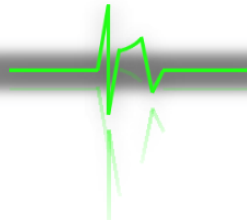


**Report No. 2**  
**Interaction Diagrams, Class Diagrams and System Architecture**



# **The Talking Heart**

*What's your heart saying?*

**A Personal Health Monitoring & Diagnosis System**

**Software Engineering I**

**11/19/2012**

**Development Team #2**

Prasoon Mishra

Vinayak Pothineni

Bhumika Singh

Jay Takle

Anu Liz Tom

Anusha Vutukuri

## Table of Contents

1. Individual Contributions	3
2. Interaction Diagrams	4
3. Class Diagram and Interface Specification	20
a. Class Diagram	20
b. Class Diagram Attributes and their Description	21
c. Traceability Matrix	24
4. System Architecture and System Design	25
a. Architectural Styles	25
b. Identifying Subsystems	26
c. Mapping Subsystems to Hardware	27
d. Persistent Data Storage	28
e. Network Protocol	29
f. Global Control Flow	29
g. Hardware Requirements	29
5. Algorithm	30
6. User Interface Design and Implementation	32
7. Design of Tests	35
8. Project management and Plan of Work	39
9. References	41

## 1. Individual Contributions

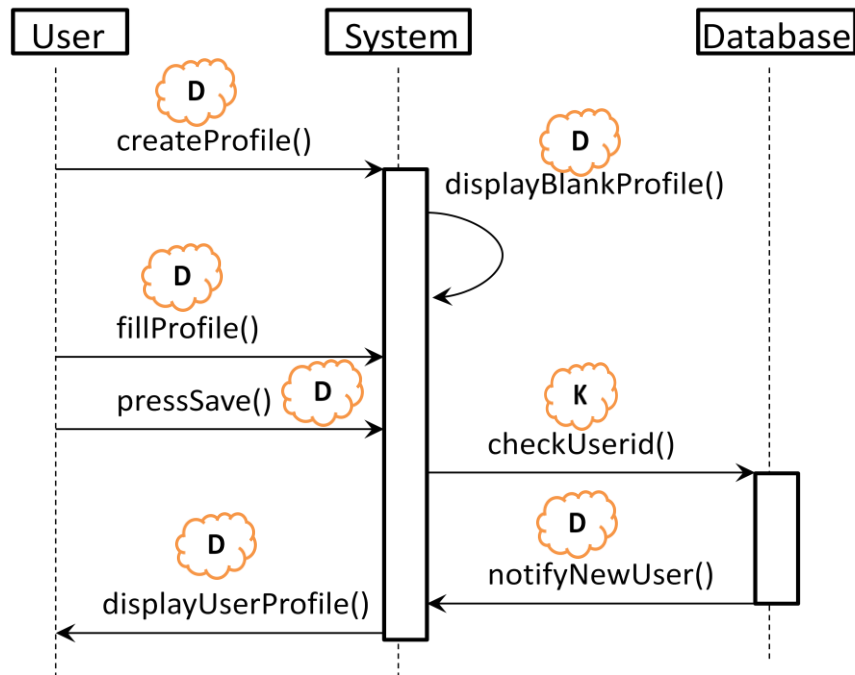
Responsibilities	Prasoon Mishra	Vinayak Pothineni	Bhumika Singh	Jay Takle	Anu Liz Tom	Anusha Vutukuri	Total
Project management			50	50			100
Interaction Diagrams	20	20	10	10	20	20	100
Class Diagram and Interface Specification	30		30	20	20		100
System Architecture and System Design		30	20		20	30	100
Algorithms and Data Structures	10	30			30	30	100
User Interface Design and Implementation	30	10	30	20		10	100
Design of Tests	10	10	10	50	10	10	100

## 2. Interaction Diagrams:

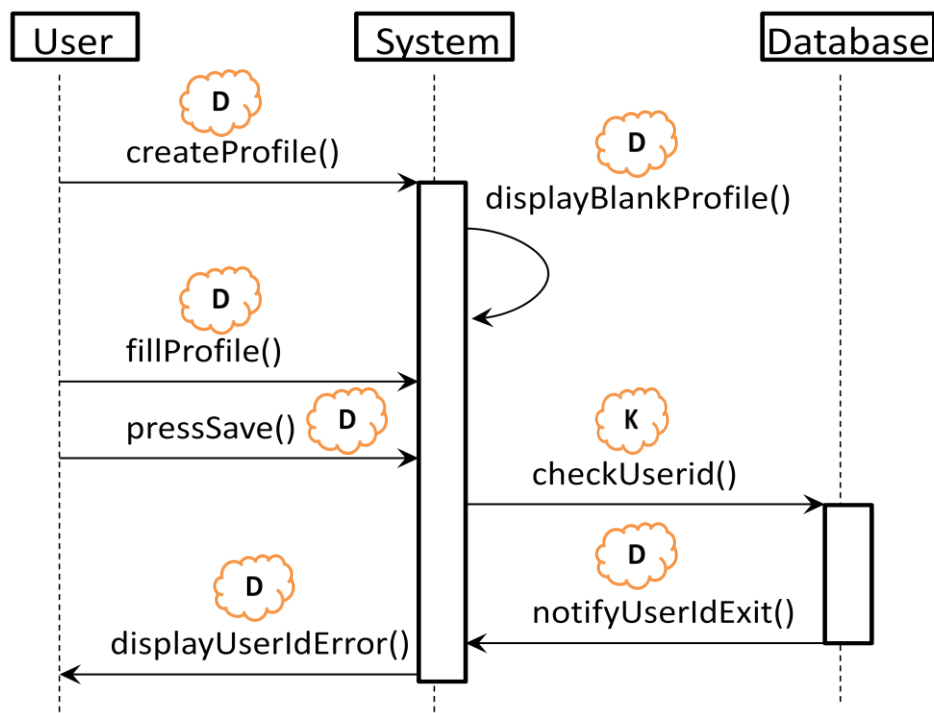
While assigning responsibilities we have differentiated concepts on the basis of doing responsibilities, and knowing responsibilities. “Knowing” concepts are denoted by a “K” bubble while “doing” concepts are denoted by the “D” bubble.

### UC1 CreateProfile

Success Case:

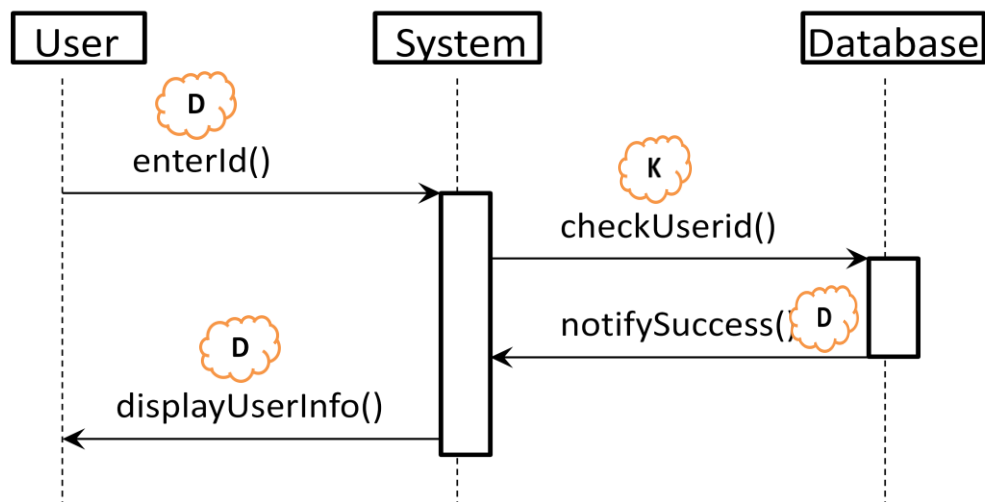


Alternate Case:

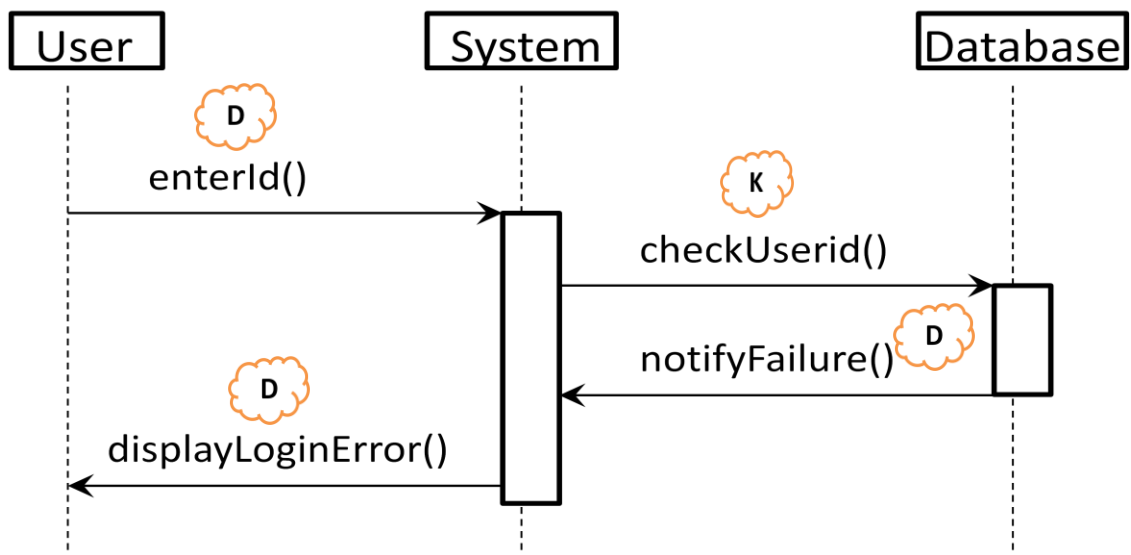


**UC2 Login**

Success Case:

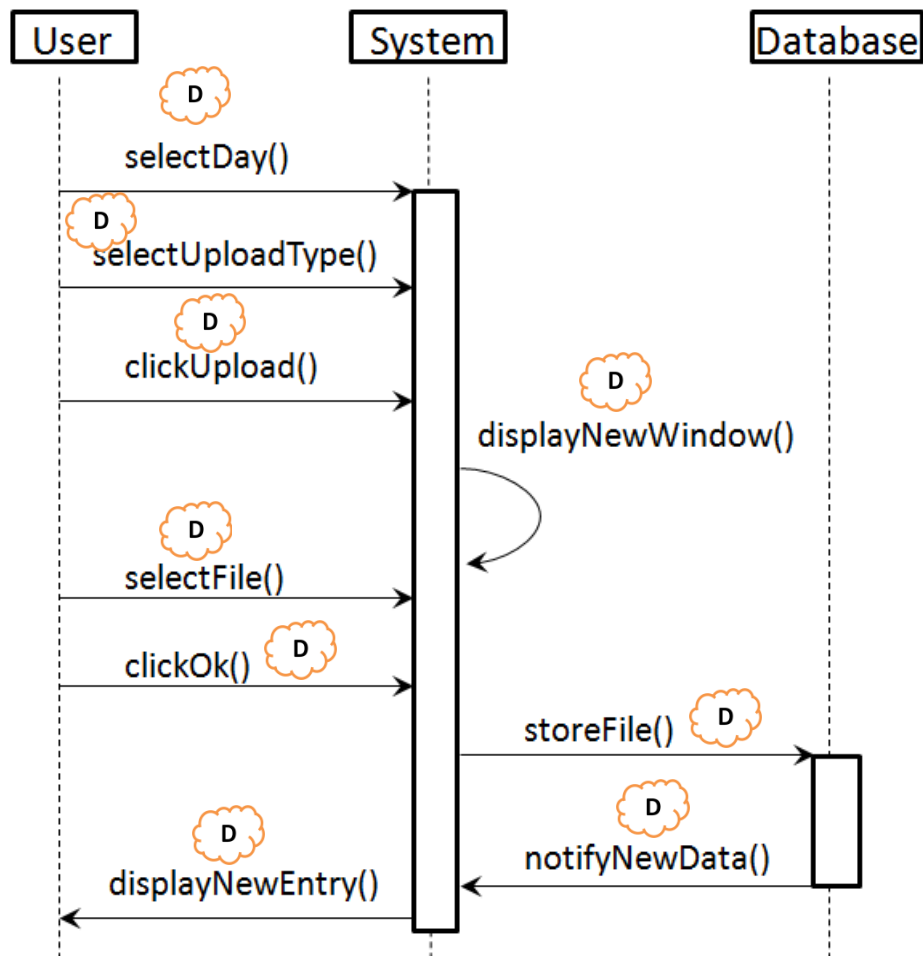


Alternate Case:

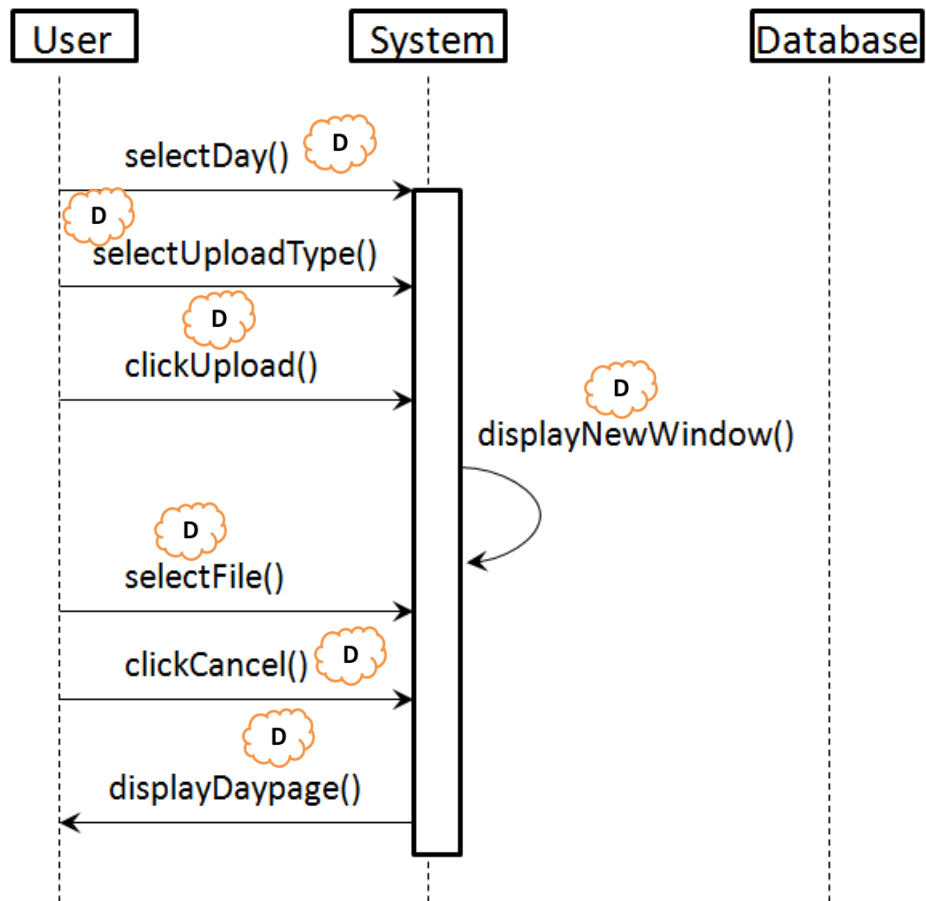


**UC3 UploadFile**

Success Case:

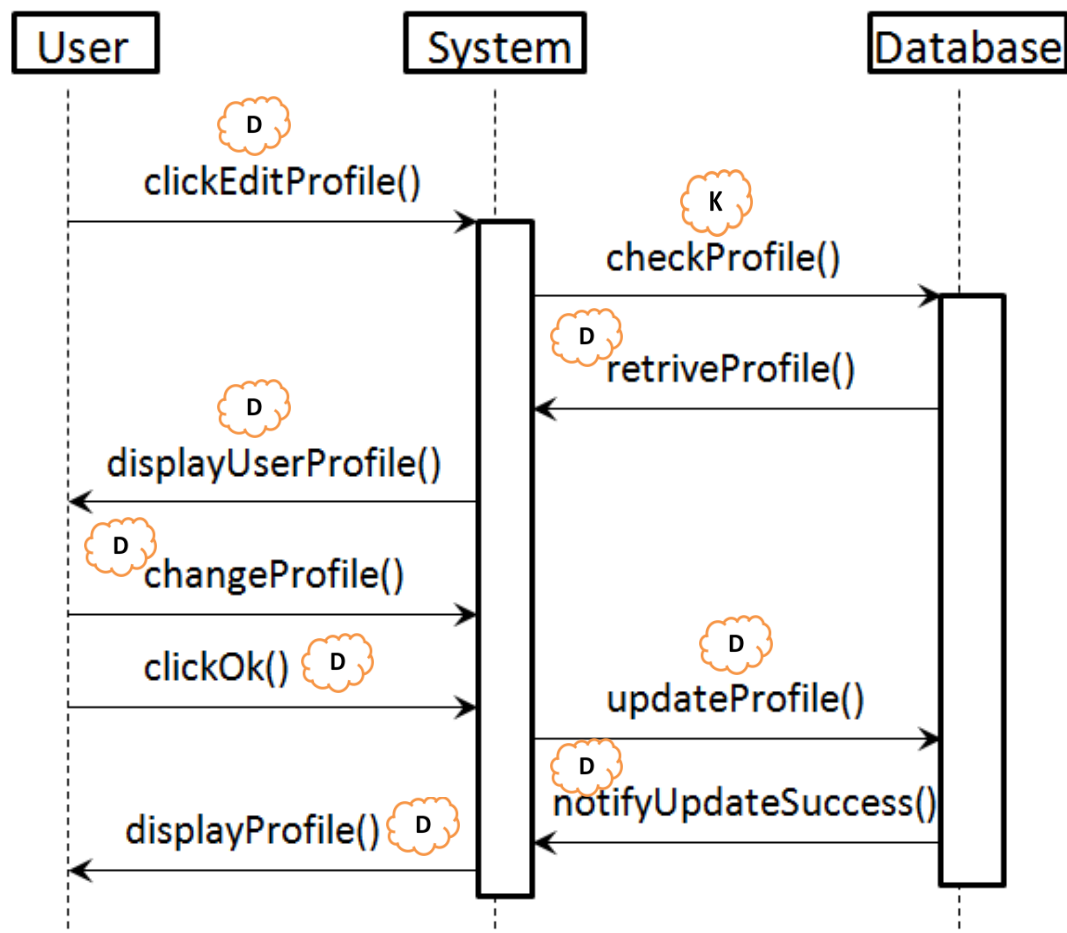


## Alternate Case



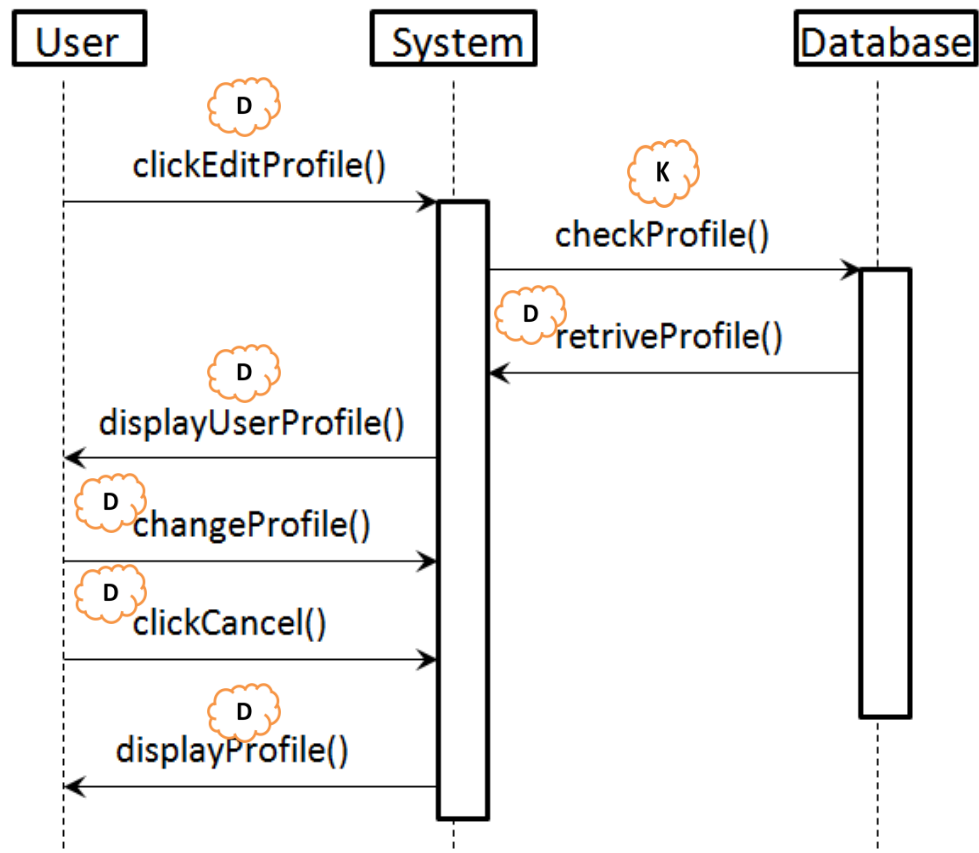
**UC4 ChangeProfile**

Success Case:



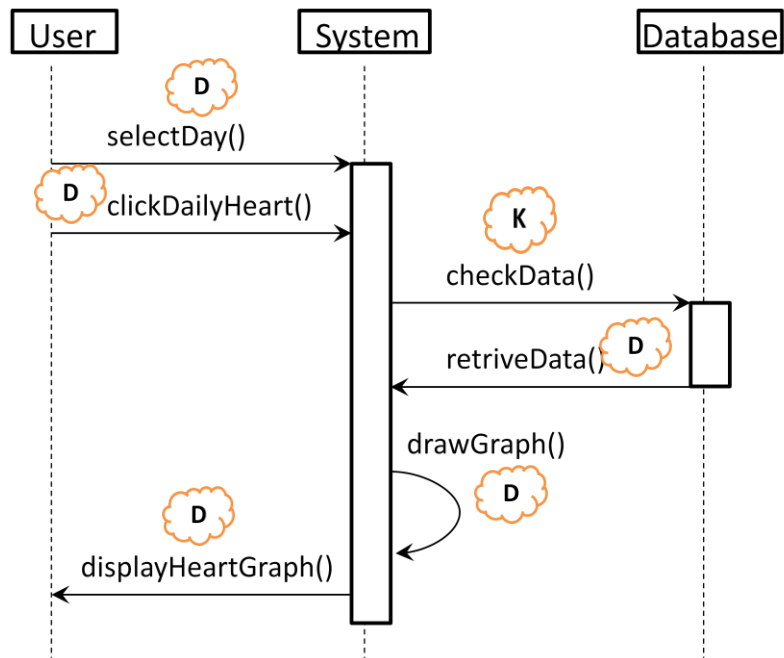


Alternate Case:

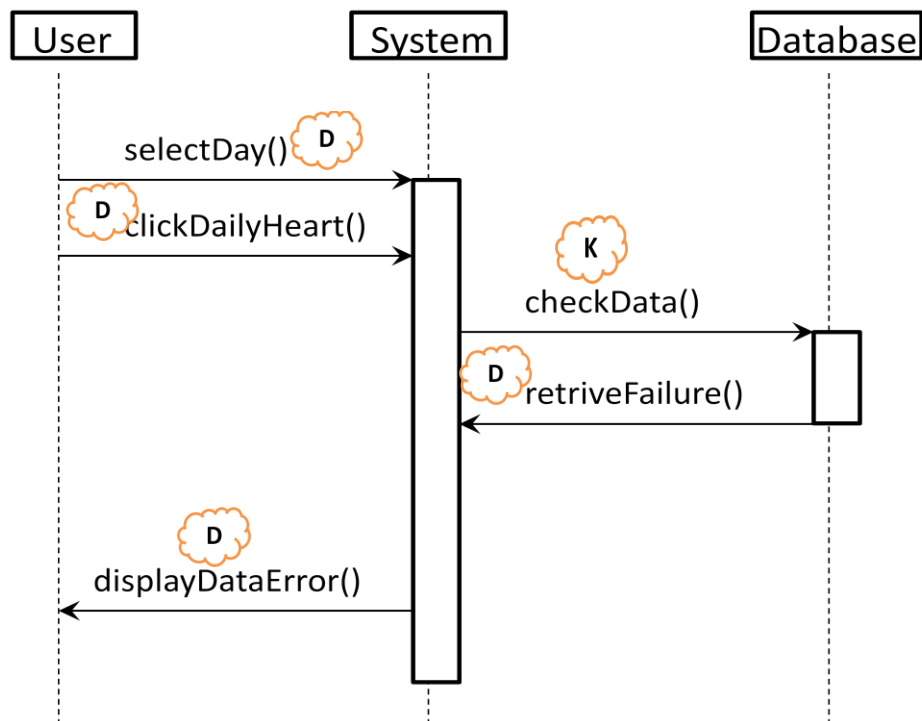


**UC5 ViewDailyHeartRate**

Success Case:

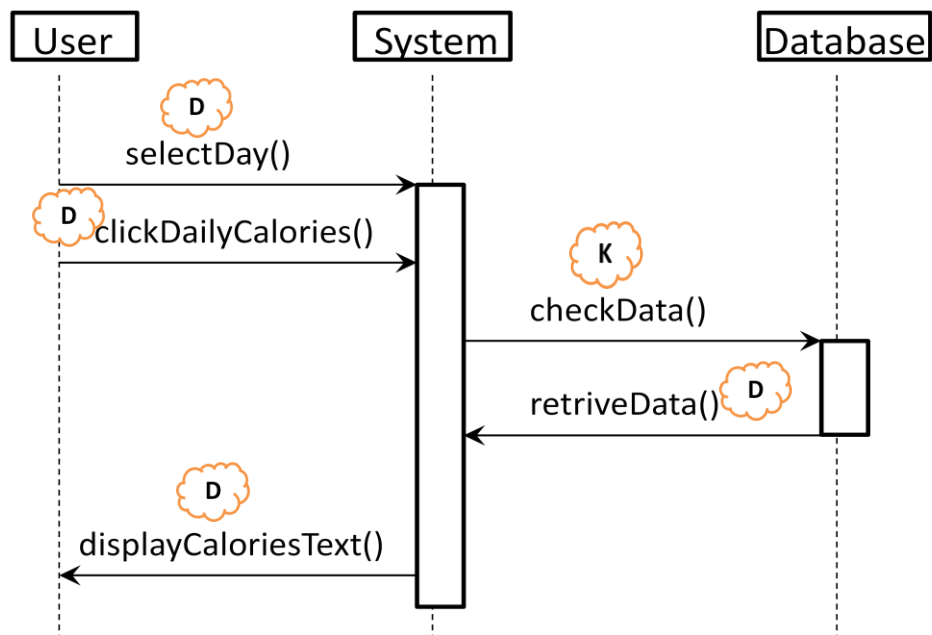


Alternate Case:

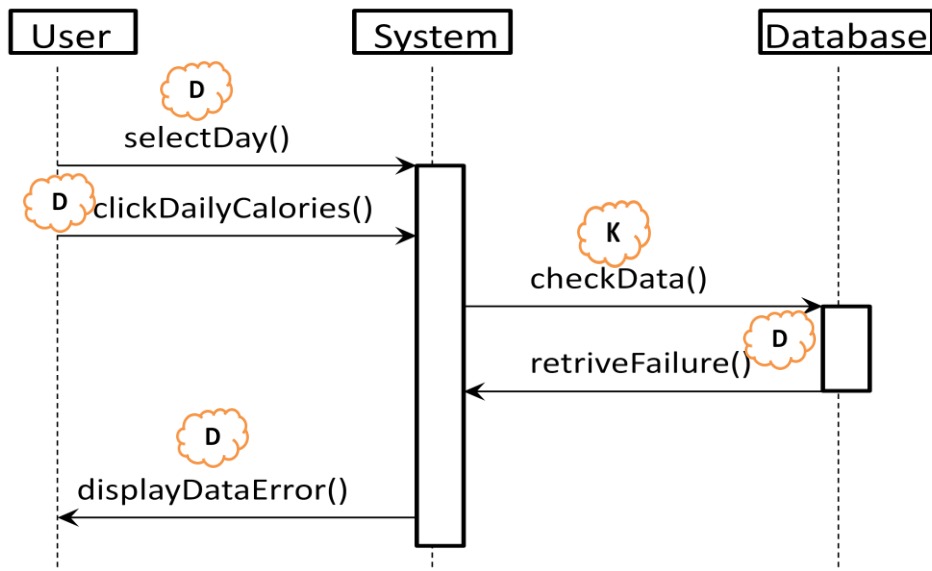


**UC6 ViewDailyCaloriesBurnt**

Success Case:

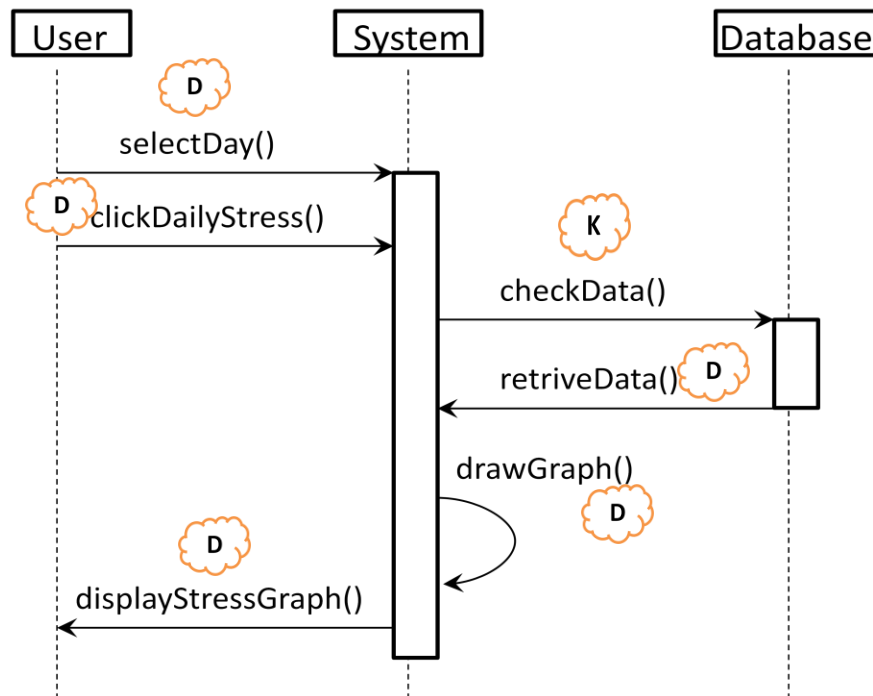


Alternate Case:

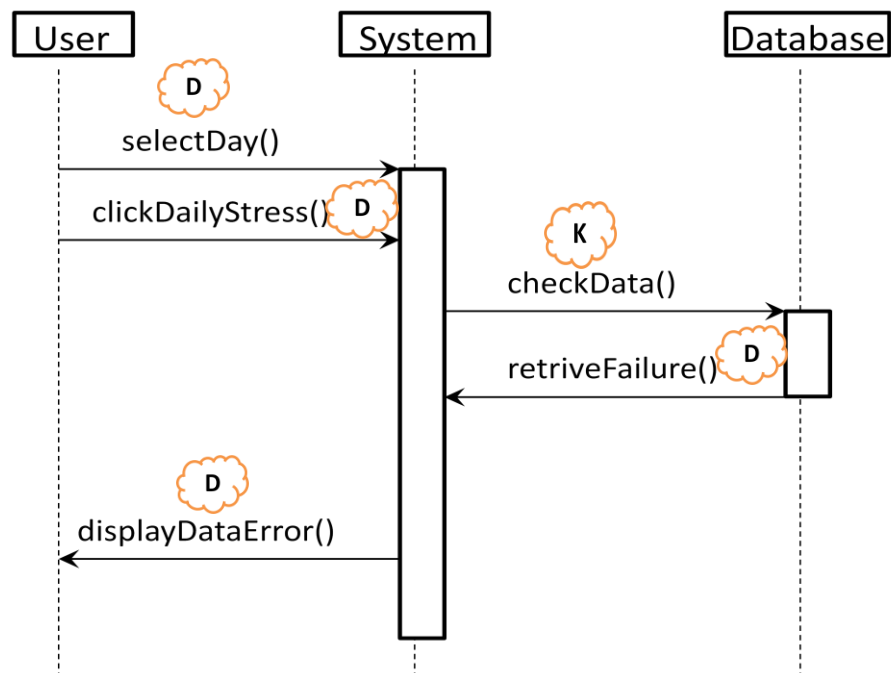


**UC7 ViewDailyStressLevel**

Success Case:

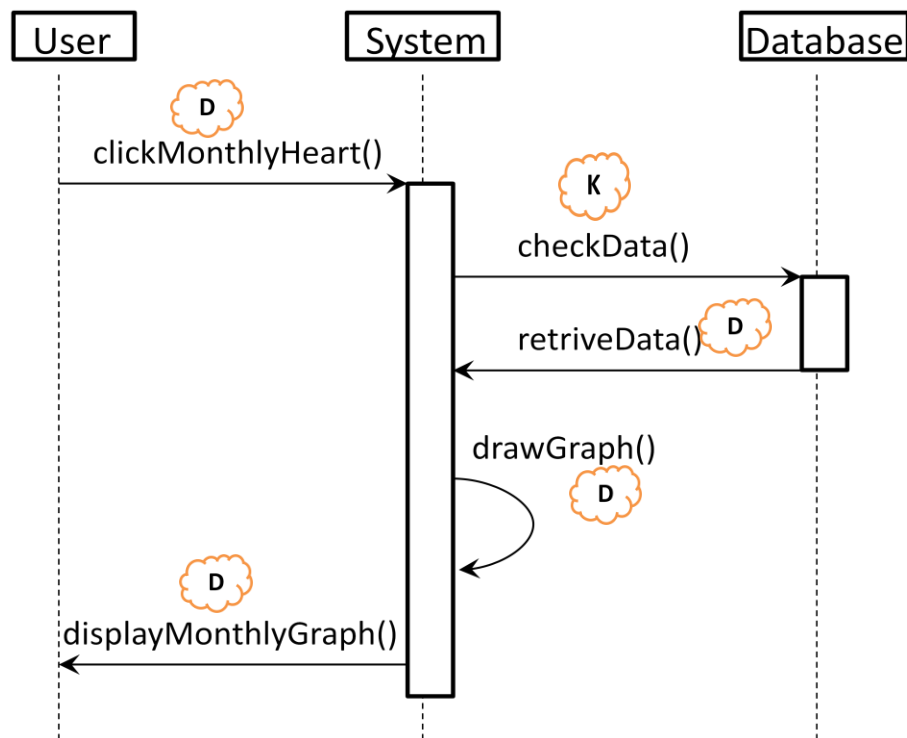


Alternate Case:

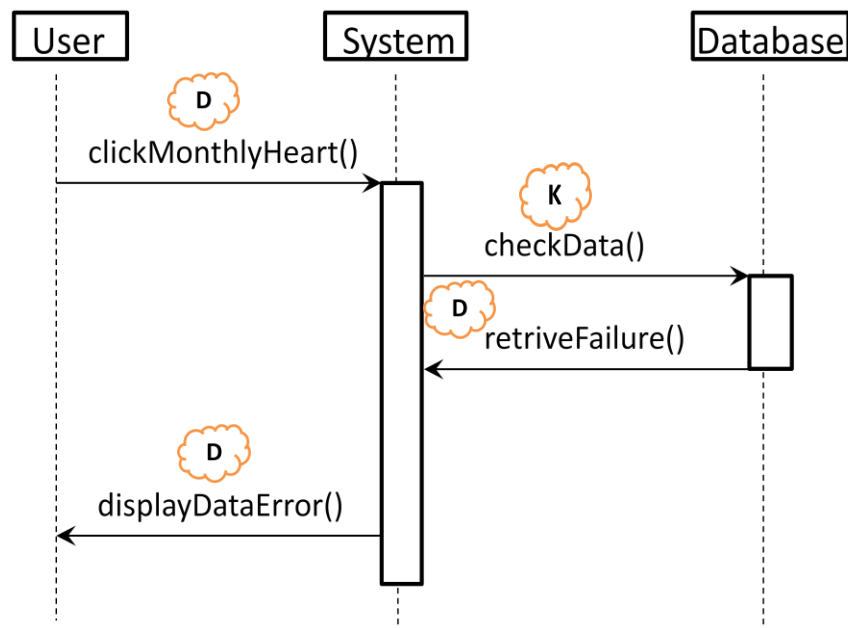


**UC8 ViewMonthlyHeartRate**

Success Case:

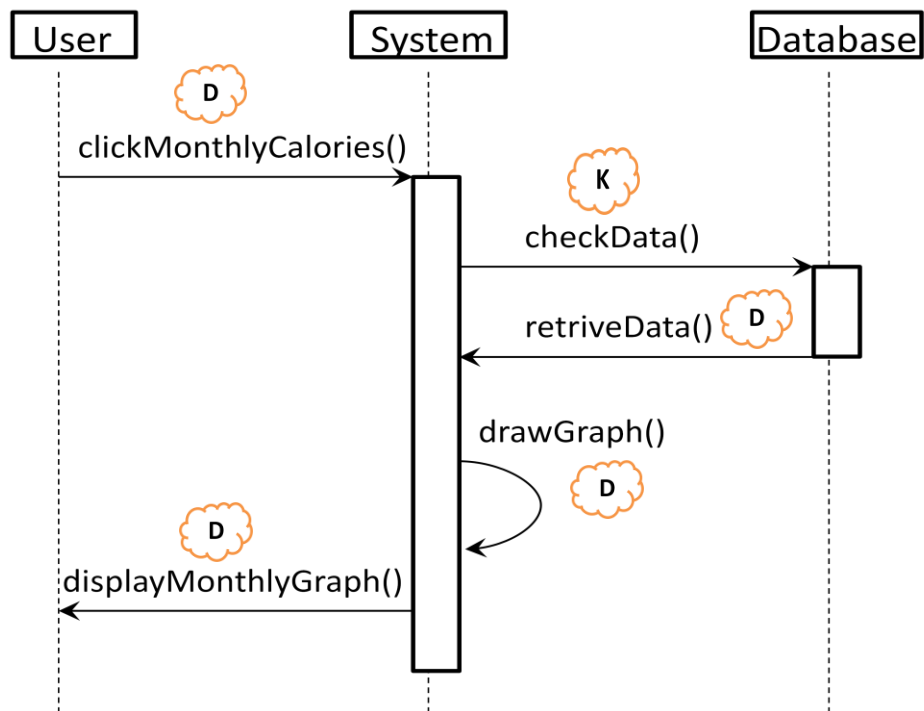


AlternateCase:

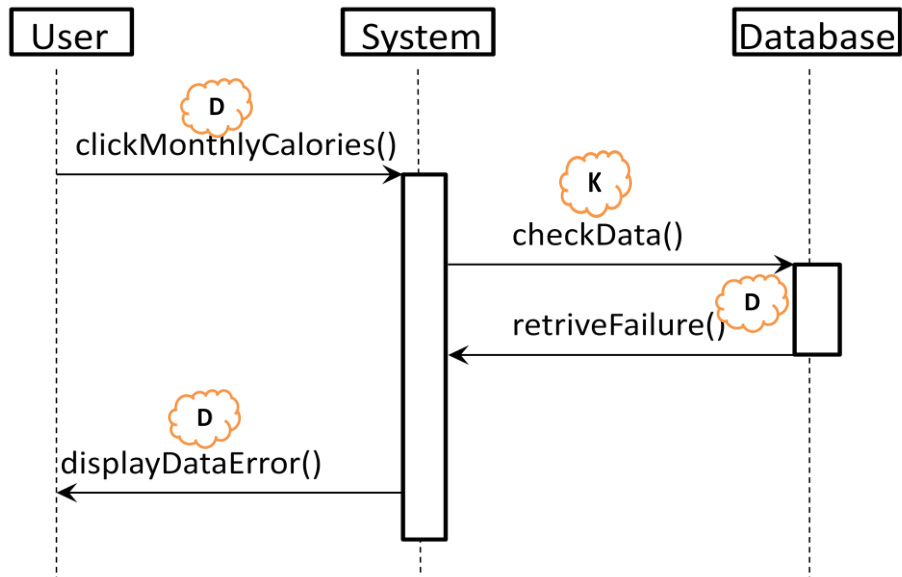


**UC9 ViewMonthlyCaloriesBurnt**

Success Case:

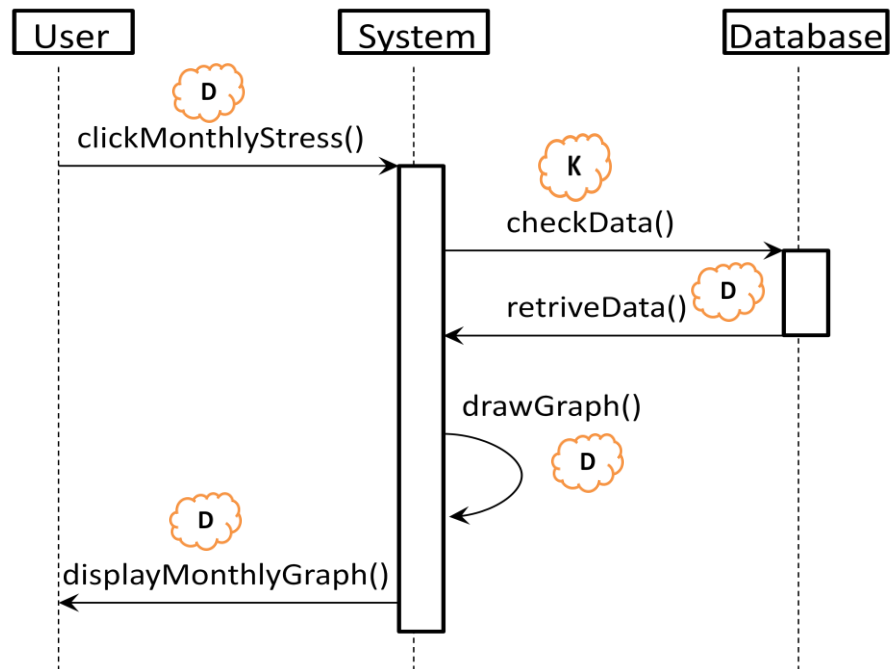


Alternate Case:

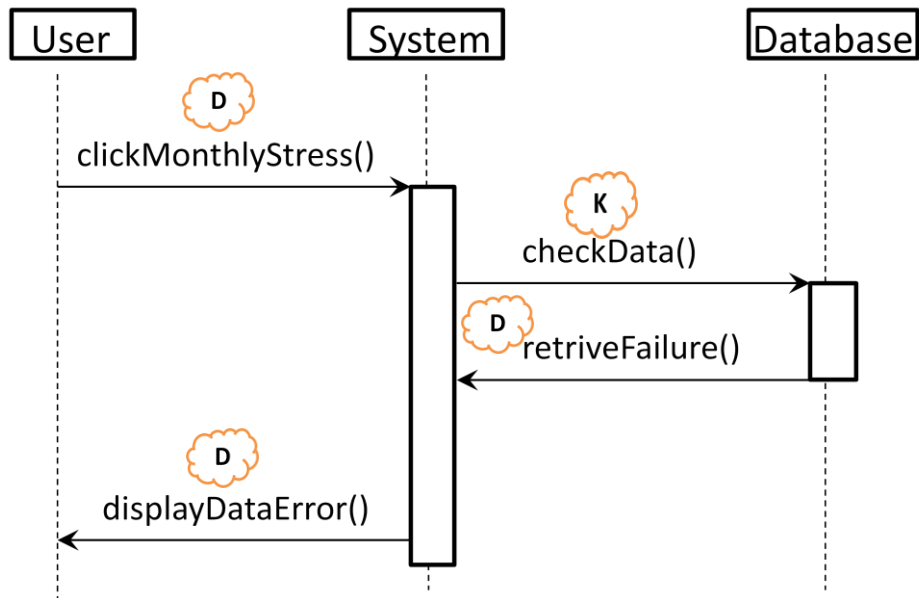


**UC10 ViewMonthlyStressLevel**

Success Case:

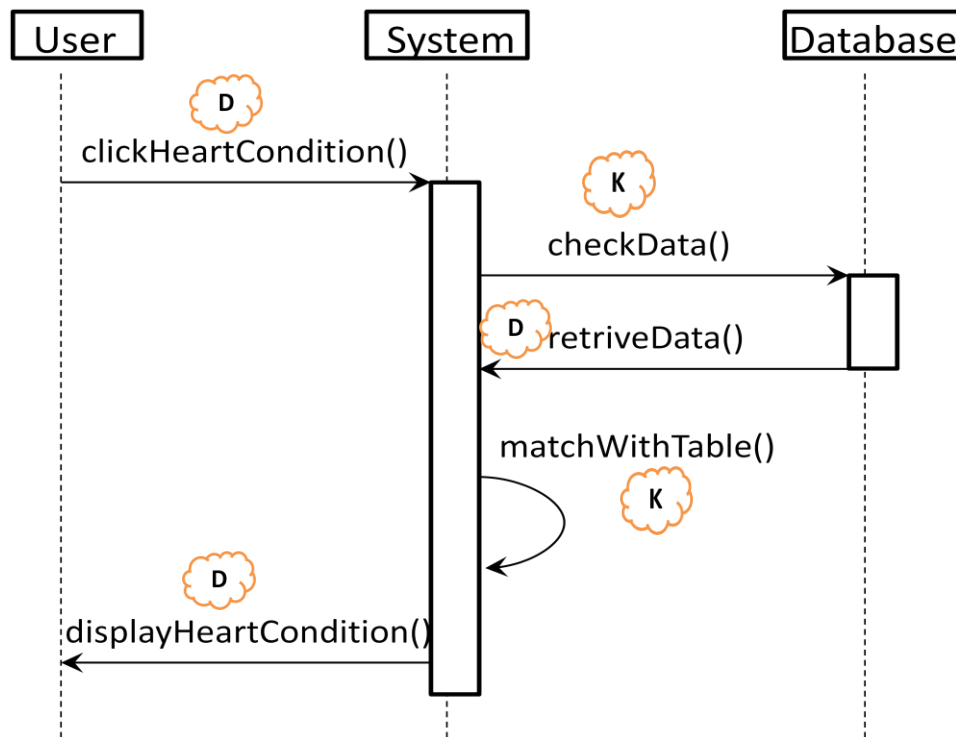


Alternate Case:

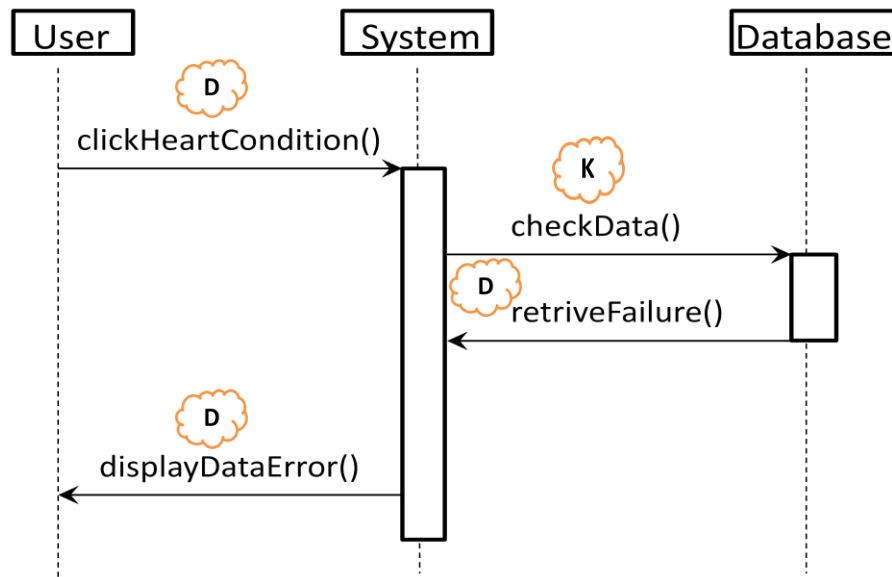


**UC11 ViewHeartCondition**

Success Case:



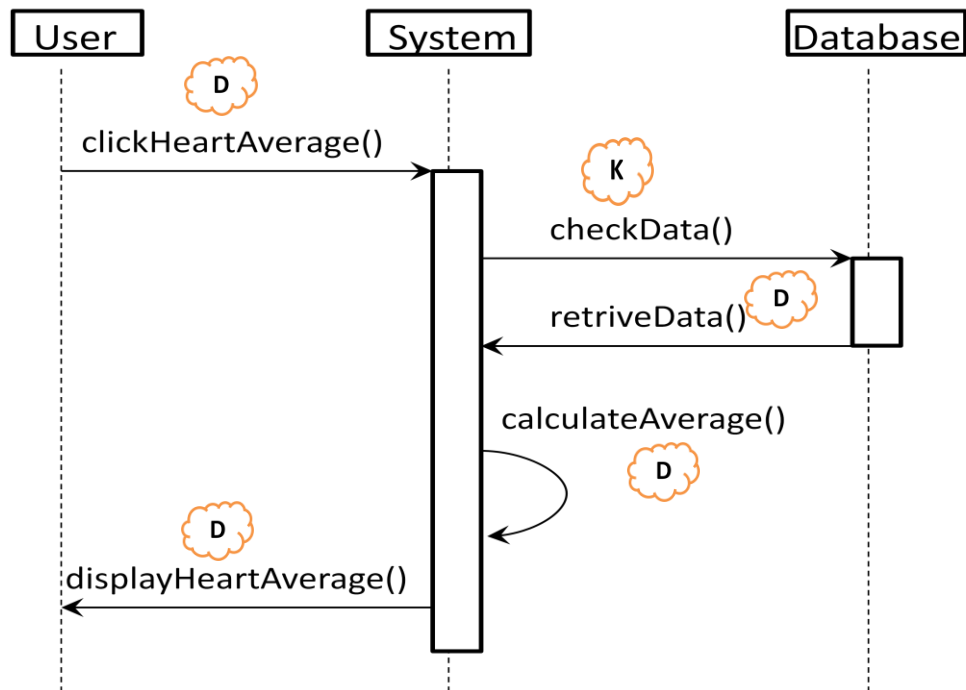
Alternate Case:



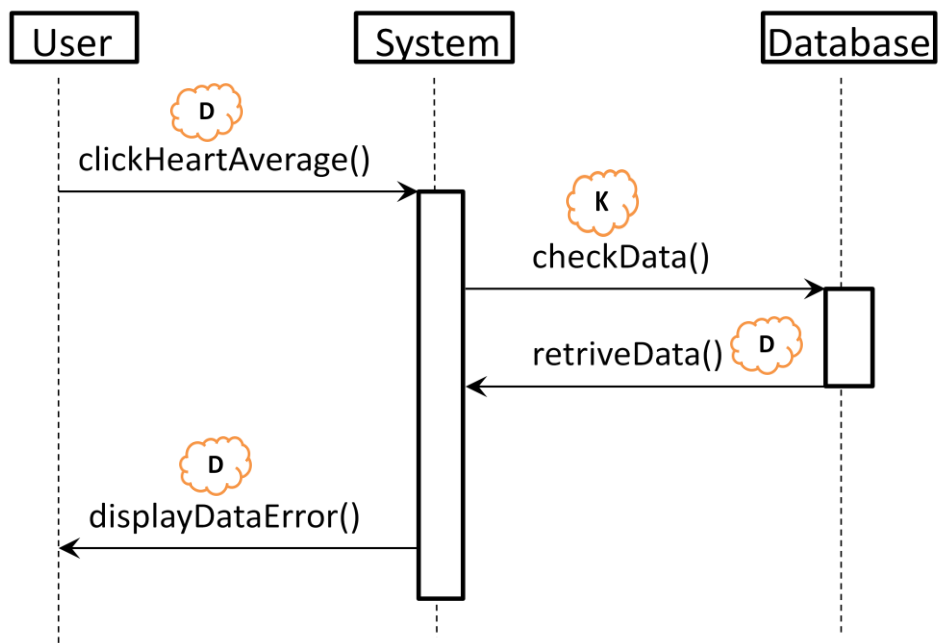


**UC12 CalculateDailyAverageHeartRate**

Success Case:

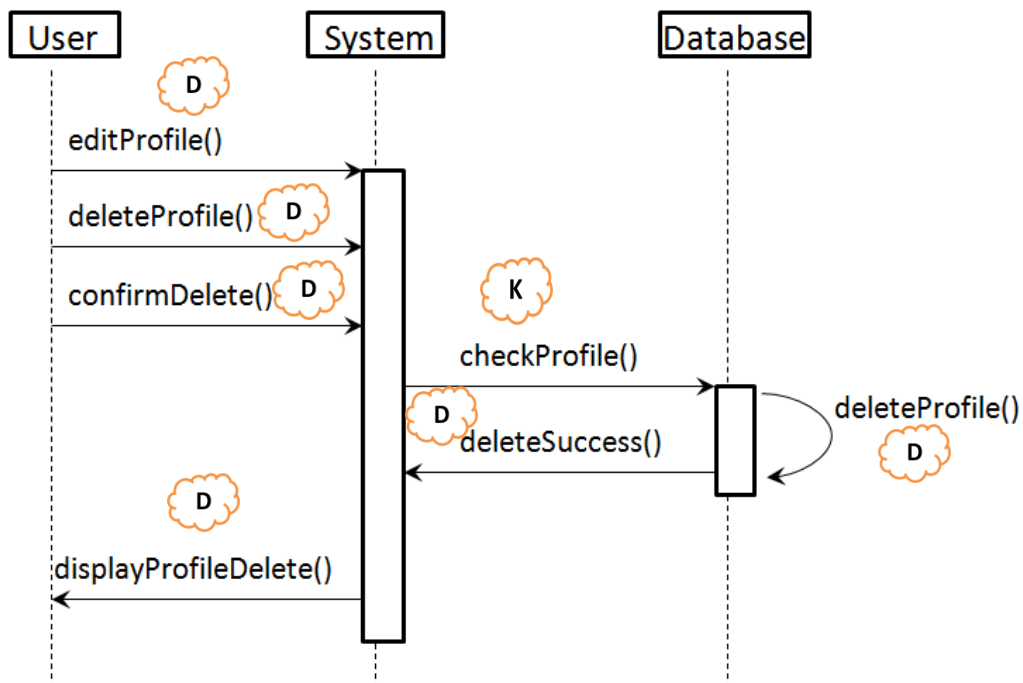


Alternate Case:

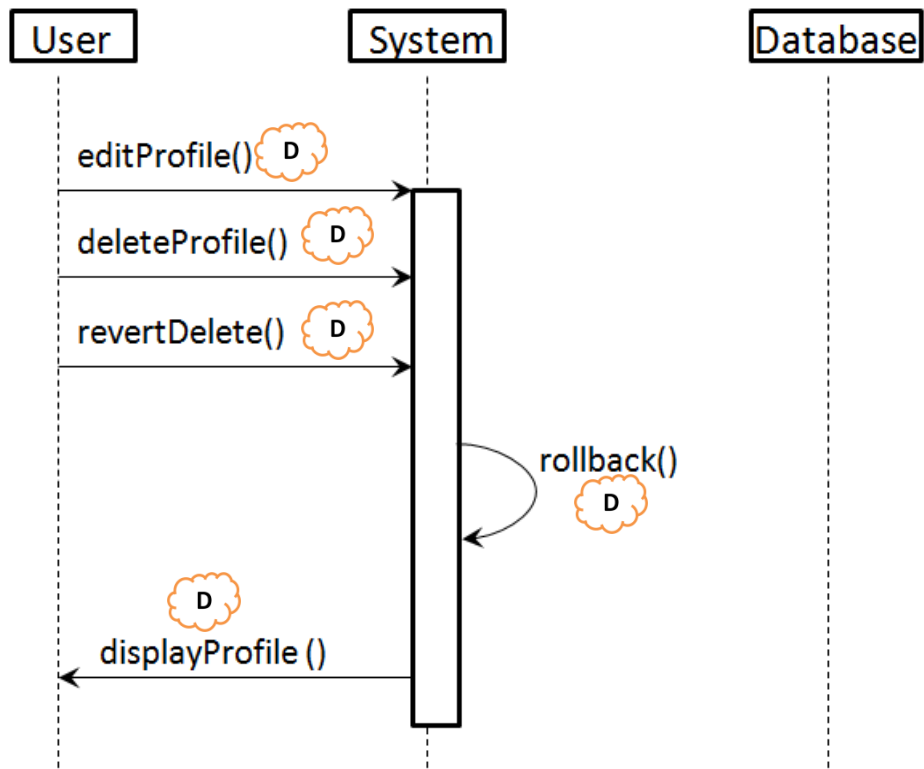


**UC13 DeleteProfile**

Success Case:

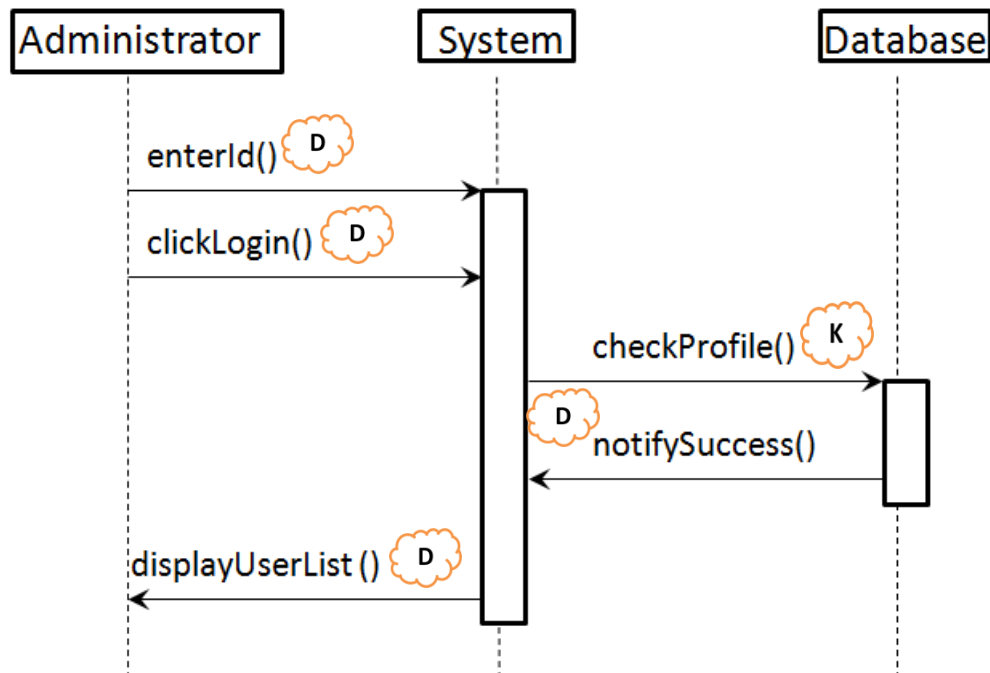


Alternate Case:

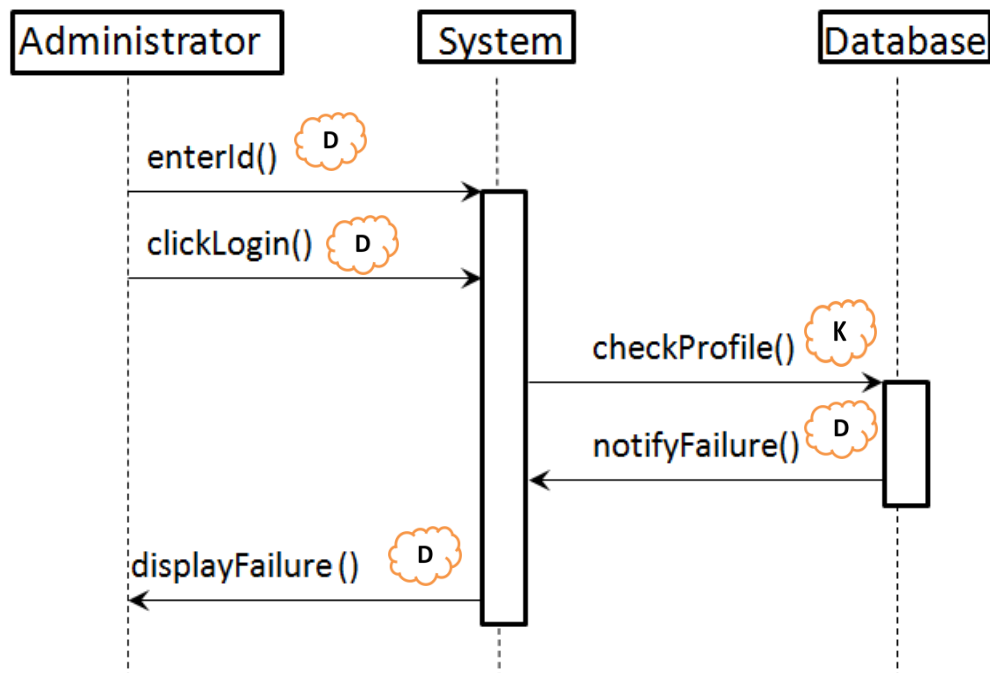


**UC14 RetrievePassword**

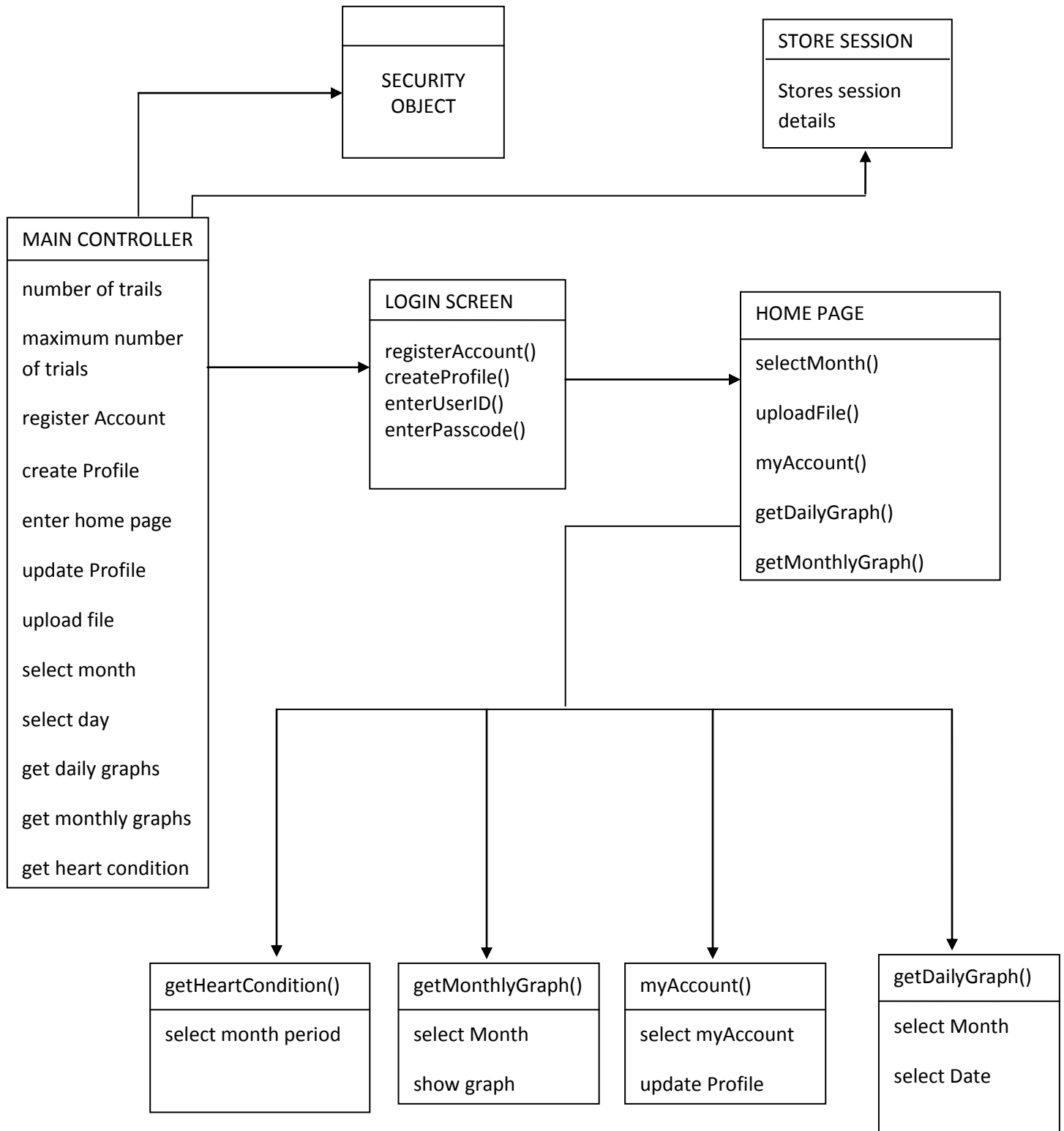
Success Case:



Alternate Case:



### 3. System Class Diagram



### 3.a. Class Diagram Description

NAME	DOCUMENTATION
MAIN CONTROLLER	The main controller controls all the activities of the application like login, controlling the activities of the web pages and accessing them.
SECURITY OBJECT	It authenticates if proper user is entering the website.
LOGIN SCREEN	It's the first page any user will see when they start the system. It will prompt for userId and password. If "new user" then registers profile/create personal profile.
HOME PAGE	When the user enters correct credential, the system takes the user to this screen. This includes all the month, account info etc. for the user to check.
SESSION STORER	This stores session for a user for each login.

### 3.b. Class Diagram Attributes and their Description

MAIN CONTROLLER	
number of trails	it describes the number of attempts a user is allowed to make
maximum number of trials	it describes the number of attempts a user is allowed to make
Register Account	Ask the user to give userId and password to create a new account.
Create Profile	User enters his personal information such as name, age, gender , weight, height and previous heart condition
enter home page	after successful login the user is taken to the enter home page
select month	select the month of which the data we want to see
Upload file	the user enters his data file into the system
select day	select the day of which you want to see the data
get daily graphs	daily graph showing the heart rate, min heart rate, maximum heart rate , average heart, stress level and calories burnt
get monthly graphs	Monthly graph showing the heart rate, min heart rate, maximum heart rate , average heart, stress level and calories burnt
get heart condition	heart rate condition that is if a person is suffering from Bradycardia, Tachycardia or Cardiovascular Mortality is shown.

LOGIN SCREEN	
enter userId	the user needs to enter his user ID
enter password	the user needs to enter his password

SECURITY OBJECT	
check login	It authenticate the userID
check password	It authenticate if the userID has given the correct password corresponding to the userID

SESSION STORE	
store session	It stores the user session for each login. After the user logout the back button will not allow the user to login without entering his credentials

SERVICE PAGE	
get Heart Condition	This function diagnoses heart conditions using pre-defined table in the database
get Monthly Graph	This function produces the monthly graph for heart rate, calories burnt and stress level
get Daily Graph	This function produces the daily graph for heart rate, calories burnt and stress level
my Account	This has the personal information for the user.

## 3. c. Traceability matrix

Class Domain concept	Create Account	update Profile	Upload file	select day	get daily graphs	get monthly graphs	get heart condition
RegisterProfile	X						
MonitorDataRead		X			X	X	X
MonitoringDataStorer		X	X				
MonitorUpdateProfile		X					
ViewDailyRecord				X	X		
CalculateMonthlyCondition						X	
ViewMonthlyRecord						X	
AnalyseHeartCondition							X
CalculateHeartRate					X	X	



## 4. System Architecture and System Design

### 4.a. Architectural Styles: 3-Tier:

The heart monitoring system uses the three- tier architecture. The three tier architecture has, as the name suggests, 3 layers. The Presentation layer which deals with the user interface or the look of the website. It helps the user to interact the system in a smooth manner giving the ability to understand the results. The logic tier is the "brain" of our system. It helps in processing various logical computations and algorithms which give user an understandable result of the raw data uploaded by the user. The data tier stores all the user data and also the logic tier retrieves data from this tier for processing.

In our system the presentation tier is handled by Drupal [4] which is a content management framework. The logical tier is handled by PHP script and database is MySQL.

The figure 1 shows a basic 3-tier architecture design.

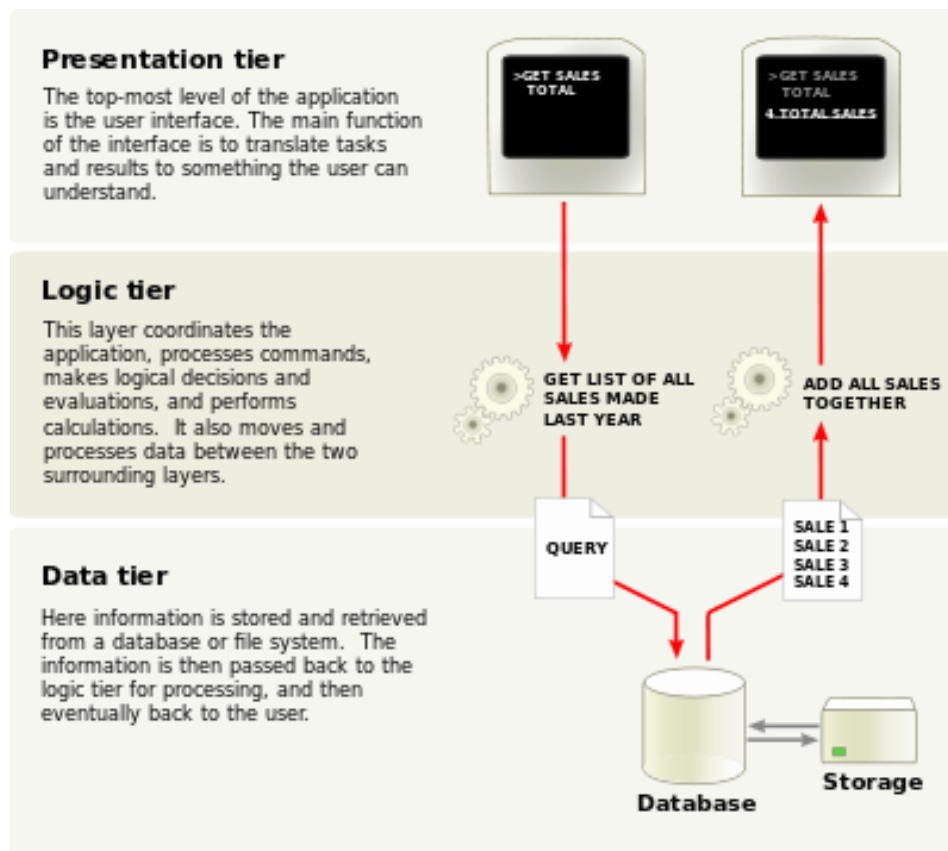


Fig 1. 3-tier architecture design

#### 4.b. Identifying subsystems:

The website is an online service and it requires the client to interact with the user and a server to maintain records of all the operations. The user accesses the web browser through a http protocol. The client uses PHP script and makes function calls to the server in background. Within the server, the client updates or retrieves the tables in the MySQL database whenever requested by the user.

The subsystems can thus be classified as User, which interacts with the client to perform required operation, Client, which interacts with the database and performs functions in the background to present the information requested by the user and Server, which is essentially a database maintaining all the information. The following diagram demonstrates how the user-client- server interactions take place.

**UML Package Diagram**

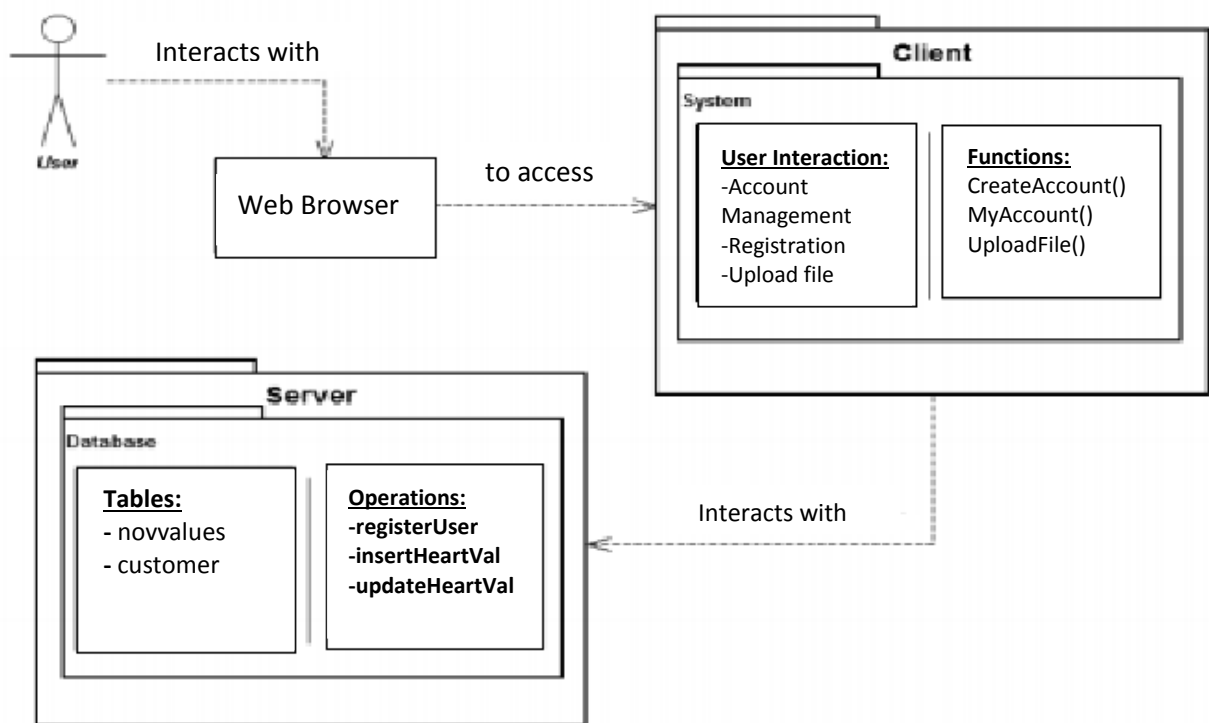


Fig 2: Client-server interaction diagram

**4.c. Mapping subsystems to hardware:**

The data collected in this system is stored in a database, which is located in the server. The server also stores the graphical user interface. The server that we use is a web server which keeps track of the customer information, heart rate values and stress levels of each customer and heart rate history.

The client computers interact with the server and will be able to run the system on the server. So, the client computers access the web server using a browser to perform certain operations like creating a user account, updating the user information and uploading a file.

#### 4.d. Persistent Data Storage:

MySQL was chosen as the database for our system. The basic tables used in the database are as follows:

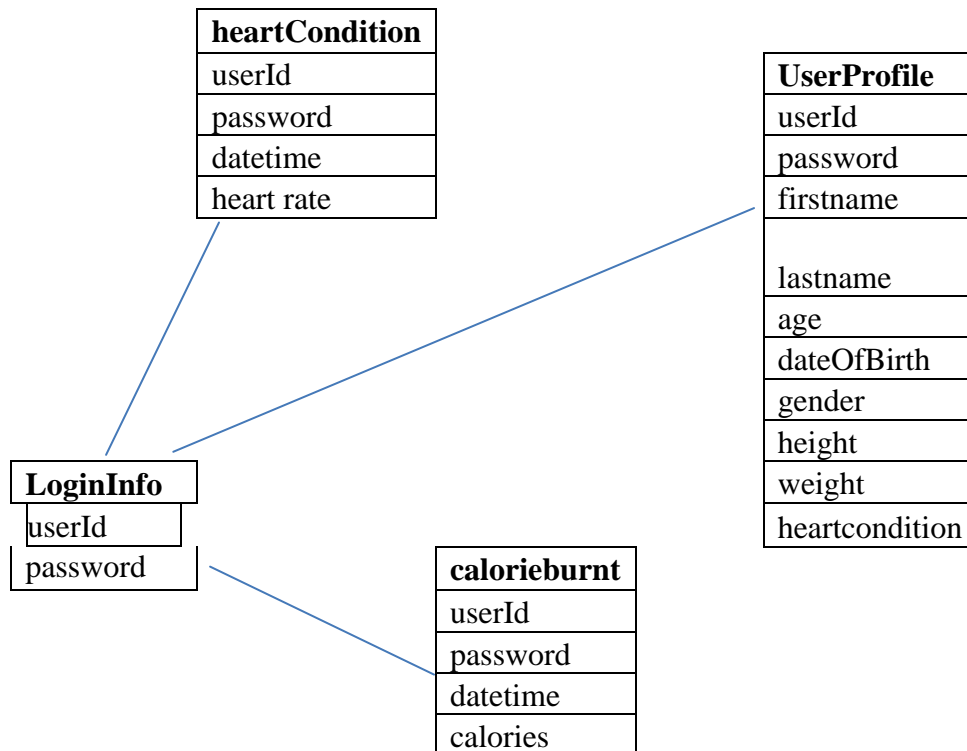
UserProfile

heartCondition

calorieburnt

LoginInfo

As our logic is in our application we can directly determine the stress level and heart condition using the heart rate. Hence we need not store this information daily in the database.



#### 4.e. Network protocol

HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the *client* [5] and an application running on a computer hosting a web site may be the *server* [6]. The client submits an HTTP *request* message to the server. The server, which provides *resources* such as HTML files and other content, or performs other functions on behalf of the client, returns a *response* message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

##### Security needs

As we need to encrypt our calls or authenticate our client, we decided to use HTTP-based application, whether that is an ASP.NET application or a remote application. .

##### Interoperation

HTTP application is best suited for heterogeneous systems and as our functionality is over web, we need a protocol which is interoperable over different inter networks.

##### Scalability

One challenge working with website is the number of users increase exponentially and we need a protocol which is scalable and as HTTP protocol is highly scalable which would suit over requirements.

The message formats for a HTTP protocol are clearly described in the below link.  
<http://www.w3.org/Protocols/rfc2616/rfc2616-sec4.html>

#### 4.f. Global control flow

The system is event driven. The user interacts with the system by first registering his user name and password. The user log in and the next step is the user upload the necessary data. This data gets stores into the database. Once the user clicks on the data he wants to see the proper output is shown in the form of daily and monthly graphs. The system allows multiple users to login and upload data.

#### 4.g. Hardware Requirement

The operating system required by our application is Windows XP/Vista 7, Mac. The web server is WAMP. Internet connection is necessary for the application. The application needs a minimum of 2Gbytes hard disk storage. Colour display is mandatory with a minimum resolution of 640 X 480 pixels as we are using graphs and charts in our user interfaces. These are the main hardware requirement for our system.

## 5. Algorithms

*The time for action is now. It's never too late to do something.* -Carl Sandburg

The heart is the most vital part of our body so it is important that we monitor it to keep our body at the best of its health. Heart rate is determined by the number of times your heart beats each minute, is an important measure of your health. How hard your heart has to work during various activities can tell you a lot about your overall physical condition. The normal heart rate of the human body is 60-100 beats per minute [7]. Generally, a lower heart rate at rest implies more efficient heart function and better cardiovascular fitness. For example, a well-trained athlete might have a normal resting heart rate closer to 40 beats a minute.

Age	18-25	26-35	36-45	46-55	56-65	65+
Athlete	49-55	49-54	50-56	50-57	51-56	50-55
Excellent	56-61	55-61	57-62	58-63	57-61	56-61
Good	62-65	62-65	63-66	64-67	62-67	62-65
Above Average	66-69	66-70	67-70	68-71	68-71	66-69
Average	70-73	71-74	71-75	72-76	72-75	70-73
Below Average	74-81	75-81	76-82	77-83	76-81	74-79
Poor	82+	82+	83+	84+	82+	80+

Fig 3: normal heart rate for men with corresponding ages

To measure your heart rate, simply check your pulse. Place your index and third fingers on your neck to the side of your windpipe. To check your pulse at your wrist, place two fingers between the bone and the tendon over your radial artery — which is located on the thumb side of your wrist.

When you feel your pulse, count the number of beats in 15 seconds. Multiply this number by 4 to calculate your beats per minute.

Keep in mind that many factors can influence heart rate, including:

- Activity level
- Fitness level
- Air temperature
- Body position (standing up or lying down, for example)
- Emotions
- Body size
- Medications

Although there's a wide range of normal, an unusually high or low heart rate may indicate an underlying problem. If your resting heart rate is consistently above 100 beats a minute it is a condition called tachycardia or if it is below 60 beats a minute it is called bradycardia. It could be accompanied by other symptoms such as fainting, dizziness or shortness of breath.

Hence it is very important that we detect these conditions at the earliest stage possible. Our system offers this by the continuous monitoring of the heart rate data.

Based on the age of the customer, we compare the heart rate values with average heart rate values and diagnose the heart condition of the customer.

Algorithm:

- Record the heart rate readings of the customer at resting time and active time

- Get the age of the customer from the user profile.

- Compare the age and relative heart rate with the ideal value for that age with the provided table.

- If it falls within the ideal range intimate the user that his heart is working perfectly

- If the value falls outside the particular range check whether it is higher or lower than the ideal range.

  - If it is lower ( $<50$ ) declare to the user that he has bradycardia

  - If it is higher ( $>100$ ) declare to the user that he has tachycardia.

## 6. User Interface Design and Implementation

In our report #1 we showed the screen mock-ups in which the user was able to see his heart rate, stress level and calories burnt in different screen as shown below. But now we have incorporated all the three graphs into one page. Thereby leading to simpler design and less user effort.



Fig 4 : Initial plan for screen



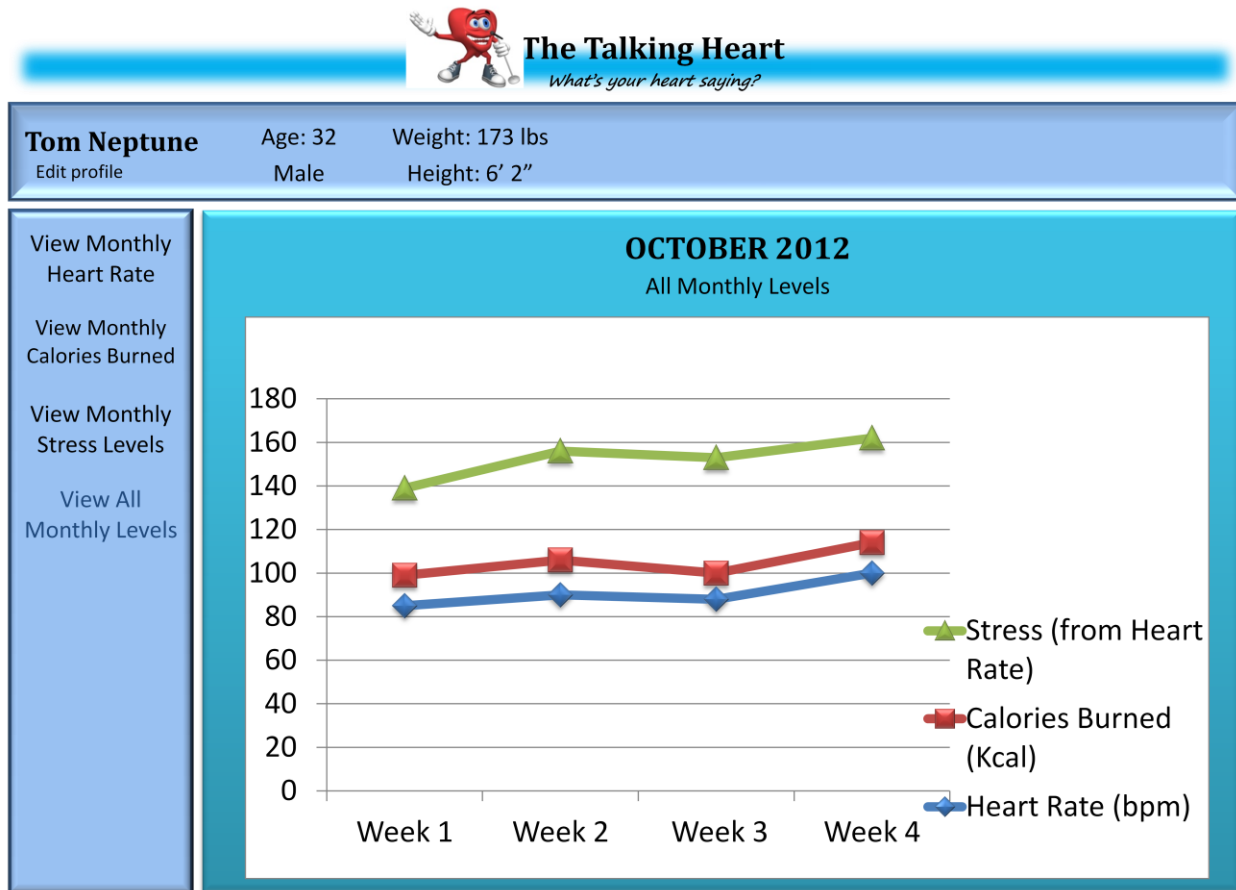


Fig 5 : Initial plan for screen

Also we have included minimum average and maximum heart rate and instead of just writing them in a text form we are showing it visually by the help of dials. In these dials we have red areas which are the "danger zones" signifying heart levels which are considered to be unhealthy.

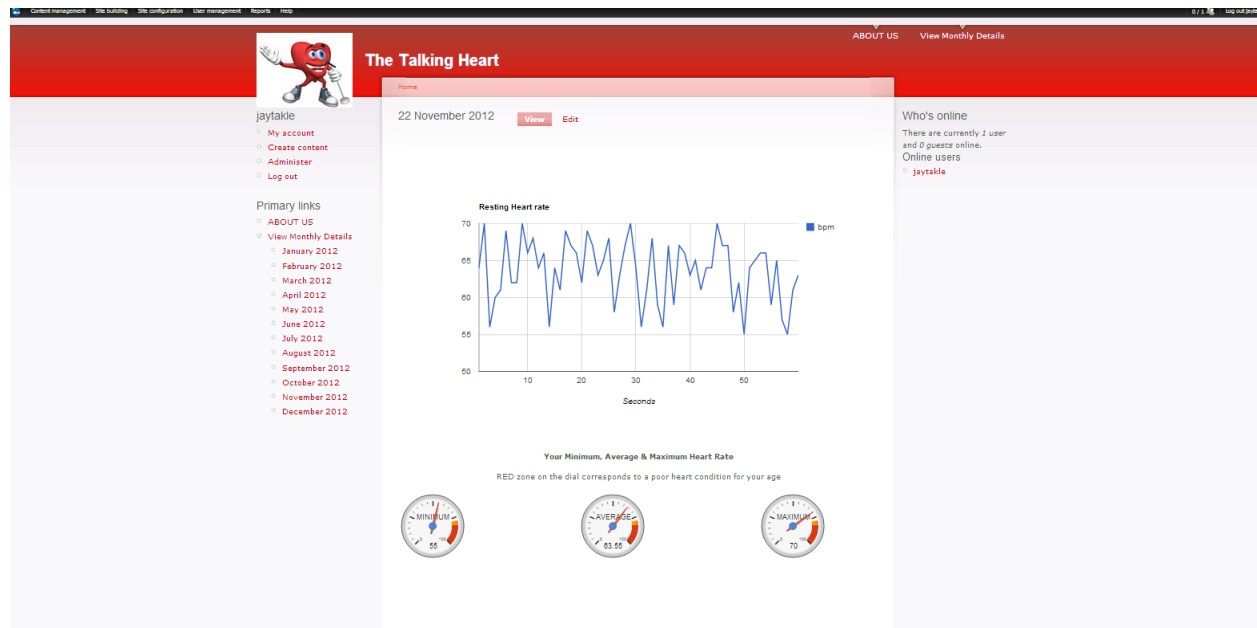


Fig 6 : current screen

There is another enhancement made to the screen mockups. We have added a bar graph. In this we have our average resting heart rate which is compared with a table which provides the average heart rate for athletes, an excellent /good/average/poor heart rates according to your age group. So instead of providing that table and making it difficult to understand we provide the bar graph as shown below. The first bar shows the users heart rate followed by the average heart rates for various categories as described above.



Fig 7 : current screen

## 7. Design of test

Below are the design of the test cases as per required by our project. These are the basic test cases. We need to check thoroughly that all the test cases are working properly, the screens are displayed in a proper format and there is no discrepancy in displayed graphs. We also need to check the code thoroughly for the correctness of algorithm and control flows.

### 1. Test ID : TC1\_Login

Assumption: the user is at login screen.

Input Requirement	Expected Output	Pass/Fail	Comments
Valid                      userId ,Password	Login successfully	Pass if the user is able to login and sees the home screen	This is to test that the system allow valid user to enter the website
Invalid                      UserId ,password	Display                      error message "Invalid userId/Password"	Pass if the user is unable to see the home screen	This test is to determine that invalid user should not enter the website

### 2. Test ID: TC2\_Register:

Assumption: the user is at the login screen.

Input Requirement	Expected Output	Pass/Fail	Comments
the user enters a userId which is unique	Register successfully	Pass if the user is able to register his UserId password.	This is to test that the system allow only unique user in the system with the unique userID
the user enters a userId which is pre-existing	Display                      error message "User Id already exist. Please choose another ID"	Pass if the user is unable to create a userId	This is to test that the system does not allow two user with the same userID

## 3. Test ID: TC3\_Upload

Assumption: the user is login and on the upload file screen

Input Requirement	Expected Output	Pass/Fail	Comments
the user tries to load a file with .csv file format and one column of heart reading	file upload successful	Pass if the user is able to upload the file and shown message "Successful upload"	This test is to determine that the system allows only .csv file to be loaded successfully
the user tries to load a file with .xls file format and one column of heart reading	Display error message "Incorrect file "	Pass if the user is unable to load the file and shown message "Unsuccessful load"	This test is to determine that the system does not allow other files to be loaded.
the user tries to load a file with .csv file format and two or three column of reading	Display error message "Incorrect file "	Pass if the user is unable to load the file and shown message "Unsuccessful load"	This test is to determine that the system allow only .csv file with proper format to be uploaded.

## 4. Test ID: TC4\_viewDailyData

Assumption: User is login and at the calendar screen.

Input Requirement	Expected Output	Pass/Fail	Comments
the user clicks on the day to see the uploaded data	the user is taken to the day screen and graphs are displayed with proper markings on x-axis and y-axis	Pass if the user is able to see his daily heart rate graph, daily minimum maximum heart rate and stress level.	This test is to determine that the user is able to see his heart graphs properly.
the user clicks on the day where no data is uploaded	Display message "no data uploaded"	Pass if the user is unable to see any graph	This test is to determine that the system does not display graph for day when no data is uploaded.

## 5. Test ID : TC5\_viewMonthlyData

Assumption: User is login and at the calendar screen.

Input Requirement	Expected Output	Pass/Fail	Comments
the user clicks on the month to see the uploaded data	the user is taken to the month screen and graphs are displayed with proper markings on x-axis and y-axis	Pass if the user is able to see his monthly heart rate graph, monthly minimum maximum heart rate and stress level.	This test is to determine that the user is able to see his daily heart graphs properly.
the user clicks on the month where no data is uploaded	Display message "no data uploaded"	Pass if the user is unable to see any graph	This test is to determine that the system does not display graph for a month when no data is uploaded.

## 6. Test ID: TC6\_viewHeartCondition

Assumption: the user is at login screen and at the home screen.

Input Requirement	Expected Output	Pass/Fail	Comments
The user clicks on view heart condition.	The user is shown his heart condition.	Pass if the user is able to see his heart condition	This test is to determine that the user is able to see their heart condition properly.
the user clicks on view heart condition when no data is uploaded for the month	Display message "no data uploaded"	Pass if the user is unable to see his heart condition	This test is to determine that the system does not display heart condition when no data is uploaded.

## 7. Test ID: TC7\_updateProfile

Assumption: the user is at login screen and at the “my account” screen.

Input Requirement	Expected Output	Pass/Fail	Comments
The user clicks on update profile and updates proper data.	the user is able to update his profile successfully	Pass if user is able to update his profile	This test is to determine that the system allows the user to update his profile successfully.
The user clicks on update profile and updates with incorrect data.	Display message "incorrect format please try again"	Pass if user is unable to update his profile with incorrect data.	This test is to determine that the system allows the user to update his profile successfully with correct format data.

## **8. Project management**

### **What we did for Interactions diagrams:**

We had already worked on most of the sequence diagrams in the report #1. For this report we referred to report #1 to improvise. For better understanding of the concepts of UML diagrams we referred to online documentation and Dr. Marsic's Software Engineering book.

We had a team meeting where we went over the peer reviews for report #1 focusing mainly on the suggestions on improvement of the sequence diagrams. We also had a brain storming session on missing use cases and the alternate scenarios. We enumerated a possible list of use cases and every member was assigned two use cases to improvise and draw interactions diagrams.

After collating all the interaction diagrams each of us reviewed the diagrams for any errors. We rectified all the errors.

### **What we did for Class diagram**

We went through the book as well as the internet and learnt how to draw the class diagram. After getting enough information for the same we looked into the classes and methods which are implemented in our project. After identifying the classes we identified the relationship between various classes and drew the diagram.

### **Merging the contribution from individual team member**

We faced challenge this time as all of us had some or the other commitments merging the individual contribution from each member on time became a hassle due to which we were unable to submit the report on time. But finally we managed to complete the task and deliver this report.

### **Project Coordination and Progress Report**

We have completed the UC1, UC2, UC5, UC7, UC12 and are working. Currently we are working monthly display of information i.e. UC8, UC9, UC10. We will work on completing the heart condition UC11 and will eventually complete the remainder of the use cases.

## Plan of Work

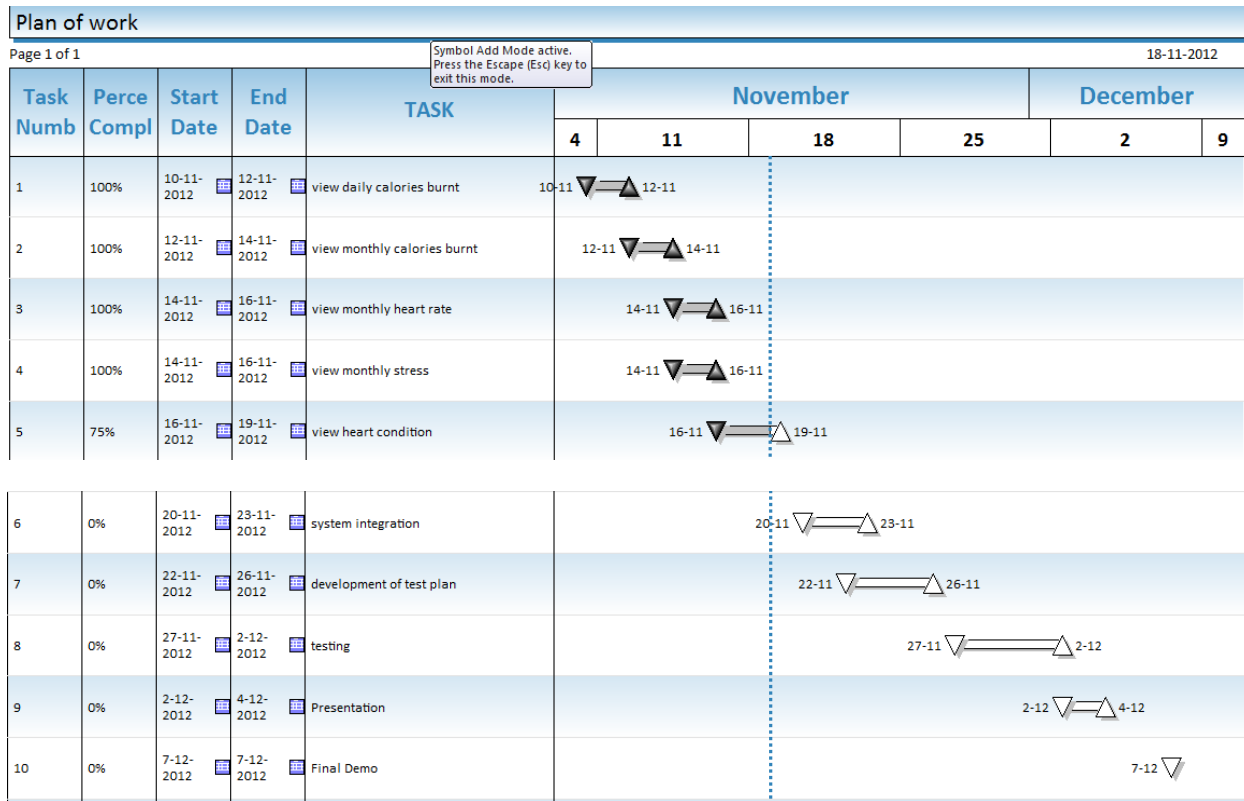


Fig 8 : Plan of Work

### Breakdown of Responsibilities:

The team comprises of 6 members.

The following are the modules developed or currently working on.

Login&Registration

UploadFile

viewDailyHeartRate

viewDailyStress

viewHeartCondition

viewMonthlyStress

viewCaloriesBurnt

viewMonthlyHeartRate

viewProfile

updateProfile

Anusha/Anu

Prasoon/Vinayak

Jay/Anu

Jay/Bhumika

Jay/Bhumika

Prasoon/Anusha

Jay/Anusha

Jay/Bhumika

Pasoon/Vinayak

Vinayak/Anu

We have equally divided the work among ourselves. The integration will be done by Jay, Prasoon and Bhumika. The integration testing will be performed by Anusha, Vinayak and Anu.



## 9. References

- [1] **Software Engineering , Dr. Ivan Marsic**
- [2] **"Unified Modelling Language"** [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)
- [3] **"Interactions Diagrams"** <http://www.cs.unc.edu/~stotts/145/CRC/Interactions.html>
- [4] **"Drupal"** <http://en.wikipedia.org/wiki/Drupal>
- [5] **"Client"** [http://en.wikipedia.org/wiki/Client\\_\(computing\)](http://en.wikipedia.org/wiki/Client_(computing))
- [6] **"Server"** [http://en.wikipedia.org/wiki/Server\\_\(computing\)](http://en.wikipedia.org/wiki/Server_(computing))
- [7] **"Heart rate"** <http://www.mayoclinic.com/health/heart-rate/AN01906>