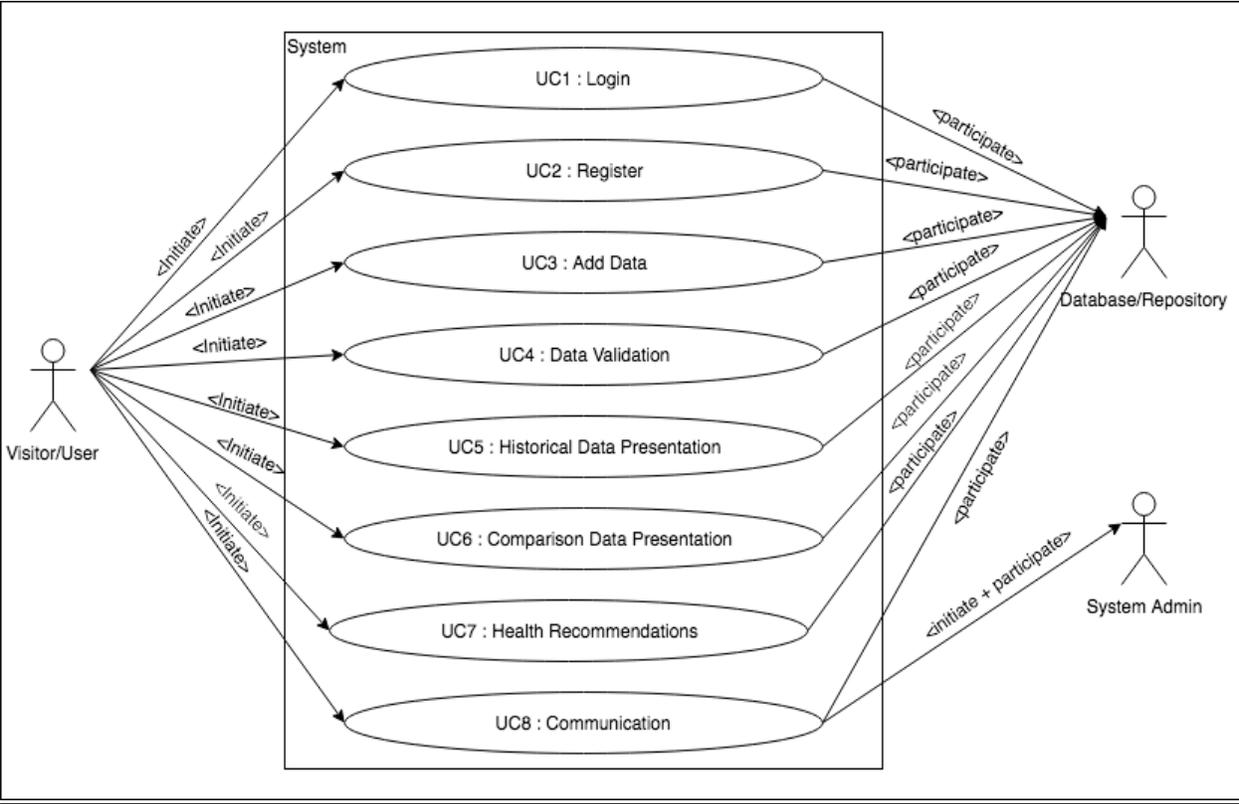
UC#7 Health Recommendation

Our system is capable of analyzing the users health data and suggesting certain health recommendations to appropriate users. These are general health recommendations which will include exercises, YouTube video recommendations etcetera to give users a chance to normalize their health parameters.

UC#8 User Communication

Our system is capable of providing a forum to let users communicate with each other and give us, the development team, feedback and report bugs.

ii. Use Case Diagram



iii. Traceability Matrix

REQ'T	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
REQ1	10	X	X						
REQ2	8			X					
REQ3	7			X					
REQ4	10			X					
REQ5	9				X				
REQ6	7			X					
REQ7	5			X					
REQ8	6							X	
REQ9	4								X
REQ10	6					X			
REQ11	10		X						
REQ12	9						X		

REQ13	10						X		
REQ14	3								X
Maximum Weight	10	10	10	9	6	10	6	4	
Total Weights	10	20	37	9	6	19	6	7	

iv. **Fully-Dressed Description**

Use Case UC#2:	<i>New Registration</i>
Related Requirements:	REQ-1,REQ-11
Initiating Actor:	Users
Actor's Goal:	To allow user to have secure login. To allow user to unregister and delete the data.
Participating Actors:	Active users,System Admin,System
Precondition:	System should be active and running. User should have established connection with the system. User should meet the requirement for registering.
Postcondition:	User can enter his data in the system. User can update or delete his data as per his requirement. User will have option to share/hide his data from other users.
Failed end condition:	User entering wrong credentials, re-enter valid credentials Unable to correct to system,check system status
Flow of events for success scenario:	

1. → **User** enters his credentials into the portal
2. ← The credentials are verified by the system.
3. → The **user** is allowed to access the portal.

Flow of events for extensions:

User enters invalid/out-of-bounds data.

1. ← **System** detects the error and sends an error message to the user prompting to enter the data again.

User enters a weak password while registering.

1. ← **System** should suggest a strong password to the user in order to keep the data more secure.

Use Case UC#3:

User Data Addition

Related Requirements:	REQ-2, REQ-3, REQ-4, REQ-6, REQ-7
Initiating Actor:	Users
Actor's Goal:	To add and update data which is within appropriate range. To append encrypted data into the blockchain with Verification from all users. To inform the user if any new parameter is available in the System for monitoring.
Participating Actors:	User fitness device System, other Active Users, System admin
Precondition:	User should be active on the system User's communication link with the system should be online The blockchain must be initialised and active.
Postcondition:	New encrypted data blocks should be appended to the blockchain and updated across all the users present in the network.
Failed end condition:	Data input unsuccessful, check data range Unable to connect with system, check connection

Flow of events for success scenario:

1. → **User** enters the web application and chooses option to input data.
2. → **User** gives permission to the system to access the **fitness device**.

3. → **User** inputs data.
4. **System** uses the data input by the **user** to make a new block
5. include:: ValidationOfData (UC-4)
6. ← **System** adds the new validated block to the blockchain and gives user success message

Flow of events for extensions:

3a. **User** inputs invalid/out-of-bounds data

1. ← **System** detects error and sends an error message to the **user** asking to re-input the data.
2. → **User** inputs correct data

5a. Failure to validate new block from more than 50% active **Users**.

1. ← **System** a) detects error, b) marks failed attempt, c) signals to **System Admin**
2. ← **System Admin** will try to detect issue (maybe problem with hashing function)
3. → **User** data gets validated.

Use Case UC#6: *Comparison of Data Report*

Related Requirements: REQ-12, REQ-13

Initiating Actor: Users

Actor's Goal: To provide users the option to choose to share their data for statistical comparison.
 To generate the relevant histograms, pie charts, scatter plots so that the user can compare his data with other users

Participating Actors: Active users, Other Users, System

Precondition: User should be active on the system.

User must have the links available for selecting or deselecting the information sharing option and to view their comparison report.

System must have the active users who have selected the option of sharing their data for comparisons.

User must have actively participated in the block chain system based on which the user must have maintained the statistical data block chains for other users.

Postcondition: User must be able to be or not be

a part of the comparison subsystem of our system and can accordingly view the comparison report while having the comparison of its own data with the statistical information maintained based on other users' data.

Failed end condition: Not able to view the comparison data properly (maybe only his own data without any comparison or with partial comparison) ,User unable to connect to comparison page (Error message page displayed)

Flow of events for success scenario:

1. → User can select/deselect the option of being part of the comparison sub system.
2. ← System is able to provide the user with the services accordingly.
3. → User click the link for accessing the comparison report page.
4. ← System provides the user with the comparison report.
5. → User can view the comparison report comparing his/her data with the statistical information derived out of other users' data.

Flow of events for extensions:

I. When users select the option for doing some necessary selection to be the part of comparison of subsystem and some abnormality happens.

1. User can then raise the issue with the system administrator.

II. User click the link and error message is displayed or improper comparison report is visible.

1. User can raise the issue with the system administrator. The user can check its connection and troubleshoot system from its end.

c. System Sequence Diagrams

I. Use Case - 2

sd UC2 New
Registration

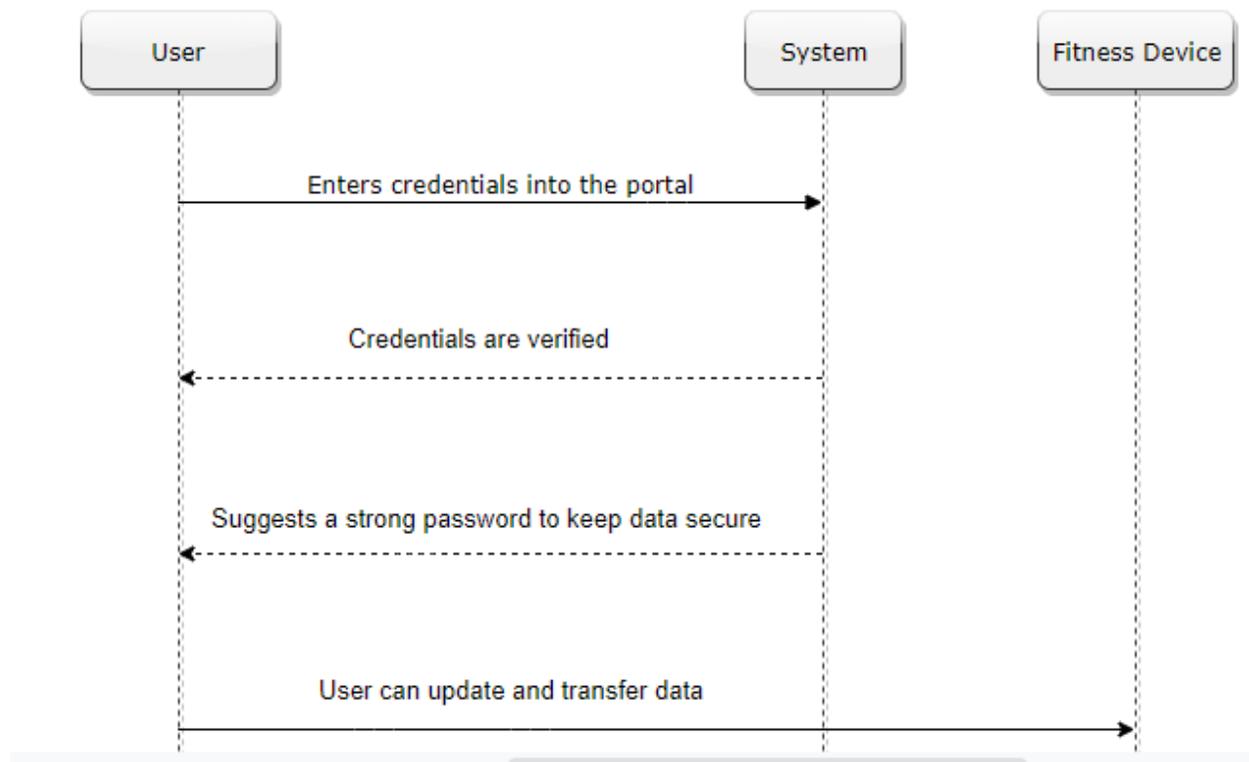


Fig 1: The above figure depicts the sequence diagram for **Use Case 2**, which is “New Registration”. The user will enter his credentials into the web portal which will be verified by the system. Once the credentials are verified, the user has to enter a unique username and password, which will serve as his login. The system should suggest a strong password to the user to keep the user’s data secure. Once he successfully enters this, he will be allowed access into the system, where he can view his data, get health recommendations, view comparison reports, among other operations.

II. Use Case - 3

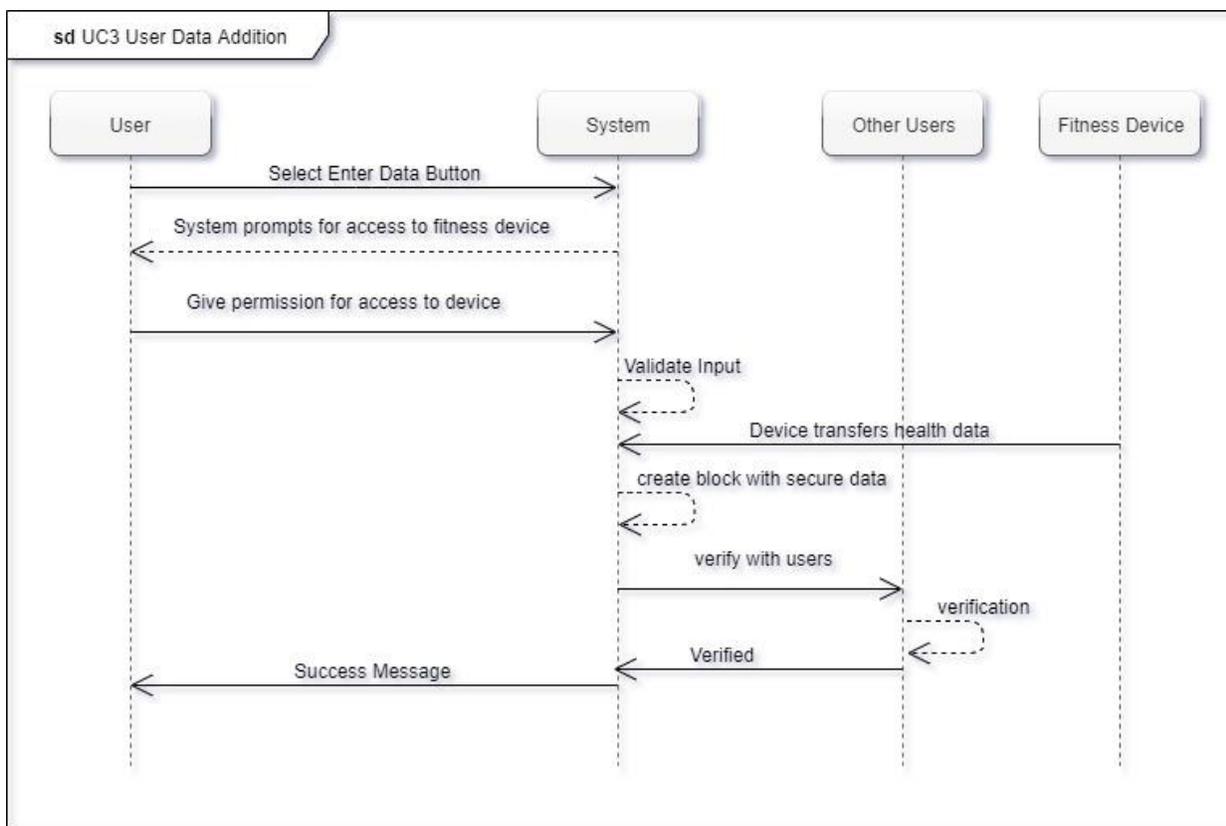


Fig 2-a shows the sequence diagram of **Use Case 3** which is “User Data Addition”. The user will select the option to enter the data to the blockchain. The System then prompts the User to input the data and give permission to access data from the fitness device. The

user then grants the system access to the fitness device and also manually inputs the data into the system. The System verifies if none of the data is an outlier. It then creates a block that will be added to the blockchain. The block is then sent to be verified by all the other users in the network and if it is verified by more than 50% of the users, it is added to the blockchain and a success message is then sent to the user.

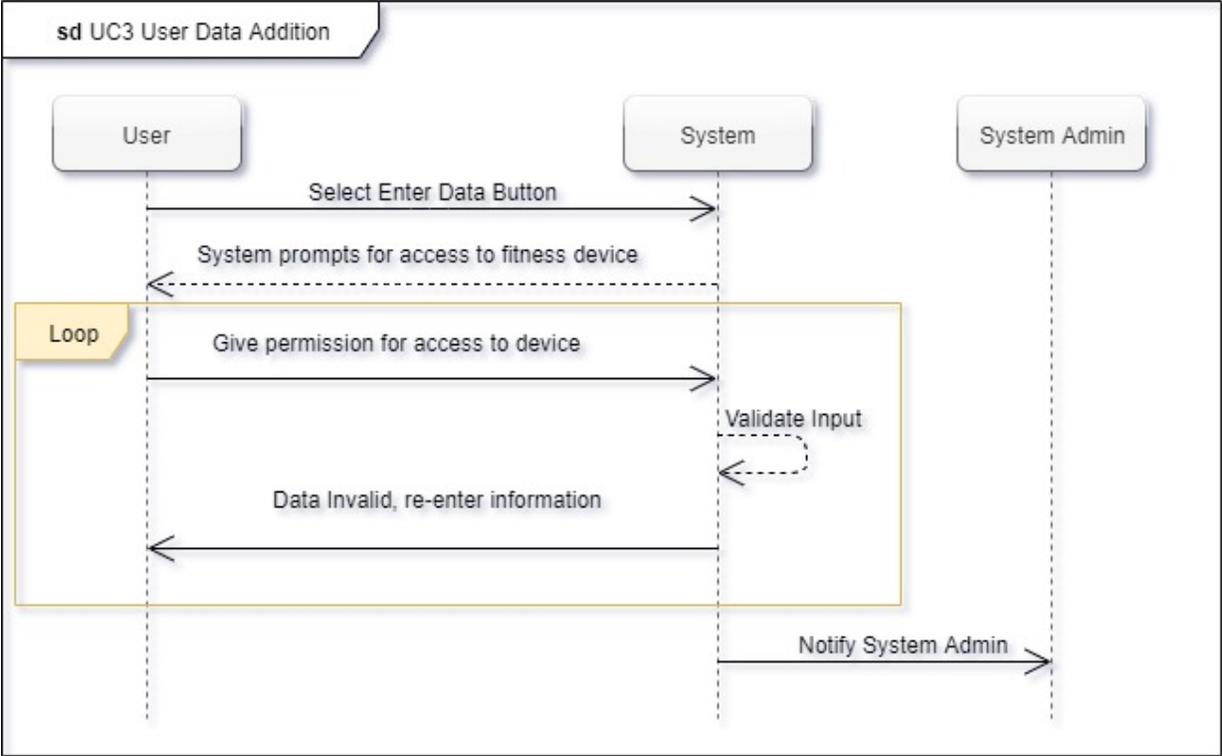


Fig 2-b shows the error scenario when the user entered data fails validation checks performed by the system. Out of range data or outlier data entry could be the root cause of this issue. In this scenario the system shows an error message to the user and requests the user to re-enter appropriate information. Also, the system admin is notified of this situation such that it can monitor the further transactions of the user with the system.

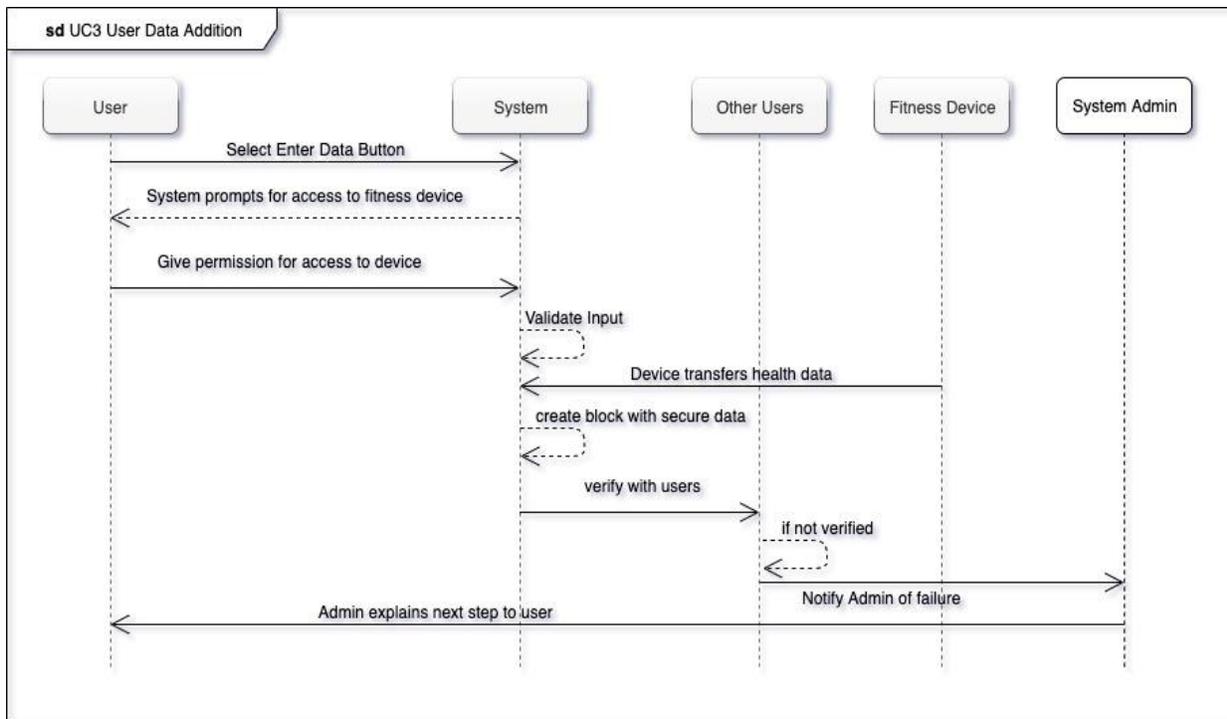


Fig 2-c shows the error scenario when the block created by the system is not validated by more than 50% of the users of the blockchain. This is a major issue which may be because of a problem in the hashing function. The system notifies the system admin who will inspect why the error occurred and then will tell the user what steps have to be taken next (maybe reenter the data etc.)

Local Storage

SuperNodes' Storage

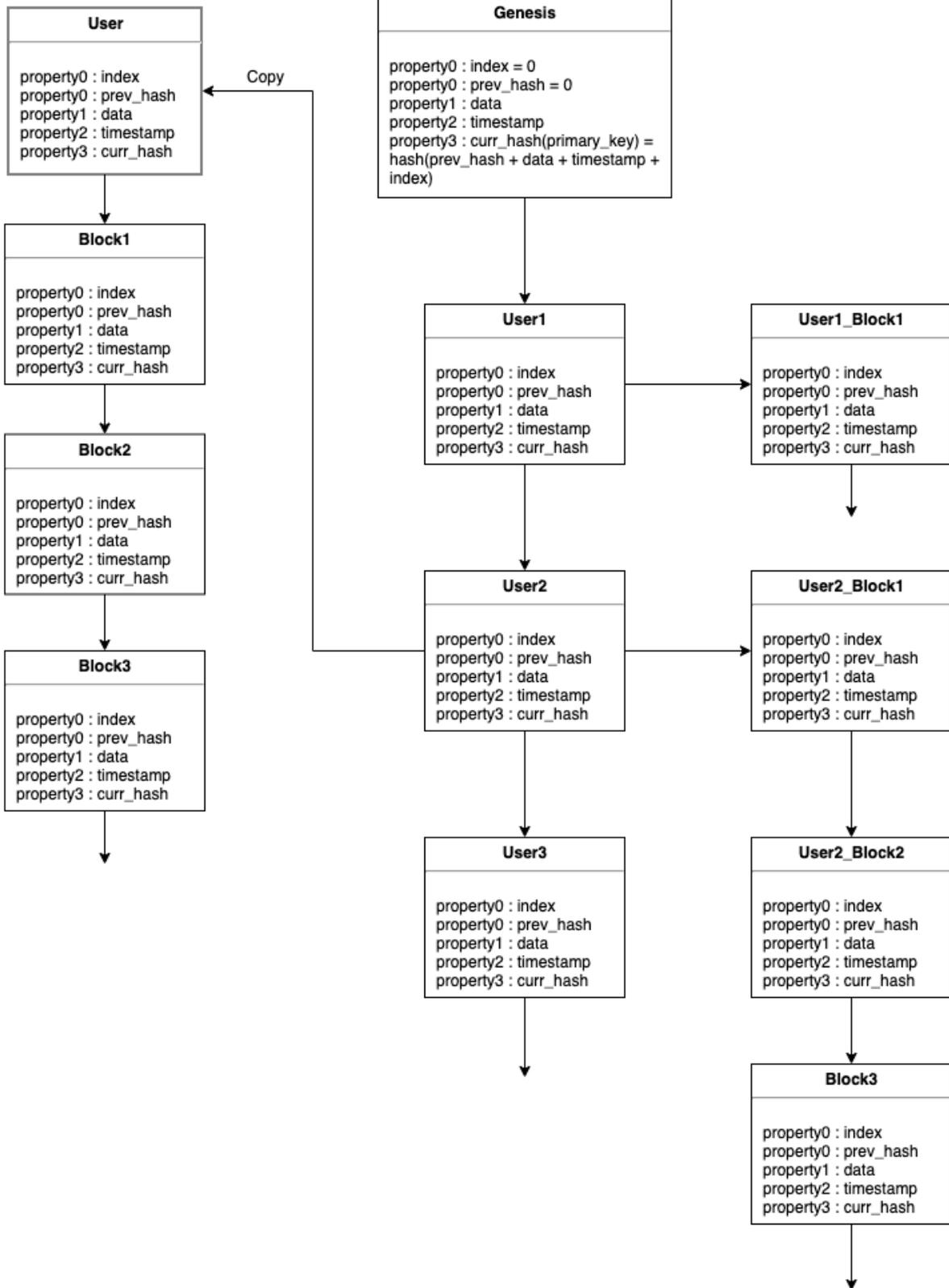


Fig 2-d shows structure of our Block-chain based storage. We store users' data both locally and remotely. On users' local app, we store and generate data blocks; on the system consist of supernodes and other users, they store the whole data blocks and verify newly adding blocks. If we have a total of m users, each user have n data blocks on average, we can reduce the complexity from m multiply n to m plus n . Also, we can process multiple users' verification without conflict.

III. Use Case - 6

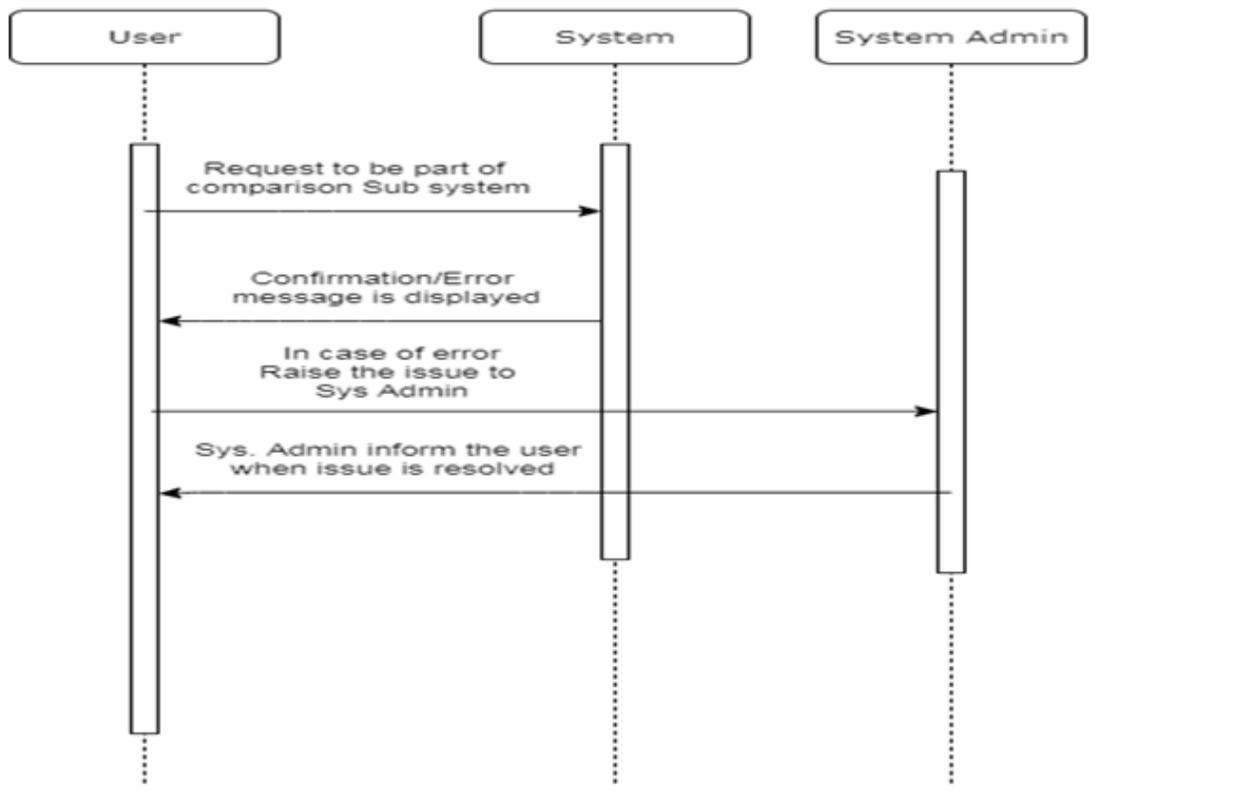


Fig. 3-a depicts **Use Case (UC) 6**, where the user has the option to be a part of the comparison subsystem. In case some issue occurs or the confirmation of selection is not shown to the user, he/she can raise the issue with the system administrator and after resolving the issue the system administrator will revert to the user.

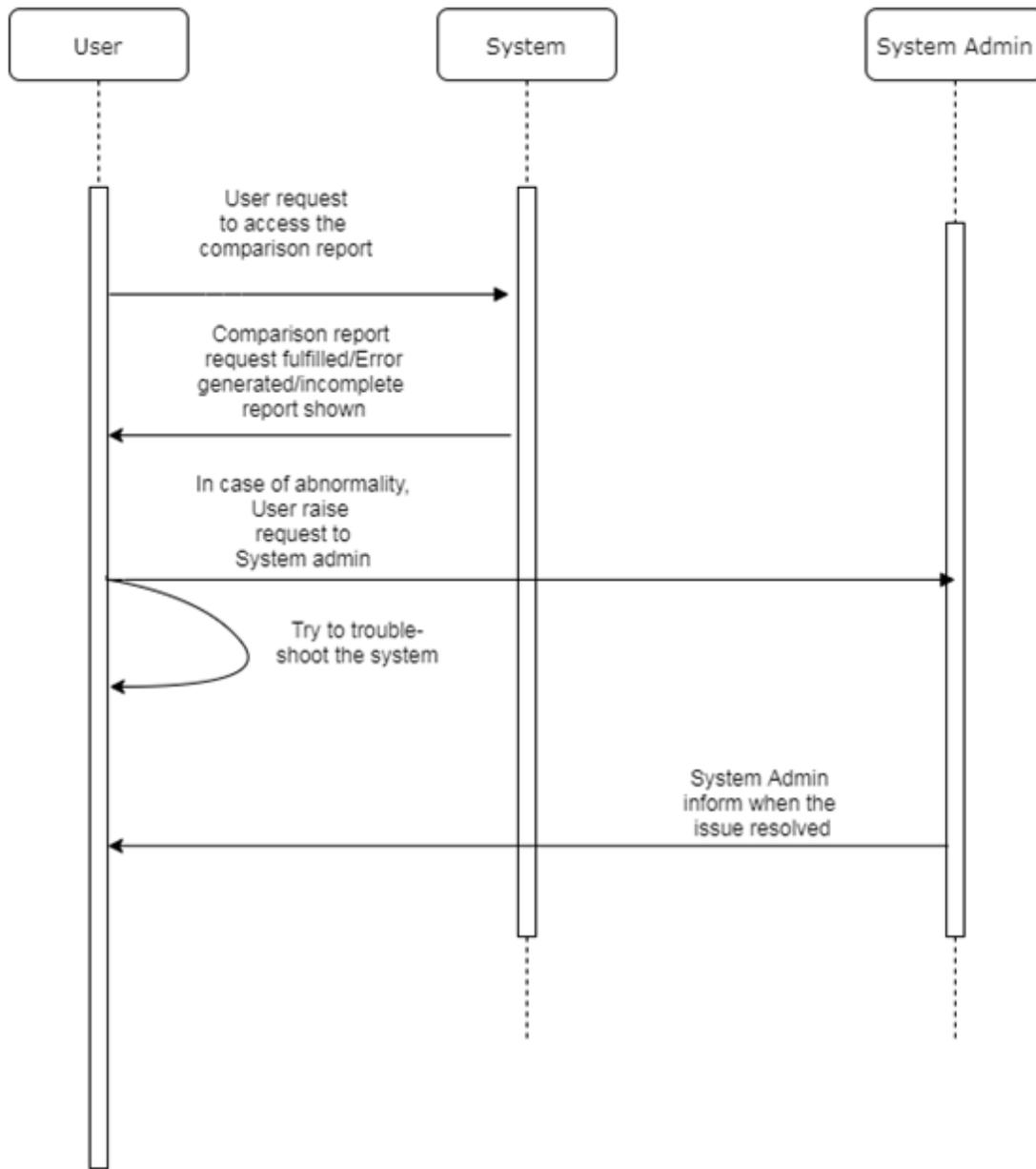
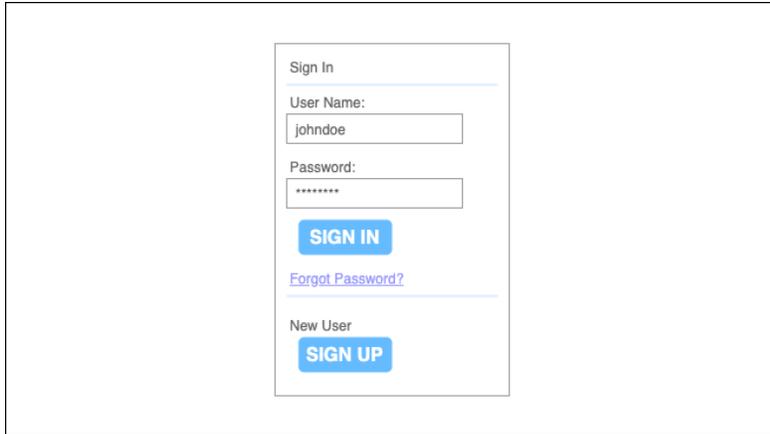


Fig. 3-b Here, the user tries to access the comparison report and in case any issue/abnormality crops up, he/she can raise the issue with the System administrator who can try to troubleshoot the system from its end . After resolution of the issue, the system administrator will revert to the user with the appropriate status message.

IV. USER INTERFACE SPECIFICATION

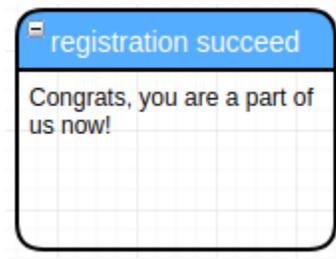
a. Preliminary Design

1. Registration :

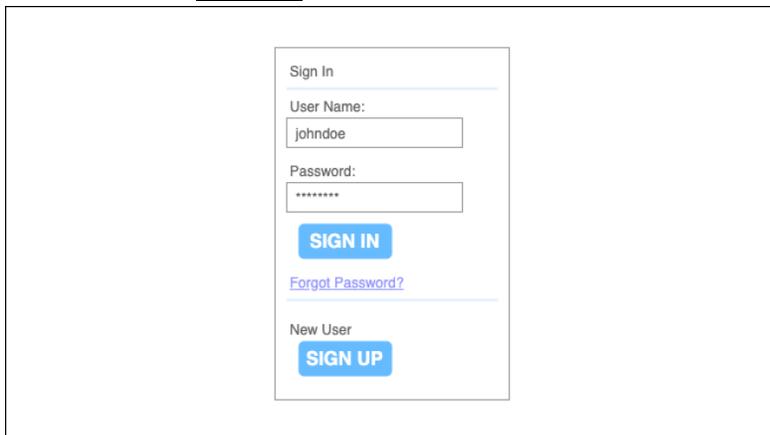


The image shows a registration form UI. It has a title "Sign In" at the top. Below it are two input fields: "User Name:" with the value "johndoe" and "Password:" with "*****". There are two blue buttons: "SIGN IN" and "SIGN UP". A link "Forgot Password?" is also present.

Once a valid input (user ID and Password) is submitted, a “registration succeed” message should be shown via the UI.

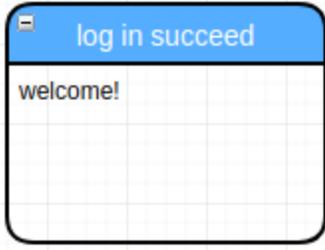


2. Log in:



The image shows a login form UI. It has a title "Sign In" at the top. Below it are two input fields: "User Name:" with the value "johndoe" and "Password:" with "*****". There are two blue buttons: "SIGN IN" and "SIGN UP". A link "Forgot Password?" is also present.

If a registered ID and correct password are submitted, a “login succeeded” message should be shown.



If a registered ID and wrong password are submitted, a “incorrect password” message will be shown.

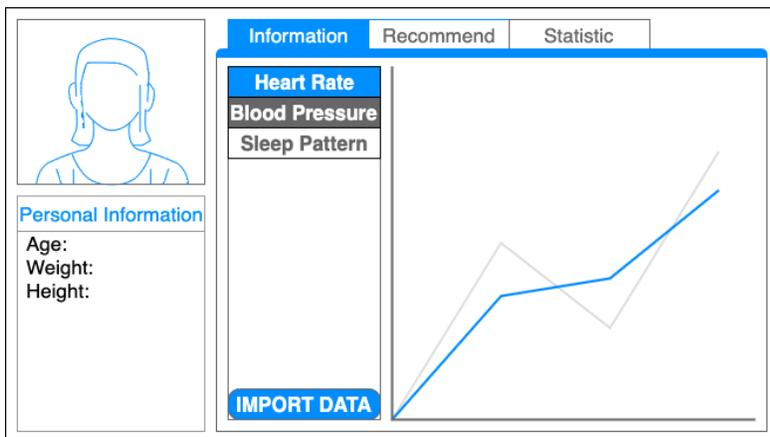


If an unregistered ID is submitted, no matter what the password is, it will give an “ID no found” message.



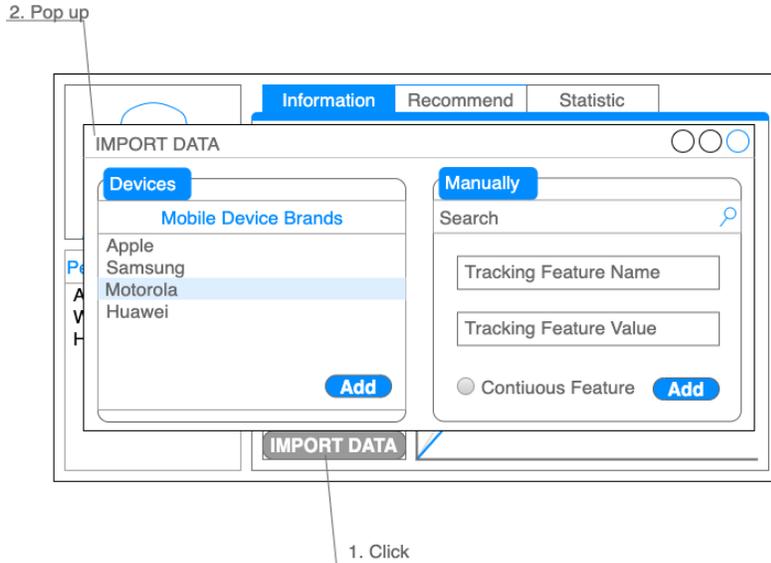
3. Personal information

Once a user successfully logged in, the personal information should be shown directly, along with the historical diagram.



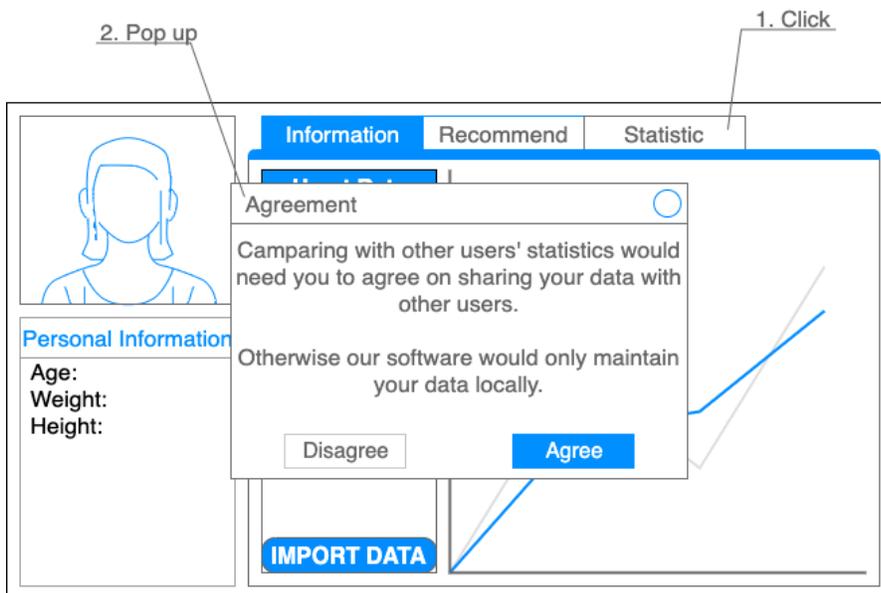
4. Data import

Once the personal information page is displayed, users could click on the “import data” button, and a data input page would be shown where users could import data freely.

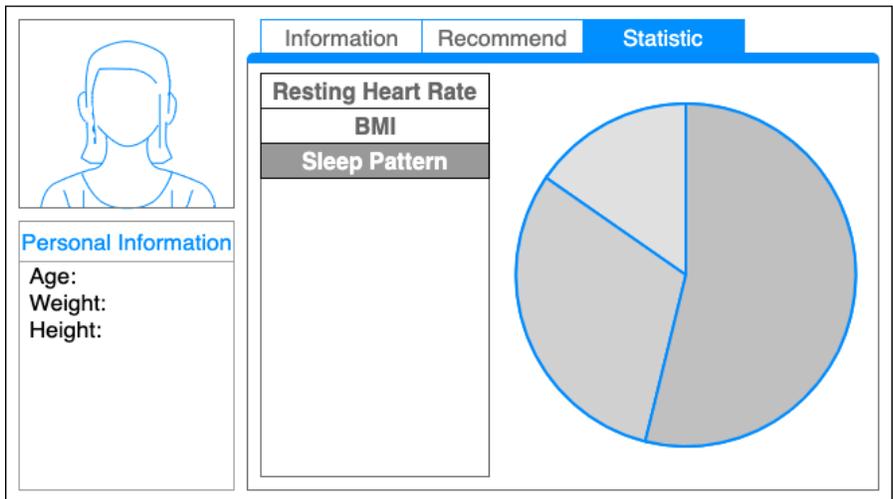
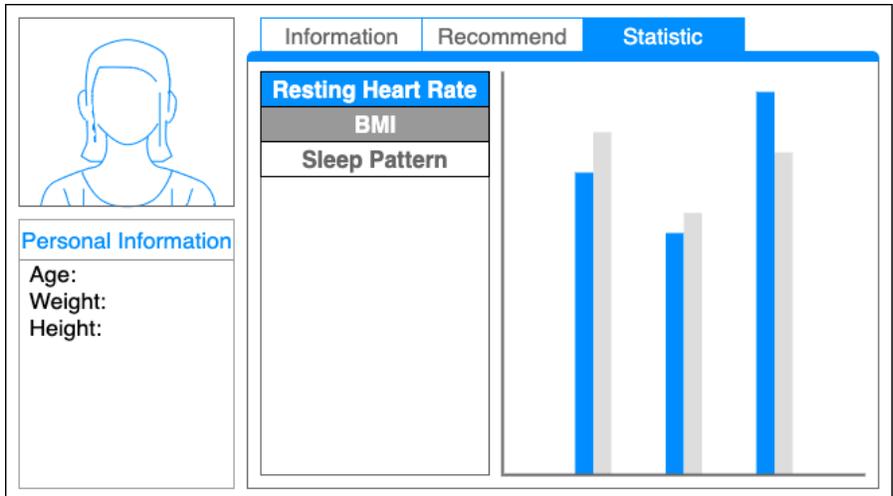


5. Statistics

Users could click on the statistic page to view the comparison, but a consent page that inquire users if they are willing to share their personal data for the statistics would pop up.

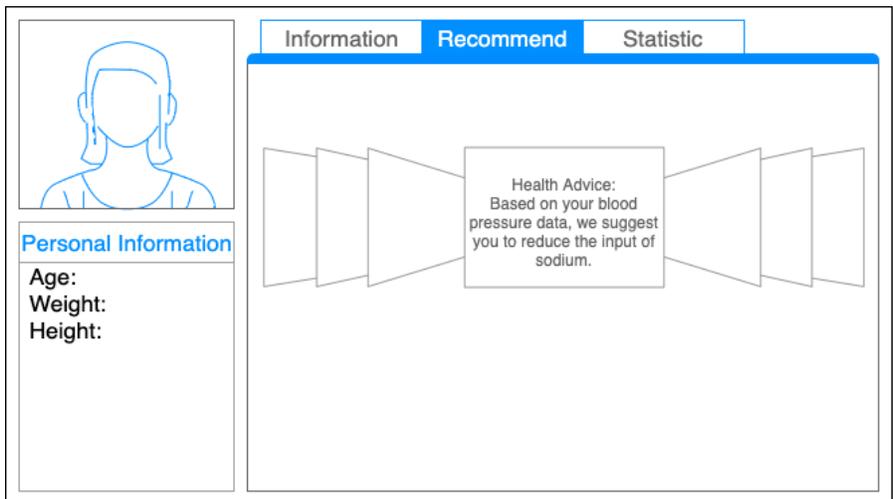


After that the statistic page would be shown



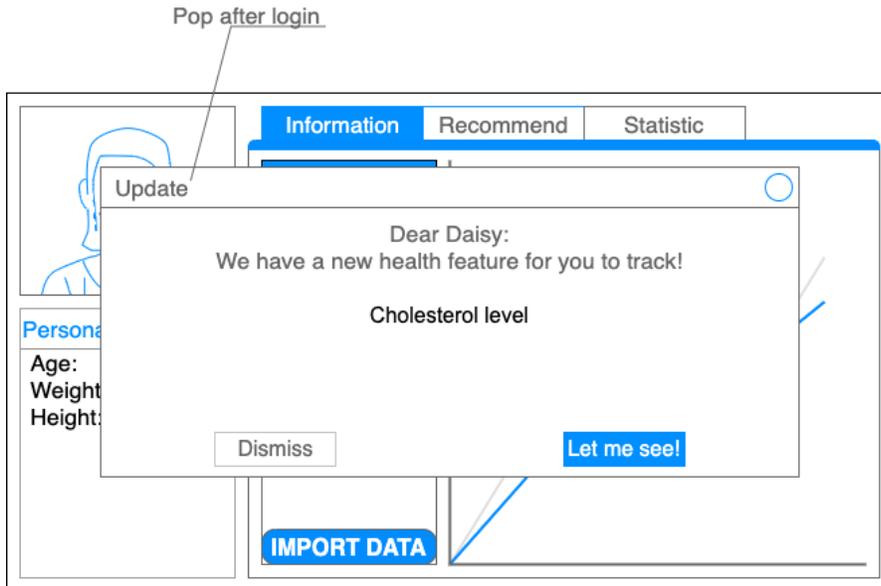
6. Recommendation

The user could click in the recommendation and the UI would provide the health recommendation for the him or her based on his or her health data.



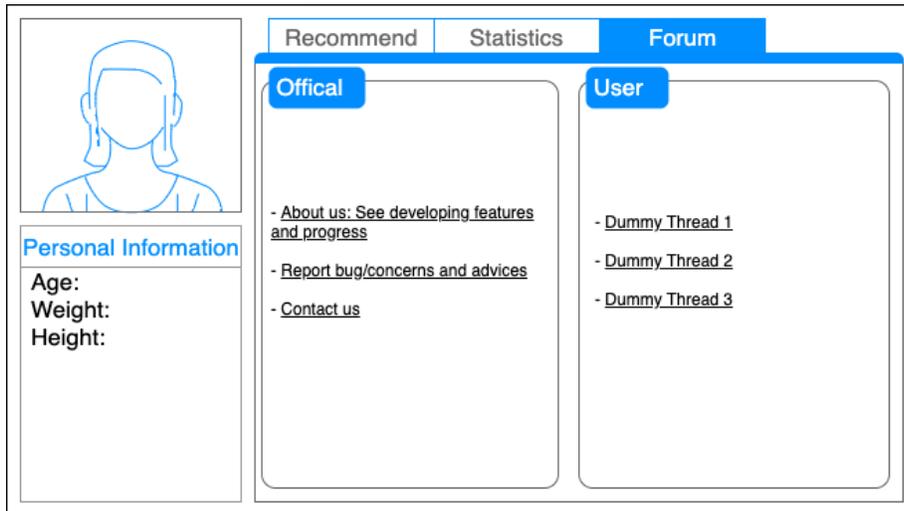
7. New update

If a new parameter is updated, the UI would notify the user with a notification page.



8. Forum

A forum page would be popped once a user click on the forum button, the support, communication and system announcements are all there.



b. User Effort Estimation

1. Registration :



Sign In

User Name:

Password:

SIGN IN

[Forgot Password?](#)

New User
SIGN UP



Sign Up

User Name:

Password:

Password Again

OK

Navigation: total 2 mouse clicks, as follows

- Click "SIGN UP" button
- Finish data entry in the new page
- Click "OK" to finish registration

Data Entry: total 3 keystrokes, as follows

- Press keys to input username
- Press keys to input password
- Press keys to input password again

2. Log in:

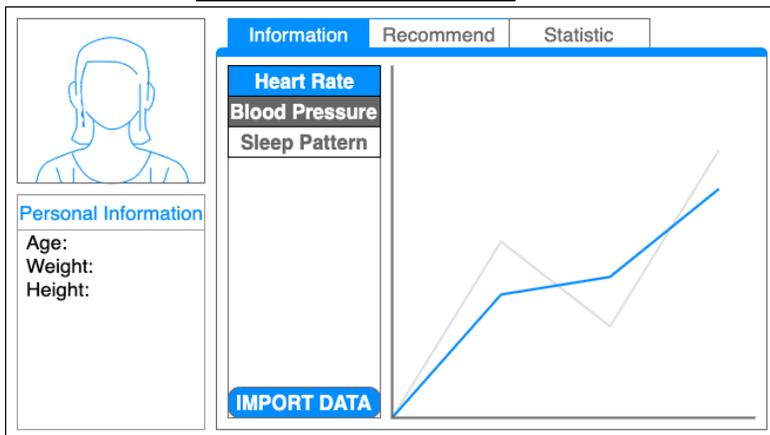
Navigation: total 1 mouse clicks, as follows

- a. Finish data entry in the new page
- b. Click “OK” to finish registration

Data Entry: total 2 keystrokes, as follows

- a. Press keys to input username
- b. Press keys to input password

3. Personal information

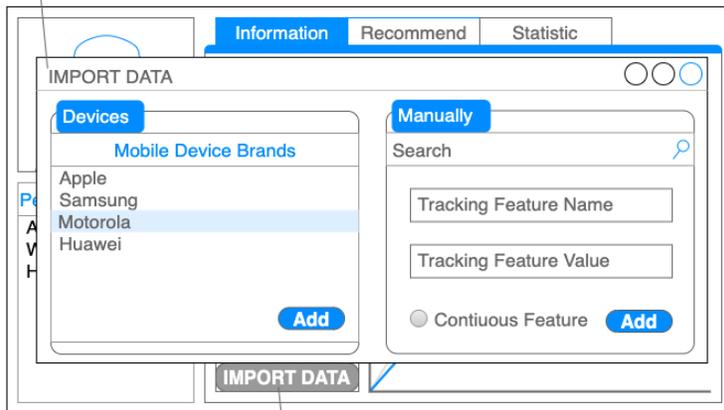


Navigation: total 2 mouse clicks, as follows

- a. Click “Information” page
- c. Click on whatever parameter that the user wants to see
- d. The static parameter will always be showed on the left

4. Data import

2. Pop up



1. Click

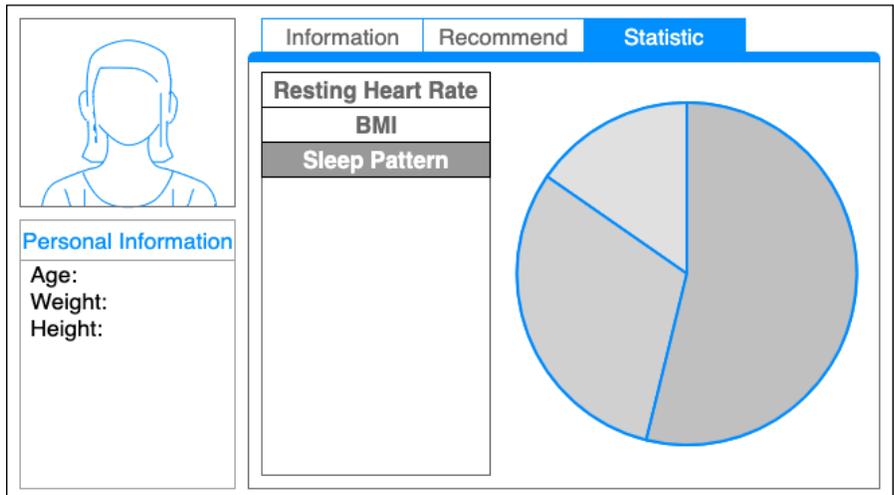
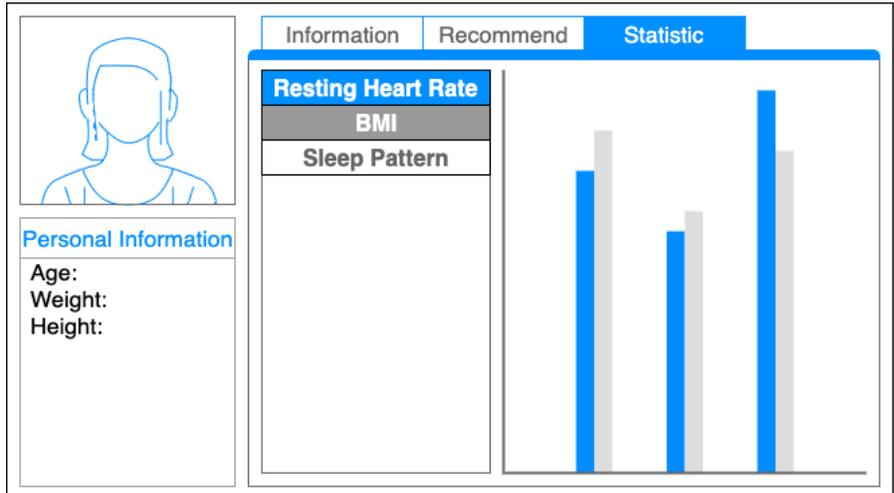
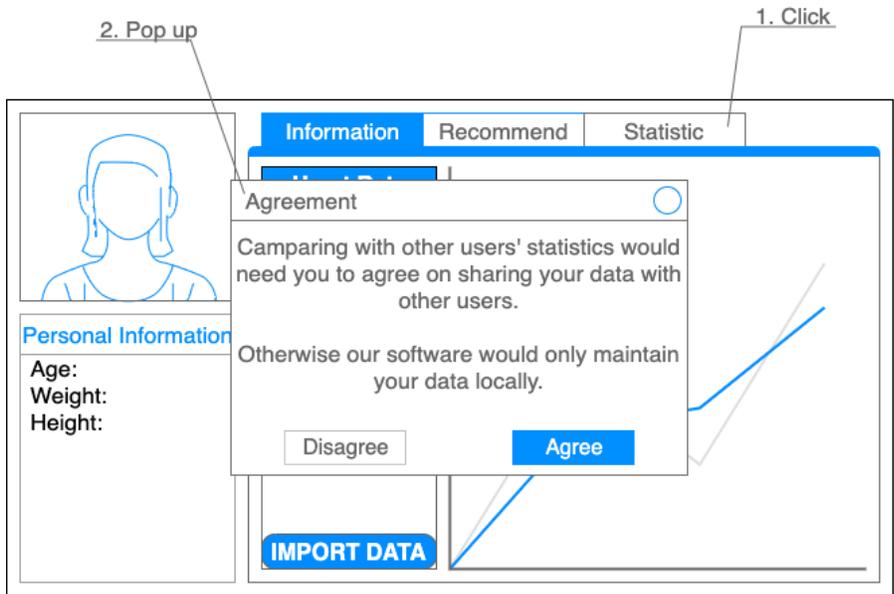
Navigation: total 2 mouse clicks, as follows

- a. Click "IMPORT DATA" button
- b. Finish data entry in the new page
- c. If the user has a health device, then select the right device to go further
- d. Click "Add" to finish data input

Data Entry: total 2 keystrokes and 1 mouse click, as follows

- a. Press the keys to input the parameter name
- b. Press the value of that parameter
- c. Click the continuous feature box if the parameter has continuous value

5. Statistics

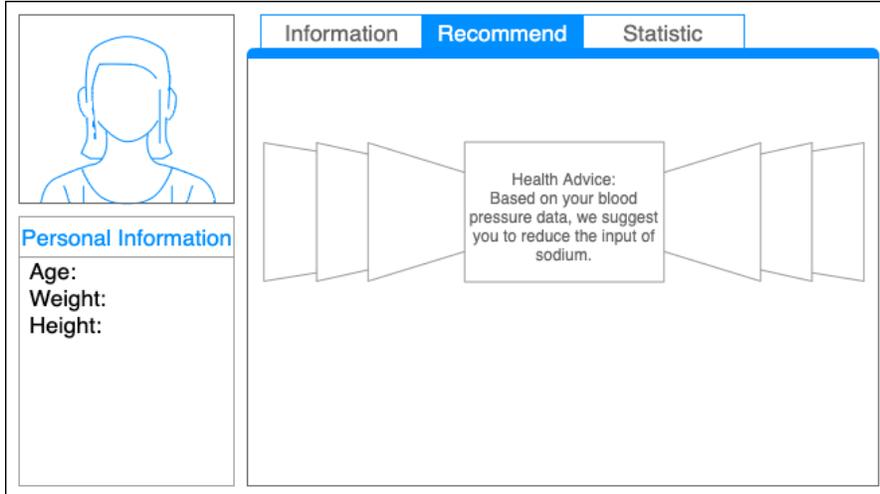


Navigation: total 3 mouse clicks, as follows

- a. Click "Statistic" page

- b. Finish agreement in the new page
- c. Click on the parameters that the user wants

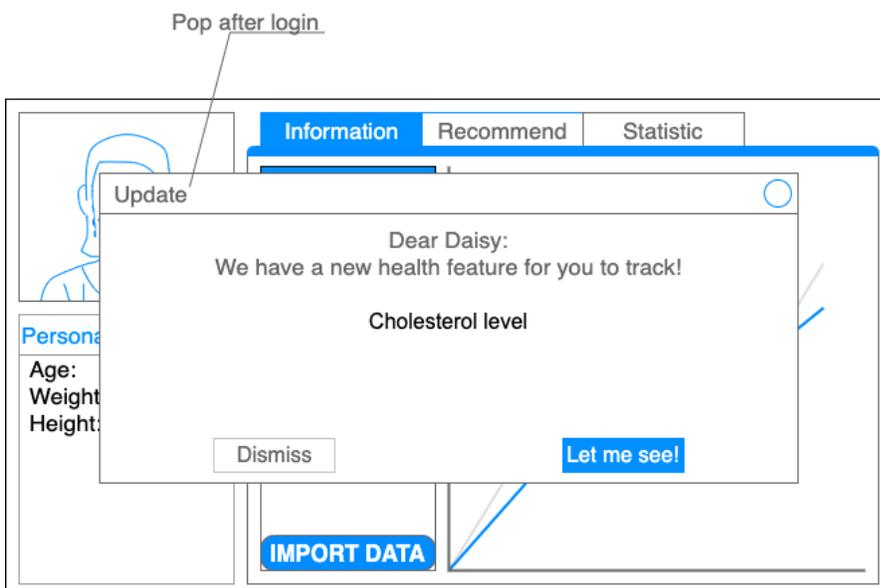
6. Recommendation



Navigation: total 1 mouse clicks, as follows

- a. Click "Recommend" page
- b. The recommendations will show on this page

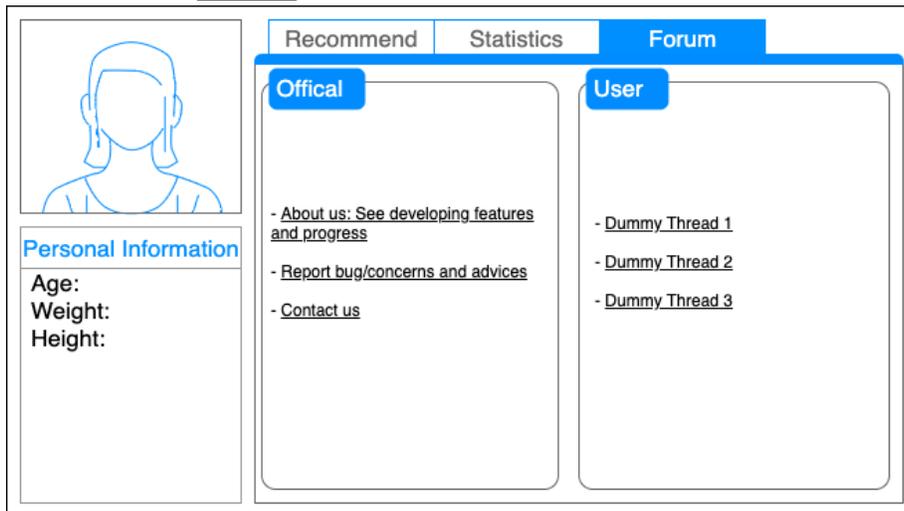
7. New update



Navigation: total 1 mouse clicks, as follows

- a. The new parameter update page will pop up after the user logged in
- b. Click "Dismiss" or "Let me see!" to move further

8. Forum



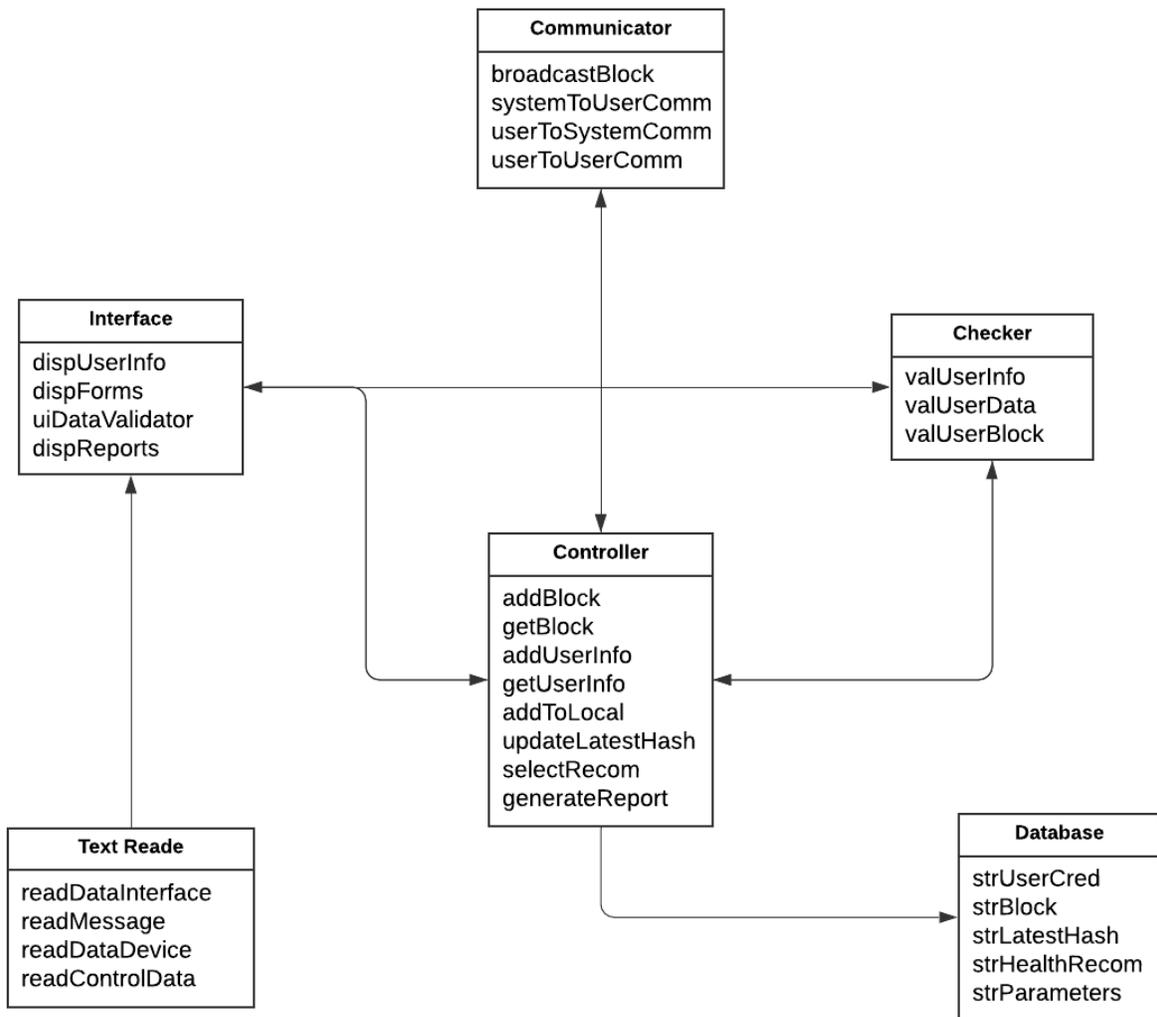
Navigation: total 1 mouse clicks, as follows

- a. Click "Forum" page
- b. The forum will show on this page.

IV. DOMAIN ANALYSIS

A. Domain Model

I. Concept Definitions



Responsibility	Type	Concept
R1: Display data and user information	D	Interface
R2: Display forms for user interaction	D	Interface
R3: Validation of input data before submission of web forms	D	Interface

R4: Display statistical and comparison report and recommendations	D	Interface
R5: Validate user entered user id and password	D	Checker
R6: Validate data entered by user (validate outliers)	D	Checker
R7: Validate new block from all the users	D	Checker
R8: Read health data, credentials and personal information of the user	D	TextReader
R9: Read messages sent by user to support team/other users	D	TextReader
R10: Read streaming data from the device (E.g. Apple Watch)	D	TextReader
R11: Read customized parameters for report generation	D	TextReader
R12: Broadcast hashed block and new parameter to all users	C	Communicator
R13: Send messages, recommendations, personal information, historical and comparison data to users	C	Communicator
R14: Send messages to system for block validations and feedback/requests	C	Communicator
R15: Send messages to other users	D	Communicator
R16: Add block to the database	D	Controller
R17: Get block from the database	D	Controller
R18: Add user information or get user information	D	Controller
R19: Get user information	D	Controller
R20: Add block to local blockchain	D	Controller
R21: Update latest hash in blockchain	D	Controller
R22: Select recommendation to user user	D	Controller
R23: Generate comparison report or historical report	D	Controller
R24: Store user credentials	K	Database
R25: Store blockchain data	K	Database
R26: Store latest hash for each user	K	Database
R27: Store health recommendations	K	Database

R28: Store existing parameters	K	Database
--------------------------------	---	----------

II. Association Definitions

Concept Pair	Association Description	Association Name
Controller → Interface	Controller gets request from the interface and sends the result for display	Send result
Interface → Controller	Reads user and device data input and messages	Read Input
Interface → Checker	Sends data to checker for verification (userID, data)	Check Info
Checker <-> Controller	Checker sends validation result to controller	Validation message
Communicator <-> Interface	Send and receive messages between users and system	Get message Send message
Controller <-> Communicator	Controller responds to requests from communicator	Get request Send response
Controller → BlockChain DB	Controller send appropriate data to DB. Controller acquires data from DB.	Generate requests Append data

III. Attribute Definitions

Responsibility	Attribute	Concept
R1: Display data , user information	dispUserInfo	Interface
R2: Display forms for user interaction	dispForms	Interface
R3: Validation of input data before submission of web	uiDataValidator	Interface

forms		
R4: Display statistical and comparison report and recommendations	dispReports	Interface
R5: Validate user entered user id and password	valUserInfo	Checker
R6: Validate data entered by user (validate outliers)	valUserData	Checker
R7: Validate new block from all the users	valUserBlock	Checker
R8: Read health data, credentials and personal information of the user	readDataInterface	Text Reader
R9: Read messages sent by user to support team/other users	readMessage	Text Reader
R10: Read streaming data from the device (E.g. Apple Watch)	readDataDevice	Text Reader
R11: Read control parameters for report generation	readControlData	Text Reader
R12: Broadcast hashed block and new parameter to all users	broadcastBlock	Communicator
R13: Send messages, recommendations, personal information, historical and comparison data to users	systemToUserComm	Communicator
R14: Send messages to system for block validations and feedback/requests	userToSystemComm	Communicator
R15: Send messages to other users	userToUserComm	Communicator
R16: Add block to the database	addBlock	Controller
R17: Get block from the database	getBlock	Controller
R18: Add user information	addUserInfo	Controller
R19: Get user information	getUserInfo	Controller
R20: Add block to local blockchain	addToLocal	Controller
R21: Update latest hash in blockchain	updateLatestHash	Controller
R22: Select recommendation to user user	selectRecom	Controller

R23: Generate comparison report or historical report	generateReport	Controller
R24: Store user credentials	strUserCred	Database
R25: Store blockchain data	strBlock	Database
R26: Store latest hash for each user	strLatestHash	Database
R27: Store health recommendations	strHealthRecom	Database
R28: Store existing parameters	strParameters	Database

IV. Traceability Matrix

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
TextReader	x	x	x	x				
Interface	x	x	x	x	x	x	x	x
Controller	x	x	x	x	x	x	x	x
Checker	x	x		x				
Communicator	x	x		x	x	x	x	x
Database	x	x	x		x	x	x	

SYSTEM OPERATION CONTRACTS

Register

1. PRE-CONDITION - No pre-condition
2. POST-CONDITION - The user is registered to the system.

Add Data

1. PRE-CONDITION - The user is registered.
2. POST-CONDITION - Health information of the user is stored by the system.

Comparison Data Presentation

1. PRE-CONDITION - The user is registered
2. POST-CONDITION - The user could see health parameters compared with population descriptors.

PLAN OF WORK

After submitting the first report for the project, we plan to implement the blockchain on a Windows platform for storing the health parameters for each user on the network.

In the first phase of our project, we are going to implement the basic blockchain-based health system. This phase is meant to form the most basic and essential functions of our system: adding data block to the main and local block-chains, communicating among users and supernodes, and simple user interface. We will add three sample users to the network. One of the three users can add their data to the blockchain and others can verify this data. If any outliers exist in the data entered by the user, the user will not be allowed access to the system. The data validator of the system will check if any data entered by the user is invalid and if so, an error message pops up prompting the user to re-enter his data. We are using the JavaScript and NodeJS, for the creation of the blockchain and backend part, where we can save the user's data and share it among other users. We will be finishing this phase by October 29th, 2019.

A supernode is made available within this network, which is used to store the blockchain data of all the users and serves as the database for the users. This will be a relational database using SQL.

In the second phase of the project, we are going to finish the user interface to facilitate users to register and login to our system. While registering with the system, the user is given an option to share his personal data with other users and only if he consents, the data is broadcast to the other users who can compare the data to verify it. Once the user logs into the system, his credentials are saved onto the system and he can update his health data. If the user adds new data, all the other users on the network will be notified. Users can also generate historical reports, which gives details of his past health data and they can also compare their data with the data of the other users. The second phase will include provisions for the data analysis that will be done on the data stored in the blockchain. Necessary upgrades will be made to the user interface. We will be using open-source library like Electron to create this sort of desktop application.

This phase will be completed by November 20th, 2019. By this time we will have the application that will be able to perform the majority of main functionalities that are desired from the application.

In the third phase, we will be addressing the scalability issues related to the network we have created. In other words, we will build the system so as to accommodate a large number of users on the network. We will also provide recommendations to users based on their present health conditions. Each metric will have its own priority depending upon the seriousness of the health conditions. For example: heart rate and blood pressure will have importance compared to sleep patterns. A forum for the users to communicate with other users will also be created during this stage.

If the user is facing any problems with logging into the system, a support system will be used to address the issues. A support system also addresses issues related to login, data updation, health parameter comparison etc.

The table below describes how we plan to move forward until the next report:

Task Name	Duration	Start
Proposal	7 days	22/09/2019
First Report	25 days	22/09/2019
<i>Part 1</i>	11 days	29/09/2019
Customer Statement	4 days	11/09/2019
System Requirements	4 days	15/09/2019
Project Management	3 days	18/09/2019
<i>Part 2</i>	7 days	06/10/2019
Functional Requirements	4 days	20/09/2019
User Interface Specification	3 days	22/09/2019
<i>Part 3</i>	7 days	13/10/2019
Domain Analysis	3 days	25/09/2019
Plan of Work	2 days	27/09/2019
Summary	2 days	28/09/2019

REFERENCES

1. "Use case", Wikipedia
http://en.wikipedia.org/wiki/Use_case
2. "System requirements", Wikipedia
http://en.wikipedia.org/wiki/System_requirements
3. "User interface specification", Wikipedia
http://en.wikipedia.org/wiki/User_interface_specification
4. "Health data", Wikipedia
https://en.wikipedia.org/wiki/Health_data
5. "Software Engineering book", Ivan Marsic
http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf
6. "Traceability matrix", Wikipedia
https://en.wikipedia.org/wiki/Traceability_matrix
7. "Requirement engineering" Wikipedia
https://en.wikipedia.org/wiki/Requirements_engineering