

**16:332:567 Software Engineering**

# **Biometric Health Monitoring**

**Project Website -** <https://sites.google.com/site/healthmonitorinsystem/>

**Group 3**

Project Members:

Ajinkya Bilolikar

Amanbir Singh

Jagbir Singh

Siddhesh Surve

Swapnil Sarode

Yanze Zhang

## Table of Contents

1. Individual Contributions Breakdown.....	3
2. Customer Statement of Requirement.....	4
a) Problem Statement.....	4
b) Glossary of Terms.....	8
3. System Requirements.....	10
a) Enumerated Functional Requirements.....	10
b) Enumerated NonFunctional Requirements.....	12
c) On-Screen Appearance Requirements.....	13
d) Acceptance Tests.....	14
4) Functional Requirements Specifications.....	17
a) Stakeholders.....	17
b) Actors and Goals.....	17
c) Use Cases.....	18
d) System Sequence Diagrams.....	26
e) Use Case System Sequence Diagrams.....	28
5. User Interface Specification.....	34
a) Preliminary Design.....	34
b) User Effort Estimation.....	40
6. Domain Analysis.....	45
a) Domain Model.....	45
b) System Operation Contracts.....	54
7) Plan of Work.....	57
8) References.....	58

## 1. Individual Contributions Breakdown

Responsibilities	Ajinkya Billolikar	AmanbirSingh Grewal	Jagbir Singh	Sidheesh Surve	Swapnil Sarode	Yanze Zhang	Total	Possible Points
Project Management	0	30	55	0	15	0	100	
Customer Statement of Requirements	25	0	0	50	25	0	100	
System Requirements	25	0	0	38	37	0	100	
Functional Requirements Specifications	17	25	15	20	23	0	100	
User Interface	0	15	15	0	0	70	100	
Plan of work	65	0	10	0	0	25	100	—

## 2. Customer Statement of Requirements

### a) Problem Statement

Biometrics devices can take unique information about you from your eye, or your hand prints, or your thumb prints and use it to identify you. This information can be used to obtain the valuable healthcare information. The personal health monitoring of each individual is considered very important because of rise in health problems in today's world. The increasing stressful lifestyle is taking maximum toll on the public health. With the ever increasing queues at hospitals and ever increasing number of patients, the doctor fees have sky-rocketed which is affecting especially those patients who cannot afford the fee or who are not suffering from major ailments but get to know so only after paying a hefty fee to the doctor. The researchers and surveys often demonstrate that most of the major health ailments are the result of careless attitude towards the minor health ailments. Majority of these issues can be solved by just following a good diet, proper sleep pattern and regular exercising. But how does a patient know what diet is good or what exercise he/she should follow and more importantly whether the plan that he is following is working effectively for him. The absence of such a mechanism makes the task of patient a difficulty, thus landing him with an option either to go to the doctor that means heavy fee or ignore the ailment that is more dangerous.

Moreover the increase in patients has also led to the decrease in the relative number of doctors per patient which results in vicious cycle where ignored or delayed diagnostics of an ailment makes the patient more dependent on doctor's check-up. But is it necessary that every time the user faces some issue or requires advice, that are not that serious, like somebody wants to get rid of some extra fat or somebody wants to improve his/her stamina, he/she is required to go to doctor?

Well, though it is advisable to visit the doctor whenever possible, but as discussed above if due to unavailability of specialist due to some reason, the Health Monitoring Devices offer and effective alternative. These days it is advisable to each individual to monitor and maintain good health by using biometric health monitoring devices and keep modifying their diet so as to improve their health stats. Thus there is a need for software that utilizes the data available from the device, uploads it to the website, gets feedback from the doctors via internet and show health reports. Doctor should be able to get data anytime he wants for analysis.

The biometric health devices can record the various data like metabolism rate, sleep hour, sedentary activity while being in contact with the user and this data from user can be used by physicians to recommend any changes to user's routine. Our project attempts to use the information obtained using such devices to give the detailed analysis of health of a patient/individual that can help in getting a prompt and timely advice from a doctor.

Currently healthcare monitoring is extensively doctor depending .Our System is basically designed for those customers who can self-monitor their health indicators to check the progress that they are making through a diet plan change or exercise routine modification. An option will also be provided to customer to send the report to the doctor for expert advice in case the customer feels he/she is not completely satisfied by the diet routine, sleep hours or any other general habit he is acquiring. Different types of health indicators would be used for the analysis.

### **Assumption:-**

1. The User already possess a biometric health device that would be used to record the health parameter. These recorded parameters are internal to the device and using our application software user would be provided with some real time user friendly parameters that are defined under parameters sections and these parameters would be displayed on user screen upon request.
2. This device need to be worn by the user/patient on his arm for a particular duration of time, for which he wants his health parameters to be monitored.
3. This device is capable of recording physical activity parameter based on which user would be provided with the information mentioned in section so and so.
4. User would be able to connect this device to his/her PC and would be able to generate the excel sheet in agreed format using the device software, this excel would be kept in user machine and application developed by our team will process data in this excel once after uploading it.

### **Device Operation:-**

The device used during this project is The Sense Wear Armband. The user will be wearing this device for some period of time. This device contains sensors that collects the body movements of the user and record data based on those body movements and body temperature and sweating to provide information about what type of activity user is doing. Details of device monitoring are given below:-

- Motion: The Armband contains an accelerometer, a device that measures motion.
- Steps: The Armband counts your steps, using the accelerometer to measure the distinct patterns created by walking and/or running.
- Galvanic Skin Response: This measures the electrical conductivity of the skin, which changes in response to sweat and emotional stimuli.
- Skin Temperature: A sensitive electronic thermometer measures the temperature of the skin.
- Heat Flux: Measures the amount of heat dissipating from the body.

Once the user has worn the device and he/she wishes to get his activity log over a period of time using some more real-life parameters, user can connect this device to his Personal computer and the device software (the software internal to the device and not this project software) will provide user with an excel file that would contain the information recorded by the device. This information contained in the excel would be used by our application to display the parameter explained later in this text.

Once the user has the generated this excel, the use would log into his account on our application and reach his/her homepage. User will have an option to upload this excel and once the excel is uploaded to proceed with the execution to process the information. Once the processing is done the user will be provided with the information mentioned in next section.

#### **Parameters provided :-**

1. **Total energy Expenditure:** - This field will display the total no. of calories burned by the user over a period of time. The vigorous activities would lead to faster rate of burning and this rate would be depicted better through the graphical representation.
2. **No. of Steps :** - Total no. of steps taken by the user over the window period of wearing the device.
3. **Lying Down Time:** -Total no. hours the user has spent lying down. This time will also include the sleeping time
4. **Sleeping time:** - The total time that user has spent while sleeping.
5. **Sedentary:** - The total time spent while sitting by the user.
6. **Physical Activity Duration:** - The total time for which the user performed the physical activity over a period of time. Any activity with 3 MET(see glossary) would be considered as Physical Activity.

The System will basically consists of three different users that would be interacting with the aim of providing better healthcare service through mutual utilization of self-monitoring and the consultation from a specialist. The mutual interaction exists because user will have the option of sending the data for analysis to the doctor, getting the feedback and then acting on his advice. The three users are:-

1. Patient
2. Doctor
3. Administrator

The Patient will register himself with system initially by providing the various personal details that includes his name, age, sex ,etc., along with the doctor's name with which he wants to consult ,if at all. The registration part is mandatory before the user is able to use the system for his task completion. Then the user proceeds with login using his/her unique id and password. Upon logging into the application the user will have a window screen with few options. One of the option user will have is to upload the file that the system will be using to display the data to the user interface. This data file is generated as a result of the data collected by the health monitoring device that the user wears on his arm-band for a particular duration of time. This file can be uploaded into the user desktop by connecting device through Bluetooth or USB cable. It is assumed that the data would be available in the Excel File. The data in this file is not present in user friendly manner and therefore is not of much use to the customer. The application that is being developed will make this data suitable for user display and analysis and also make provisions to send the data to the doctor. Once this file is uploaded into the system, system extracts the data. This would result in the user getting a clear picture of all the parameters which were recorded by the device over that particular period of time.

The total duration that the user slept during that time would also be provided, including other activities like number of steps taken, that is especially useful for patients aiming at weight reduction monitoring. The working professionals whose workplace activities involve lot of sedentary activities would be interested in knowing the time that they spent sitting. The Sedentary parameter would be available for the user to monitor that. Such diversified parameters would make the application cater to the requirements of various categories of people, thus achieving higher customer base. The customer should have an additional feature to view the whole data captured in a graphical form for a better analysis as we believe graphs are much easier to analyze and understand as compared to other forms, hence making the customer's and doctor's job much easier.

The role of doctor is vital to any healthcare process and we would provide the provision in our project to get doctors feedback .The patient should be able to send the report to the doctor, who was specified by the customer during his/her registration. But this functionality should be entirely upon the discretion of the user that whether he/she chooses doctor's analysis or not. If the customer feels, he/she is satisfied with his improvement and performance after monitoring the data provided by our application, then patient can logout of the system.

The Doctor should be able to register for the application using the same process and then log in the system using his unique credential. Once the doctor gets logged in, he/she should be able to see the data of a patient through a user interface that allows him to select the patients. Once the patient gets selected the data should be available for the doctors analysis as it was presented to the customer, with the exception that doctor should be able to add his/her comments or feedback after seeing the details of the patient thus issuing his/her advice. It must be noted that the data available for monitoring should not be editable either by the patient or the doctor. Once the doctor's feedback is posted, user/patient would be able to see that once after logging into his account.

**Doctor Analysis and Feedback :-** The doctor would analyze the parameter mentioned in above specific to particular patient. Doctor would take into consideration the information of patient medical history maintained in patients profile during the registration of patient.

*For Ex.:- A patient is suffering from obesity and is undergoing process to reduce his/her weight, the doctor would check the weight from user profile and compare the amount of physical activity that user is undergoing. Based on whether the patient is doing enough or less, doctor can suggest his/her advice. For this doctor would be provided a field in his profile and after checking the user data he/she would be able to enter the prescription/advice in that field and send to the user. This would provide for an efficient and productive communication between the patient and doctor.*

The role of administrator should be limited to the validation of authenticity of the patient or the doctor. The parameters that admin would use for validating the authenticity of the doctors are external to the project and not in its scope.

The designed application will be extremely user friendly since the users can be of any age. Simplicity in presenting the application to the user will be the key to success of this system.

## **b) GLOSSARY OF TERMS:**

**User-** A person who wants to upload his personal health information on the web site.

**Doctor-** A person who monitors the health information of the user and provides feedback to the user.

**Database** – Entity that stores all the system's information.

**Administrator-** A person responsible for the installation, configuration, upgrade, administration, monitoring and maintenance of databases in an organization.

**Website** – An interface that the user and doctor can use to register, and upload or download the health information.

**Graph-** A graph is an abstract representation of a set of objects where some pairs of the objects are connected by links.

**Report-** A report is a textual data of the user's health made with the specific intention of relaying information or recounting certain events in a widely presentable form.

**Acknowledgement-** A feedback given by the doctor to the user to make changes in his/hers diet.

**Software application-** An application is computer software designed to help the user to perform specific tasks which manage and integrate a computer's capabilities.

**Web Server-** a server is a physical computer dedicated to running one or more services to serve the needs of the users of other computers on the network.

**Unique ID-** Identification is a process whereby the subject assimilates an aspect, property, or attribute of the other and is transformed, wholly or partially, after the model other provides.

**Diet-** the sum of the food consumed by an organism or group.

**Heart Beat-** Heart rate is the number of heartbeats per unit of time, typically expressed as beats per minute (bpm).

**Metabolic Rate-** It is the amount of energy expended daily by users at rest or sleep .

**Energy Expenditure-** It refers to the amount of energy (calories), that a person uses to breathe, circulate blood, digest food, and be physically active.

**Speed-** It is the distance travelled per unit time.

**Number of Steps-** It is the total number of steps travelled by the user to cover a particular distance.

**Threshold-** A limiting value of a parameter below which the health of the user gets degraded.

**MET :** - The **Metabolic Equivalent of Task (MET)**, or simply **metabolic equivalent**, is a physiological measure expressing the energy cost of physical activities and is defined as the ratio of metabolic rate (and therefore the rate of energy consumption) during a specific physical activity to a reference metabolic rate

**Alert-** A message which a doctor receives when a particular parameter of a user falls below the threshold value.

**Emergency Services-** are organizations which ensure public safety and health by addressing different emergencies.

### 3. System Requirements

#### a) Enumerated Functional Requirement

REQ1a	5	The system shall allow new users to register an account on the website.
REQ1b	5	The doctor and the patient have to register themselves with the administrator.
REQ2a	5	The system should provide a specific Login ID to every Registered user and doctor.
REQ2b	5	The system should be able to differentiate between the users and doctors while login.
REQ2c	4	The system shall identify the user based on his login ID and Password.
REQ3a	3	The system should be able to upload the data in excel form using a user interface upon user request and process it.
REQ3b	4	The system should be able to convert the raw data from excel and store it in the database.
REQ4a	2	The system should be able to display the data to the user in a tabular form. This data would include energy expenditure, metabolic rate, sleep rate, etc.
REQ4b	2	On user request the System should be able to display the graphs corresponding to data displayed in the REQ7
REQ4c	1	The system should provide the user with the option to save the data on his PC.
REQ5a	4	The system should send the data to the doctor.
REQ5b	3	The user should provide the doctor name he wishes to send the data to during registration.
REQ6a	5	The system should display the list of all the users mapped to a particular doctor.
REQ6b	5	The system shall allow the patient data to be received by only one doctor to avoid conflict of feedbacks from doctors.
REQ6c	5	The system shall allow the doctor to view only one patient data at one point

		of time.
REQ7a	4	The system should display the data of the selected patient.
REQ7b	3	The data can be either in tabular or graphical format.
REQ8a	3	The system should check if the feedback from the doctor is available for the particular patient.
REQ8b	5	The system should be able to display the feedback from doctor to the patient.
REQ8c	5	The system should allow the patient to view the feedback written by the doctor.
REQ9	3	The doctor should inform the emergency services if patient data crosses the threshold value.
REQ10	5	The system should not allow the patient or doctor to modify the data.

The REQ1a has a high priority as only registered patients and doctors will be able to communicate with the system. The REQ2a also has a high priority as unique identity has to be given to the users. REQ2b is paramount as during the login, the system has to be able to differentiate between a patient and a doctor. Depending on the type of user separate windows and functionalities are provided. REQ3a has a lower priority because excel sheet is not the only application to store the patient data. REQ4c has the least priority since the patient being able to save the report to this pc is not our main agenda. REQ9 has been given a lower priority as we may have to include its functionality depending on the time and resources available. REQ10 implements that the users should not be able to change the data as it would result in the incorrect analysis of the patient information.

**THE NOT LIST-What This Project is not about-** In this project we only work on the information captured by the health monitoring device. We are not concerned with how the device works or how the user actually manages to upload his data onto the system. We also do not pass the patient record to all the registered doctors. Only the doctor whose name has been entered by the patient gets the patient record. The doctor does not receive an alert on his mobile phone whenever a patient uploading his data, references that particular doctor. He only gets to know about it when he logs into his account.

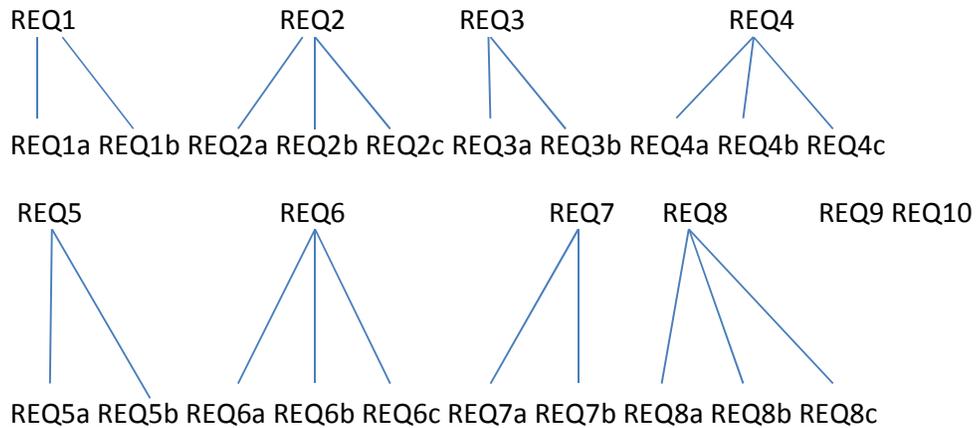
### **Business Policy:**

**TTT-BP01:** No more than one patient data can be viewed at a time by a doctor.

While the doctor is viewing a patient's record, he must not be allowed to open another patient's record. Only after the doctor has given his feedback for the current patient can he access the next patient's record.

**TTT-BP02:** The patient can view the feedback only after the doctor uploads it.

If the doctor still has not prescribed his medication then the patient should see “no comments available” on his screen.



**b) Enumerated Non Functional Requirements:**

ID	PW	Requirement
REQ11	5	Authorized access to user is provided as a patient, doctor or System Administrator
REQ12	4	User is allowed to change the password and notified when he attempts to change his password.
REQ13	3	User is logged, off if he is idle for most of his login duration.
REQ14	2	User satisfaction is ensured by providing quick computing, and importing and exporting data.
REQ15	4	The number of transactions per hour does the system need to be able to handle
REQ16	4	The amount of data does the system need to be able to store

**c) On-Screen Appearance Requirements:**

ID	PW	Requirements
REQ17	5	The system is designed such that it can work on Windows, iOS and Linux operating systems.
REQ18	3	The user is shown his expected results onscreen as well as provided with an option for printing.
REQ19	2	The system must have a consistent look across different browsers and screen resolutions.

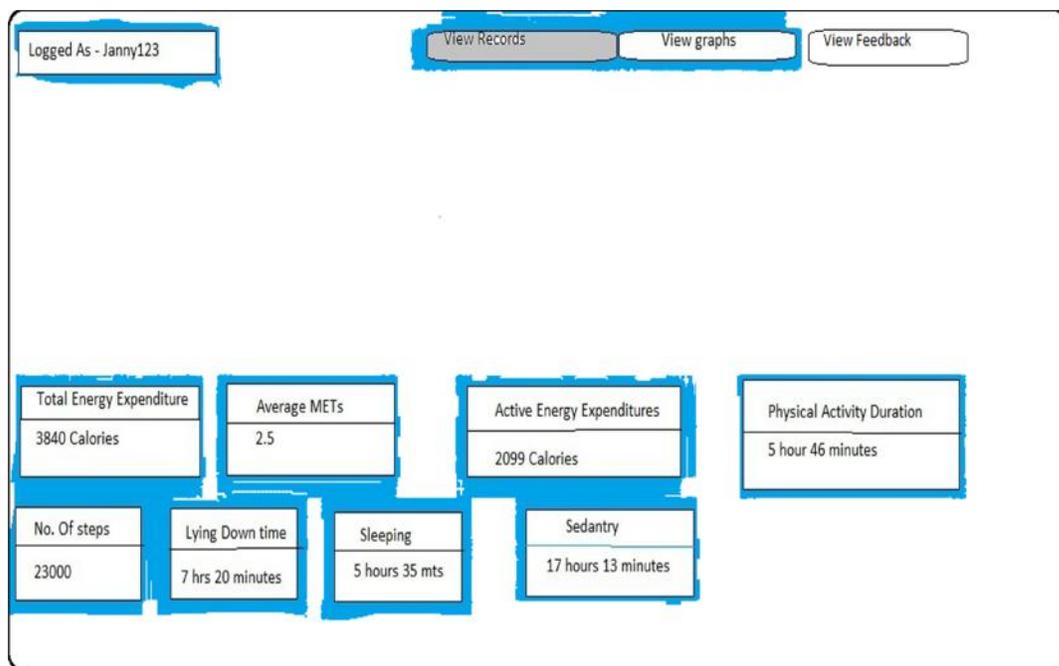


Figure above shows a customer-provided initial sketch of the user interface appearance. This appearance pertains to patient view of homepage. The screen Homepage contains different tabs displayed as explained below.

**Tabs:-**

1. **View Records** : - This tab will enable the user to see the records in tabular form. This is also the default view and the records would be displayed in this form until the user choose to go to graphical form.
2. **View Graphs** : -This tab will show the user parameters in graphical form to provide a better idea of user activity over a larger span of time.
3. **View feedback** :- This button is used by the patient for viewing the feedback of the doctor on his reports.

**d) Acceptance Tests****Acceptance test case for REQ1a:**

ATC 1a.01: Ensure connectivity between user and network which hosts the webpage.

ATC 1a.02: Ensure connectivity remains intact until registration of user completes.

**Acceptance test case for REQ1b:**

ATC 1b.01: The system must make sure that all the required fields are filled up the registering user.

**Acceptance test for REQ2a:**

ATC 2a.01: Every user (patient or doctor) should be assigned with a unique Login ID to prevent overlapping of login information.

ATC 2a.02: Ensure that the user's password is associated with his respective Login ID.

**Acceptance test for REQ2b:**

ATC 2b.01: Differentiate between a Doctor, a User or an Administrator during login.

**Acceptance test for REQ2c:**

ATC2c.01: After login ensure that correct user profile is selected based on login information.

**Acceptance test for REQ3a:**

ATC 3a.01: Ensure every user can upload his data in excel and process the needed parameters from the user interface.

ATC 3a.02: Ensure that the data uploaded by a user is associated with his profile.

Note that the process in ATC 5.01 can be a process like check how many hours of sleep the user had or keep a tab on sedentary work hours. The profile of user in ATC 5.02 is his registration information.

**Acceptance test for REQ3b:**

ATC 3b.01: Ensure that the data from excel is entered into the database where it is stored for analysis.

**Acceptance test for REQ4a:**

ATC 4a.01: Ensure validity of data stored display it to user and get confirmation from the user.

ATC 4a.01 can be a simple test where the user confirms that the data entered by him which was later transferred to the database is the same and correct data.

**Acceptance test for REQ4b:**

ATC 4b.01: Ensure correct graphical representation by displaying graphs of parameters requested by the user.

**Acceptance test for REQ4c:**

ATC 4c.01: Ensure backup of the data by allowing user to have one, for future references.

**Acceptance test for REQ5a:**

ATC 5a.01: Ensure that the doctor associated with users profile receives the data when user sends it.

**Acceptance test for REQ5b:**

ATC 5b.01: Ensure that the patient selects the doctor name from the drop down menu.

**Acceptance test for REQ6a:**

ATC 6a.01: Ensure that the list of users handled by a doctor is updated every time there's a new registration with the same doctor.

ATC 6a.02: The Doctor should be able to view any of his user's (patient's) data.

**Acceptance test for REQ6b:**

ATC 6b.01: Ensure that the patient data is received by only one doctor.

ATC 6b.02: The doctor's comments are unique to that patient so only one doctor shall comment for a particular patient.

**Acceptance test for REQ6c:**

ATC 6c.01: Ensure that only one patient data can be accessed at a time by the doctor.

ATC 6c.02: The doctor should write his comments to the current patient before being allowed to access the next patient.

**Acceptance test for REQ7a, REQ7b:**

ATC 7.01: Ensure that the doctor is viewing the data of the correct user.

ATC 7.02: Doctor should be able to choose whether to view data in tabular format or graphical format.

**Acceptance test for REQ8a:**

ATC 8.01: Ensure that the doctor gives feedback to the correct user.

The test confirms that the comments made by doctor on profile he is currently working on are linked to the correct user.

**Acceptance test for REQ8b:**

ATC 8a.01: Ensure that feedback provided by doctor is associated with user under observation.

**Acceptance test for REQ8c:**

ATC 8c.01: Ensure that user gets the feedback provided by the doctor.

**Acceptance test for REQ10:**

ATC 10.01: Ensure that data is not altered once stored on the database either by the patient (user) or by the doctor.

## 4. Functional Requirement Specifications

### a) Stakeholders

- Patient
- Doctor
- Administrator
- Emergency Services
- Web page designers
- Database Managers

### b) Actors and Goals:

#### ❖ Patient-

- Performs his daily routines with a device attached to his body.
- Registers himself with the acknowledgment from administrator.
- Uploads the data collected by the device into the database.
- Views his health readings.
- Views the suggestions prescribed by doctor.

#### ❖ Doctor-

- Registers himself with the acknowledgment from administrator.
- Views the patient record uploaded.
- Monitors if the patient condition is improving or not.
- Prescribes solution to the patient by writing a report.
- Provides patient with a graphical visual of his readings.

#### ❖ Administrator-

- Stores patient and doctor details while registering.
- Creates the database tables
- Upgrades the application
- Monitors the application

**Participating:**

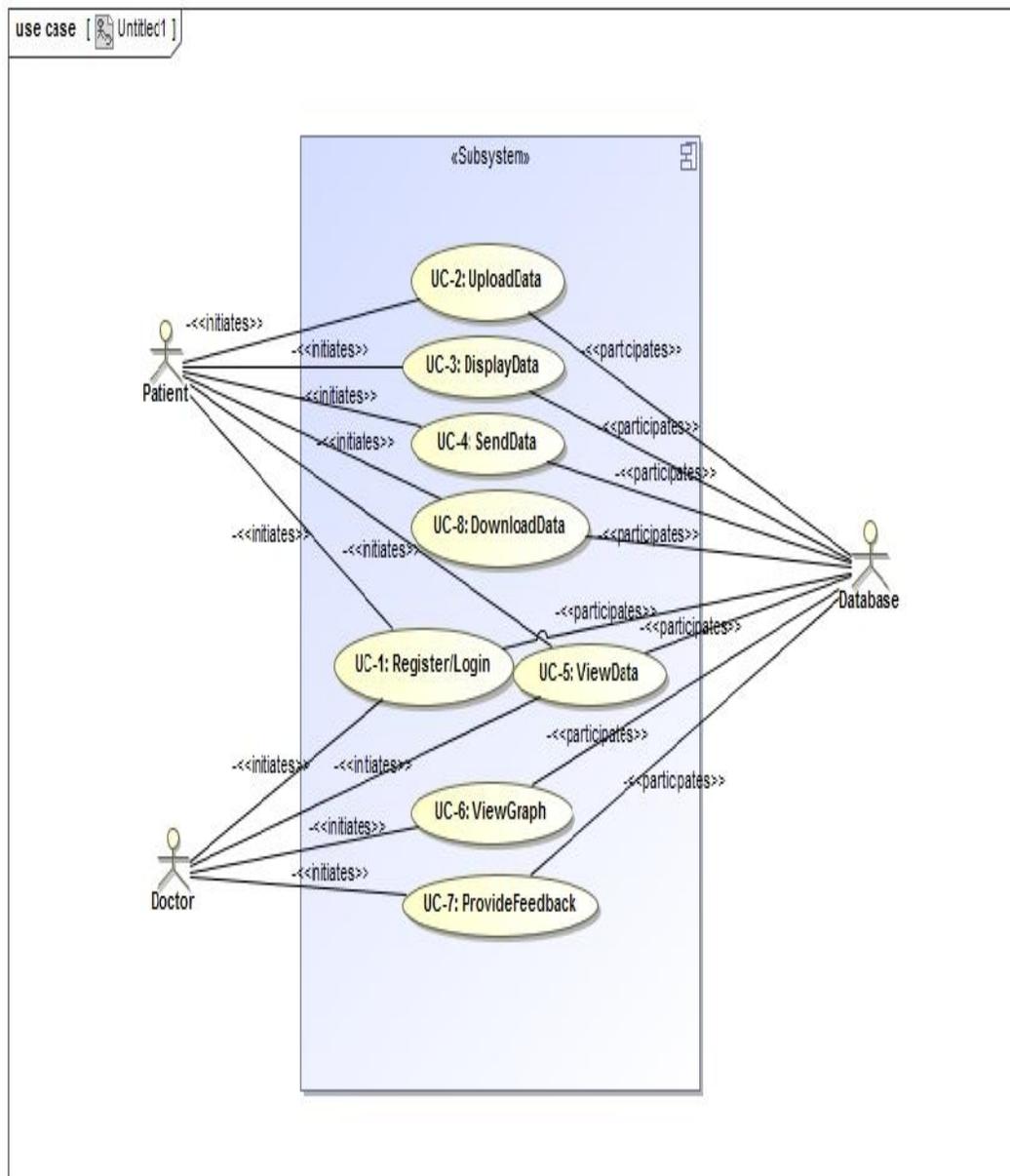
- ❖ Device-
  - Captures the patient activity and stores it in the form of data.
- ❖ Web Page
  - Allows the doctor and patient to register with the system.
  - Allows the doctor and patient to interact.
- ❖ Database-
  - Maintains a record of the patient registration.
  - Maintains a record of the doctor registration.
  - Maintains a record of the patient data and doctor feedback.

**c) Use Cases****i. Casual Description:**

Use Case	Name	Description	Requirement
UC-1	Register/login	To Register a user/doctor to the website	REQ1a, REQ2a, REQ2b, REQ2c, REQ11, REQ12, REQ13, REQ15
UC-2	UploadData	To upload the raw data file for system to proceed with extraction	REQ3a, REQ3b, REQ16
UC-3	DisplayData	To display the data like energy expenditure, sleep rate, etc. in user friendly manner	REQ4a, REQ4c, REQ10, REQ17, REQ18, REQ19
UC-4	SendData	To send the health information to the doctor	REQ5a, REQ5b, REQ14
UC-5	ViewData	To view the health information of a user by the doctor	REQ7a, REQ7b, REQ6a, REQ6b
UC-6	ViewGraph	To plot the graph of the data	REQ4b, REQ7a, REQ7b
UC-7	ProvideFeedback	To provide the desired feedback to the user	REQ8a, REQ8b
UC-8	Downloaddata	To get the feedback	REQ8c

		from the doctor and save it.	
--	--	------------------------------	--

ii. Use Case Detailed Description:-



**Use-Case Diagram**

**Use-case Details** :- As in UC1, User will try to login the application with his/unique username and password if he is already registered with the system. The application/browser will communicate with the database to authenticate the user. If user is authentic he/she will be allowed to continue his/her session. If the credentials are mismatching, the user will be returned with an error saying the the username and password do not match. The new users will have to register with the application and information will be stored in the database. Like UseCase 2 shows correctly logging in with username and password, the user will be able to upload the data in excel sheet. This excel should be in proper format as sepecified with device specifications in customer requirements. This data uploaded will be saved in the system and the acknowledgment would be sent to the browser that the data is correctly uploaded and saved in the system. In case of failure, proper error would be reported. The user would be able to process the file and all the specified parameters would be shown on user screen. In Use case4, the user after viewing his health parameters would be able to send his report to the doctor for getting feedback on his progress. User will choose the button and send report and all his details would be sent to the doctor. A message acknowledging the data sending would be sent to the patient screen. In case of failure the reported error with failure message would be displayed. Doctor would be able to see the data in tabular and graphical forms as depicted in use case diagram in Use Case 5 and 6 .Once after the data is analysed by the doctor , the doctor will write his feedback comments in the field provided for feedbacks. After entering the comments, when saved. the database would be updated with the doctors comments and are read for user view. User would be able to see this feedback in his profile after logging in.

#### ❖ **Use case 1: Register/login**

**Initiating Actor-** User

**Actor Goal-** To register a new user, user should be able to register himself with the application

**Requirement Addressed** - REQ1a, REQ2a, REQ2b, REQ2c, REQ11, REQ12, REQ13, REQ15.

**PRE-CONDITION-** No precondition

**POST CONDITION-**The user is successfully registered with the system

**Flow of events :**

- The user, if new, requests to registers himself with the system
- ←The system asks for the personal details of the user.
- User enters the personal details including chosen credentials, i.e Username/password and submit the registration form
- ←System differentiates between different users, and accordingly saves them in the system database and registration is successful.

**❖ Use case 2: Upload Data**

**Initiating Actor-** User

**Actor Goal-** To upload the raw data file in the system so that data can be extracted to a user friendly form.

**Requirement Addressed-** REQ3a, REQ3b,REQ16

**PRE-CONDITION-** User is already registered

**POST CONDITION-**The raw data file is uploaded and ready to use for the system.

**Flow of events:**

- user logs into the account
- user request the system to upload the raw data file from the local upload address
- ←the file is successfully uploaded into the system

**❖ Use Case 3: Display Data**

**Initiating Actor-** User/Patient

**Actor Goal-** To extract the data from the user uploaded file and display it in user-friendly manner

**Requirement Addressed -** REQ4a,REQ4c, REQ10,REQ17,REQ18,REQ19

**PRE-CONDITION-** the user has already uploaded the raw data file to the system

**POST CONDITION-**the data is displayed to the user on screen

**Flow of events**

- the patient triggers the data extraction from uploaded raw file by clicking the execute button
- ← system displays the data for user analysis in tabular form
- user requests data in graphical form
- ← data is displayed in graphical form

**❖ Use Case 4: Send Data**

**Initiating Actor-** User/Patient

**Actor Goal-** To send the data displayed in UC-3 to the doctor for analysis

**Requirement Addressed-** REQ5a,REQ5b,REQ14

**PRE-CONIDTION-** the user has already extracted the data and data has been displayed on the system

**POST CONDITION** - data displayed will be saved in the database for doctors analysis

**Flow of events:**

- the patient opts for sending the data to the doctor
- ← system saves the data for the customer

**❖ Use Case 5: View Data**

**Initiating Actor:** Doctor

**Actor Goal:** To analyze the data reported by the user through application and user interaction

**Requirement Addressed:** REQ7a,REQ7b,REQ6a,REQ6b

**PRE-CONDITION:** Patient has requested the data to be sent to the doctor and asked for feedback.

**POST CONDITION** – The Doctor Selects the desired patient for providing feedback.

**Flow of events**

- The doctor registers his account on the webpage.
- The doctor makes login to his account by using his login ID and password.
- The doctor selects the patient from an available list of patients.

❖ **Use Case 6: View Graph**

**Initiating Actor:** Doctor

**Actor Goal:** to analyze the data reported by the user through application using graphical representation.

**Requirement Addressed:** REQ4b, REQ7a,REQ7b

**PRE-CONDITION:** The data for specified patient is already present in tabular form.

**POST CONDITION:** The data is displayed using the graphical representation

**Flow of events**

- Doctor requests the data of that patient to see.
- ←Data is displayed in tabular form.
- Doctor requests the data of that patient in graphical form.
- ←Data is displayed in graphical form after analysis.

❖ **Use Case 7: Provide Feedback**

**Initiating Actor:** Doctor

**Actor Goal:** To provide the feedback on user health after analyzing the data available for that patient in tabular and graphical form

**Requirement Addressed:** REQ8a,REQ8b

**PRE-CONDITION:** The data for specified patient is already present.

**POST CONDITION:** Feedback comments would be saved in the system for patient use.

**Flow of events**

- The doctor analyze the data using different forms.
- Doctor posts his/her feedback comments after the data analysis
- ←Feedback is saved in the database.

**❖ Use Case 8: Download Feedback**

**Initiating Actor:** Patient

**Actor Goal:** to see the feedback on user health after analyzing the data available for that patient in tabular and graphical form

**Requirement Addressed:** REQ8c

**PRE-CONDITION:** The Feedback comments for specified patient is already present in system

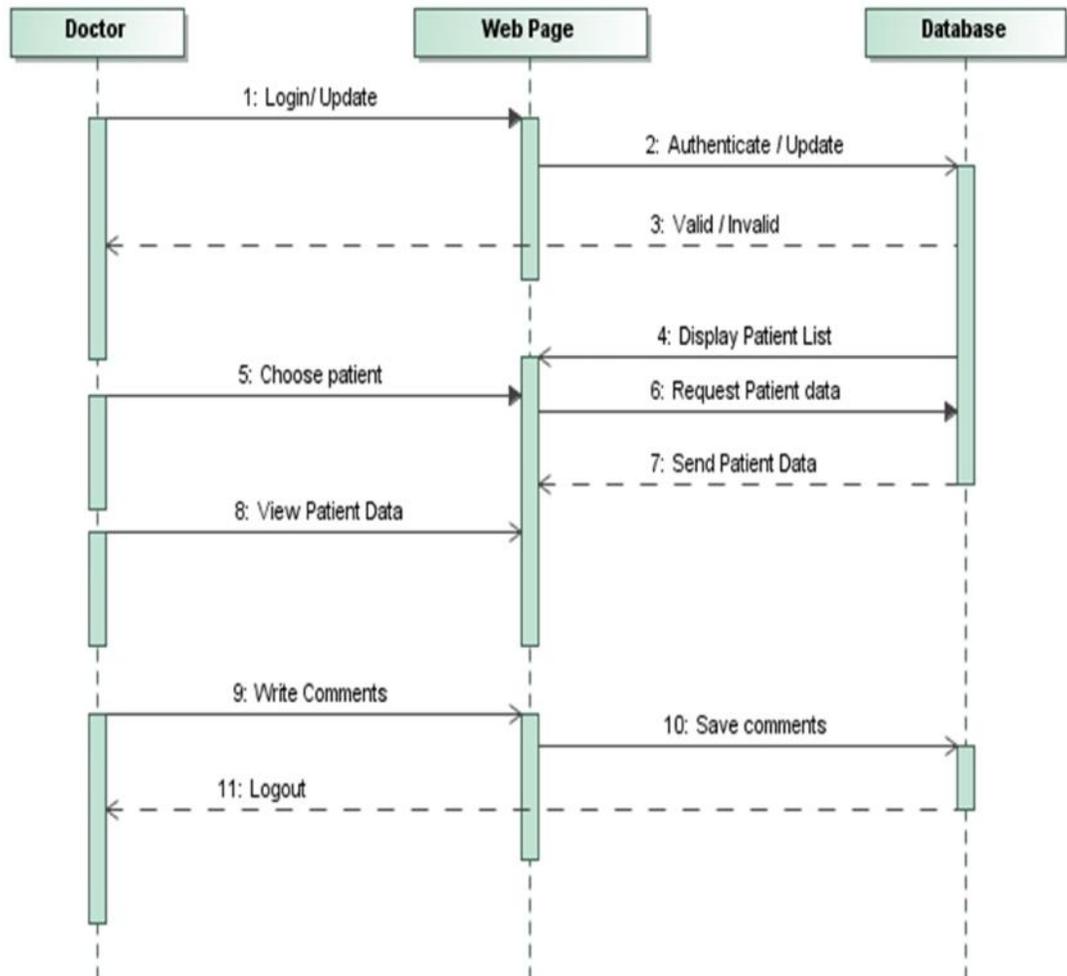
**POST CONDITION:** Patient would be able to see the feedback comments through user interface.

**Flow of events**

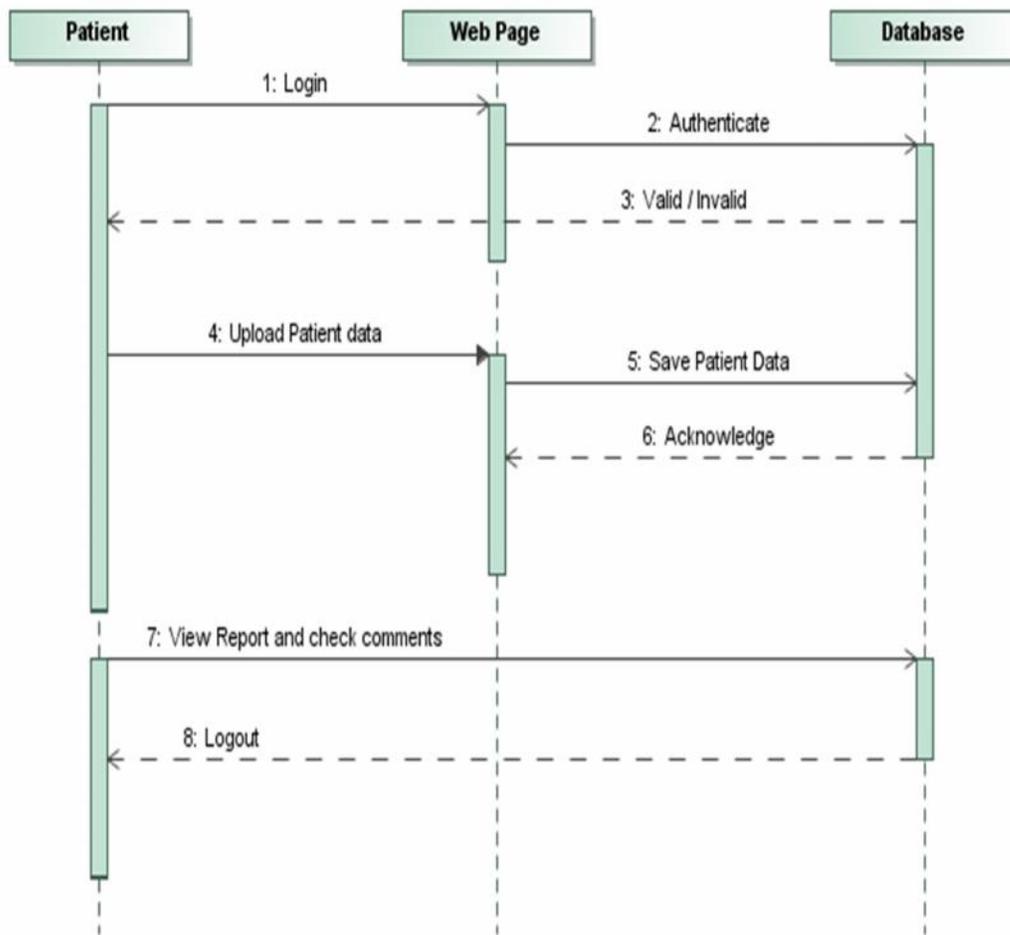
- The user requests to see the feedback comments.
- ←Feedback comments are displayed on the user interface.

**iii. Traceability Matrix:**

	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8
REQ1a	X							
REQ2a	X							
REQ2b	X							
REQ2c	X							
REQ3a		X						
REQ3b		X						
REQ4a			X					
REQ4b						X		
REQ4c			X					
REQ5a				X				
REQ5b				X				
REQ6a					X			
REQ6b					X			
REQ7a					X	X		
REQ7b					X	X		
REQ8a							X	
REQ8b							X	
REQ8c								X
REQ10			X					
REQ11	X							
REQ12	X							
REQ13	X							
REQ14				X				
REQ15	X							
REQ16		X						
REQ17			X					
REQ18			X					
REQ19			X					

**d) SystemSequenceDiagrams**

**Sequence followed for User: Doctor**



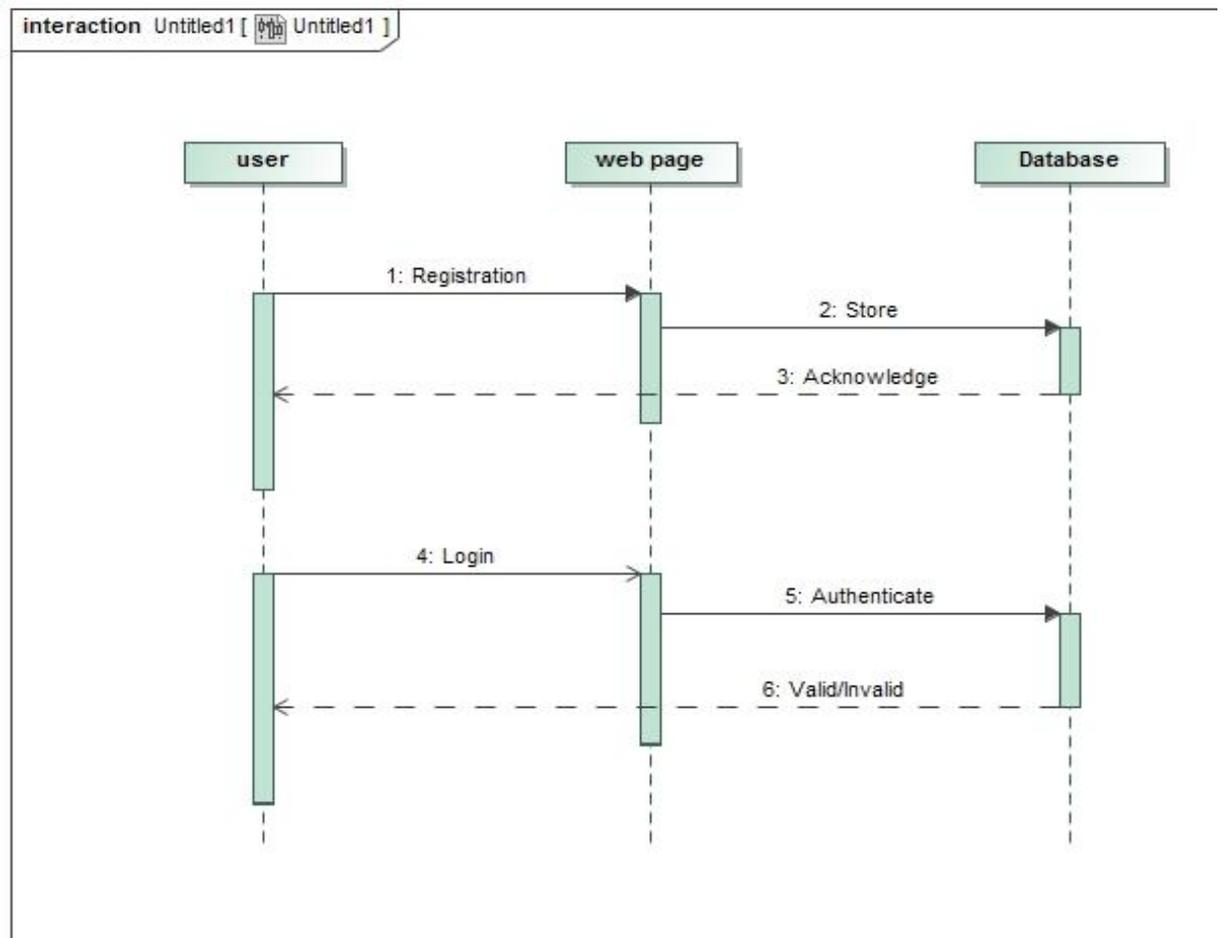
**Sequence followed for User: Patient**

## e) Use Case System Sequence diagrams

### Use case 1: Register/login

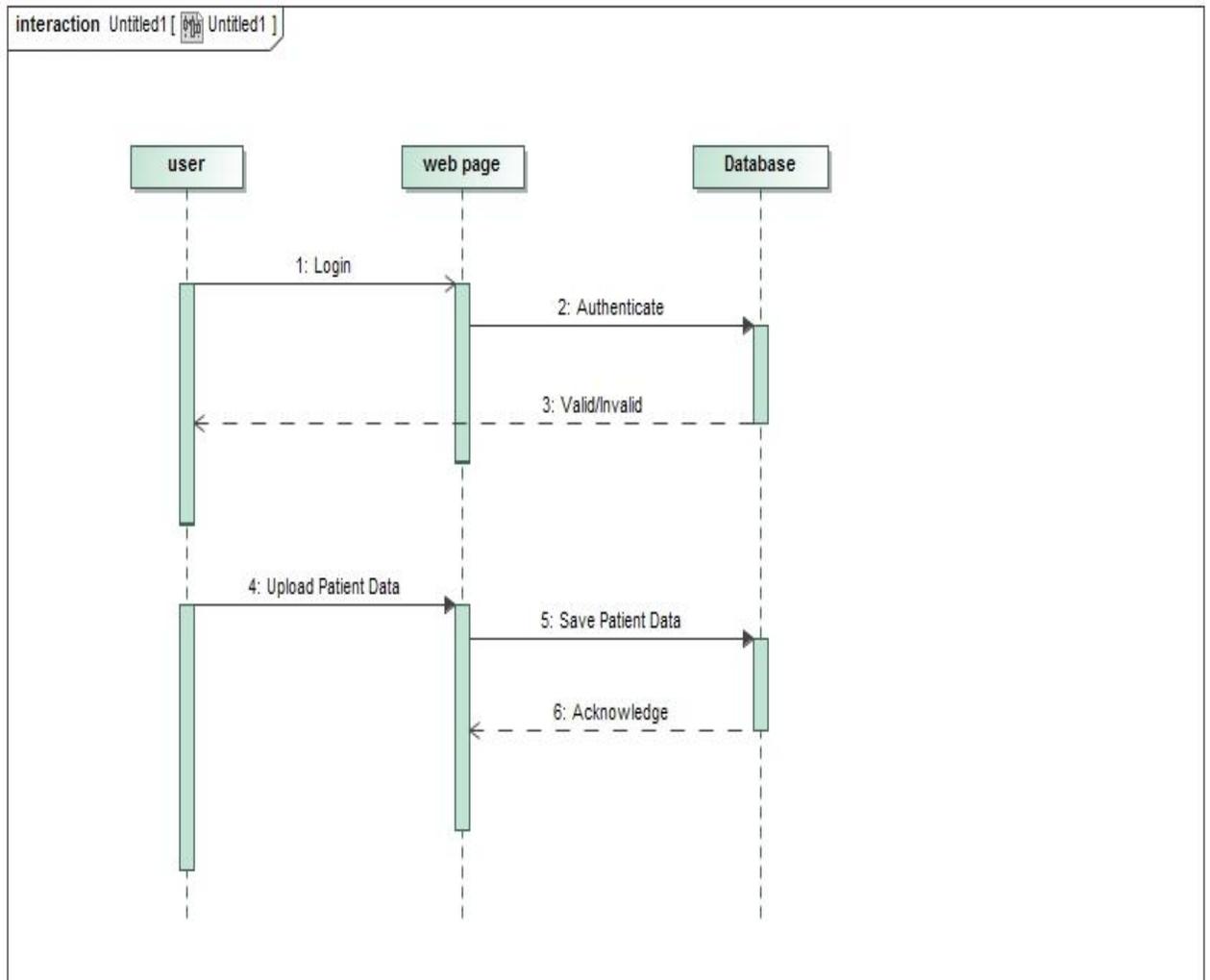
In this case User will try to login the application with his/unique username and password if he is already registered with the system. The application/browser will communicate with the database to authenticate the user. If user is authentic he/she will be allowed to continue his/her session. If the credentials are mismatching, the user will be returned with an error saying the the username and password do not match.

The new users will have to register with the application and information will be stored in the database.



## Use case 2: Upload Data

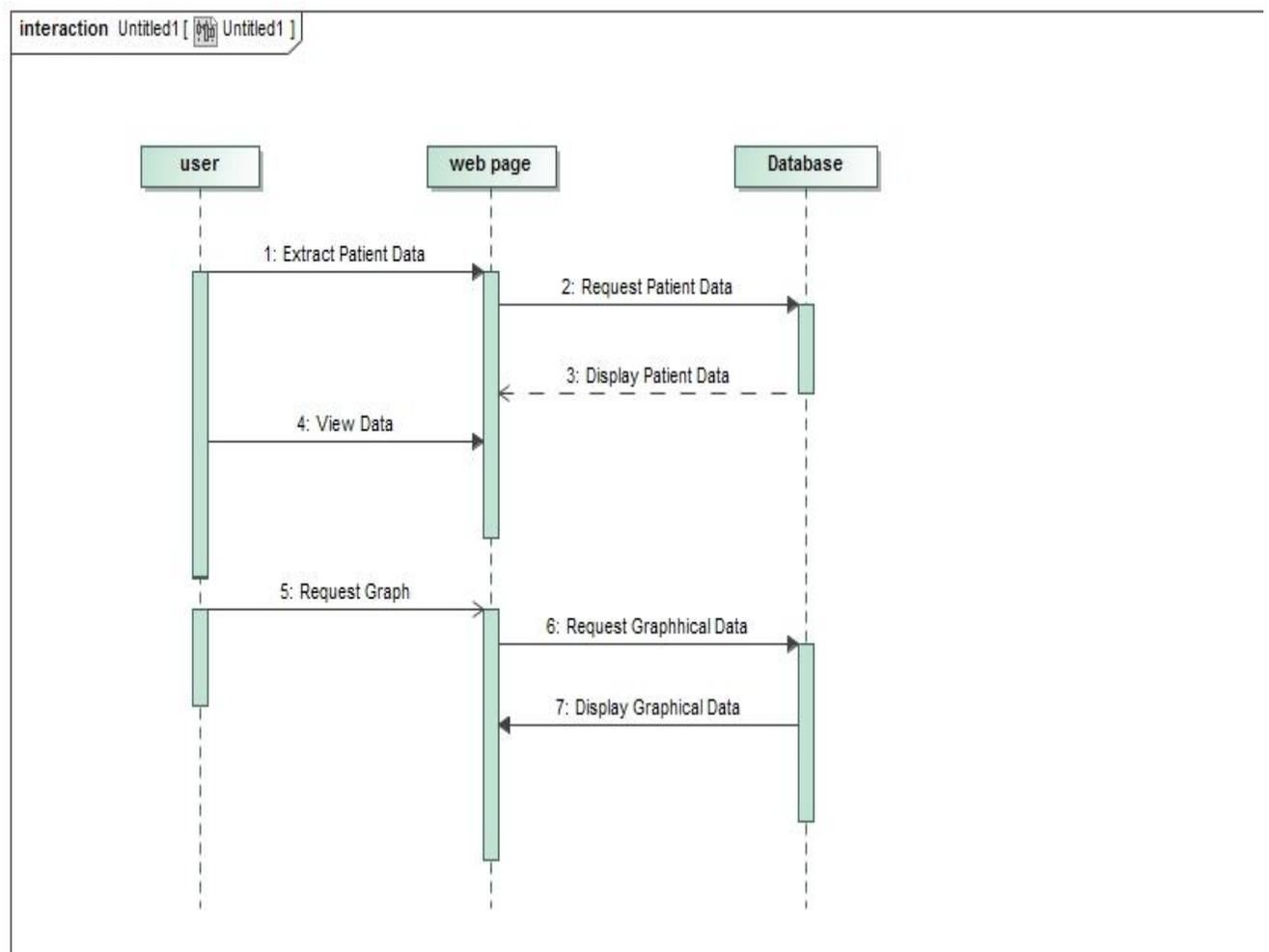
In this case, after correctly logging in with username and password, the user will be able to upload the data in excel sheet. This data uploaded will be saved in the system and the acknowledgment would be sent to the browser that the data is correctly uploaded and saved in the system. In case of failure, proper error would be reported.



## Use Case 3: Display Data

After uploading the data if the user wish to see the parameters defined customer requirements in graphical form, user will be able to chooses that and the parameters would

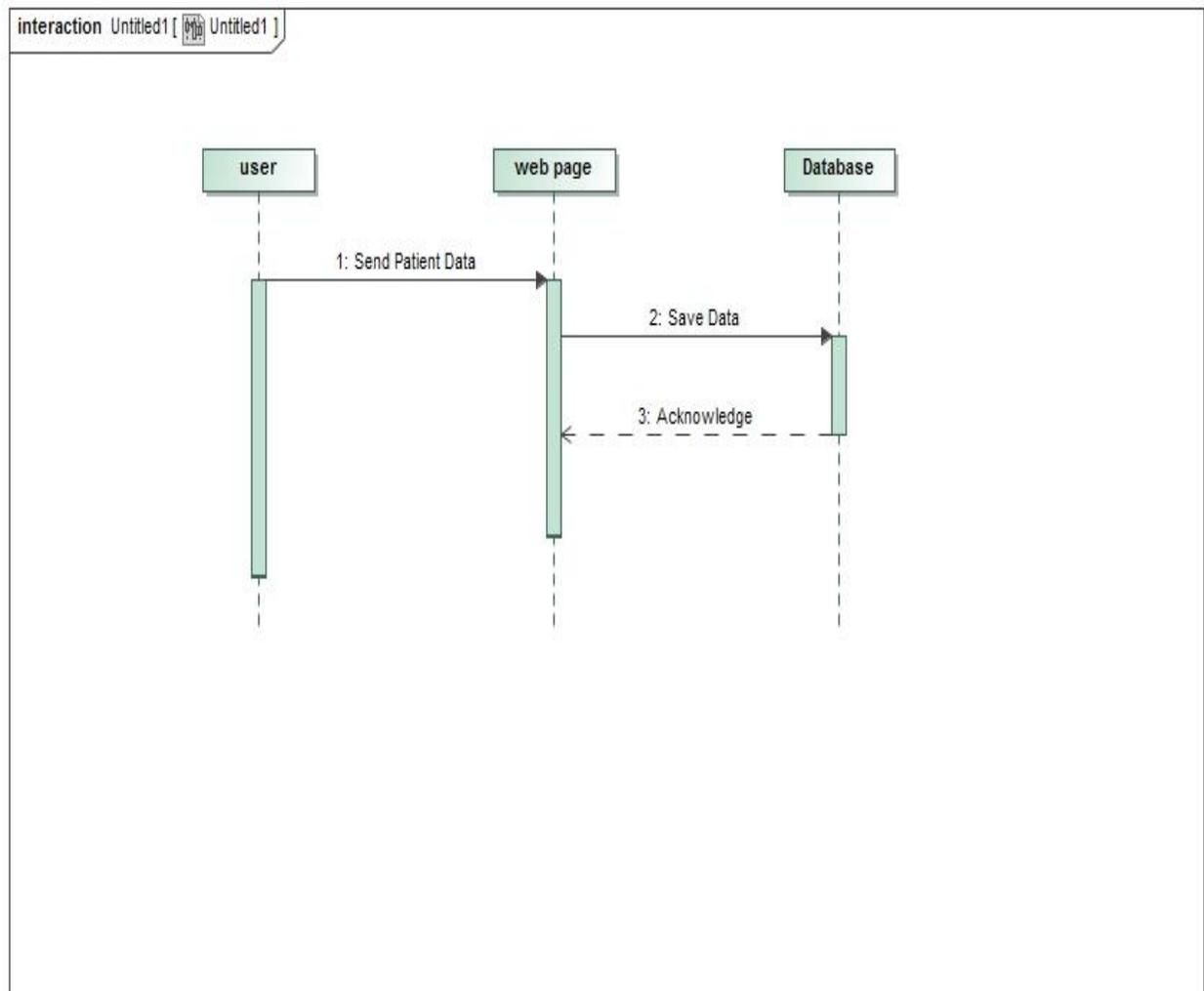
be displayed in graphical form, displaying how the user activity has varied over a period of time.



#### Use Case 4: Send Data

The user after viewing his health parameters would be able to send his report to the doctor for getting feedback on his progress. User will choose the button and send report and all his details would be sent to the doctor. A message acknowledging the data sending would be

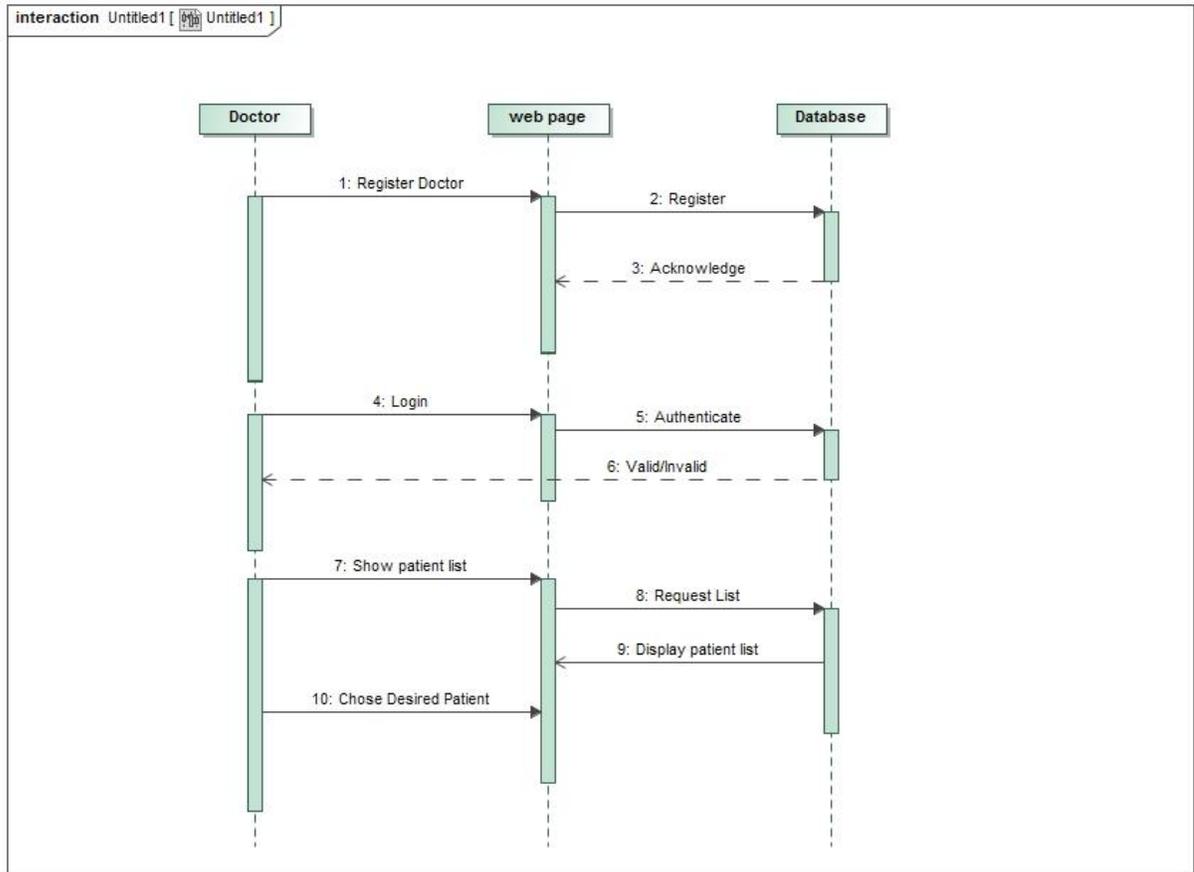
sent to the patient screen. In case of failure the reported error with failure message would be displayed.



### Use Case 5: View Data

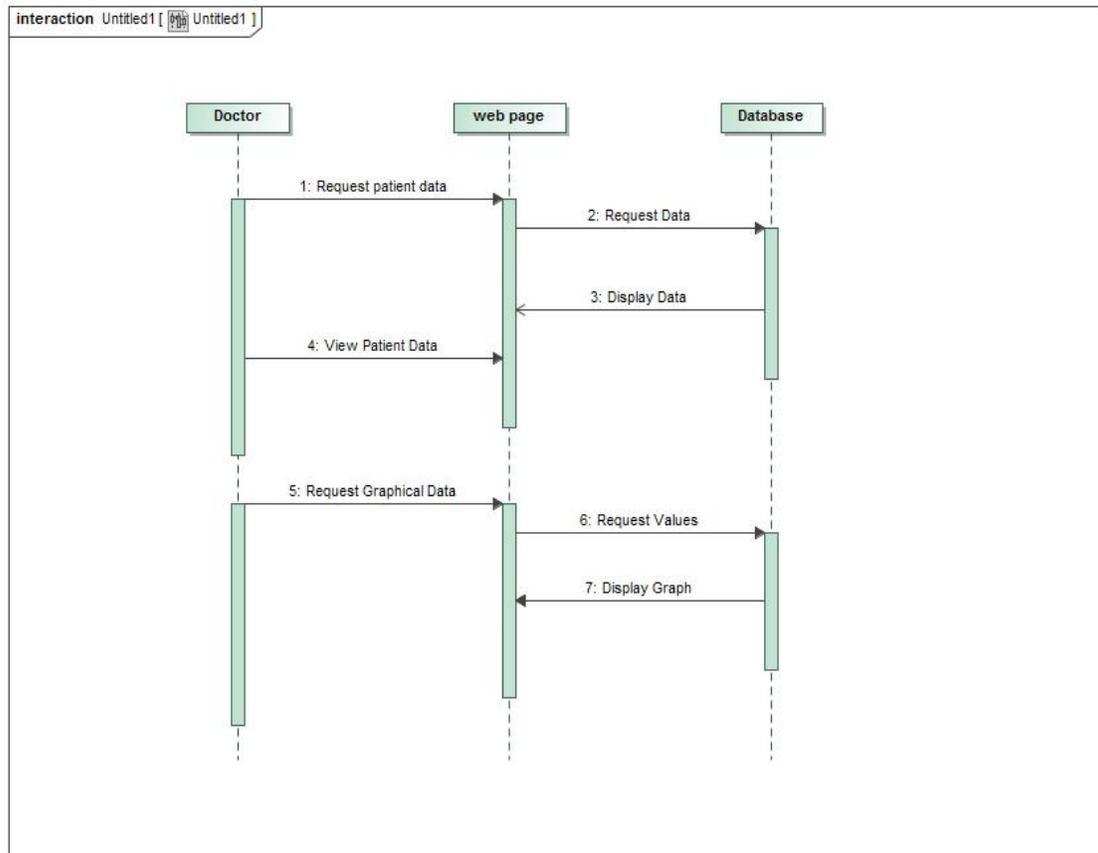
The Doctor would be able to view the data pertaining to a specific patient after logging in with his/her details. Doctor would get choose from within the a list of patients that would be displayed at his interface and then choose the patient. The data would be displayed as

soon as the patient is selected. A message saying “No data present ” would be displayed in case no data is present for the chosen patient.



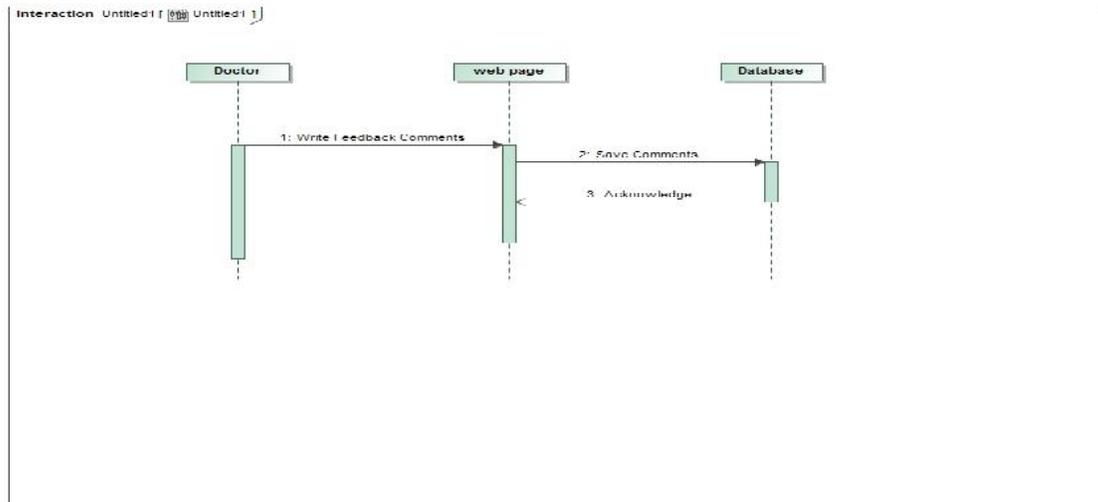
### Use Case 6: View Graph

When the doctor is not satisfied his data analyses using the tabular data, Doctor would be able to same data in the graphical forms. Doctor would request the data from browser and required graphs would be displayed.



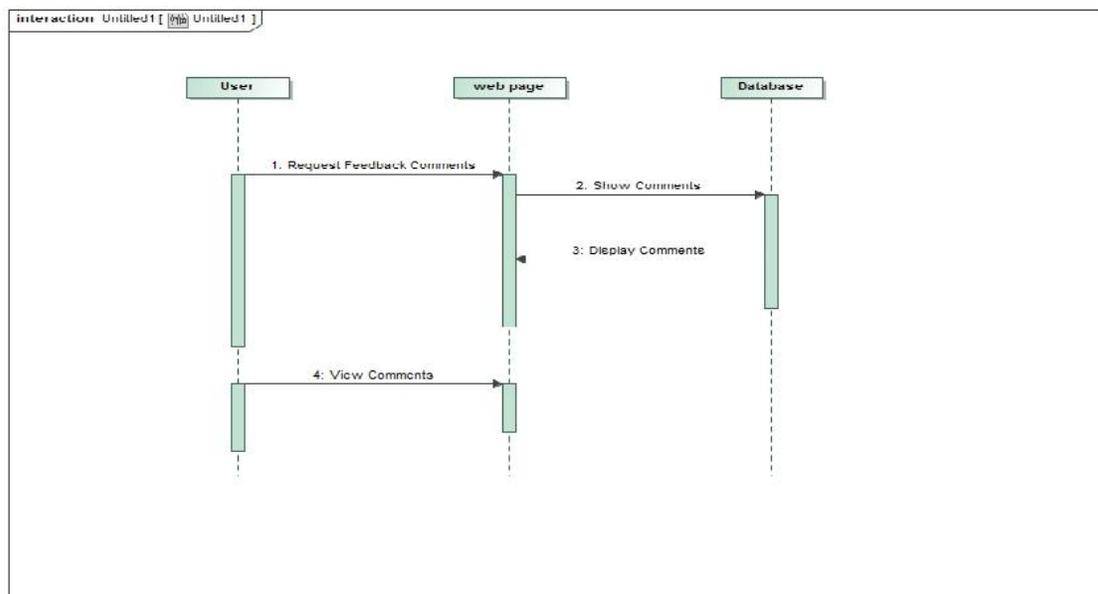
### Use Case 7: Provide Feedback

Once after the data is analyzed by the doctor, the doctor will write his feedback comments in the field provided for feedbacks. After entering the comments, database would be updated with the doctors comments and are ready for user view.



### Use Case 8: Download Feedback

After logging in, the patient would be able to choose his report for which he wants to see his/her feedback. If the comments are available patient would be able to see the feedback comments. However there is no separate communication provided that whether the doctors' comments are available or not. If the comments field is empty it is understood that the doctors' comments are not available.



## 5. User Interface Specification

User interface used for human-machine interaction, and User interface design become more and more important on software development. In this project we need to design User Interface for a computer software and an android application to let our user can easy to use our application to monitor their health situation. Consider about the user who need to monitor health are old people, they normally cannot use such electrical device very well. So we need to make our interface as clearly as possible. We hope by a good user interface design, our user can feel comfortable and have a good user experience.

### a) Preliminary Design

#### i. Patient Use Case

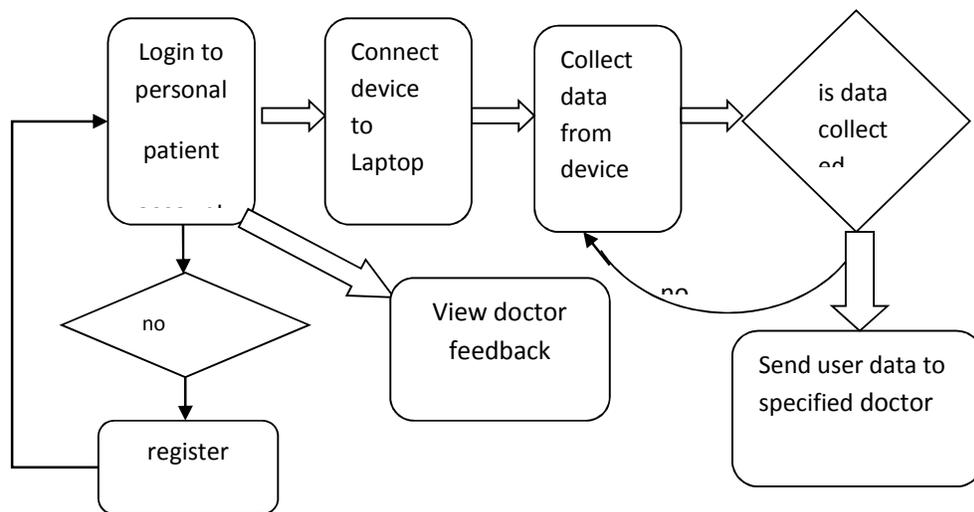


Figure a.1 This chart shows that the flow of patient send their data collected by device to doctor.

For patients, after they use device to monitor and collected their health parameter, they need to choose a doctor to send their data. In our system patients need to login their own account, if their do not have an account they should register at first. After that they can view doctor's feedback or connect their device to PC and use application to collect data. When system successfully get patient's data, patients could choose a doctor and send their data to the specified doctor.

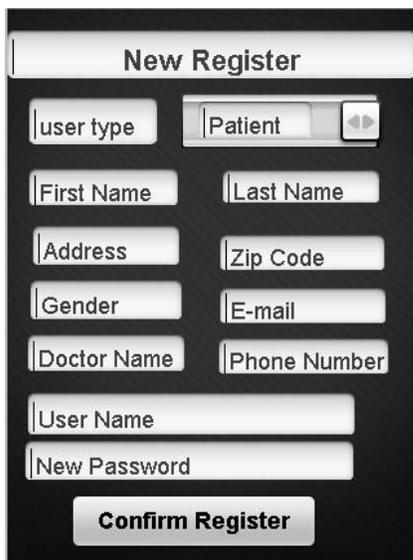
### User Interface of Patient



The screenshot shows a mobile application interface titled "Health Monitor". It features a dark background with white text and input fields. At the top, there is a header bar with the text "Health Monitor". Below the header, there are two input fields for "Account ID" and "Password". To the left of these fields are three radio buttons labeled "Patient", "Doctor", and "Administrator". Below the radio buttons are two buttons: "Login" and "register".

Figure a.2 Login Interface

This interface is for user to login system. For a patient, he or she can choose "Patient" selection and enter his or her ID, password, then hit "Login" button he or she can login to system. If a patient does not have an account ID, he can hit "register" button to create his own account, after that he can login to system.



The screenshot shows a mobile application interface titled "New Register". It features a dark background with white text and input fields. At the top, there is a header bar with the text "New Register". Below the header, there is a "user type" label and a dropdown menu showing "Patient". Below this are several input fields: "First Name", "Last Name", "Address", "Zip Code", "Gender", "E-mail", "Doctor Name", and "Phone Number". At the bottom, there are two more input fields: "User Name" and "New Password". Below these fields is a button labeled "Confirm Register".

Figure a.3 New Register Interface

As the Figure2.3 shows, user could do register on New Register Interface. Patient should use spinner to select a user type. After that enter all of text fields, user need to click "Confirm Register" button to finish register and will back to login interface.



Figure a.4 Patient Interface

When patient login to system, they will see a interface like figure2.3. They can see their name, gender, age and photo. If their personal information is correct, they can hit "Collect Data" button to collect data from their device and choose a doctor from spinner then hit "Send Data" button to send their data to a specified doctor. For the "View Feedback" button, patient can see their doctor's feedback.

**ii. Doctor Use Case**

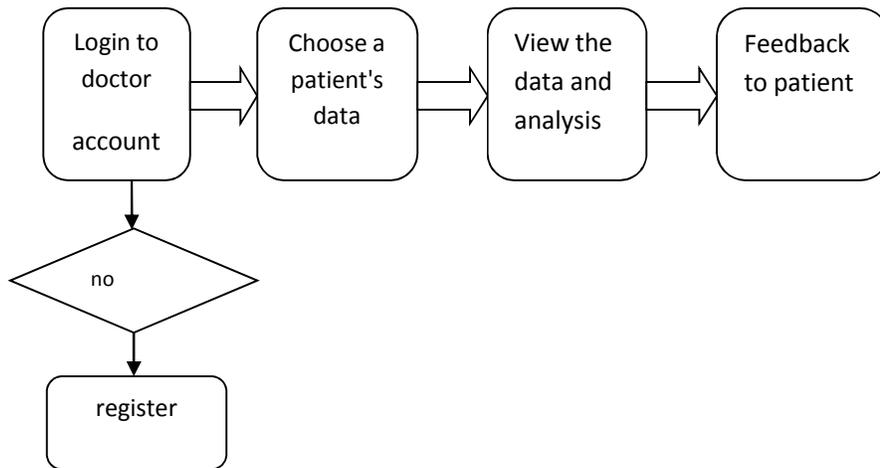


Figure a.5 This chart shows that the flow of doctor to view patient's data and feedback to patient.

For doctor, they also need to login their own account or register at first. Then they can choose a patient and view patient's data. After that doctor could select a patient to view his or her data, and do analysis. Then doctor can give feedback to their patient.

**User Interface of Doctor**

The login interface is same as figure2.2, for doctor, they need to select "doctor" selection when they login.



Figure a.6 Doctor interface

In doctor interface show by 2.6, doctor could choose a patient by name, gender and age. When doctor hit "View Data" button they could see patient health parameter.



Figure a.7 Patient data interface

In patient data interface, doctor can see the patient's data and use those data to analyze patient health situation. And "Feedback" button is for doctor to write feedback for their patient.

**iii. Administrator Use Case**

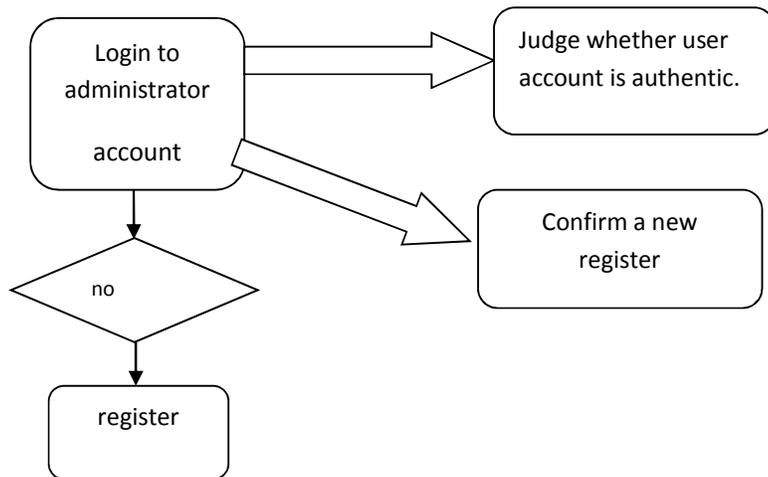


Figure a.7 This chart show flow of Administrator to judge account whether authentic and confirm new register

For administrator, they need login system by their administration account. After that administrator could judge whether an account is authentic or to confirm a new register.

### User Interface of Administrator

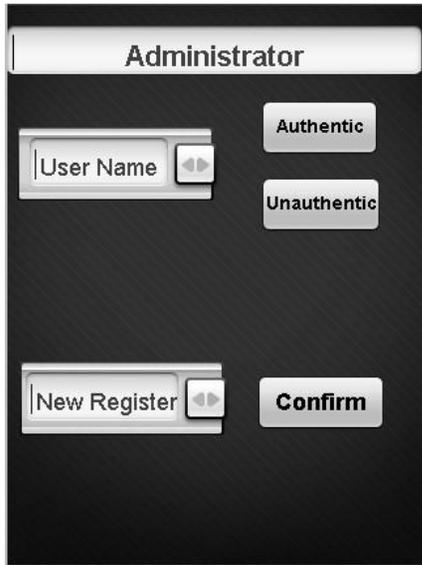


Figure a.8 Administrator interface

In Administrator interface show on figure, administrator can use spinner choose user name in order to judge whether an account is authentic or not. And hit "Authentic" and "Unauthentic" to make judgment. And use "New Register" spinner to choose a new register then hit "Confirm" to pass a register.

#### **b) User Effort Estimation**

In this section, we will estimate three part of user effort, include: patient user effort, doctor user effort, administrator user effort.

##### **i. Patient User Effort estimation**

Typical usage scenario 1: A patient already has an account. He wants to send his data to a doctor. And his account ID is "123456", password is "000000".

(1) NAVIGATION: total 5 mouse clicks, as follow.

a. Click "Patient" radio button.

--- after completing ID and password entry as shown below ---

b. Click "Login" button.

c. Click "Collect Data".

- d. Click "Doctor Name" spinner.
- e. Click "Send Data".

(2) LOGIN: total 1 mouse click and 13 keystrokes, as follow.

- a. Click cursor to "Account ID" text field.
- b. Press the keys "1" "2" "3" "4" "5" "6".
- c. Press the key "Tab" to next text field.
- d. Press the keys "0" for six times.

Typical usage scenario 2: A patient already have an account. He want to check his feedback from doctor. And his account ID is "123456", password is "000000".

(1) NAVIGATION: total 3 mouse clicks, as follow.

- a. Click "Patient" radio button.

*--- after completing ID and password entry as shown below ---*

- b. Click "Login" button.
- c. Click "View Feedback" button.

(2) LOGIN: total 1 mouse click and 13 keystrokes, as follow.

- a. Click cursor to "Account ID" text field.
- b. Press the keys "1" "2" "3" "4" "5" "6".
- c. Press the key "Tab" to next text field.
- d. Press the keys "0" for six times.

**Worst-case scenarios:** A patient does not have account ID, he need to do register first then he can login and send his data to a doctor. Assume he want to use "123456" as his account ID, and "000000" as his password, his first name is "Tom" , last name is "Jack" address is "NJ", zip code is "00000", gender is "Male", email is "Tom@gmail.com", doctor name is "Peter" and phone number is "7777777" .

(1) NAVIGATION: total 6 mouse clicks, as follow.

- a. Click "Register" button.

--- after completing register and back to login interface ---

b. Click "Patient" radio button.

--- after completing ID and password entry as shown below ---

c. Click "Login" button.

d. Click "Collect Data".

e. Click "Doctor Name" spinner.

f. Click "Send Data".

(2) REGISTER: total 2 mouse click and 56 keystrokes, as follow.

a. Click spinner to choose patient selection.

b. Click cursor to "First Name" text field.

c. Press the keys "T" "o" "m".

d. Press the key "Tab" to next text field.

e. Press the keys "J" "a" "c" "k".

f. Press the key "Tab" to next text field.

g. Press the keys "N""J".

h. Press the keys "Tab" to next text field.

i. Press the keys "0" for 5 times.

j. Press the key "Tab" to next text field.

k. Press the keys "M" "a" "l" "e".

l. Press the key "Tab" to next text field.

m. Press the keys "T" "o" "m" "@" "g" "m" "a" "i" "l" "." "c" "o" "m".

n. Press the key "Tab" to next text field.

o. Press the keys "P" "e" "t" "e" "r".

p. Press the key "Tab" to next text field.

q. Press the keys "7" for seven times

r. Press the key "Tab" to next text field.

- s. Press the keys "1" "2" "3" "4" "5" "6".
- t. Press the key "Tab" to next text field.
- u. Press the keys "0" for six times.
- v. Click the "Confirm Register" button.

(2) LOGIN: total 1 mouse click and 13 keystrokes, as follow.

- a. Click cursor to "Account ID" text field.
- b. Press the keys "1" "2" "3" "4" "5" "6".
- c. Press the key "Tab" to next text field.
- d. Press the keys "0" for six times.

## ii. Doctor User Effort estimation

Typical usage scenario: A doctor already have an account. He want to check patient's data and make a feedback. And his account ID is "123456", password is "000000".

(1) NAVIGATION: total 7 mouse clicks, as follow.

- a. Click "doctor" radio button.

*--- after completing ID and password entry as shown below ---*

- b. Click "Login" button.
- c. Click "Patient Name" spinner to choose a patient.
- d. Click "Gender" spinner to choose patient's gender.
- e. Click "Age" spinner to choose patient's age.
- f. Click "View Data" button to see patient's data.
- g. Click "Feedback" button to write a feedback.

(2) LOGIN: total 1 mouse click and 13 keystrokes, as follow.

- a. Click cursor to "Account ID" text field.
- b. Press the keys "1" "2" "3" "4" "5" "6".
- c. Press the key "Tab" to next text field.
- d. Press the keys "0" for six times.

**iii. Administrator User Effort estimation**

Typical usage scenario 1: A administrator already have an account. He want to confirm a new register. And his account ID is "123456", password is "000000".

(1) NAVIGATION: total 4 mouse clicks, as follow.

a. Click " administrator " radio button.

*--- after completing ID and password entry as shown below ---*

b. Click "Login" button

c. Click "New Register" spinner to choose a new register.

d. Click "Confirm".

(2) LOGIN: total 1 mouse click and 13 keystrokes, as follow.

a. Click cursor to "Account ID" text field.

b. Press the keys "1" "2" "3" "4" "5" "6".

c. Press the key "Tab" to next text field.

d. Press the keys "0" for six times.

Typical usage scenario 2: A administrator already have an account. He want to judge an account whether authentic or not And his account ID is "123456", password is "000000".

(1) NAVIGATION: total 4 mouse clicks, as follow.

a. Click " administrator " radio button.

*--- after completing ID and password entry as shown below ---*

b. Click "Login" button

c. Click "User Name" spinner to choose a user.

d. Click " Authentic " or "Un authentic ".

(2) LOGIN: total 1 mouse click and 13 keystrokes, as follow.

a. Click cursor to "Account ID" text field.

b. Press the keys "1" "2" "3" "4" "5" "6".

c. Press the key "Tab" to next text field.

d. Press the keys "0" for six times.

## 6. Domain Analysis

### a) Domain Model:-

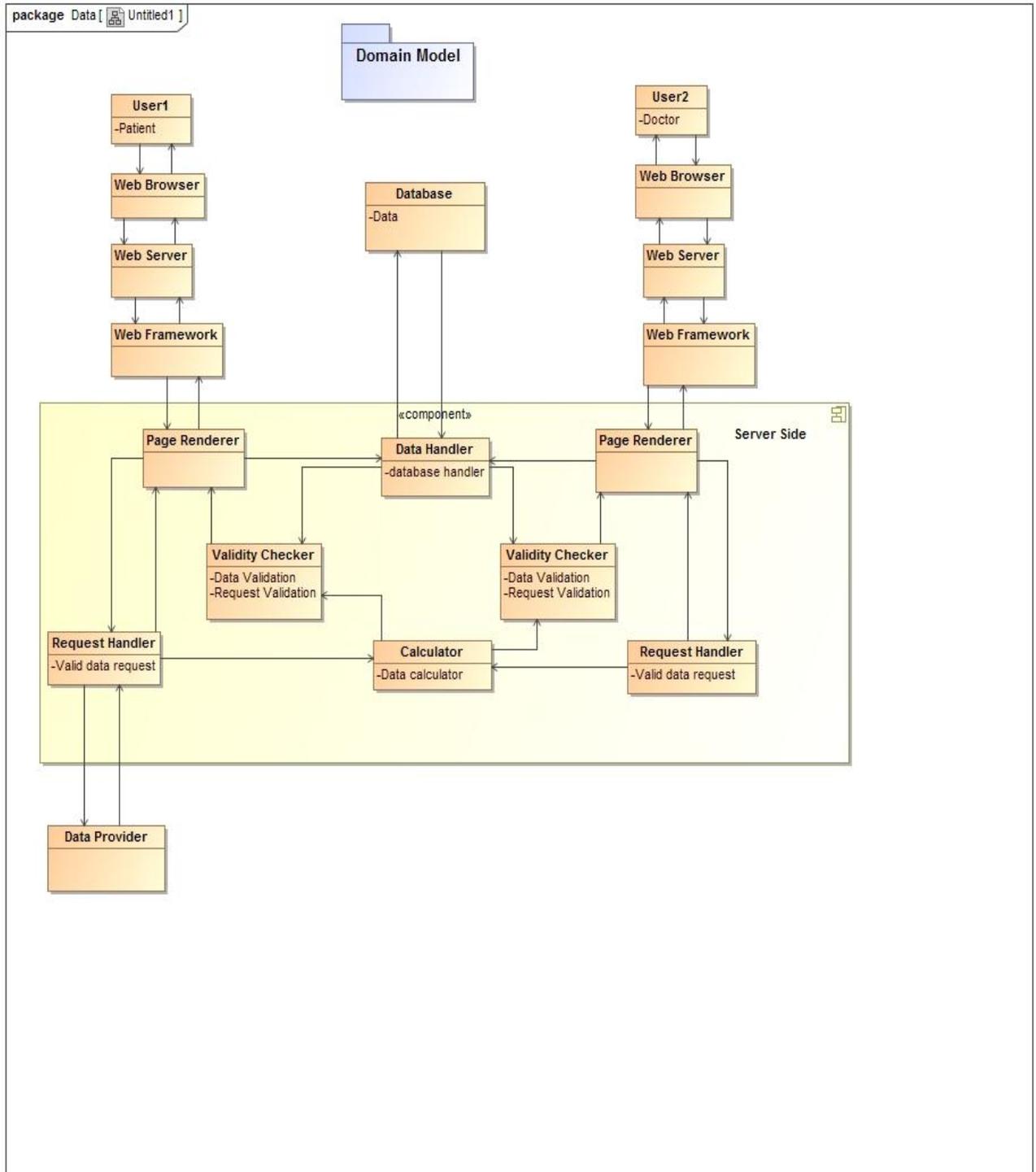




Figure will represent UC1, UC2, UC3. It represents login, register upload data information, calculate and display information. The user will request for data to be uploaded and the request goes to request handler from web Framework via page renderer. The request handler will communicate with the system from which the data will be fetched and then it is uploaded to the database after processing by the calculator. Also the user requests for data to be calculated and displayed. The request will be send from web framework to the page renderer and then to the request handler which checks which type of request to be process. For calculating result reqest then goes to data handler which fetches the stored data from the database and then this data goes to the calculater. The calculater process the data and determine various parameters like total energy expenditure, sleep time, steps etc and then validity checker checks if the results calculated are right.

. Then the calculated results are send to page rendered which generates a user friendly page to display data. The web framework then process the page and diplayed to the user.

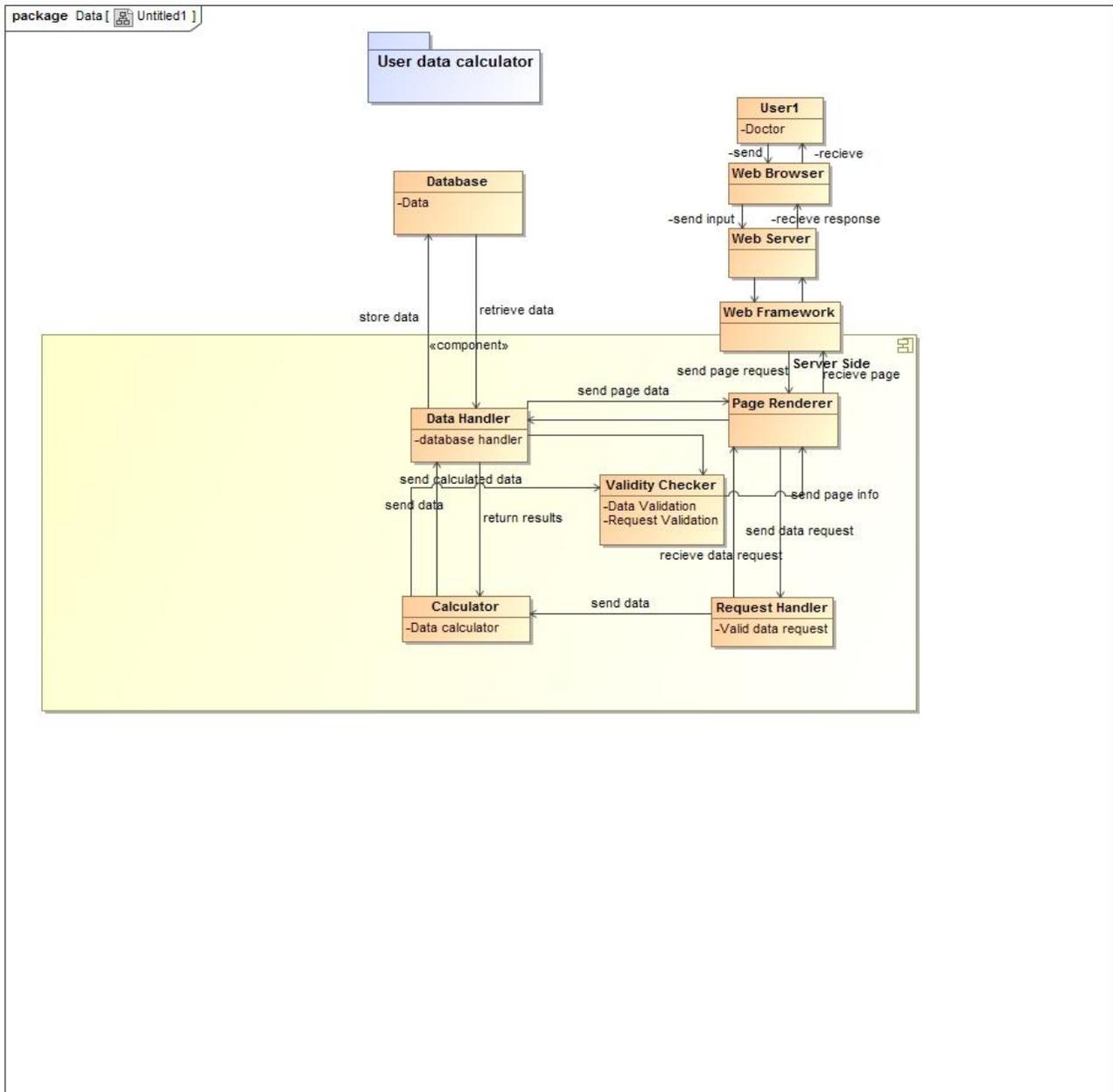


Figure will represents UC1, UC5, UC6, UC7. It represents login, register of doctor, view data by doctor and feedback. The user(doctor) will send the request and the request goes to request handler from web Framework via page renderer. The request handler checks which type of request to be process. For calculating result request then goes to data handler which fetches the stored data from the database and then this data goes to the calculator. The calculator process the data and determine various parameters like total energy expenditure, sleep time, steps etc and then validity checker checks if the results calculated are right. Then the calculated results are send to page rendered which generates a user friendly page to display data. The web framework then process the page and diplayed to the user.

To give feedback the user request for feedback window and then writes the feedback which is then goes to data handler. It then saves the feedback for that particular patient in the database which can be viewed by him.

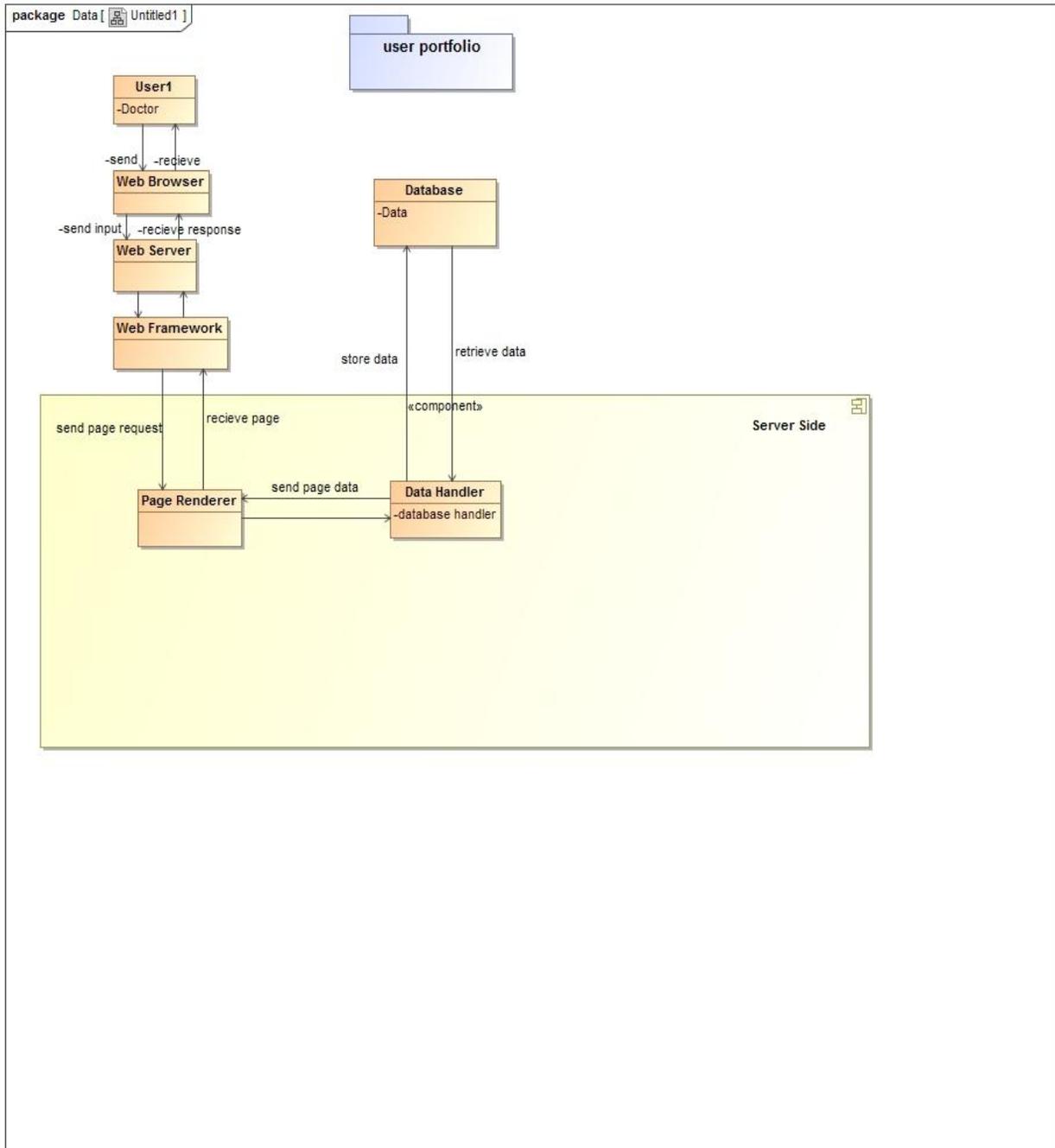


Figure represents UC1. The user queries about the profile view and then the request is sent from the web framework to the page renderer which then goes to data handler. The data handler fetches the reused data from the database and it is sent back to the page renderer. The page renderer generates the page for the web framework to send to the user for viewing.

### i. **Concept Definition**

The Definitions of our concepts are as follows:

#### **User**

**Definitions:** A player playing Bears & Bulls.

Responsibilities:

- \_ Manage portfolio
- \_ Make requests for feedback
- \_ send data
- \_ Navigate through website

#### **Web Browser**

**Definitions:** The user's browser which runs from the user's device.

Responsibilities:

- \_ Take requests from the user
- \_ Send requests to the Web Server
- \_ Get responses from the Web Server
- \_ Display the response from the Web Server

#### **Web Server**

**Definitions:** HTTP web server, running on a web host

Responsibilities:

- \_ Receive requests from Web Browser
- \_ Send requests to Web Framework
- \_ Get responses from Web Framework
- \_ Send responses to the Web Browser

#### **Web Framework**

**Definitions:** APIs to help display user-friendly output

Responsibilities:

- \_ Receive requests from Web Server
- \_ Sends request to appropriate handler: application or database
- \_ Receive rendered pages in the form of structured data
- \_ Send responses to the Web Server

#### **Page Renderer**

**Definitions:** Takes user requests and creates a page which is user-friendly

Responsibilities:

- \_ Determine the information required to be rendered and request it
- \_ Receive the required information

- \_ Convert the information into user-friendly format
- \_ Send rendered pages to the Web Framework

**Request Handler**

**Definitions:** Application conducting transactions of stocks

**Responsibilities:**

- \_ Receive requests from Web Framework
- \_ Determine what the request is and readies for manipulation
- \_ Request updated calculated data info
- \_ Transmit necessary information to other concepts

**Calculator**

**Definitions:** Calculate the total or average of the real time Data

**Responsibilities:**

- \_ Receive requests for calculation
- \_ Request information from request handler
- \_ Retrieve information from data handler
- \_ Send real-time calculated data to be stored for application's use

**Validity Checker**

**Definitions:** Checks if a trade is valid

**Responsibilities:**

- \_ Receive updated order information
- \_ Request and receive portfolio data
- \_ Determine if correct data is provided by the user
- \_ Determine if calculated data is correct for the desired user
- \_ Send updated portfolio information if necessary

**Data Handler**

**Definitions:** Communicates with Database to service data requests

**Responsibilities:**

- \_ Receive and send every kind of data used in system
- \_ Request data from Database
- \_ Send data to be stored in Database

## ii. Association Definitions

The following association definitions are provided for the domain models that model not only for the important use cases, but also any alternative models for said use cases

Concept Pair	Association Description	Association Name
Web Browser Web Server	User interacts with browser	send input, send response
Web Framework \$ Request Handler	Passes data, User ID and league ID	send data request
Web Framework \$ Page Renderer	Request to visit page, sends rendered page in form of data	send page request, send page
Page Renderer \$ Data Handler	Requests data to correctly render page, passes necessary data	send data request, send page data
Page Renderer \$ calculator	Asks for calculated data, send calculated data	request data data, send data
Page Renderer \$ Validity Checker	Passes necessary data for the page to be rendered	send page info
request Handler \$ Stock Query	Passes necessary data	send data
Validity Checker \$ Data Handler	Asks for user's portfolio information for validity purposes, passes user's portfolio information, passes updated portfolio information following	request portfolio data, return portfolio data, send new portfolio data
Validity Checker \$ calculator	Sends updated data to be checked	send updated data
calculator \$ Data Handler	Sends calculated information, returns new value	send info, return request
Data Handler \$ Database	Stores incoming data, request certain data, retrieve needed data	store data, request data, retrieve data

### iii. Attribute Definitions

Most of our concepts do not need to hold their own data, as our website is dynamic and web-based. We also have not yet made the decision to cache data.

Thus, nearly all data is stored in a single database. The sparse attributes that must be accounted for are as follows:

Concept

Concept	Attribute	Meaning
Data Handler	databaseHandle	Interacts with the database.
Database	data	Stores data for future use. Includes all data used in the system, including League ID, User ID, stock volume and price data, league settings, fund settings, and portfolio data such as transaction history.
calculator	Calculate data	Calculates the average value or the total value
Page Renderer	su_cientRenderData	Determines if the required data to render the page is there.
Request Hander	validOrderRequest	Checks to see if there is all the required data.
Validity Checker	Data valid,user valid	Compares users with database checks settings to make sure a user and data are valid. Determines if identity is a success.

**iv. Traceability Matrix**

UC	PW	User Patient	User Doctor	Web Browser	Web Server	Web Framework	Page Renderer	Request Handler	Calculator	Data Handler
UC01-03		X		X	X	X	X	X	X	X
UC05-07			X	X		X	X	X	X	X
UC1		X		X	X	X	X			X

**b) System Operation Contracts****i. Register User:****Preconditions: None****Post conditions:**

- User has a profile linked to database and every user has a doctor who will review his profile.
- Database keeps track of user's information.

**ii. Upload Data:****Preconditions:**

- Make sure enough space is available on in database.
- User is logged in.

**Post conditions:**

- Data is updated to database.

**iii. Display data:****Preconditions:**

- Database has valid information.
- Parameters have been calculated.

**Post conditions:**

- Data is displayed in user friendly version.
- Graphical representation is used for easy understanding.

#### **iv. Send to Doctor:**

##### **Preconditions:**

- Doctor is logged in.
- Parameters have been calculated.
- User has requested a review from his Doctor.

##### **Post conditions:**

- Data is sent to the doctor specified by the user during registration.

#### **v. Feedback:**

##### **Preconditions:**

- Doctor has received data pertaining to user who has requested a review.
- Doctor has viewed all graphical charts of user.

##### **Post conditions:**

- Doctor writes comments for user based on his report.
- Feedback is uploaded to the database.

#### **vi. Graphical Representation:**

##### **Precondition:**

- User/Doctor is logged in.
- Valid information is present in the database.

##### **Post condition:**

- Graphical version of the user information is generated.

**vii. Feedback to User:****Precondition:**

- User is logged in.
- Doctor has updated feedback to the database for user.

**Post condition:**

- User gets the feedback meant for him/her from the database.

**viii. Downloadable Feedback:****Precondition:**

- Database contains the information requested by the user.

**Post condition:**

- Information is downloaded and saved on user's system.

ix.

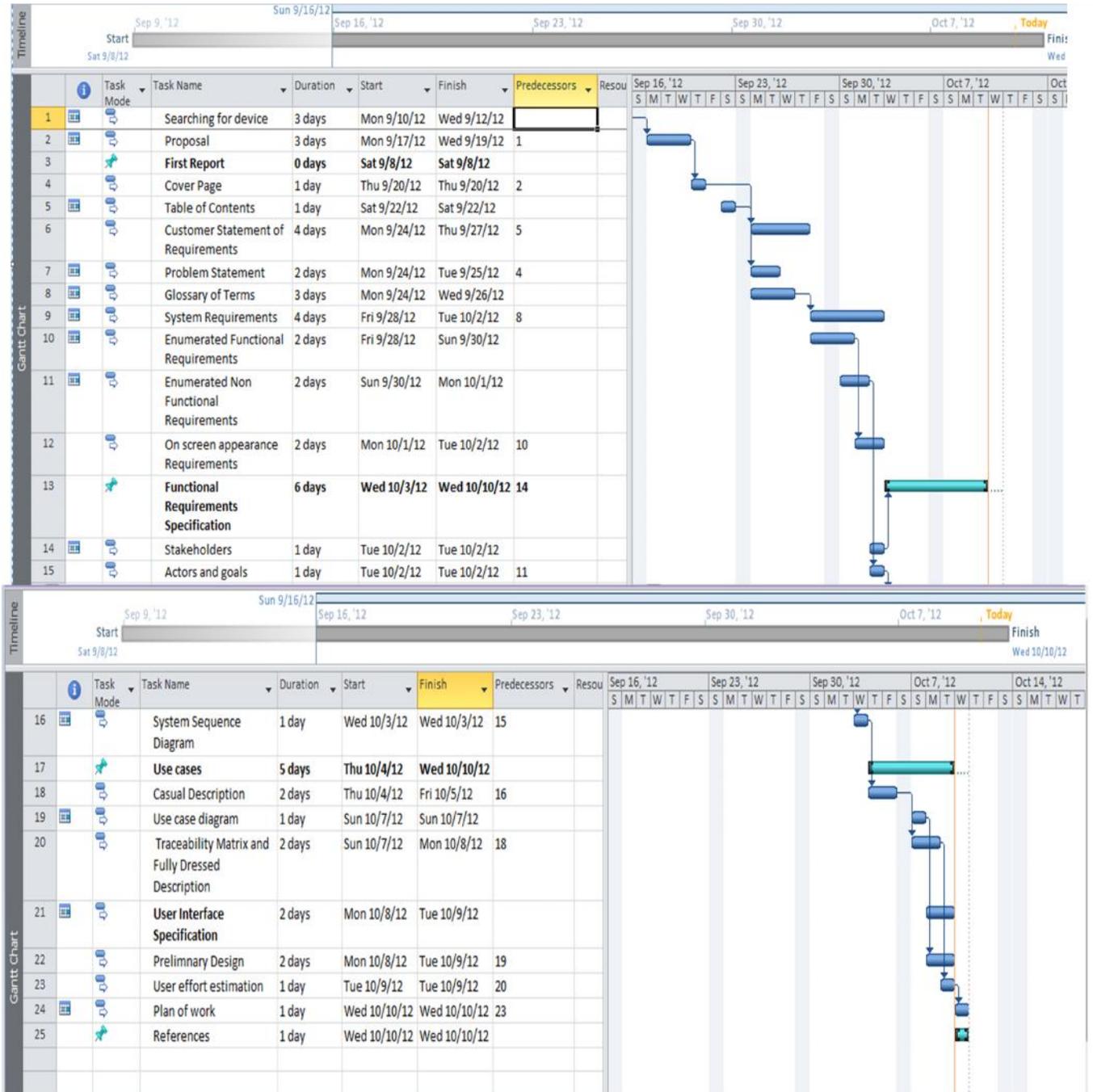
**View and Edit Information:****Precondition:**

- User/Doctor is logged in.

**Postcondition:**

- Profile Information is modified and updated in database.

## 7) Plan of Work



## 8) References

Marsic, Ivan. *Software Engineering*. 2012.

Bruegge, Bernd, and Allen H. Dutoit. *Object-oriented Software Engineering: Using UML, Patterns, and Java*. Boston: Prentice Hall, 2010

Bardi, James A. *Hotel Front Office Management*. Hoboken, NJ: John Wiley & Sons Inc., 2007.

<http://sensewear.bodymedia.com/>

<http://medicaldevice-network.com/>

<http://google.com>,<http://Wikipedia.org>

Ramakrishnan, Raghu, and Johannes Gehrke. *Database Management Systems*. Boston: McGraw-Hill, 2003.

Website Template – [sites.google.com](http://sites.google.com).