SOFTWARE ENGINEERING

Sleep Quality Assessment

..........How well do you sleep???

Report 3

(Submitted on 12/12/2013)

Cagdas Karatas	cagdas@cac.rutgers.edu
Chenyun Lan	lan.chenyun@hotmail.com
Jianjun Zhu	zjjjun1991@hotmail.com
Mridula Krishnapur	mridu89@gmail.com
Yangkai Zhou	zhouyangkai1991@hotmail.com
Zeya Zhang	tooth6612@gmail.com

Project Website: here

All team members contributed equally.

Contents

SUMMARY OF CHANGES

Section 3.3, 3.3.2, 3.4, 3.5 : Change and rearrangement of Use Cases and Sequence Diagrams

Section 5.1: Removal of certain concept definitions in Domain Model to avoid redundancy

Section 6.0: Corresponding Change in Interaction Diagrams to match with Use Cases

Section 9.0: Fine tuning of Algorithms used

Section 10.0 : Updated User- Interface

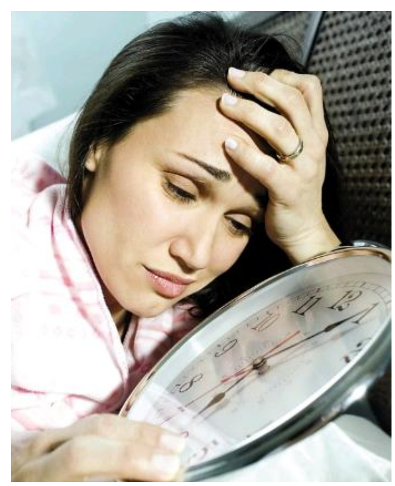
Section : Change in requirements: Removal of validation of My Zeo and MOTOACTV results. To justify please read section 12: Comparing My Zeo and MOTOACTV: an indepth analysis

1. CUSTOMER STATEMENT OF REQUIREMENTS

1.1 Introduction

"Healthy brains depend on healthy sleep; Healthy bodies depend on healthy sleep"

These are two of the Golden Sleep Principles from the website of World Sleep Day Org. The World Sleep Day is an annual event organized by the World Sleep Day Committee of the World Association of Sleep Medicine (WASM) since 2008. It is aimed to celebrate the benefits of good and healthy sleep and to draw society attention to the burden of sleep problems and their medicine, education and social aspects; to promote sleep disorders prevention and management.Through the World Sleep Day the WASM tries to raise awareness of sleep disorders and their better understanding and preventability, and to reduce the burden of sleep problems on society that constitute a global epidemic and threaten health and quality of life for as much as 45% of the world's population.



Good sleep can eliminate fatigue and resume physical strength, protect the brain, restore energy, enhance immunity, anti-aging and promote longevity. However, many people are suffering the insomnia. Insomnia has become a serious disease in the last decade, 45% of the world's population are suffering this disease. Insomnia is the inability to sleep or inability to stay asleep, leading to lack of sleep. It is also known as initiating and maintaining sleep disorders, for a variety of causes difficulty falling asleep, sleep depth or frequency through the short, early awakening, and inadequate or poor quality sleep. Insomnia is a common disease. It often makes patients feel great pain and psychological burden. Abusing drugs will damage patients' bodies and hurt many other aspects. And the social activities will be negatively affected during daytime. These show that a high quality of sleep is very important to everybody.

1.2 Background

A good sleep can be quantified and detected based on ECG and heartbeats.When people have a good sleep, their recorded heartbeats are stable without abnormal pulses. Scientifically there are five stages of sleep:

Stage 1: Beginning of sleep, relatively light stage of sleep, can be considered as a transition period between awake and sleep. It lasts about 5 ~ 10 mins.

Stage 2: It lasts about 20 mins, heart rate begins to decrease.

Stage 3: Transition period between light sleep to very deep sleep.

Stage 4: Stage 4 is a deep sleep that lasts approximately 30 mins, bedwetting and sleepwalking usually happen at the end of stage 4 sleep.

Stage 5: Most dreaming occurs during this stage, known as Rapid Eye Movement(REM) sleep. During this stage, body system become more active, and heart rate is supposed to rise up. On an average, we enter the REM stage approximately 90 mins after falling asleep. The lasting time of REM stage might get longer with each sleep cycle, up to an hour as sleep progresses.

There exists a clear relationship between the heart rate while asleep VS. awake. Dr. K. Krrauchi, in a study reported in “Neuropsychopharmacology” (2001) detected an average drop from 64 to 52 beats per minute from lights off till you reach light, continuous sleep. Paper concludes that subjects’ heart rates vary between sleep and awake. The low frequency power as well as the high frequency power was lower when the subjects were asleep. There is a relationship between the variation of heart rate and a specific sleep stage. Basically, the larger variation usually goes with the REM sleep. Difference of heart rate variation between REM sleep and Non-REM sleep may be used to distinguish the sleep stages. On the other hand, in the frequency domain, one study has showed that compared to Non-REM sleep, low frequency band power has decreased and low frequency (LF) to (HF) ratio has significantly increased during REM sleep.

Based on the scientific findings as cited above, we aim to create a model to tell apart sleep stage and awake stage in a whole sleep cycle. We intend to use MOTOACTV which gives us a log of the heart rate of a person to identify the stages of sleep in the customer and make use of this data to diagnose a possible sleep disorder, report the quality of sleep and finally suggest improvement in sleeping habits.

1.3 Project Overview

Sleep Quality Assessment as a product must provide the following experience to the user:

The user must be create a log in with SQA's interface;

The user then puts in important contact information like phone number, time he usually sleeps etc;

The user presses start button on MOTOACTV when he/she lies down;

When the user wakes up in the morning, he/she presses stop;

The data is logged and analyzed at the server and a report of the night's sleep is emailed to the user;

The user receives a report with suggestions to improve sleep quality and possible anomalies;

The same process repeats as long as the user wishes to.

This system uses two devices for collecting the sleeping data from user, they are Motoactv and Zeo. We use Motoactv as the main device, and Zeo to be the supplement device. Here are the introduction for these two deceives.

(1)MOTOACTV

Motoactv is an exciting portable product for fitness use. It’s a GPS-enabled watch with a touch screen, and is able to connect wirelessly to

a heart rate monitor. Armed with this product, we can get the walking distance, steps being taken, as well as the calories consumed.

By using the heart rate monitor, a chest strap, we can get the important heart rate data which would be used to fulfill our “qualified sleep” goal. So far, we have been able to get the actual heart rate data and export the data to our computer through the Motoactv website. However, it will definitely be inconvenient to the users if every time they want to do some sleeping quality analysis, they need to download the data from website. So here comes our task: to come up with a feasible method (a program) to get the data from the website conveniently, in which scenario the users only has to do one clicking to get the data they want. Then they can use the data as input, and use our software: Sleep Quality Assessment, they can easily get the result of their sleep quality.

(2)zeo sleep monitor

Even though we can extract the sleep patterns from Heart Rate, we want to incorporate this data with EEG data. Although there are many existing commercial products are available we choose MyZeo Sleep Manager for two reasons:

Device is by itself a sleep manager and it has capabilities of extracting sleep patterns from EEG data. By this feature we can even use MyZeo as a backup solution or verification purposes.

Device availability: As we consider limited time and budget of this project we had to choose something cheap and easily accessible. MyZeo Sleep Manager is provided by the customer at zero-cost.

Although aforementioned benefits of MyZeo Sleep manager the biggest challenge is device is no longer available and officially not supported. Fortunately, a couple of developers provide APIs which had been officially provided by the producer until they went out of business. As the team we accept MyZeo part of the project as a risk and to minimize this risk, we define ECG as *core feature* and EEG as a *supplementary feature*.

1.4 Glossary of Key Terms

EEG – EEG is the recording of electrical activity along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain. third clinical use of EEG is for studies of sleep and distinguishing sleep stages.

ECG – ECG is an interpretation of the electrical activity of the heart over a period of time, as detected by electrodes attached to the surface of the skin and recorded by a device external to the body.

Resting Heart Rate – The heart rate, measured in beats per minute (bpm), measured when the subject is awake, but has not performed physical activity. The resting heart rate is an indicator of general health. It is also a reference to compare the heart rate between sleeping and awake.

Maximum Heart Rate – The highest rate of heart beats per minute that is achieved during sleeping.

Minimum Heart Rate – The lowest rate of heart beats per minute that is achieved during sleeping.

Sleeping stage – Sleep proceeds in cycles of rapid eye movement (REM) and non-rapid eye movement (NREM), usually four or five of them per night. And NREM can be further divided into three sub-stages.

Sleep apnea – Sleep apnea is a type of sleep disorder characterized by pauses in breathing or instances of shallow or infrequent breathing during sleep.

Sleep report – A recording of data associated with sleep including the number of times users wake up during sleep, the time slept, the sleep quality index and a figure showing the stages of sleep over the sleep time.

REM – REM sleep is a normal stage of sleep characterized by the rapid and random movement of the eyes. REM sleep is considered the lightest stage of sleep so it would be better to wake someone up during this stage.

NREM – NREM(Non-rapid eye movement) sleep is, collectively, sleep stages 1–3, There are distinct EEG and other characteristics seen in each stage. Unlike REM sleep, there is usually little or no eye movement during this stage.

Health devices – some devices that can help your training or sleep by giving a report of your activities. Here we use Motoactv to record heart rate during sleep and use Zeo sleep monitor to extract sleep patterns from EEG data.

Database – The storage of all relevant Sleep Quality Assessment data, including users’ profiles (name, age, gender, weight), heart rate data and the time that users sleep and wake.

User level – The mark distinguishes whether a user have access to the sleep quality of other users’。

2. SYSTEM REQUIREMENTS

2.1 Requirements Summary

To fulfill the user requirements for the problem statement, Sleep Quality Assessment can collect the heart rate data from the mobile device that allow users to detect their sleep quality. The basic requirement for any user to be able to use Sleep Quality Assessment is that heart rate data must be recorded with a heart rate monitor during the user is sleeping. The report of the sleep quality with suggestions should be provided when the user finish the data collection. It is suggested the user to collect more sleeping data as possible as they can, so the application can provide a more accurate sleeping assessment. Also, the interface should be designed user-friendly, to confirm the user can use the application easily.

2.2 Function Requirements Table

ID	Priority Weight	Requirement
REQ-1	5	The system shall be able to collect the heart rate data from user.
REQ-2	5	The system shall be able to store the heart rate data in the database.
REQ-3	5	Compare My Zeo and MOTOACTV's results
REQ-4	4	Download patterns and display patterns from Myzeo
REQ-5	5	The system should provide suggestions to user to improve the sleep quality.
REQ-6	2	The system should be able to diagnose sleep apnea
REQ-7	4	Download heart rate data from Motoactv automatically
REQ-8	4	Display Heart Rate Data
REQ-9	4	The system shall be able to distinguish different sleep stages(including awake, asleep, deep sleep, REM, NREM)
REQ-10	2	The administrator should be able to access the user account data.

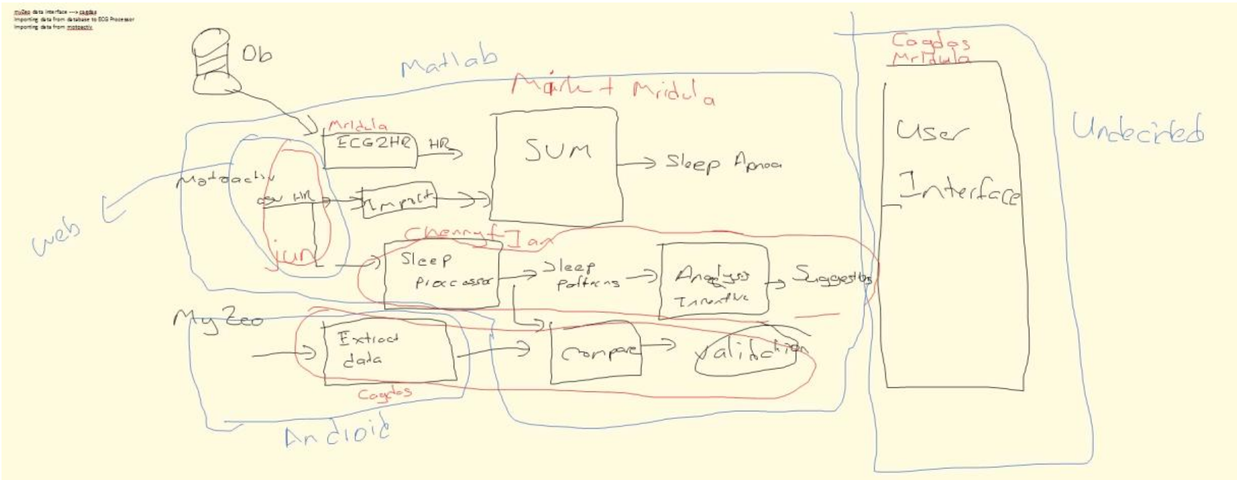
REQ-11	2	The administrator should be able to retire a user account.
REQ-12	2	The system shall be able to divide user account into high-level and low-level. And high-level user could have access to check the sleep quality of low-level users.

2.3 Non-Function Requirements Table

ID	Priority Weight	Requirement
REQ-13	2	The system should be easy to use for all users.
REQ-14	2	The interface of the system should be friendly to user.
REQ-15	4	Each user must have a separate login
REQ-16	4	The privacy of the user is of utmost importance
REQ-17	5	User shall be not allowed to modify any data in the database.
REQ-18	5	User shall be not allowed to access the accounts of other users'.
REQ-19	4	The system should be maintained at a appropriate frequency.

2.4 On-Screen Appearance Requirements

We intend to create a secure user interface for every user to access his or her report. Given the fact that a central server can be used, the project is easily scalable. We need to gather more clarity on our user interface as we are concentrating on the actual data acquisition and classification of the project as of now.



ID	Priority Weight	Requirement
REQ-1	3	Log-In page
REQ-2	3	Main Activity page after successful log-in that displays options to the user: Hear Rate Manager, Sleep Pattern Manager and Suggestion Manager
REQ-3	5	A Heart Rate Manager screen that Allows the user to automatically download data from MOTOACTV and choose the session of interest. Once chosen the, screen displays the heart rate graph.
REQ-4	5	A Sleep Pattern Manager screen that allows the user to access My Zeo data and graphical representation of the Sleep Pattern
REQ-5	5	A Suggestions screen that displays the sleep score based on the sessions chosen by the user and gives suggestions
REQ-6	4	On the Heart Rate Manager Screen, if sleep apnea is detected, a questionnaire that is used to calculate the sleep apnea score

3. FUNCTIONAL REQUIREMENTS SPECIFICATION

As a high quality sleeping is very important to everyone, people should consider about the quality of sleeping. Thus, this Sleeping Quality Assessment is suitable for any people. In other words, many end-users can be explored.

3.1 Stakeholders

This software is a health monitor that can be used by all the people. Everybody wants to have a good sleep at night. Thus, many people are the aim user for this Sleeping Quality Assessment. Here are some ideal stakeholders for our software.

- First, the people who have the disease of insomnia are the primary aim for this application. These people may be workers or students who suffer from the high working or studying pressure thus can’t sleep well or simply adults and kids who experiencing the insomnia or some similar diseases which destroy the sleep hours. This category includes all age groups from children to adult who are suffering the insomnia and need to improve their sleeping quality.
- Second, the people who have not insomnia, but they want to record their sleeping quality and improve it.
- Third, the set of end-users is the people who would like to keep an eye on other people’s sleep quality. They may be parents who want to supervise the sleep of their naughty kids or sons and daughters who pay attention to the sleep quality of their aged parents.
- The other possible users may include the athletes who need a good sleeping to keep their body in a good condition. Or the coaches and the experts who give advices to other people for gaining good sleep.

3.2 Actors and Goals

- **Low-level user**: a registered low-level user of the system;
- **High-level user**: a registered high-level user of the system;
- **Visitor**: an unregistered user;
- **Database**: records of all the sleep data;
- **Personal website**: the place where player can post their sleep quality;
- **Administrator**: a special user of the system who have top priority access to the system database.

3.3 Use Case Descriptions

3.3.1 Casual Description

Use Case	Name	Description	Requirements
UC-1	Import_heart_rate data_from Motoactv_website	Allows all the users download and sto store their heart rate data in Database.	REQ-1,REQ-2,REQ-7
UC-2	Show_the_heart rate_data_from database_and _detect_sleep_apnea	Shows the heart rate data in a graph and simultaneously detects the presence of sleep apnea	REQ-6, REQ-7
UC-3	Create_sleep_pattern_from_heart rate	DIstinguishes sleep in stages of REM, NREM, AWAKE, LIGHT and generates patterns from raw heart rate data	REQ-8, REQ-9
UC-4	Import_sleep_pattern_from_My Zeo_files	Downloads and displays	REQ-4

		My zeo data	
UC-5	Show_sleep_patterns_and_statistics_from_db	Displays the sleep patterns from the database based on the choice of the user	REQ-4,8
UC-6	Create_suggestions_from_sleep_patterns	Calculates sleep score and gives suggestions to the user	REQ-5
UC-7	Show_sleep_scores_graph	Displays a graph of the chose sleep scores	REQ-10
UC-8	Get_Sleep_Suggestions	Gets the sleep suggestions	REQ-5
UC-9	AddUserAccount	Allow the Administrator to add a registered user's account.	REQ-11
UC-10	DeleteUserAccount	Allow the Administrator to retire a user account.	REQ-11
UC-11	HighLevelUserPrevileges	Allows a high level user to get data from other users and simulate a case scenario	REQ-12

3.3.2 Fully-Dressed Description

UC-1 Import heart rate data from Motoactv website

Initiating Actor: Low-level and high-level users

Actor's Goal: To store/access the original data

Participating Actors: Database, System, MOTOACTV Website,

Pre-condition: The system/screen displays a download button and a combobox with already present sleep sessions in the database

Post-condition: The original sleep data, i.e, the heart rate is downloaded from MOTOACTV website and stored in the database. A combobox of all the downloaded data is presented to the user and the user chooses the session of interest.

Flow of events:

→ User clicks download from motoactv button,

← System downloads data from motoactv website

→ Uploads to database

← Repopulates heartrates sessions combobox

UC-2 Show the heart rate data from database and detect sleep apnea

Initiating Actor: Low-level and high-level users

Participating Actor: System, Database, Sleep Apnea Manager

Actors Goals: To see heart rate of the chosen session and answer's questionnaire to obtain sleep apnea index

Pre-condition: UC-1 has been executed.

Post-condition: The heart rate graph is displayed and the sleep apnea index is calculated

Flow of events:

→User clicks a heart rate session from combobox

←System retrieves heart rate from database(db)

→System checks for sleep apnea

←Shows heart rate graph and sleep apnea questionnaire(if exist) in the screen

UC-3 Create sleep pattern from heart rate

Initiating Actor: Low level and high level users

Participating Actor: Sleep Pattern Manager, Database
Actors Goals: To see the sleep patterns

Pre-condition: UC-2 has been already executed

Post-condition: The sleep patterns of the user are displayed and saved in the database
Flow of events:

→User clicks sleep patterns button

←System creates sleep patterns from heart rate

→Checks whether sleep-pattern session already exist in db. If not, saves pattern in db

←Displays sleep patterns

UC-4 Import sleep pattern from My Zeo files

Initiating Actor: Low Level and High Level Users

Participating Actors: System, DataBase, MyZeo Application
Actors Goals: To see and access the sleep patterns from MyZeo

Pre-condition: Access to My Zeo

Post-condition: Display of sleep patterns

Flow of events:

→User clicks "download from myzeo

←System asks user for the input file

→User chooses file

←Displays data and System uploads sleep patterns to database if they don't already exist in db

UC-5 Show sleep patterns and statistics from db
Initiating Actor: Low Level User and High Level User

Particitpating Actors: Database, System

Actors Goals: To obtain sleep patterns and statistics

Pre-Condition: Sleep pattern sessions are already in the database

Post-Condition: User obtains statistics

Flow of events:

→User chooses a sleep pattern session from combobox

→System retrieves sleep pattern from database(db)

→System calculates statistics

←Shows sleep patterns and statistics in the screen

UC-6 Create suggestions from sleep patterns

Initiating Actor: Low Level User and High Level User

Participating Actors: Database, System

Actors Goals: Obtain suggestions

Pre-Condition:UC-5 has been completed

Post-Condition: The sleep score and suggestions are calculated and stored in the database

Flow of events:

→User clicks suggestions button

→System calculates sleep score and suggestion from sleep pattern

→Checks whether suggestion session already exist in db.

←If not, saves suggestion and score in db

UC-7 Show sleep scores graph

Initiating Actor: User

Participating Actors: System, User, Database

Actors Goals: To obtain sleep scores graph

Pre-Condition: UC-6 is executed

Post-Condition: The sleep scores and suggestions are displayed

Flow of events:

→User opens sleep suggestion screen

← System retrieves all sleep scores from database

←Shows a graph of sleep-score history

UC-8 Get Sleep Suggestions

Initiating Actor: User

Particitpating Actors: Database

Actors Goals: To see the suggestions based on the scores

Pre-Condition: UC-6 is executed

Post-Condition: The user gets his sleep suggestions
Flow of events:

→User chooses a sleep pattern session from combo-box

←System retrieves Sleep suggestion from database and shows in UI

3.4 User Case Diagrams

The use case diagram is shown in figure 3-1. Low-level and high level User, Visitor and Administrator <<initiate>> all user cases, except for UC-3 (GetSleepPattern), UC-4 (REMandNON-REM), which are <<extend>> from UC-2 (GetSleepQuality) as sub-use-cases. Database store account information,monitoring data and relationship between low-level and high-level users, so it’s <<participate>> in all use cases.

Figure 3-1 Use Case Diagram

3.5 Traceability Matrix

Use cases are designed to meet the system requirements, the traceability matrix in Table 3-1 shows the mapping relation between system requirements and use cases of this software.

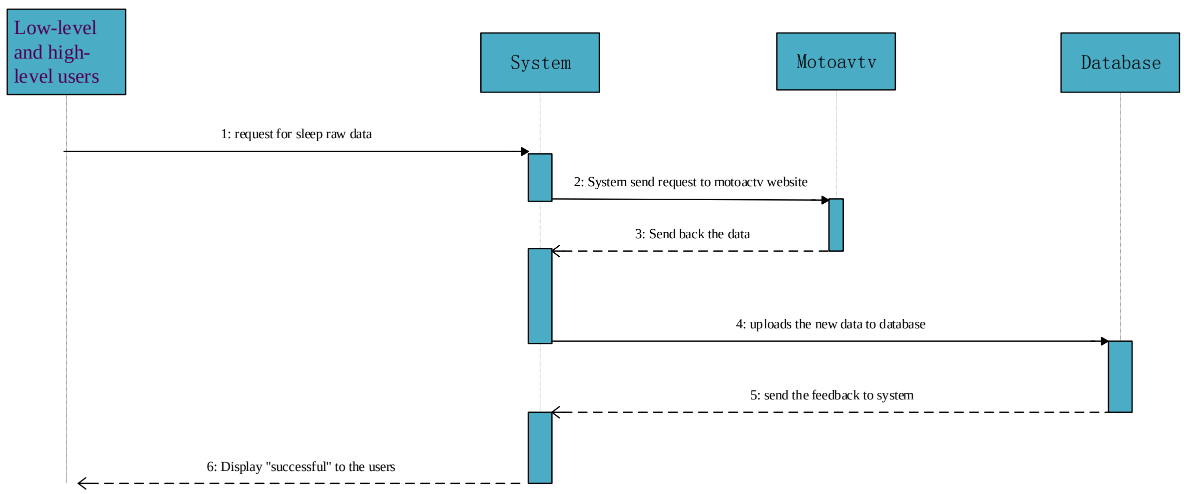
	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12
REQ-1	X											
REQ-2	X											
REQ-3		X										
REQ-4			X									x
REQ-5					X							
REQ-6				X							X	
REQ-7	X											
REQ-9		X	X	X								
REQ-10									X			
REQ-11										X		
REQ-12						X	X	X				

Table 3-1 Traceability Matrix

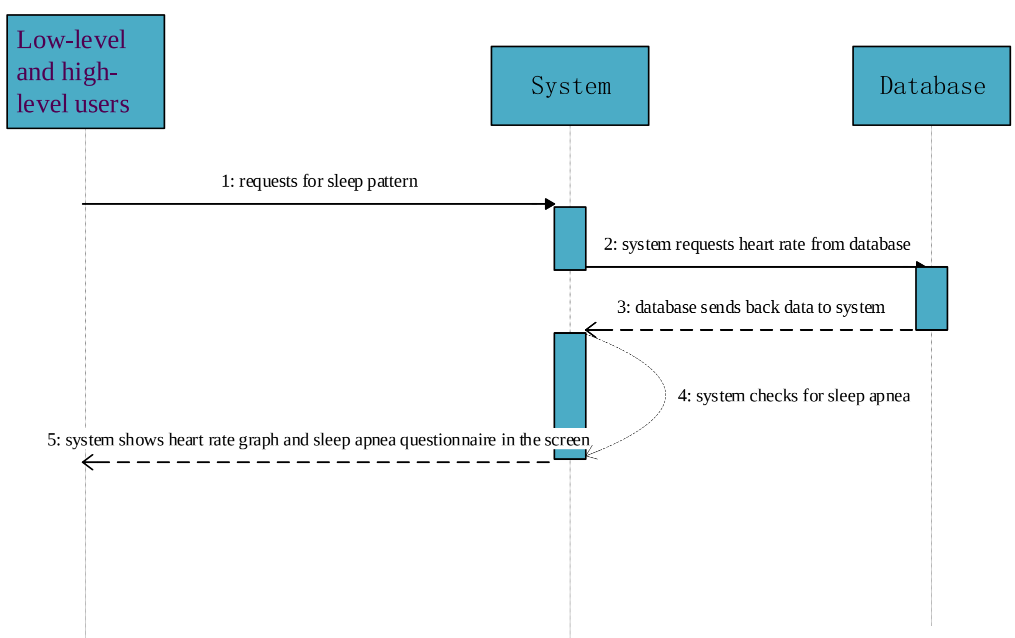
3.6 System Sequence Diagram

This section is the system sequence diagrams for the four important use cases described above.

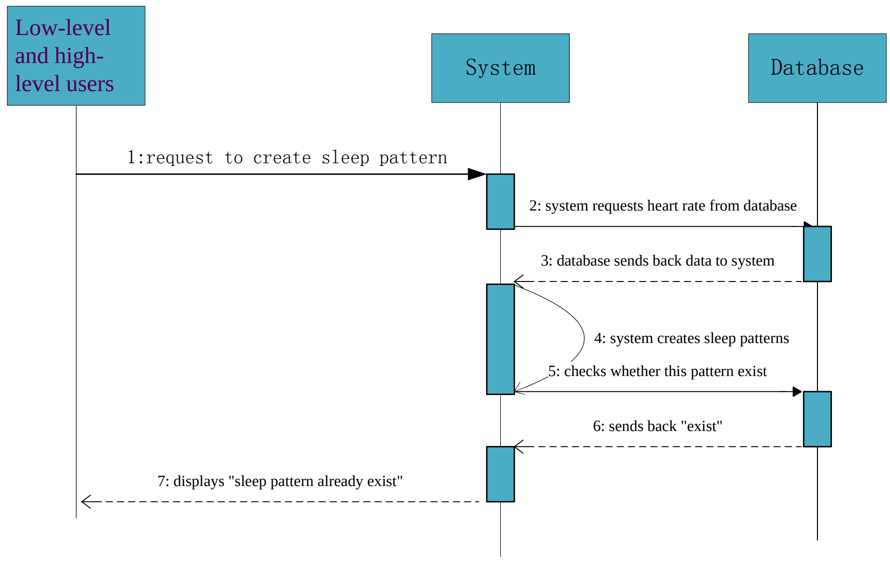
System Sequence Diagram for UC-1

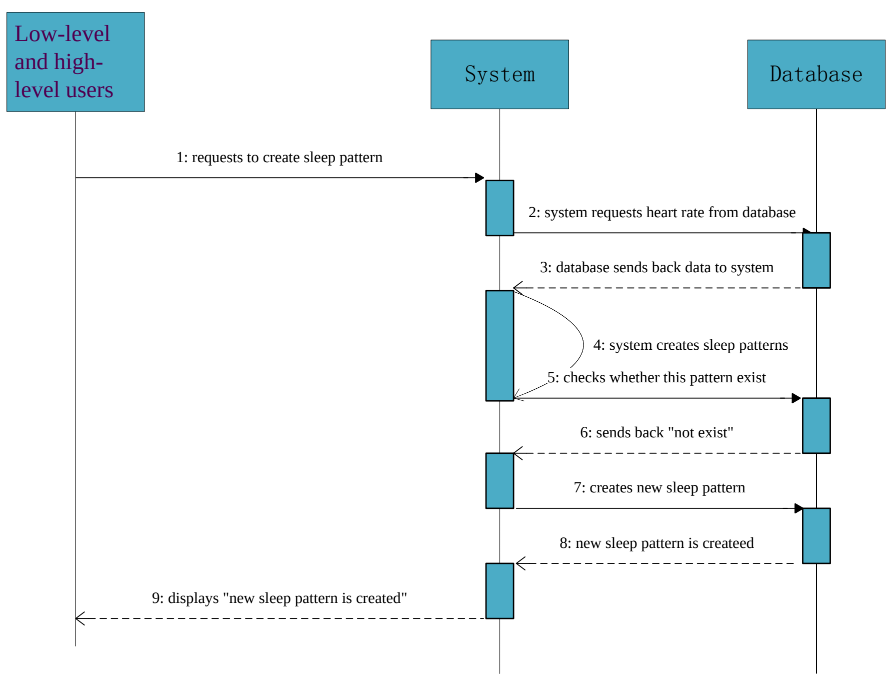


System Sequence Diagram for UC-2

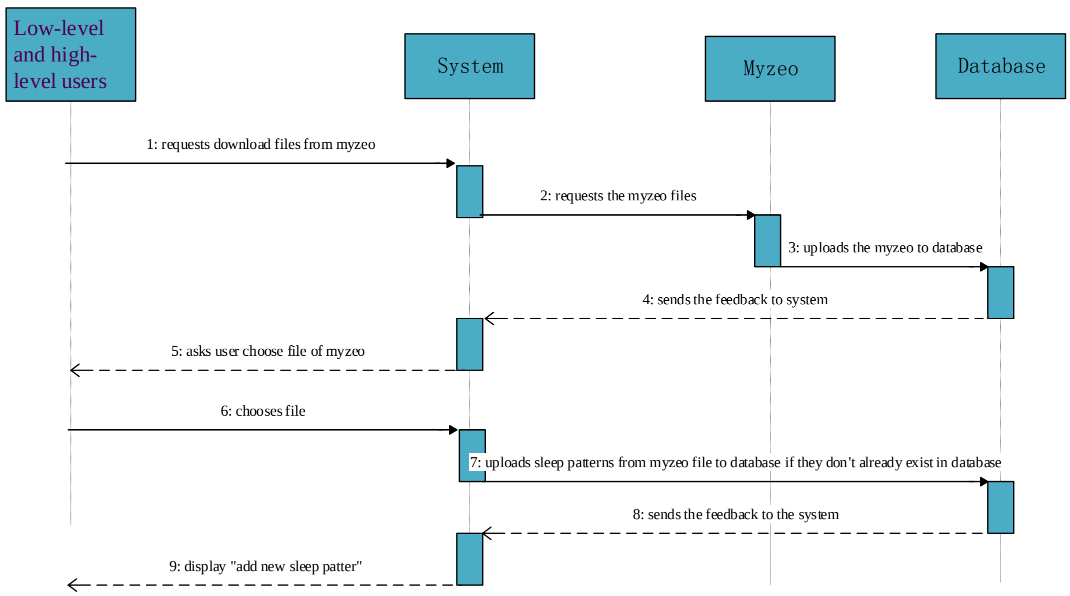


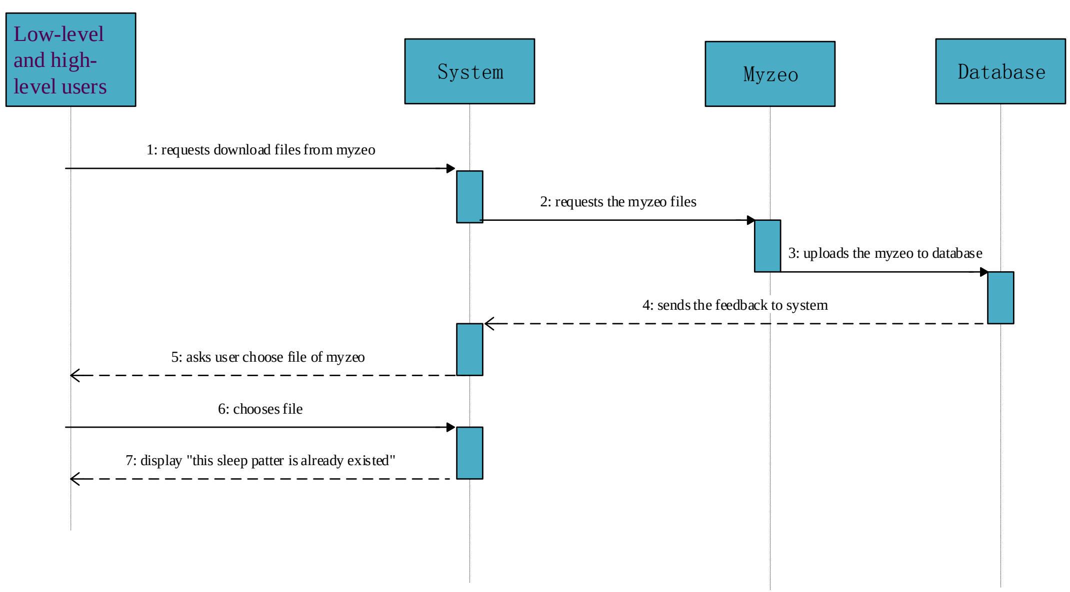
System Sequence Diagram for UC-3



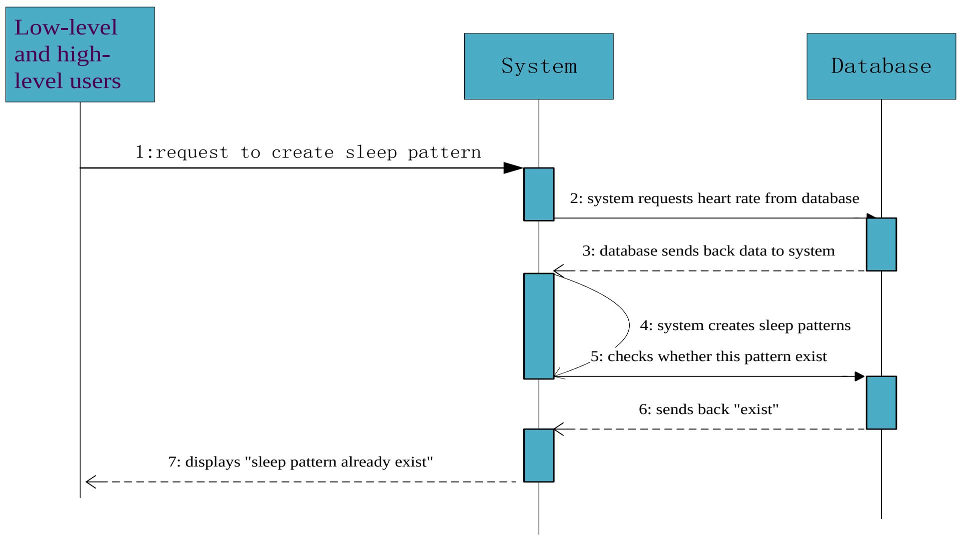


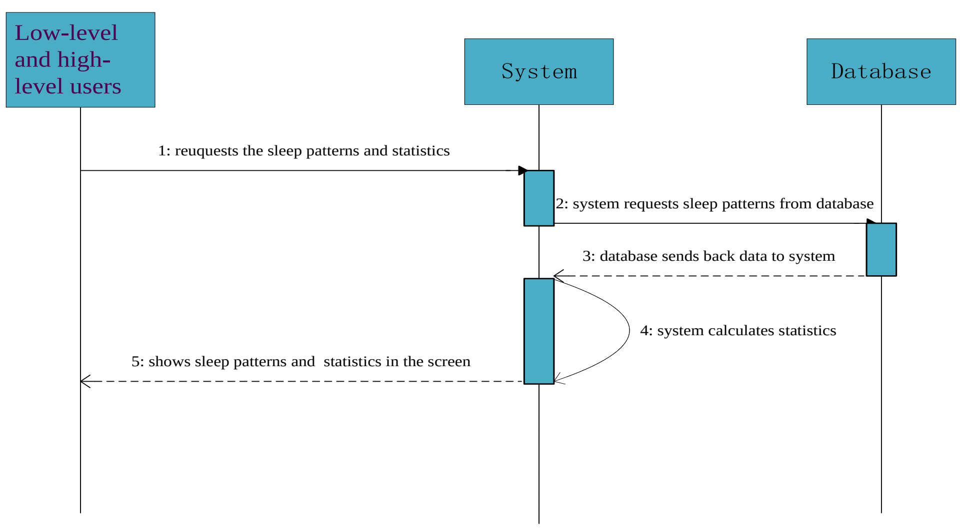
System Sequence Diagram for UC-4



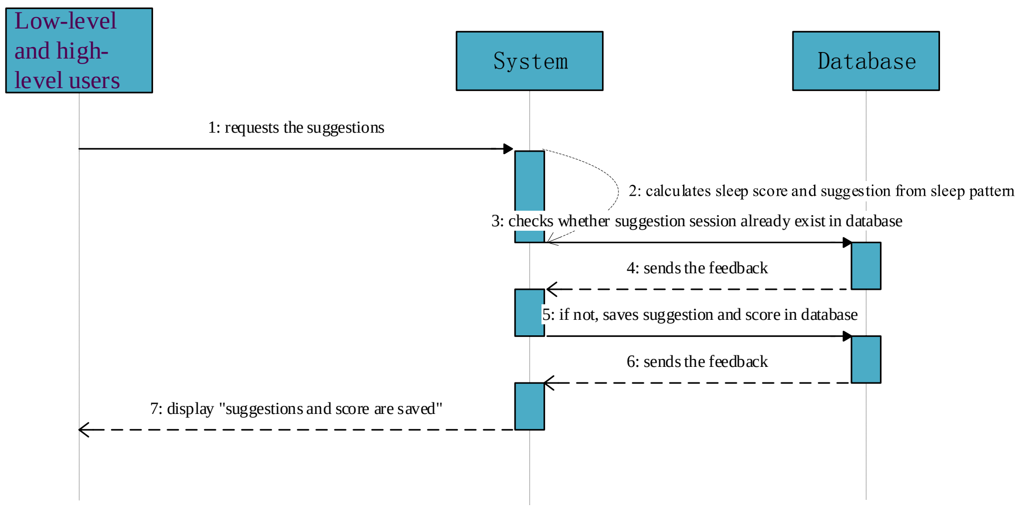


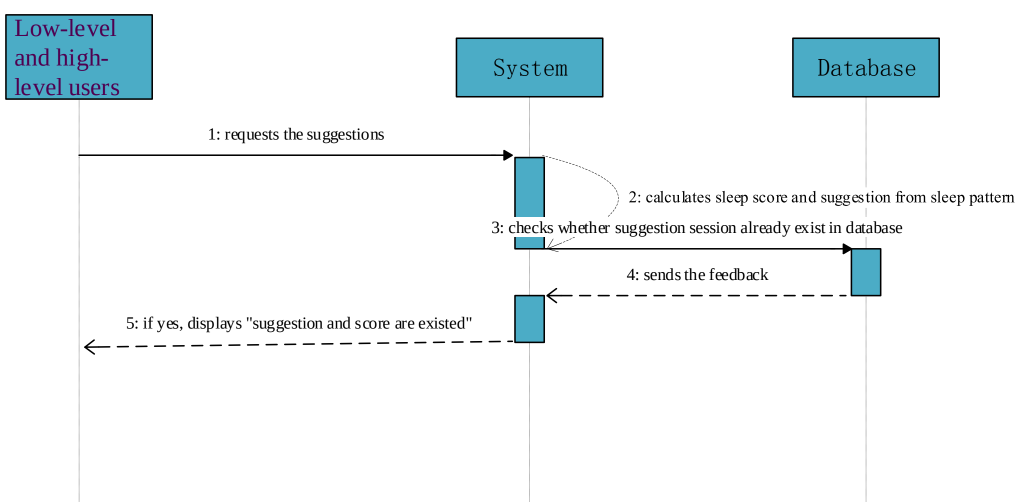
System Sequence Diagram for UC-5



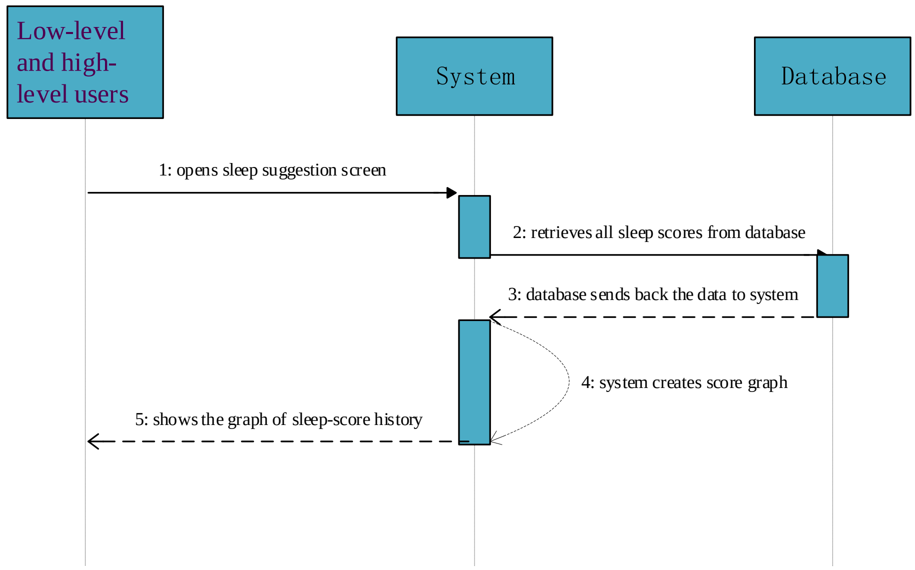


System Sequence Diagram for UC-6

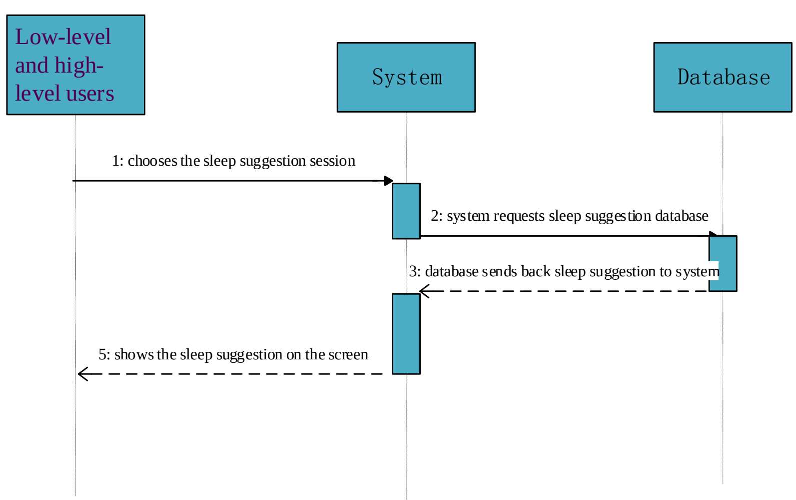




SYSTEM SEQUENCE DIAGRAM FOR UC-7



SYSTEM SEQUENCE DIAGRAM FOR UC-8



4.0 EFFORT ESTIMATION USING USE CASE POINTS

Here we use the following equation to calculate the Use Case Points (UCP).

UCP = UUCP* TCF *ECF

Where Unadjusted Use Case Points (UUCPs) are computed as a sum of these two components:

1. The Unadjusted Actor Weight (UAW), based on the combined complexity of all the actors in all the use cases.

2. The Unadjusted Use Case Weight (UUCW), based on the total number of activities (or steps) contained in all the use case scenarios.

The table below shows the Unadjusted Actor Weight (UAW).

Unadjusted Actor Weight (UAW)

Table: Actor classification and associated weights.

Actor name	Complexity	Weight
Users	Complex	3
Motoactv	Average	2
database	Average	2
MyZeo Application	Simple	1

UAW = 3+2+2+1= 8.

The UUCW is derived from the number of use cases in three categories: simple, average, and complex. We use the following table to record the classification of each user case.

Table: Use case classification and its weight

Use Case	Name	Description	Category	Weight
UC-1	Import_heart_rate data_from Motoactv_website	Allows all the users download and sto store their heart rate data in Database.	Average	10
UC-2	Show_the_heart rate_data_from database_and _detect_sleep_apnea	Shows the heart rate data in a graph and simultaneously detects the presence of sleep apnea	Complex	15
UC-3	Create_sleep_pattern_from_heart rate	DIstinguishes sleep in stages of REM, NREM, AWAKE, LIGHT and generates patterns from raw heart rate data	Complex	15
UC-4	Import_sleep_pattern_from_My Zeo_files	Downloads and displays My zeo data	Simple	5
UC-5	Show_sleep_patterns_and_statistics_f rom_db	Displays the sleep patterns from the database based on the choice of the user	Average	10
UC-6	Create_suggestions_from_sleep_patt erns	Calculates sleep score and gives suggestions to the user	Complex	15
UC-7	Show_sleep_scores_graph	Displays a graph of the chose sleep scores	Simple	5
UC-8	Get_Sleep_Suggestions	Gets the sleep suggestions	Average	10

UC-9	AddUserAccount	Allow the Administrator to add a registered user’s account.	Simple	5
UC-10	DeleteUserAccount	Allow the Administrator to retire a user account.	Simple	5
UC-11	HighLevelUserPrevileges	Allows a high level user to get data from other users and simulate a case scenario	Average	10

Based on the table above, we can get UUCW = 105. Therefore, UUCP = 8+105 =113.

Technical Complexity Factor (TCF)—Nonfunctional Requirements

Table: Technical complexity factors and their weights.

Technical factor	Description	Weight	Perceived Complexity	Calculated Factor
T1	The interface of the system should be friendly to user.	1	2	2
T2	Internal processing is relatively complex	2	3	6
T3	Users expect good performance but nothing exceptional	1	3	3
T4	Security is a	1	5	5

	significant concern			
T5	Easy to change minimally required	1	1	1
T6	No direct access for third parties	1	0	0
T7	Ease of use is very important	0.5	5	2.5
Technical Factor Total (TFT)				19.5

$TCF = C1 + C2 \times TFT$

Where C1=0.6, C2=0.01, TCF=0.6+0.01*19.5=0.795

Environment Complexity Factor (ECF)

The environmental factors measure the experience level of the people on the project and the stability of the project. For detail, please check the below table.

Table : Environmental complexity factors and their weights.

Environmental Facotr	Description	Weight	Perceived Impact	Calculated Factor
E1	Beginner familiarity with the UMLbased development	1.5	1	1.5
E2	Some familiarity with application problem	0.5	2	1
E3	Some knowledge of object-oriented approach	1	2	2
E4	Beginner lead	0.5	1	0.5

	analyst			
E5	Highly motivated	1	3	3
E6	Stable requirements expected	2	5	10
E7	Programming language of average difficulty will be used	-1	3	-3
Environmental Factor Total:				15

Here we use the following formula to calculate ECF,

ECF = Constant-1 +Constant-2 *Environmental Factor Total

Where Constant-1 (C1) = 1.4,Constant-2 (C2) = 0.03. Given these data, the ECF = 1.4+(-0.03*15)=0.95.

In the end, the final UCP = 113*0.795*0.95=85.3.

Assume that productivity factor is 28 hours per user case point. The effort estimation would be 2380.

5.0 DOMAIN MODEL

5.1 Concept Definitions

To analyse the domain model, we first derive domain model concepts and corresponding respo nsibilities from the formerly defined system use cases. Table 5-1 lists all the domain model concepts and corresponding responsibilities.

Table 5-1

Responsibility	Type	Concept	UseCase
Load Monitoring Data from Monitoring Device	D	MotoActiv Manager	UC-1
Load Monitoring Data from MyZeo Device	D	MyZeo Manager	UC-4
Store Monitoring Data into the DataBase	D	Database Manager	UC-1,4
Load Monitoring Data from Database to Sleep Pattern Manager, Sleep Apnea Manager	D	Data Loading Manager	UC-2,3
Load Monitoring Data from Database and get Sleep Pattern analysis result	D	Sleep Pattern Manager	UC 3,4, 5
Calculate the score of sleeping and asses quality of sleep	D	Sleep Quality Manager	UC 6,7,8
Find if patient suffers from sleep apnea using SVM and	D	Sleep Apnea Manager	UC 2

MIT-ECG database			
Login of User	K	Login Manager	All
Request Date for Data	K	Date Manager	All
Start Recording Data	D	Data Logging Manager	UC 1,2
Display the report in user interface	D	Report Manager	UC 8
Generate Report of quality of sleep with suggestions using output of Sleep Apnea Manager ,Sleep Pattern Manager and Sleep Quality Manager and forward the report generated to the Report Manager	D	Sleep Diagnosis Manager	UC 8
Manage User Permissions	K	User Account Manager	UC 9,10,11
Control the process of loading data and transferring report, etc	D	Controller	All
Provides the user with a clear and concise way of uploading sleep raw data and viewing analysis results and suggestions	D	GUI	All

5.2 Association Definitions

Some of the concepts defined above as domain concepts have to work in certain pattern to finis h some target, Table 5-2 gives the corresponding association definitions based on the defined d omain concepts

Table 5-2

Concept Pair	Association Description	Category
User->Login Manager	Allows User to Login and choose whether to see previous report or start data logging	User Interface
User->Data Logging Manager	All Login Operations are handled by the Data Logging Manager. This includes log in, request of former data, logging in of heart rate data from MOTOACTV, passing on the generated reports to the user	User Interface
Data Logging Manager-> Data Acquisition	When the start recording data button is pressed by the user, the Login Manager, generates an interrupt signal to the MotoActv Manager. The MotoActv Manager collects data from MotoACTV device and forwards it to the Data Base Manager. It also sends a beacon to the Sleep Pattern Manager and Sleep Apnea Diagnosis Manager	Data Read and Save
MotoActv Manager -> Database Manager	The recorded raw data readings are given a time stamp and forwarded from MotoActv Manager to the database Manager. The Data Base Manager stores the recorded dad	Data Read and Save
Database Manager ->Sleep Pattern Manager	Sleep pattern Manager, upon receiving the beacon signal from MotoActv Manager, queries the raw data from the database and then does its job	Query Data and process

(SPM)		
Database Manager ->Sleep Apnea Manager(SPM)	Sleep Apnea Manager, upon receiving the beacon signal from MotoActv Manager, queries the raw data from the database and then does its job	Query Data and Process
SAM and SPM ->Sleep Diagnosis Manager	Use results of Sleep Pattern Manager and Sleep Apnea Manager to create a report and forward to Report Manager	Aggregation of results

5.3 Attribute Definitions

Table 5-3

Concepts	Attributes	Attribute Definition
Login Manager	User Interface	Store user input to the system, or show analyzed result to user.
Data Logging Manager		
Report Manager		
Date Manager		
MotoActv Manager	Data Read and Store	Read in and store data from the monitoring device or database
Data Loading Manager		
Database Manager		
Sleep Pattern Manager	Data Analysis	Analyze Data
Sleep Quality Manager		

Sleep Apnea Manager		
Sleep Diagnosis Manager		
User Account Manager	Permission Management	Manage User Previleges
Controller	Control process	Control the processes of loading data and transferring report

5.4 Traceability Matrix

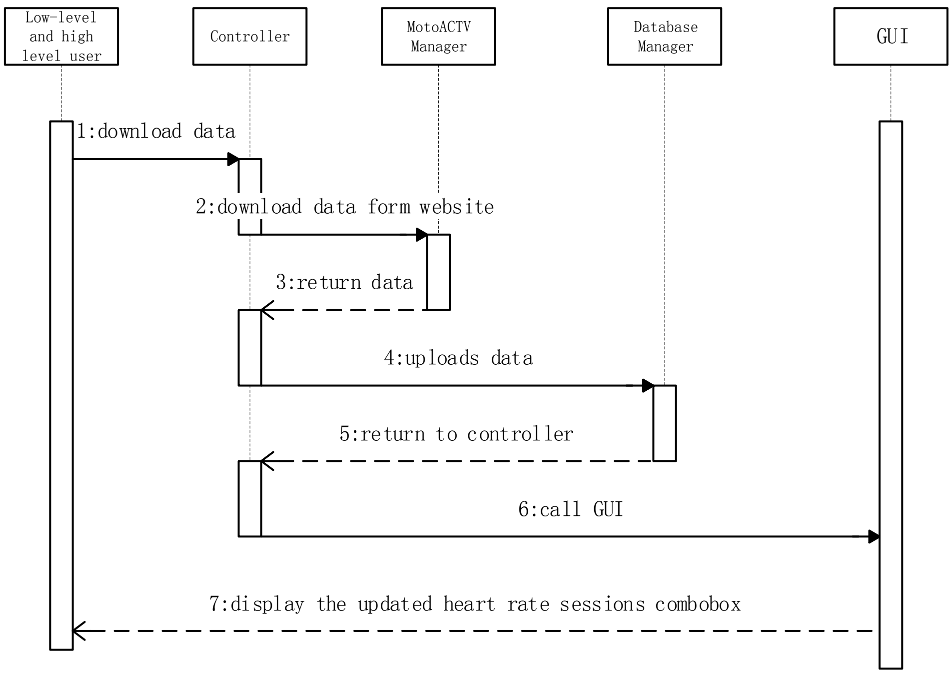
	MotoActv Manager	Database manager	Sleep pattern manager	Sleep apnea manager	Login manager	Date manager	Data logging manager	Report Manager	Sleep Diagnosis Manager	User Account Manager	Sleep Quality Manager
UC-1	x	X			x	X	X				
UC-2				X	x	X	X				
UC-3			X		x	X					
UC-4		X	X		x	X					
UC-5			X		x	X					
UC-6					x	X					X
UC-7					x	X					X
UC-8					x	X		X	X		X

UC-9					x	X				X	
UC-10					x	X				X	
UC-11					x	X					
UC-12					x	X					

6.0 INTERACTION DIAGRAMS

The following are the interaction diagrams corresponding to the use cases

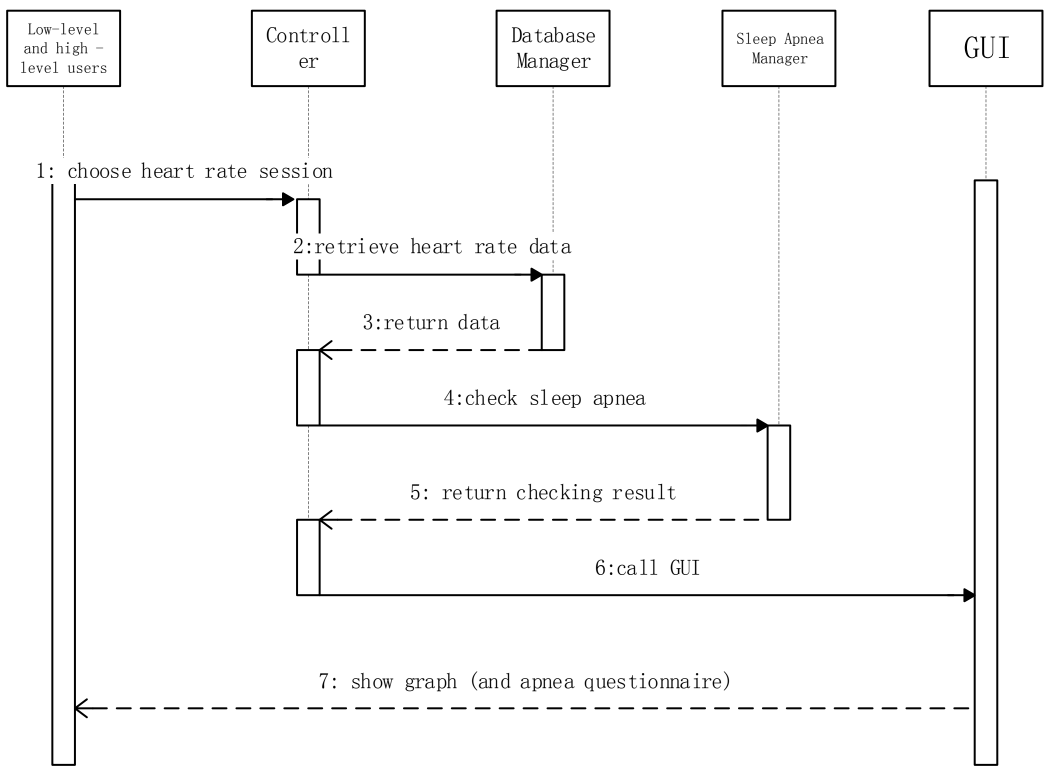
UC-1



Responsibilities Associated:

1) UC1 is mainly to collect the sleeping data from device, and download the sleeping data via GUI.
2) It can be used to make sure that the display can show the status of downloading the sleeping data.
3) MotoactvManager is responsible for loading monitoring data from MotoActv device.
4) DatabaseManager is responsiple for saving the sleeping data downloaded from Motoactv device.

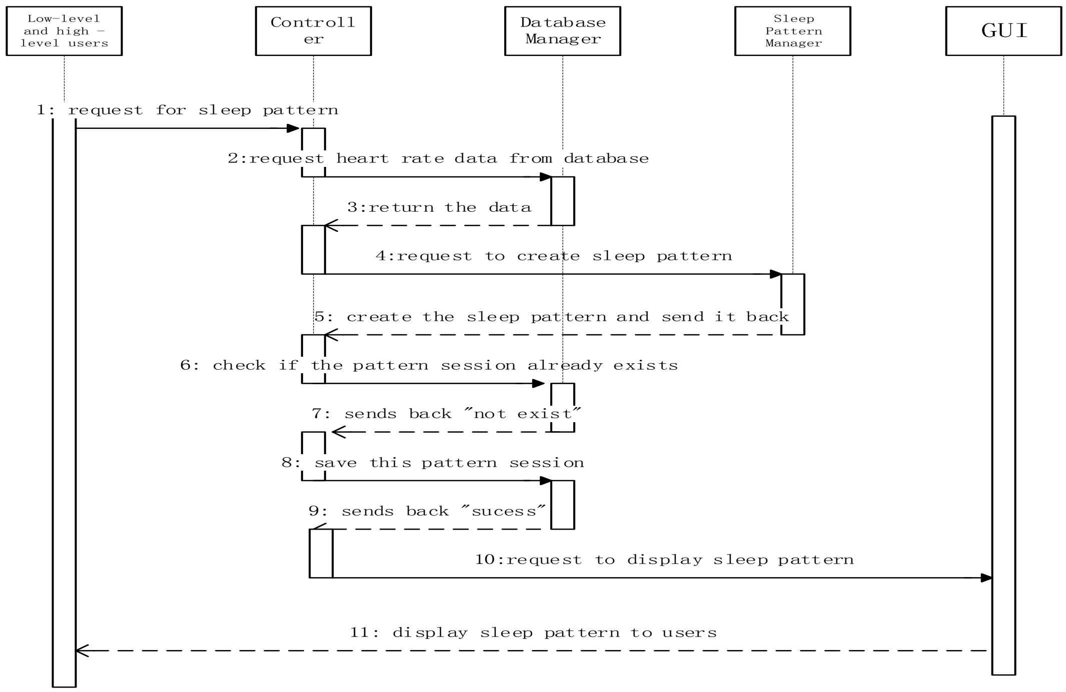
UC-2:

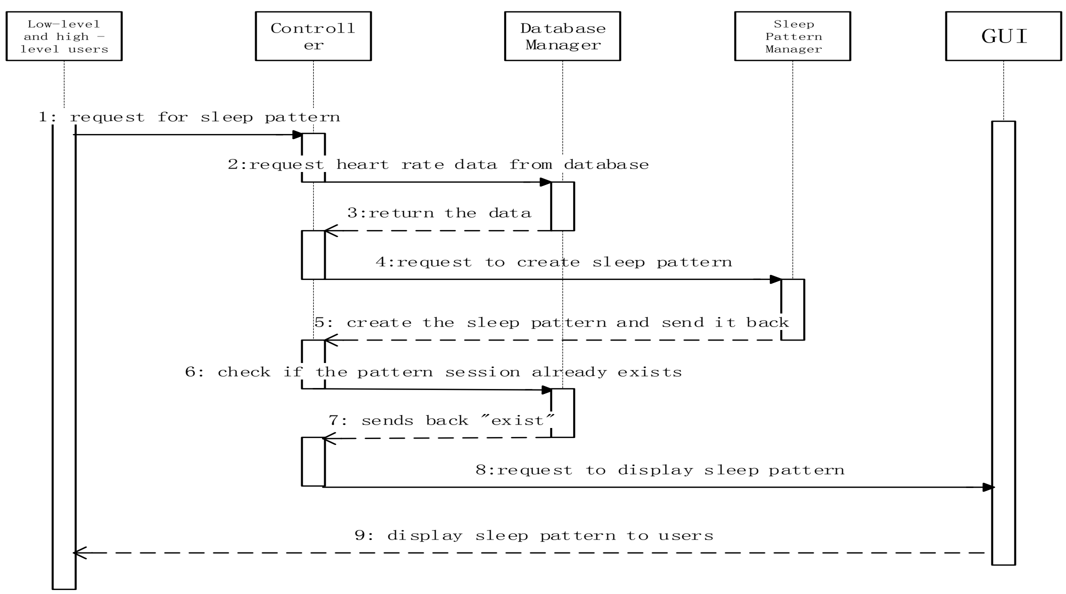


Responsibilities Associated:

1) UC-2 is mainly display the heart rate data and simultaneously check for Sleep Apnea
2) GUI is responsible for displaying the graph
3) SleepApneaManager is responsible for detecting sleep apnea and prompting the user with a questionnaire.

UC-3

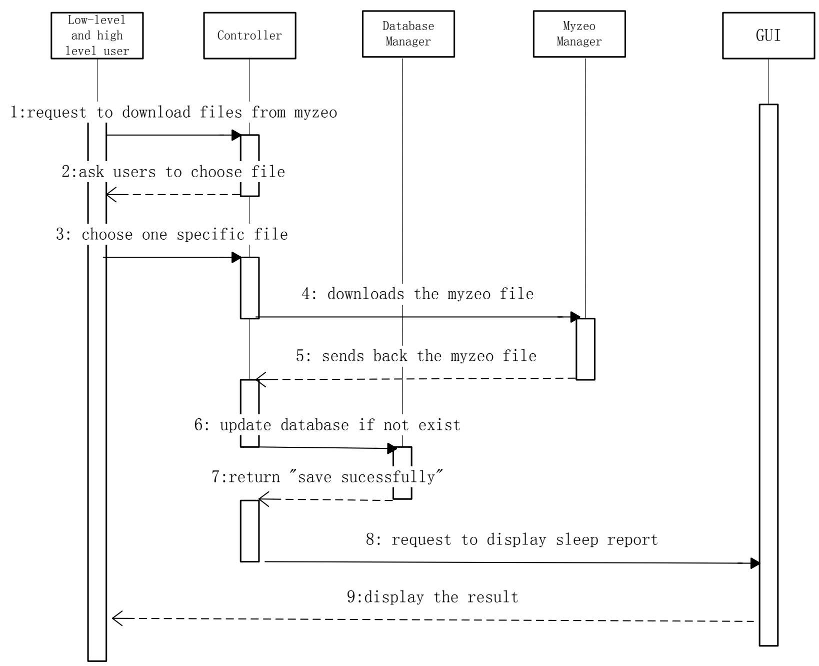


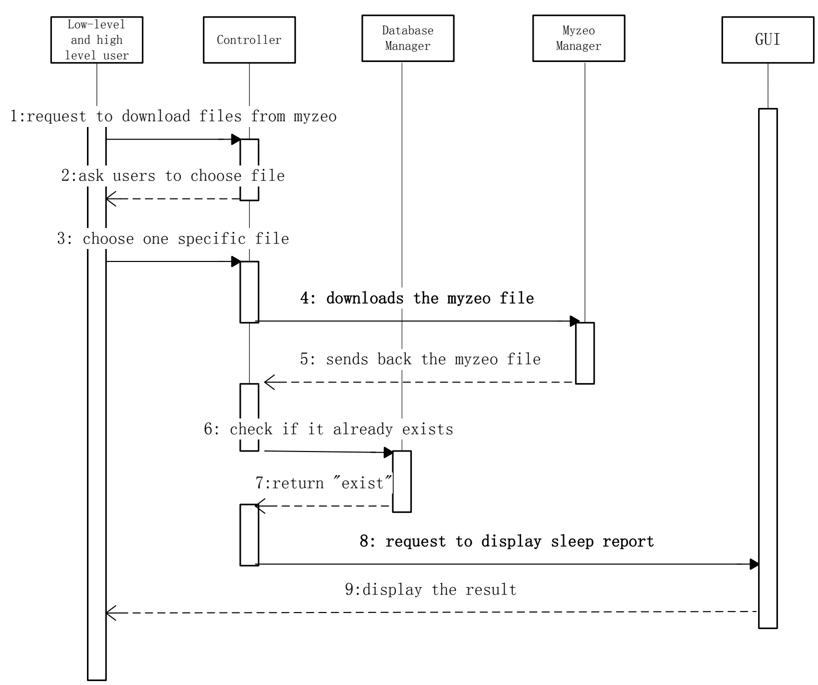


Responsibilities Associated:

The key responsibility of UC-3 is to generate sleep patterns from raw data and store in the data base if not already stored.

UC-4

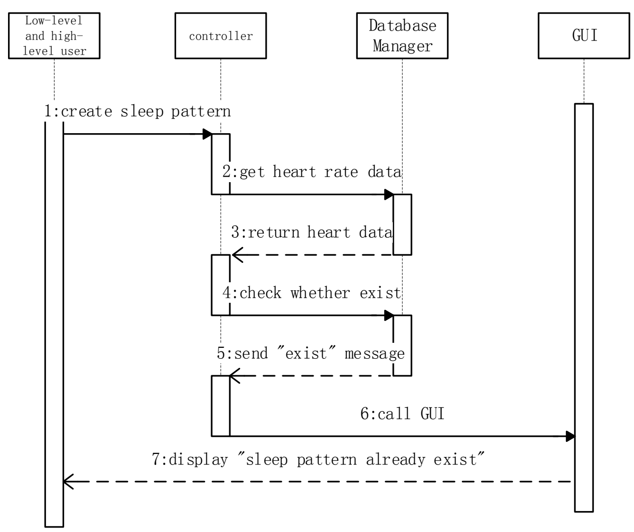

Low-level and high level user
Controller
Database Manager
Myzeo Manager
GUI
1:request to download files from myzeo
2:ask users to choose file
3: choose one specific file
4: downloads the myzeo file
5: sends back the myzeo file
6: update database if not exist
7:return "save sucessfully"
8: request to display sleep report
9:display the result

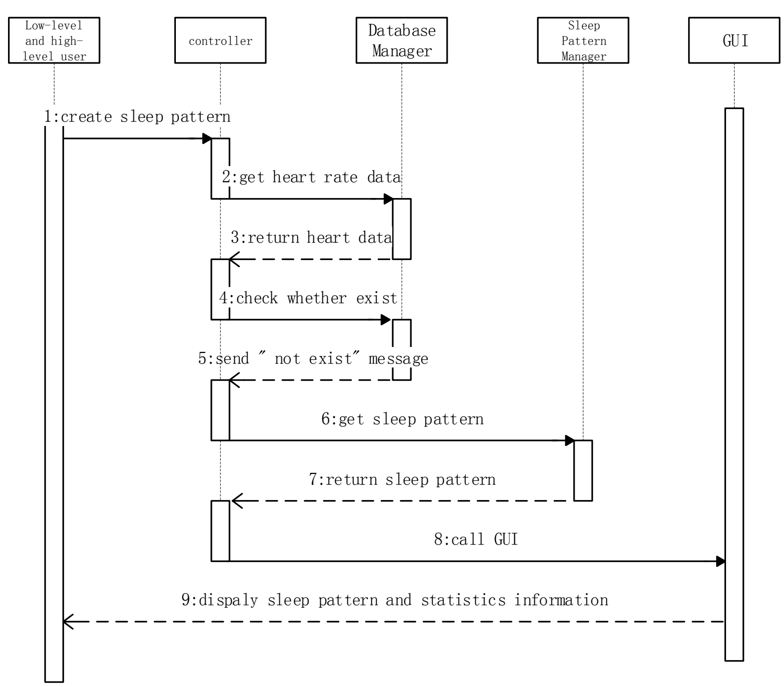

Low-level and high level user
Controller
Database Manager
Myzeo Manager
GUI
1:request to download files from myzeo
2:ask users to choose file
3: choose one specific file
4: downloads the myzeo file
5: sends back the myzeo file
6: check if it already exists
7:return "exist"
8: request to display sleep report
9:display the result

Responsibilities Associated:

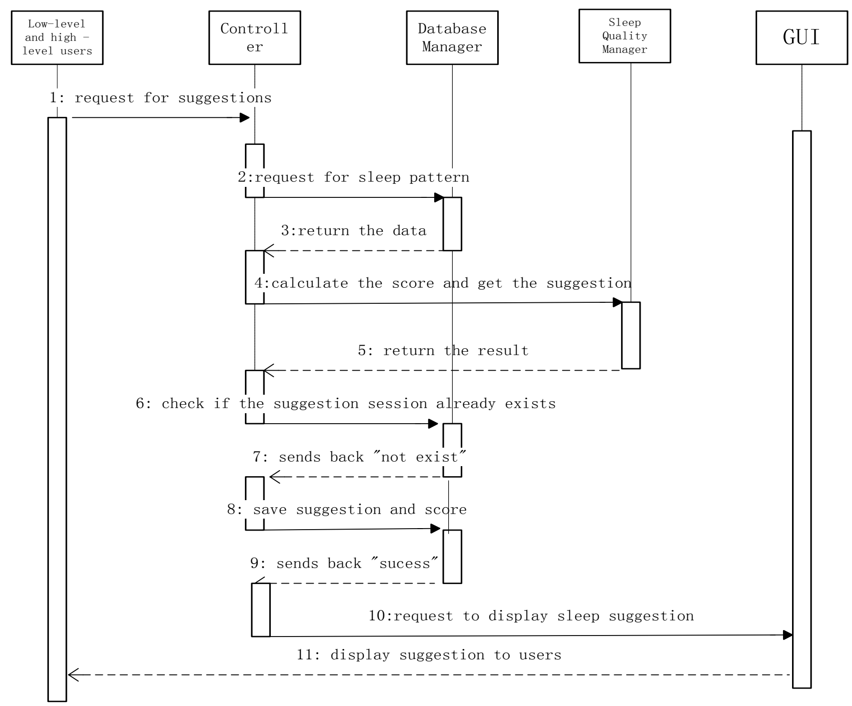
The main responsibilities of UC-4 are display the sleep patterns of my zeo's data and store in the database

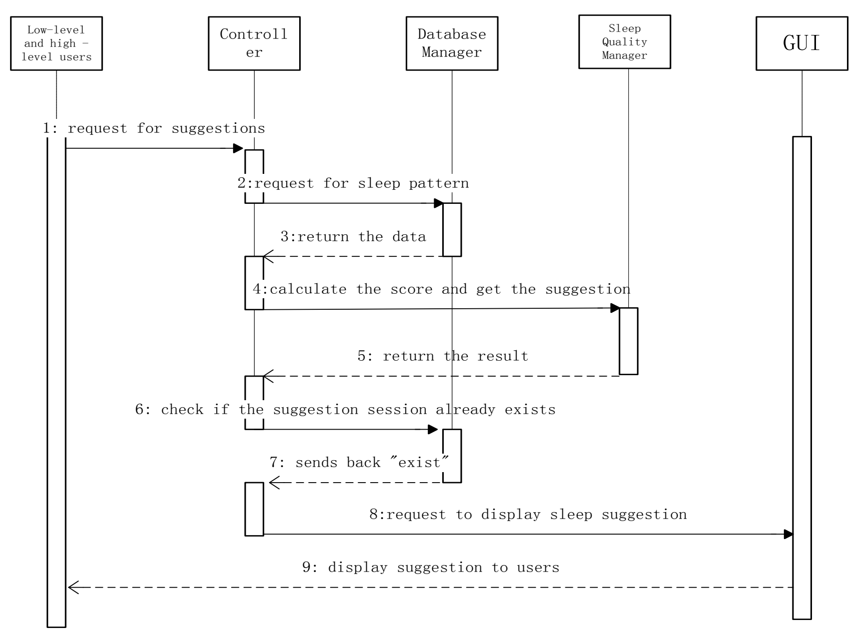
UC-5





UC-6:


Low-level and high - level users
Controll er
Database Manager
Sleep Quality Manager
GUI
1: request for suggestions
2:request for sleep pattern
3:return the data
4:calculate the score and get the suggestion
5: return the result
6: check if the suggestion session already exists
7: sends back "not exist"
8: save suggestion and score
9: sends back "sucess"
10:request to display sleep suggestion
11: display suggestion to users

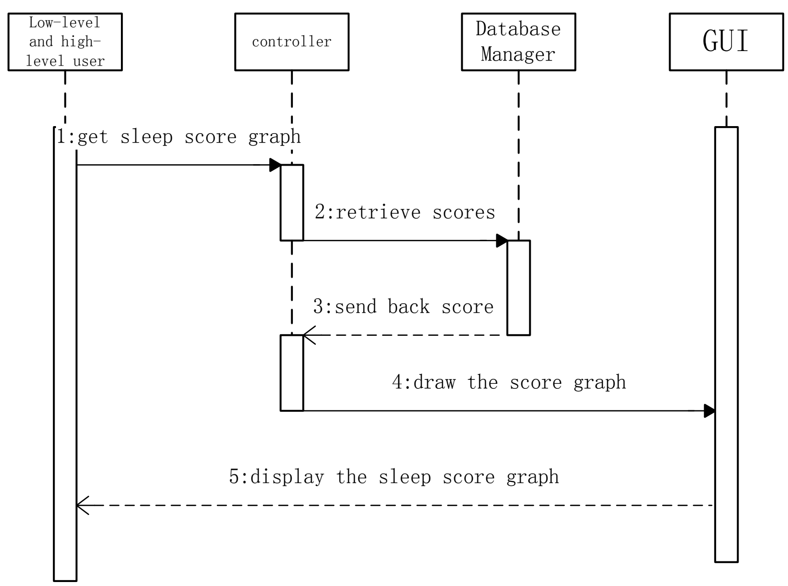

Low-level and high - level users
Controll er
Database Manager
Sleep Quality Manager
GUI
1: request for suggestions
2:request for sleep pattern
3:return the data
4:calculate the score and get the suggestion
5: return the result
6: check if the suggestion session already exists
7: sends back "exist"
8:request to display sleep suggestion
9: display suggestion to users

Responsibilities Associated:

1. The Create suggestions from sleep patterns allows will calculate the score and provide the suggestions to the users of their sleep, based on the collected sleep pattern in the database. If the suggestions and the score is not existed in the database, the system will save them in the database.

2. The Sleep Quality Manager plays the key role here. It will calculate the score based on the sleep pattern from database, do necessary analysis and provide the suggestion for the sleep quality.

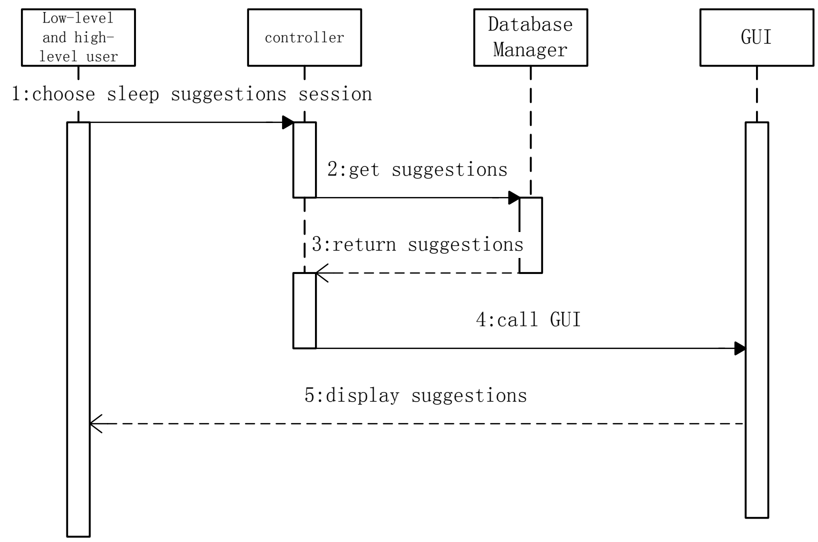
UC-7:



Responsibility associated:

1) This use case aims to give the user a picture presenting the fluctuation of his/her sleep scores during the last a few days.
2) The controller receive the initiating action from the user and then tries to retrieve the sleep scores from the database, after which it draws the graph and forward it to the GUI. Finally GUI displays the graph to user.

UC-8:



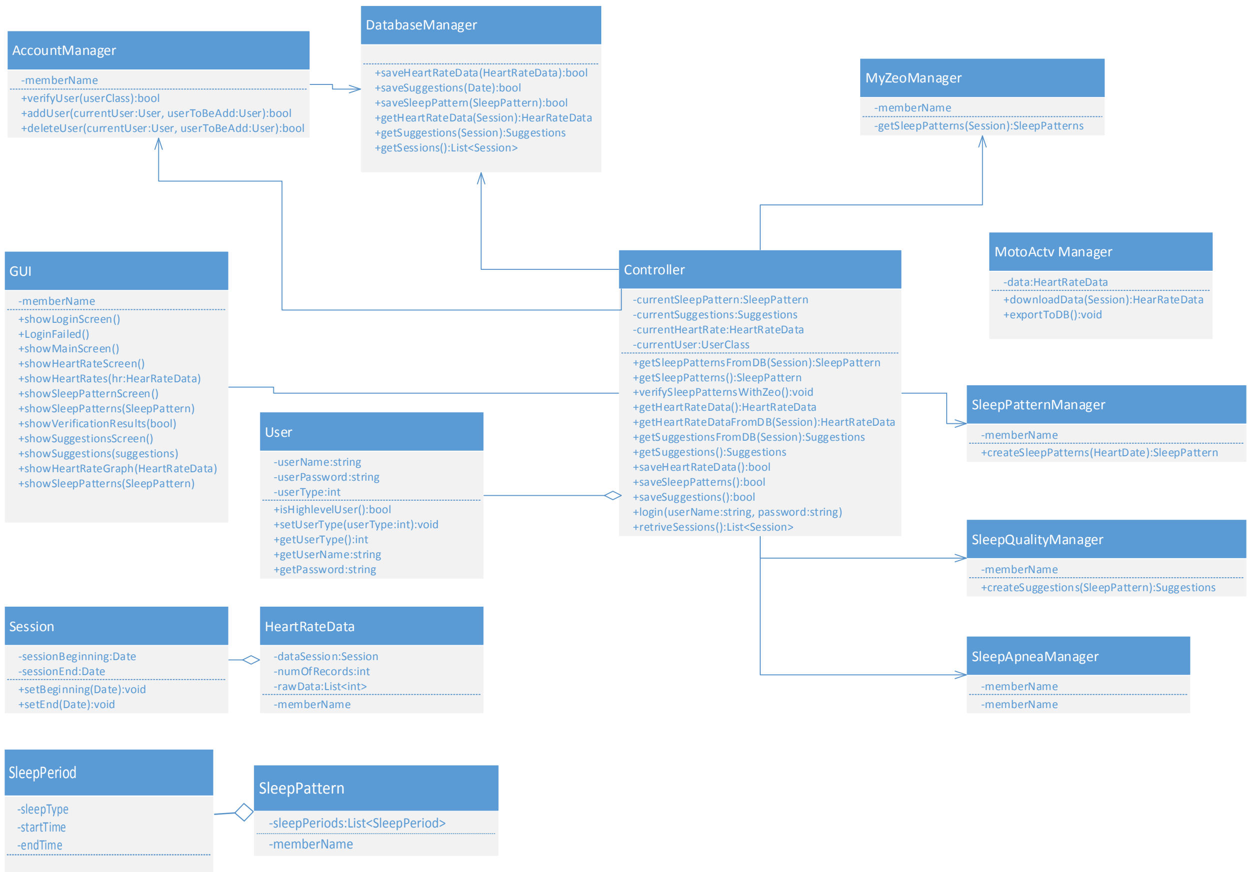
Responsibility associated:

1) This use case aims to give the user sleep suggestions based on the sleep scores his/her gets.
2) After the user choose the specific date, the controller retrieve the suggestions, which has been stored in the database after created. Finally, the controller will call the GUI to display these suggestions to user in plain text format.

6.2 Design Principles:

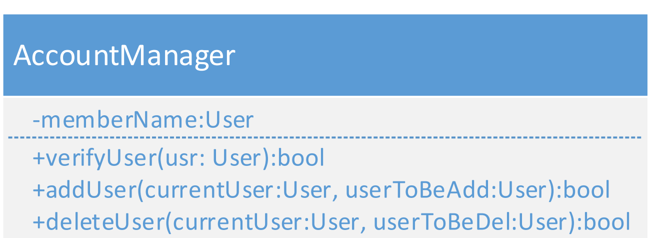
Generally, we apply the Expert Doer Principle to guide our use case design. This principle mainly state that an object who knows should do the task. For example, in the use case 2 “show the heart rate data from database and detect sleep apnea”, the Sleep Apnea Manager is exactly what we call an “expert”, so once retrieve data from database, we call the Sleep Apnea Manager to do all the analysis work. Other examples like UC-3, UC-7, UC-8, we make the Sleep Pattern Manager, Sleep Quality Manager the “expert” in the corresponding situations. The Sleep Pattern Manager is focusing on deriving sleep patterns from the raw data, whereas the Sleep Quality Manager is focusing on calculate the sleep quality scores and giving out the related suggestions.

7.1 Class Diagrams



7.2 Data Types and Operation Signatures

Account Manager



Attributes: Member Name of type User

Related Concepts: Controller, DataBase Manager

Operations:

+verifyUser, +addUser, +deleteUser

- Used within Login operation of Controller
- In the controller, Login Data from GUI is put in the form of User Data Type and compared with password and user name in database by Account Manager's operation +verifyUser
- If the result is true then the input user is made the current user (+addUser), else the flow of control is stopped in Account manager and given back to the Controller.
- Upon, successful login, then the Main Screen function is presented to the User
- When the User logs out, through the controller, the current user is deleted

Controller

Controller
-currentSleepPattern:SleepPattern -currentSuggestions:Suggestions -currentHeartRate:HeartRateData -currentUser:User
+getSleepPatternsFromDB(Session):SleepPattern +getSleepPatterns():SleepPattern +verifySleepPatternsWithZeo():void +getHeartRateData():HeartRateData +getHeartRateDataFromDB(Session):HeartRateData +getSuggestionsFromDB(Session):Suggestions +getSuggestions():Suggestions +saveHeartRateData():bool +saveSleepPatterns():bool +saveSuggestions():bool +login(userName:string, password:string) +retriveSessions():List<Session>

Attributes: CurrentSleepPattern, current Suggestions, current Heartrate, currentUser

Associated Concepts: GUI, User, Account Manager, Database Manager, Controller, My Zeo Manager, Sleep Pattern Manager, Sleep Quality Manage

System Operations

+getSleepPatternsFromDB(Session):SleepPattern

- Obtain Sleep Pattern for the particular session from the database

+getSleepPatterns():SleepPattern

- Get Sleep pattern for current heart rate data from Sleep Pattern Manager for the current heart rate data

+verifySleepPatternsWithZeo():void

- Compare Sleep Pattern with Zeo's sleep Pattern to Obtain accurate results

+getHeartRateData():HeartRateData

- Get Heart rate data from MOTOACTV website

+getHeartRateDataFromDB(Session):HeartRateData

- Get Heart rate data for the corresponding session from the database.

+getSuggestionsFromDB(Session):Suggestions

- Get the sleep suggestions from the database for the corresponding session

+getSuggestions():Suggestions

- Get suggestions from the database for the current user

+saveHeartRateData():bool

- Save Heart rate data in the database and if successful, return true, else false

+saveSleepPatterns():bool

- Save Sleep pattern in the database and if successful, return true, else false

+saveSuggestions():bool

- Save Suggestions in the database and if successful, return true, else false

+login(userName:string, password:string)

- pass the username and password to the Account Manager and proceed

+retriveSessions():List<Session>

- Get all the sessions stored on the database

Database Manager

DatabaseManager
+saveHeartRateData(HeartRateData):bool +saveSuggestions(Date):bool +saveSleepPattern(SleepPattern):bool +getHeartRateData(Session):HearRateData +getSuggestions(Session):Suggestions +getSessions():List<Session>

Attributes: Heart Rate Data, Date, Sleep Pattern

Association: Controller, Account Manager

System Operations:

+saveHeartRateData(HeartRateData):bool

- To check if heart rate is saved correctly in the database

+saveSuggestions(Date):bool

- Save suggestions to the database based on the date and return true if successful

+saveSleepPattern(SleepPattern):bool

- Save Sleep Pattern to the database

+getHeartRateData(Session):HearRateData

- Get Heart Rate data for a session from the database

+getSuggestions(Session):Suggestions

- Get suggestions corresponding to a session from the database

GUI

GUI
-memberName
+showLoginScreen() +LoginFailed() +showMainScreen() +showHeartRateScreen() +showHeartRates(hr:HearRateData) +showSleepPatternScreen() +showSleepPatterns(SleepPattern) +showVerificationResults(bool) +showSuggestionsScreen() +showSuggestions(suggestions) +showHeartRateGraph(HeartRateData) +showSleepPatterns(SleepPattern)

Associations: Controller

Attributes: MemberName

System Operations

+showLoginScreen()

- First Page, Lets the user authenticate themselves by supplying username and password in association with the Controller

+LoginFailed()

- If The Controller sends a message saying the login falied, informs the user

+showMainScreen()

- Upon successful login, the Main Screen is presented, for the user to get Heart Rate Data, To get sleep suggestions etc

+showHeartRateScreen()

- The

User

User
-userName:string -userPassword:string -userType:int
+isHighlevelUser():bool +setUserType(userType:int):void +getUserType():int +getUserName:string +getPassword:string

Attributes: Username, User Password, User Type

Associations: Controller

System Operations

+isHighlevelUser():bool

- Checks if the User is a high level user, if so , return True

+setUserType(userType:int):void

- Sets the User Type, if High Level or Low Level

+getUserType():int

- Obtain the User Type

+getUserName:string

- Get the User name

+getPassword:string

- Get the password corresponding to the user

Session

Session
-sessionBeginning:Date -sessionEnd:Date
+setBeginning(Date):void +setEnd(Date):void

Attributes: Date

Associations: GUI, Database Manager

System Operations:

+setBeginning(Date):void

- Set the beginning date of the session

+setEnd(Date):void

- Set the end date of the session

Heart Rate Data

HeartRateData
-dataSession:Session -numOfRecords:int -rawData:List<int>
-memberName

Attributes: Session, Number of Records, rawData

Association: Everything!

System Operation:

Forms the base Data Structure for this whole project

7.3 Traceability Matrix

Classes / Domain Concepts	Account Manager	Controller	DataBase Manager	GUI	User	Heart rate Data	Session	Sleep Pattern Manager	Sleep Apnea Manager	My Zeo Manager	Sleep Quality Manager	Motoactv Manager
MotoActv Manager												x
MyZeo Manager										X		
Database Manager			X									
Data Loading Manager		x	x	x	x	x	x	x			x	X
Sleep Pattern Manager			x	x	x	x		x				
Sleep Quality Manager	x	x	x	x	x	x	x	x	x		x	x
Sleep Apnea Manager									x			

classe	Login Manager	Date Manager	Data Logging Manager	Report Manager	Sleep Diagnosis Manager	User Account Manager	Controller	GUI
Account Manager	x	x		x		x		x
Controller	x			x		x	x	
DataBase Manager	x			x		x		
GUI	x			x		x		x
User	x	x				x		
Heart rate Data			x	x				
Session		X						
Sleep Pattern Manager				x	x			
Sleep Apnea Manager				x	x			
My Zeo Manager					x			
Sleep Quality Manager				x	x			
Motoactv Manager								

7.4 Design patterns

Both in UC 5, UC 6, we use publish-subscribe pattern. The GUI don't communicate directly with the managers. Instead controller acts as publishers to get the event source.

The process is as follows:

→GUI (publishers) finds the interested objects (subscribers)

→Subscribers register/unregister with publisher

→The registered subscribers (such as SleepPatternManager in UC5, SleepQualityManager in UC6) process received event notifications

→Publisher get notified, return results to event source GUI, in this case.

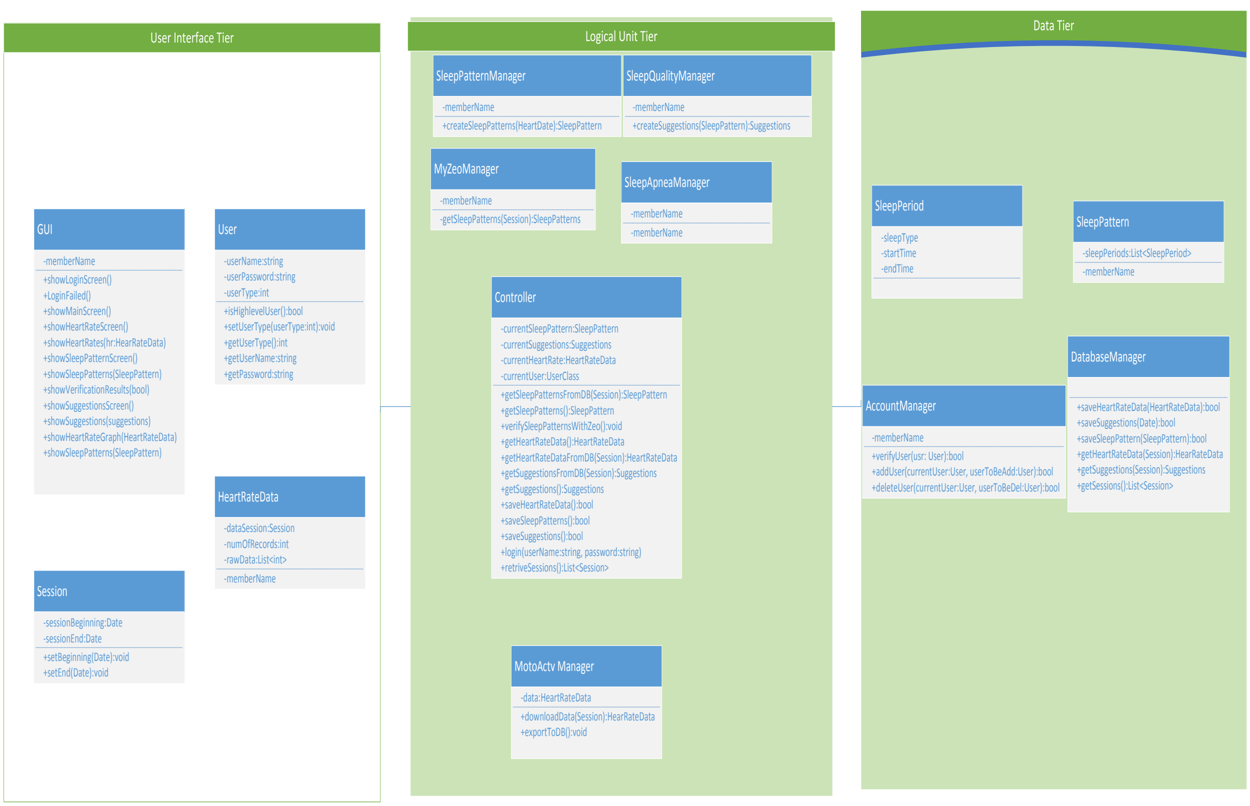
Decorator pattern is applied in the communication between UC5, UC6, UC7 and UC8. Here mainly use processing between UC6 with UC7, UC8 as an example.

From the use case descriptions, we know that the pre-condition for UC7 and UC8 is UC6 is executed. Say there's a decorator between this communication. If the UC6 is done, the decorator forwards the request to next object, may forward to the request to the controller, and controller calls the SleepQualityManager to calculate the score. And then the controller make a query to DatabaseManager to retrieve the corresponding suggestions. This helps to bulid chain between those classes, make the event flow.

8.1. Architectural Styles

Our system adopts 3-tier architecture in which the user interface, functional process logic ("business rules"), data storage and data access are developed and maintained as independent modules. The Presentation layer deals with the user interface which helps the user to interact the system in a smooth manner providing the ability to understand the results. The logic tier is the heart of our system. In the tier, the raw data uploaded by users is being processed by various algorithms and transferred to understandable results. In the end, the data tier stores all the user data and also the logic tier retrieves data from this tier for processing.

8.2 Identifying Subsystems



8.3. Mapping Subsystems to Hardware

Our system does not run on multiple computers. As we scale the system, we forsee this requirement, but as of now we shall not deal with this scenario

8.4. Persistent Data Storage

Our system must store all data that the system may use. This includes all user profile information, personal heart rate data. Our system accomplishes this by using relational database.

8.5. Network Protocol

Since our system runs on a single machine, network protocol is not applicable here.

8.6. Global control flow

The system is event driven. The user interacts with the system by first registering his/her user name and setting the password. Once the user log in and the next step is to download his/her personal heart rate data from website into database. Then the system waits for events, which could be clicking "analyze" button or selecting sleep disease diagnose function or just changing the time period. In addition, our system is not a real-time system and does not use multiple threads. Only one user can login into the system at a time.

8.7. Hardware Requirement

Our system requires the use of a MOTO Heart Rate monitor and Myzeo sleep monitor. In addition, the PC must have an internet connection to be able to download heart rate data from Motoactv website.

9 ALGORITHMS AND DATA STRUCTURES

9.1 Algorithms for User Management

9.1.1 Verify user

Required for login screen and high level user operations in account management GUI

Expected return: user Record from database with id, name, password and type fields

Possible errors: User (password_error, user_not_found)

Step 1) Create user object with name and password fields are filled

Step 2) Retrieve userRecord with user.name from database

Step 3) Does userRecord exist? If no return user_NotFound

Step 4)Does userRecord.password = user.password? if no return User_password_error

Step 5) if yes, return user object

9.1.2 Getting all account names

Required for populating deleteUser combo box in account management GUI

Expected return: All user names from database

Possible errors : AuthenticatingUser (password_error, permission_error, user couldn't be found)

Step 1) create authenticatingUser object with name and password fields

Step 2) verify authenticatingUser. If any error from verifyUser, return same error

Step 3) if authenticatingUser is low level user return Authenticating User_permission_error

Step 4) else get all users from database and return

9.1.3 Adding user

Called from account management GUI

Expected return: success

Possible errors : AuthenticatingUser (password_error, permission_error, user couldn't be found)

Step 1) create AuthenticatingUser and newUser objects with userNames and passwords

Step 2) verify AuthenticatingUser, if there is an error with verifying return same error

Step 3) if AuthenticatingUser is a low level user return AuthenticatingUser_permission_error

Step 4) otherwise user is allowed to add account, add newUser to database return success

9.1.4 Deleting user

Called from account manager GUI

Expected return: success

Possible errors : AuthenticatingUser (password error, permission error, user couldn't be found), deleteUser couldn't be found

Step 1) create AuthenticatingUser object with userNames and passwords. Also create deleteUser with only username

Step 2) verify AuthenticatingUser, if there is an error with verifying return corresponding error

Step 3) if AuthenticatingUser is a low level user return AuthenticatingUser_permission_error

Step 4)else retrieve userRecord from database from deleteUser.name

Step 5) if deleteUser doesn't exist return deleteUser_notFound

Step 6) if user exist delete deleteUser by id from database return success

9.2 Algorithms for Sleep Pattern Assessment

Healthy sleep consists of several stages: deep sleep, light sleep, and rapid eye movement (REM) sleep. According to some papers, we know that these sleep stages can be characterized and distinguished by correlations of heart rates[8]. Here we get some information during subjects' sleep from MotoACTV, and use heart rate as a main indicator of sleep quality.

In our first processing and analysis duration, we mainly apply the heart rate data into time domain analysis.

Dr. K. Krrauchi, in a study reported in "Neuropsychopharmacology" (2001) detected an average drop from 64 to 52 beats per minute from lights off till you reach light, continuous sleep[2]. The low frequency power as well as the high frequency power was lower when the subjects were asleep. There is also a relationship [4] between the variation of heart rate and specific sleep stages. Basically, the larger variation usually goes with the REM sleep. Difference of heart rate variation between REM sleep and Non-REM sleep may be used to distinguish the sleep stages. On the other hand, in the frequency domain, one study [5] has showed that compared to Non-REM sleep, low frequency band power has decreased and low frequency (LF) to (HF) ratio has significantly increased during REM sleep.

However, the time domain analysis seem to be not enough to further distinguish different sleep stages including awake, light sleep, REM sleep, deep sleep. As we have read some papers saying that frequency domain analysis may be more suitable for sleep stages analysis, we then took our data processing into frequency domain.

Power-Spectral density (PSD) contained in the main frequency bands of heart rate variability (HRV) can be used as a new technique to find identification factors for the different sleep stages. The result show that the PSD of the very-low-frequency (VLF) band and the low-frequency (LF) band are reduced as sleep stages vary from the wake state to REM sleep and further to light sleep (S1-S2) and deep sleep （S3-S4）。 The variation of the PSD in the high-frequency (HF) band is almost the opposite. The ratio of the VLF/HF PSD is found to be a good indentification factor between the different sleep stages, showing better results than other, commonly used factors such as the LF/HF and VLF/LF PSD ratios [7].

Following are some thresholds we used in our analysis.

Frequency components:

—The low-frequency (LF) power-spectral density （PSD） components, between 0.05HZ and 0.15HZ, linked to sympathetic modulation, but also including some parasympathetic influence, and

—The high frequency (HF) component, from 0.15 HZ to 0.5 HZ, which reflect parasympathetic (vagal) activity.

For demo 1:

9.2.1 Algorithm for distinguishing awake and sleep stages:

Step 1)Calculate the average heart rate of every 5 minutes, HR_i

Step 2)If HR_i > HR_{i+1} > HR_{i+2} and $avg(HR_i + HR_{i+1} + HR_{i+2})$ < θ, go to step 4

Step 3) $i = i+1$, go back to step 2

Step 4) t_{i+2} is the time when user goes to asleep stage

Where HR_i is the average heart rate in the ith 5 minutes; θ represents the threshold of heart rate based on experiments.

9.2.2 Algorithm for distinguishing REM stage and NREM stage:

Step 1) Calculate the average heart rate of every 5 minutes, HR_i

Step 2) Calculate the standard variance of heart rate in every 5 minutes, Var_i

Step 3) If $avg(Var_i + Var_{i+1}) > \theta_{var}$, go to step 5

Step 4) T_i is NREM stage, $i = i + 1$, go back to step 3

Step 5) T_i and T_{i+1} are classified as REM stage, $i = i + 1$, go back to step 3

Where Var_i is the standard variance of heart rate in the ith 5 minutes, and θ_{var} is the threshold, which we estimate it as the three-quarters of heart rate range. T_i means the ith 5 minutes.

For demo2:

9.2.3 Algorithm for distinguish different sleep stages:

Step 1) Cut the sleep heart rates data into 5 minutes

Step 2) Calculate the power spectral density (PSD) of heart rates data in 5 minutes

Step 3) Get the low frequency band and high frequency band PSD

Step 4) Get the ratio of LF/HF PSD

Step 5) Get the according sleep depth, and try to find sleep stages based on these ratios in our situation

9.2.4 Algorithm for computing the sleep quality score:

the total score is divided into four parts: light sleep, REM sleep, deep sleep, and awake.

an initial score is assigned to these four parts separately, where light sleep gets 2, REM sleep gets 3,
deep sleep gets 4, and awake gets 1.

for light sleep, if the ratio it occupies in the whole sleep hours is 0.5~0.6, then score(light sleep)=2; if not,
score(light sleep)=2-(|ratio-0.55|-0.05)*20

for REM sleep, if the ratio it occupies is 0.2~0.25, then score(REM sleep)=3; if not,
score(REM sleep)=3-(|ratio-0.225|-0.025)*80

for deep sleep, if the ratio it occupies is 0.15~0.2, then score(deep sleep)=4;
if ratio>0.2, score(deep sleep)=4+(ratio-0.2)*40;
if ratio<0.15, score(deep sleep)=4-(0.15-ratio)*60;

for awake, if the ratio it occupies <0.05, then score(awake)=1; if not, score(awake)=1-(awake-0.05)*20;

total score =8*{score(light sleep)+score(REM sleep)+score(deep sleep)+score(awake)}
if total score<0, then assign 0 to it.

9.2.5 Algorithms for Sleep Apnea Detection

Sleep Apnea is a common disorder characterised by the patient's inability to breathe for short periods of time while asleep. This causes changes in the ECG of the person[18] and thus has some amount of effect on the heart rate. The aim was to use the data available to us to characterize if the patient suffered from sleep apnea or not. If the patient is diagnosed with sleep apnea, the software prompts the user to take a questionnaire[19], usually asked by doctors, to calculate the sleep apnea score.

In order to diagnose sleep apnea, the MIT BIH [21]database of ECG samples of patients with sleep apnea was used. The first challenge was to extract heart rate data from ECG. The aim was to detect q-q peaks in the qrs complex[20] and divide by the time range.

Once the database of heart rate s with Sleep Apnea v.s non sleep apnea samples were established, a Support Vector Machine was used for classification. In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The training set consisted of equal number of samples of heart rates with obtrusive sleep apnea, borderline sleep apnea and healthy sleep. The features of each training sample, were the minimum heart rate, maximum heart rate, average heart rate and the variance of the heart rate. The reason these parameters were chosen was because sleep apnea is characterised by changes in heart rate and variations in heart rate.

The training matrix essentially was:

60.8105	84.3786	35.2941	109.091
80.9156	37.7189	36.3636	107.143
67.1186	86.4228	37.037	107.143
63.5362	60.501	41.3793	95.2381
61.6386	28.0683	27.1493	96.7742
73.0484	61.9362	49.5868	117.647
66.8224	24.267	53.0973	101.695
66.1046	26.0269	51.2821	100

65.9613	62.4843	33.5196	120
59.1599	46.4638	15.9151	117.647
58.9123	38.1725	3.8835	96.7742
65.6648	25.4774	53.5714	101.695
53.9815	58.5044	36.5854	98.3607
59.0691	40.0866	2.3603	101.695
60.7049	28.9359	41.3793	98.3607

The diagnosis of this prompts the user with a questionare[].

9.3 Data Structure

9.3.1 Data classes for account management

User

Id (int 11) : unique user id

Name (varchar(45)) : unique user name

Password(varchar(45)) : user's password

Type : user's level :

0 – High level user

1 – low level user

9.3.1 Data structure for SleepPeriod

In order to conveniently analyze sleep pattern, we construct a specific data structure called SleepPeriod, which is organized as follows:

SleepPeriod

sleepType : sleep stages :

0 – awake

1 – NREM

1 – REM

startTime : the start time of current sleep period

endTime : the end time of the current sleep period

Then we used a linked list to store them for further use and easy access.

10. USER INTERFACE DESIGN AND IMPLEMENTATION

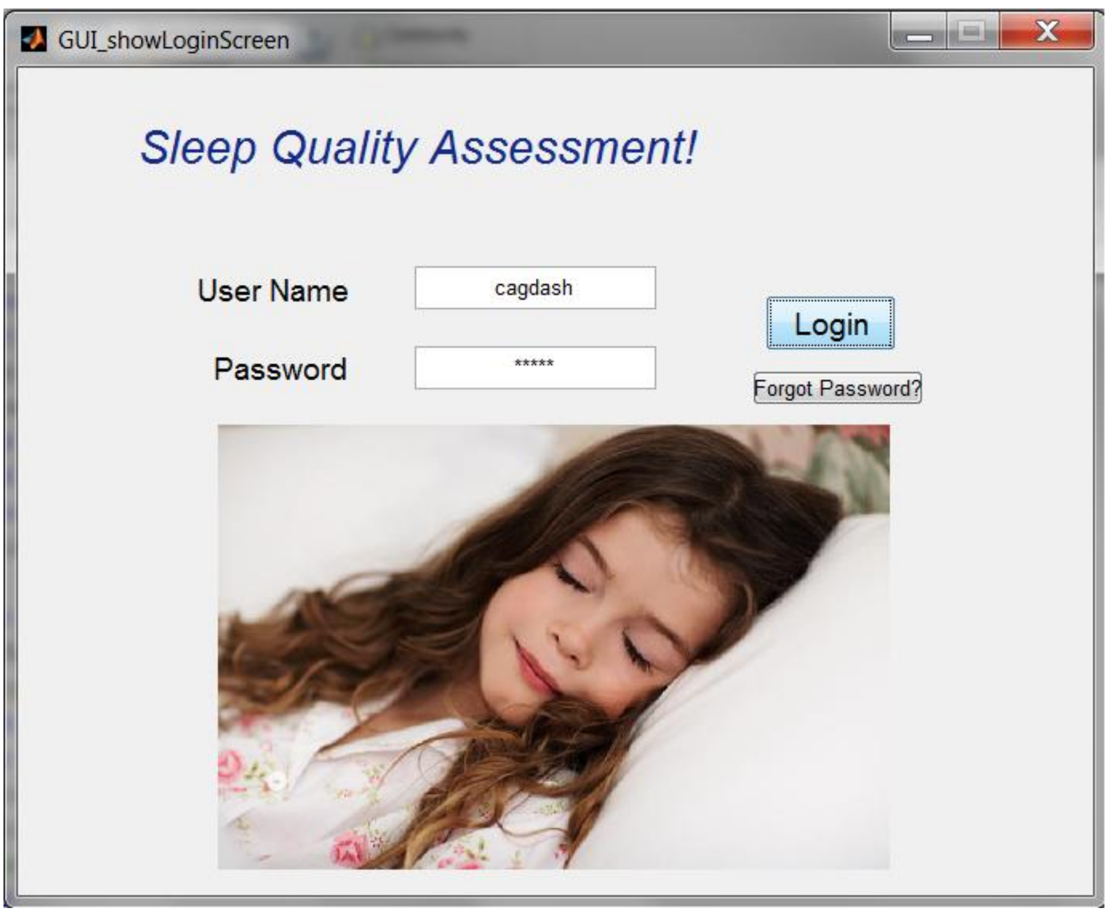
The main aim of user interface design was to keep the interface as simple as possible. We wanted the user to have as hassle free an experience as he/she could. In order to achieve this, we made the following implementable on the GUIDE toolbox of MATLAB and using JAVA:

1. Login Page:

User enters using his/her user name and password

Presses Login

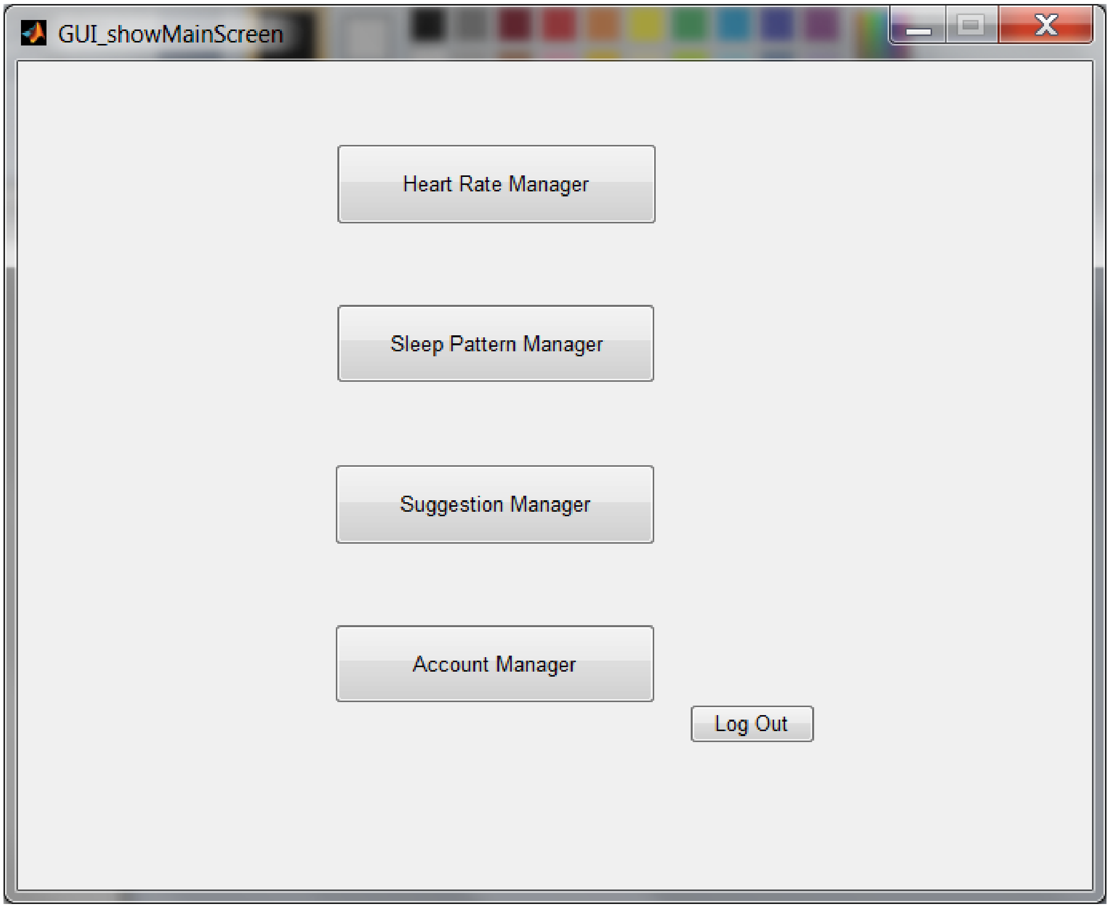
If user forgers the password, he/she can press the Forgot Password button for help



2. Main Screen Page

User has options of Heart Rate Manager, Sleep Pattern Manager, Suggestions Manager and Account manager. User can choose the function he/she wants to use in this Main Screen Page directly.

There is a Log Out button for the user to change the account.

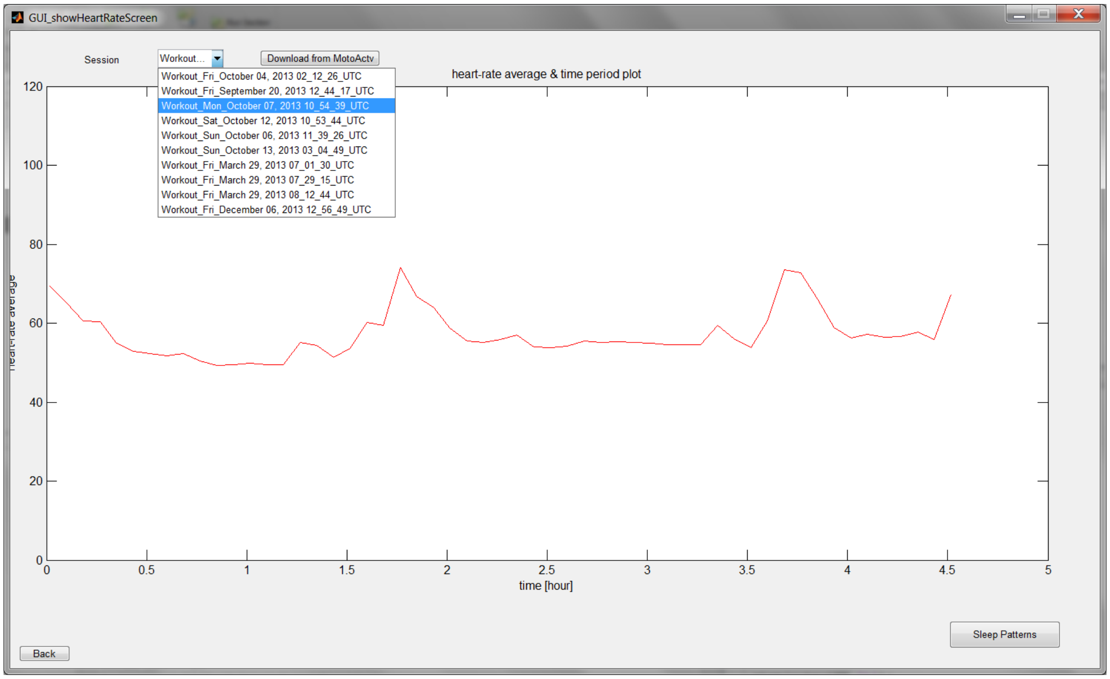


3. Heart Rate Manager

On choosing the Heart Rate Manager option, the user is prompted to a screen, where if the data has not yet been downloaded, user can press the Download button. Here a java script integrated with MATLAB directly downloads the new user data from MOTOACTV saving the user from task of going by himself to the website, logging in, searching for the download data tab and choosing the session.

Once the user chooses the session of interest, the heart rate data is displayed and if the user wants to know the sleep pattern, rather than going back to the main page, the user has been provided with a Sleep Patterns button in this page itself. This is another salient feature of our user interface design, where we have tried our best to minimize user effort.

There is a Back button for the user to get back to the Main Page.

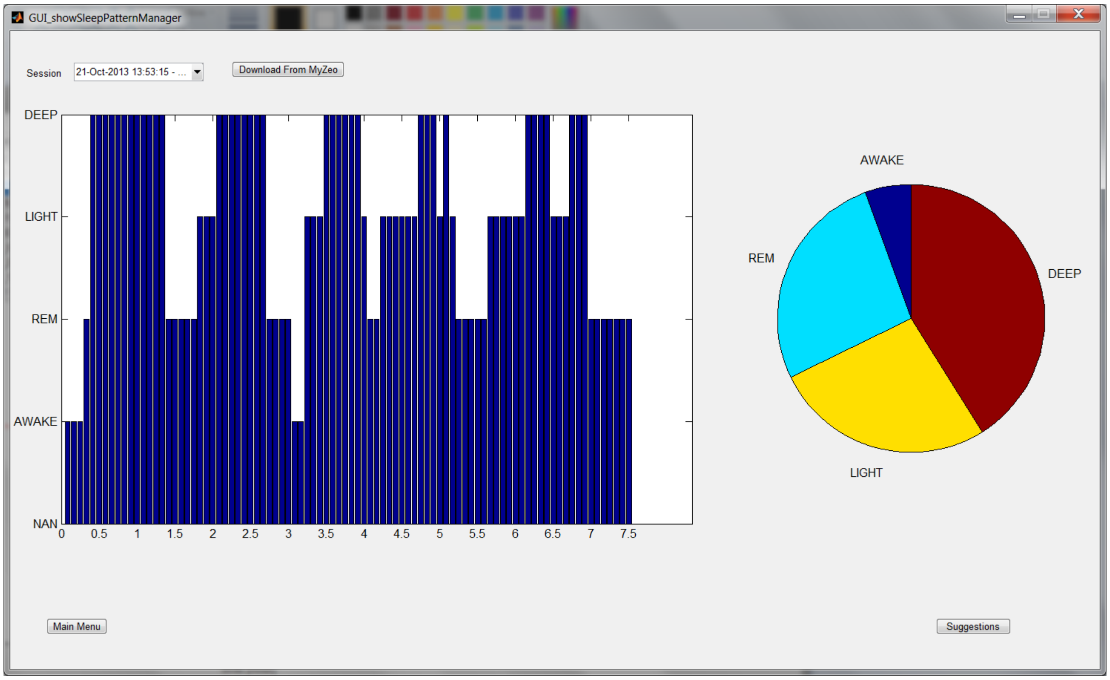


4. Sleep Pattern Manager

On choosing the Heart Rate Manager option, the user is prompted to a screen, where if the data has not yet been downloaded, user can press the Download button.

Once the user chooses the session of interest, the diagram of sleep stages is displayed, as well the pie chart of the sleep stages, so the user can know which stage took the most percentage in the user's sleep. And if the user wants to get the suggestion for the sleep, rather than going back to the main page, the user has been provided with a Suggestions button in this page.

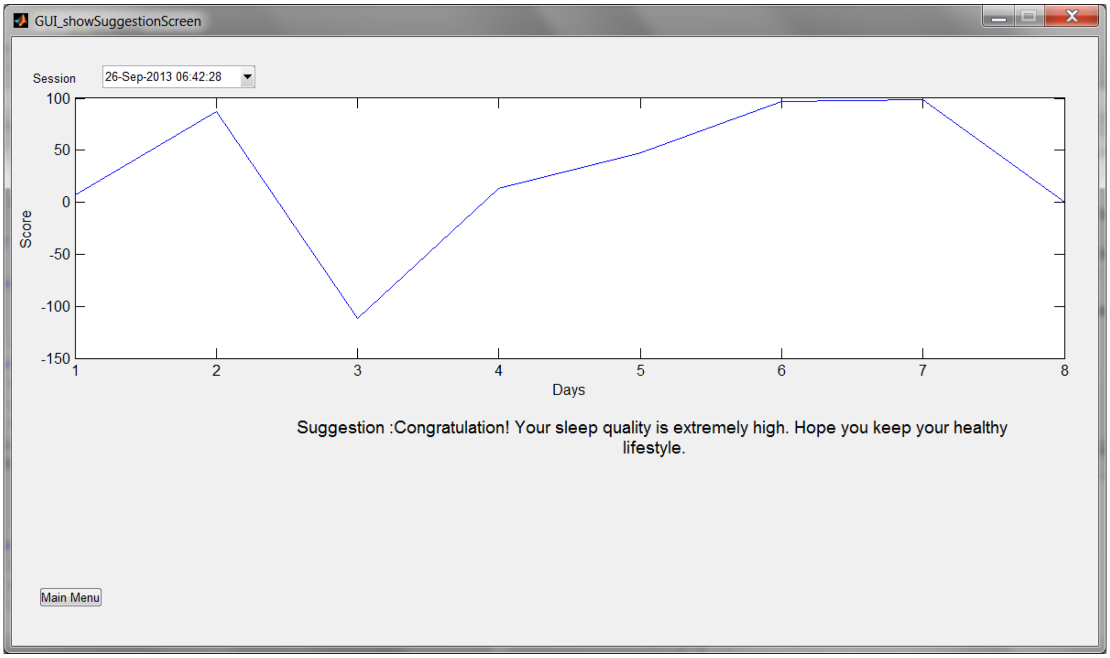
There is a Back button for the user to get back to the Main Page.



5. Suggestion Manager

On choosing the Suggestion Manager option, the user is prompted to a screen. In this screen, the user can choose the session of interest, and then the screen will show the score and the suggestion for the user in the screen.

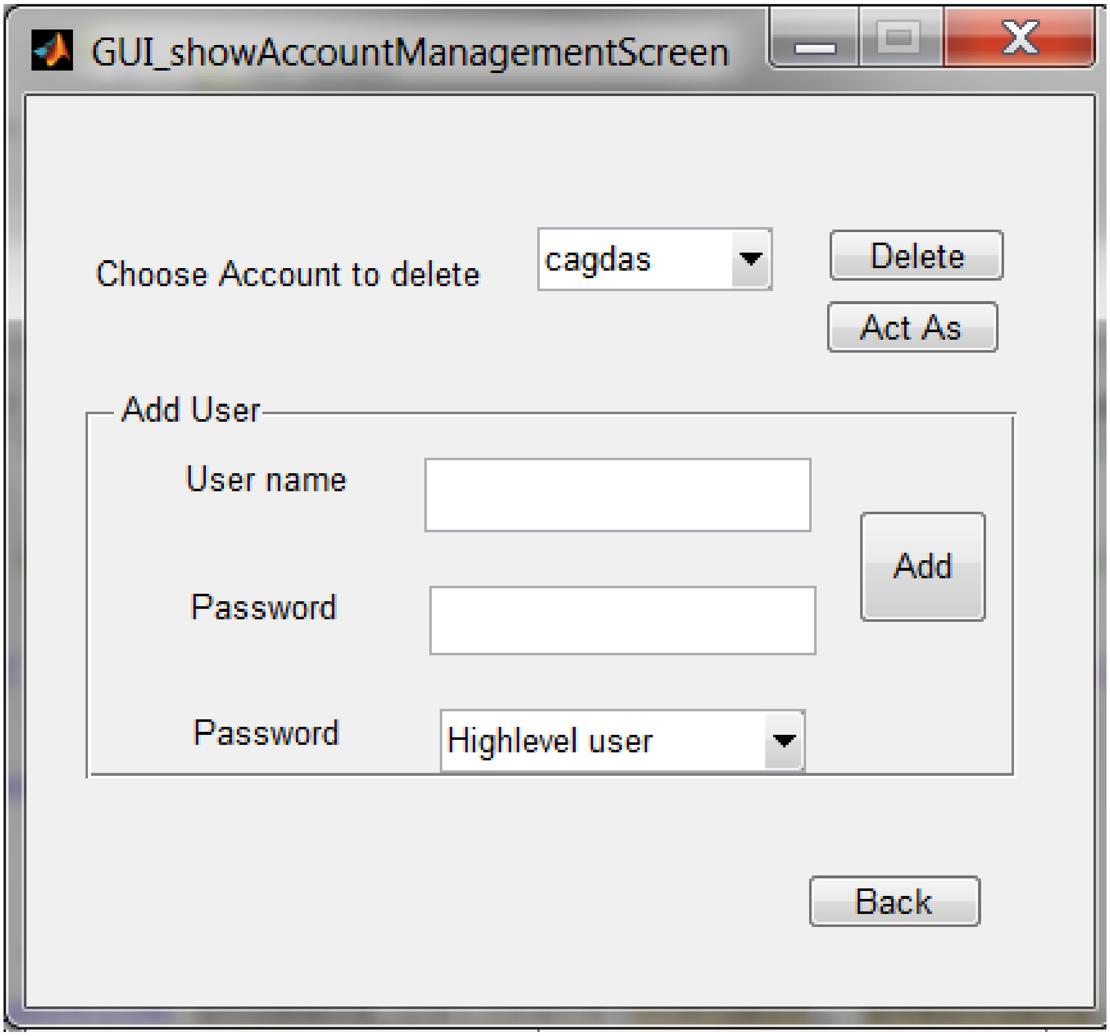
There is a Main Menu button in this screen to go back to the main screen directly.



6. Account Manager

On choosing the Account Manager option, the administrator is prompted to a screen. In this screen, the administrator can choose to add or delete a user account. Also, the administrator can choose to add a high-level user or low-level user in this screen.

There is a Back button in the screen, the administrator can go back to the main screen if he/she click the button.


GUI_showAccountManagementScreen
Choose Account to delete
cagdas
Delete
Act As
Add User
User name
Add
Password
Password
Highlevel user
Back

11 DESIGN OF TEST

The following are the necessary test cases for ensuring the correctness of the whole system.

Test case id	Account Registration
Unit to test	Account Manager, database Manager
Assumptions	The program has displayed the registration screen and is waiting for the user's input.
Steps to be executed	Input the preferred account number and password. Input the basic personal information. Click "finish" button to complete registration.
Expected result	A new user account has been added to the database.
Pass/Fail	A new user account shows up in the database / No account has been added to the database.

Test case id	login/log out
Unit to test	Account Manager, Database Manager
Assumptions	The program has displayed the log in page and is waiting for the user's input.
Steps to be executed	Input a valid user account and the corresponding password. Click "login" button. If step 2 successes, Click "log out" button.
Expected result	The user successfully logs in and logout.
Pass/Fail	The user log in the system, and then logout / the user fails to log in.

Test case id	Show heart rate
Unit to test	GUI, database manager
Assumptions	the user has logged in the system and clicked the "Heart Rate Manager" button on the main screen.

Steps to be executed	The user clicks the "Download from MotoActv". The user chooses one specific session.
Expected result	the heart rate data is displayed in the x-y coordinates as a function of time.
Pass/Fail	the graph is successfully displayed / the graph does not show up.

Test case id	Show sleep pattern
Unit to test	GUI, database manager, sleePatternManager
Assumptions	the user has logged in the system and clicked the "Sleep Pattern Manager" button on the main screen.
Steps to be executed	The user chooses one specific session. Click "show sleep pattern" button.
Expected result	The sleep pattern has been shown in terms of REM and NON-REM sleep hours.
Pass/Fail	The graph is successfully plotted / the graph fails to show up.

Test case id	Show suggestions
Unit to test	GUI, database manager , sleepQualityManager
Assumptions	The user has logged in the system and clicked the "Suggestion Manager" button on the main screen.
Steps to be executed	The user chooses one specific session. Click "show suggestion" button.
Expected result	The suggestions are displayed in the form of text.
Pass/Fail	Suggestions are generated and displayed. / no suggestion comes up.

Test case id	add user/delete user
Unit to test	Database Manager, Account Manager
Assumptions	there are valid user accounts in the database.
Steps to be executed	Choose one user account in the database. Click “delete user” button. Input a new user account and password. Click “add user” button.
Expected result	the former user account is removed from database and the new account exists in the database.
Pass/Fail	The former user account is successfully removed from database as well as new account being added. / Former account still exists in the database or new account does not show up.

Integration testing

The integration flow test we carried out is described below.

1) Login with unknown user account

-> Fail

2) Login with high-level user account with wrong password

-> Fail

3) Login with high-level user account with correct password

-> Success

4) Add high-level user

-> Success

5) Logout

-> must show login screen

6) Login with new high-level user

-> success, must show main screen

7) Logout

-> must show login screen

8) Login with high-level user

-> must show login screen

9) Remove new high-level user

-> success

10) Add low-level user

-> success

11) Logout

-> must show login screen

12) Login with new low-level user

-> must show main screen

13) Run Heart-rate Manager

-> must show an empty screen

14) Check sessions

-> There must be an empty list

15) Click download from motoactv

-> session list must be changed

16) check sessions

-> There must be a full list

17) Click one from the session list

-> Heart rate must be shown

18) Click sleep patterns

-> must show sleep patterns

19) Logout

-> must show login screen

20) Login with high-level user

-> must show main screen

21) Remove new low-level user

-> success

22) Logout

-> must show login screen

12 COMPARING MY ZEO AND MOTOACTV: AN INDEPTH ANALYSIS

One of the main aims was to compare how coherent and matching the My Zeo Sleep patterns were with our algorithm of sleep stage classification from the heart rate. The results we obtained and the analysis we carried out is as follows:

Based on papers[17] we decided to use following LF/HF ratios for thresholds to determine different stages.

Stage	Min threshold	Max Threshold
Deep Sleep	0.6	0.7
Light Sleep	0.7	0.95
REM	0.95	1.08
Wake	1.08	1.2

When we applied mentioned algorithm we got following figure (Figure 12 1)

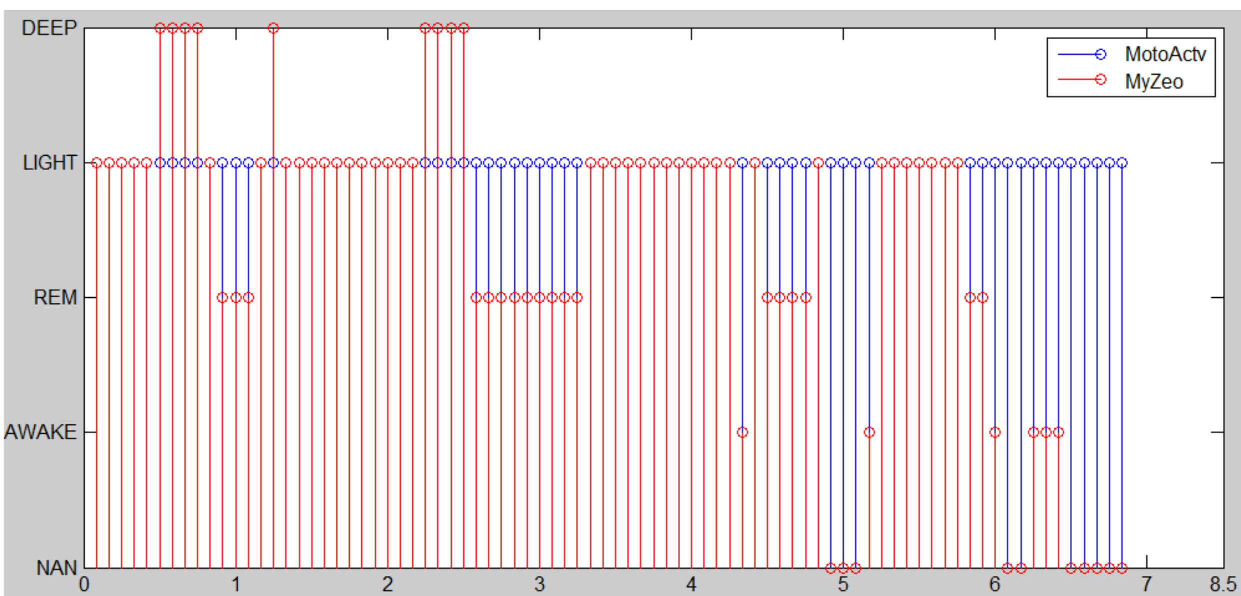


Figure 12 1 My Zeo v/s MOTO

As we can see what we got from MotoActv is actually always a deep stage. We realized the real problem as we plot LF/HF ratio graph (Figure 12 2).

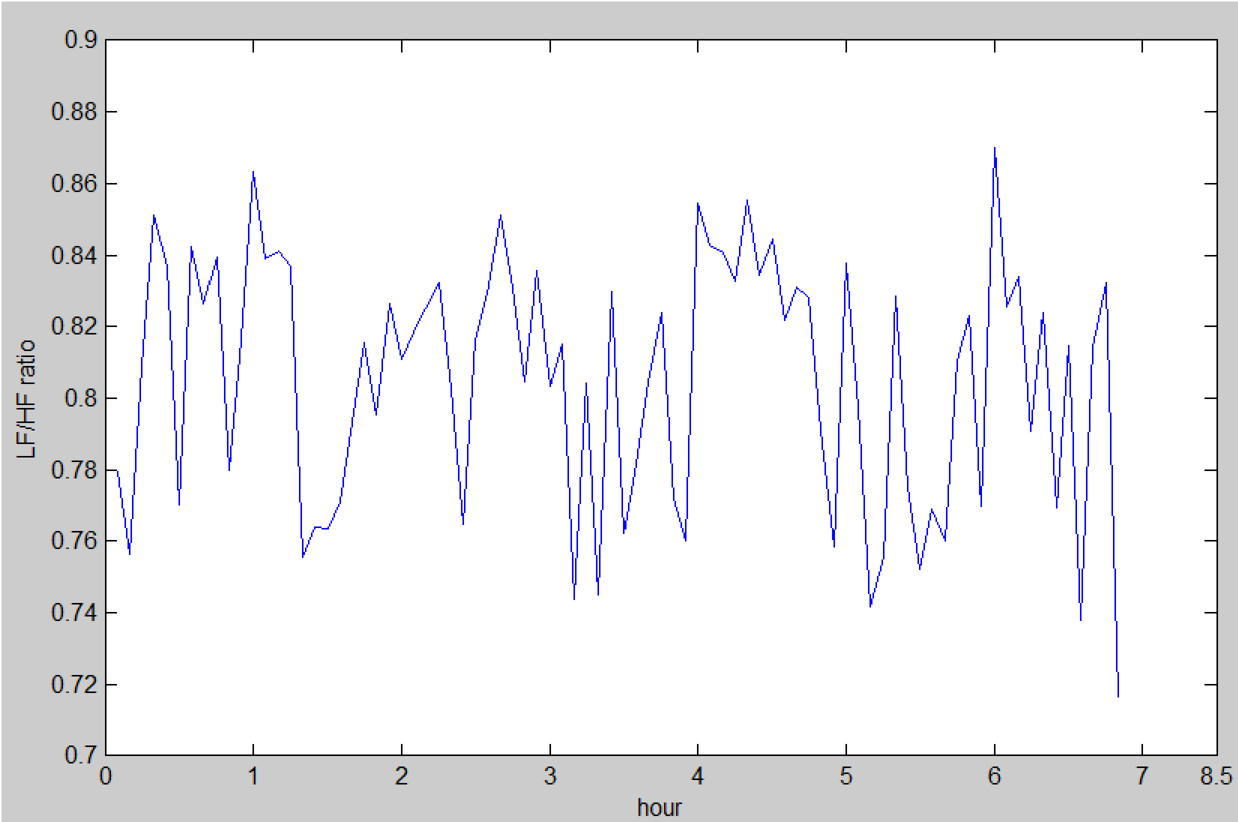


Figure 12 2 LF/HF MOTOACTV

As we can see actually LF/HF ratio always lies within 0.7 and 0.9 which corresponds to light sleep. To further investigate the situation we scaled LF/HF ratio and plot in the same screen with patterns from MyZeo.

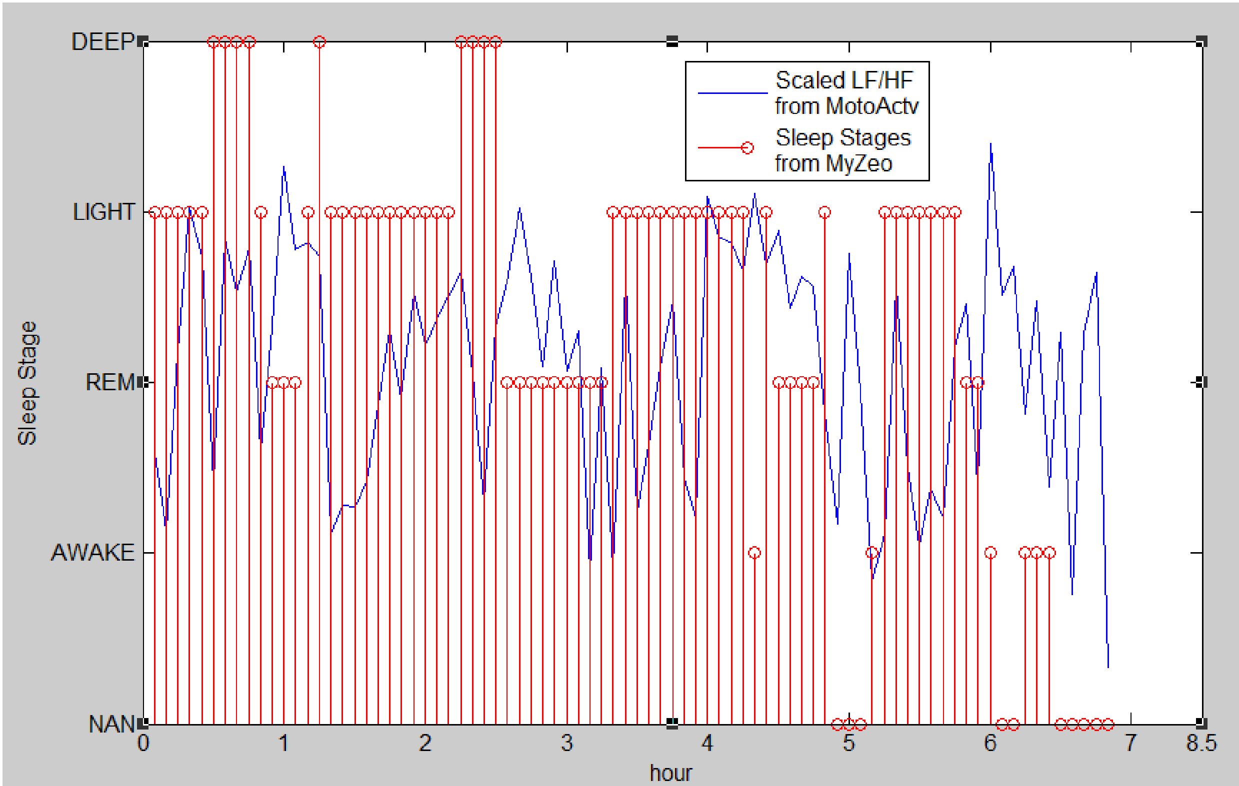


Figure 12 3 Scaled LF/HF with My Zeo

From figure we felt there might be another type of relation between LF/HF ratio and sleep stages from MyZeo. Finding that relation surely wouldn't be easy that's why we decided to try machine learning algorithms.

Our first machine learning algorithm was SVM and for features we used following four parameters, we choose those parameters based on the best practices in the papers[16,17].

- Parameter 1: Ratio of Low frequency coefficients over high frequency coefficients
- Parameter 2 :Ratio of logarithm of low frequency components) over logarithm of high frequency components
- Parameter 3: 1/Parameter 1
- Parameter 4: 1/Parameter2

We got following result

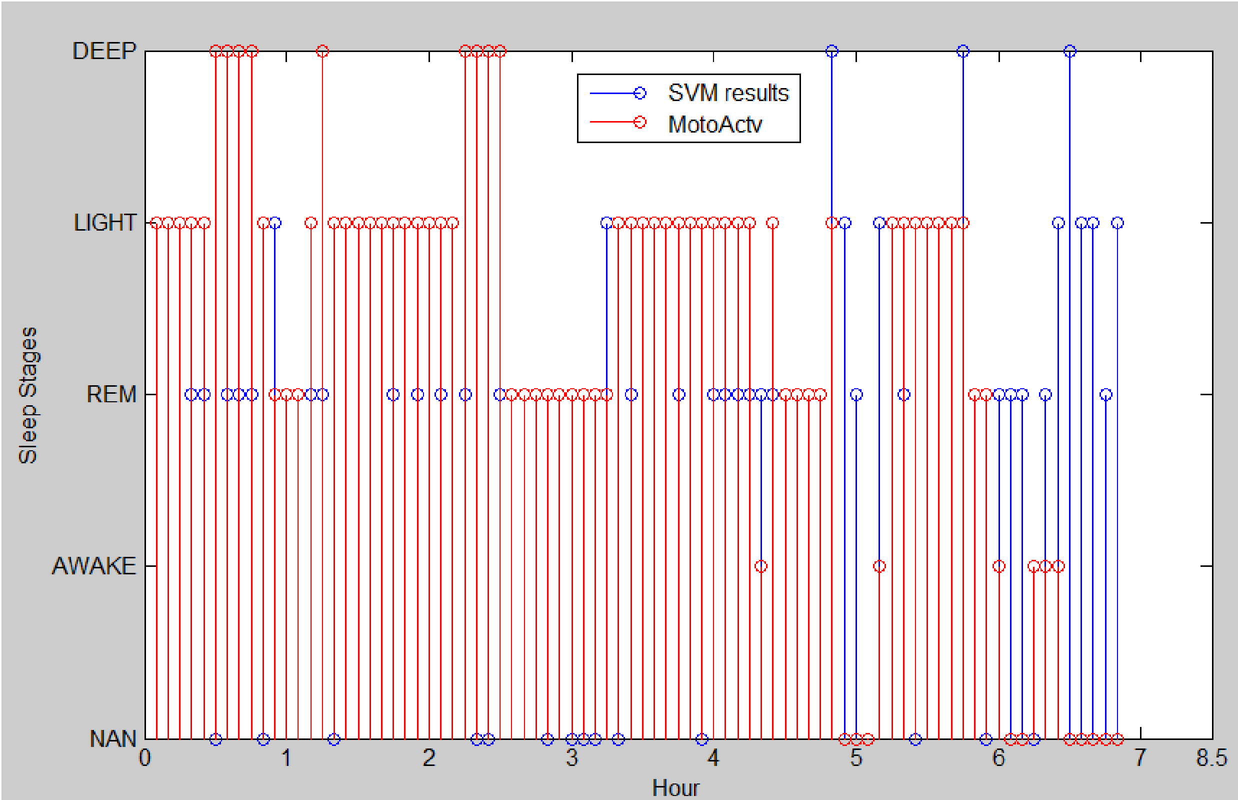


Figure 12 4 SVM for classification

Then we tried twenty different cutoff frequencies for for LH and HF discrimination and got four parameters as we mentioned in the previous section for each of cutoff frequencies. As total we train SVM with 80 features

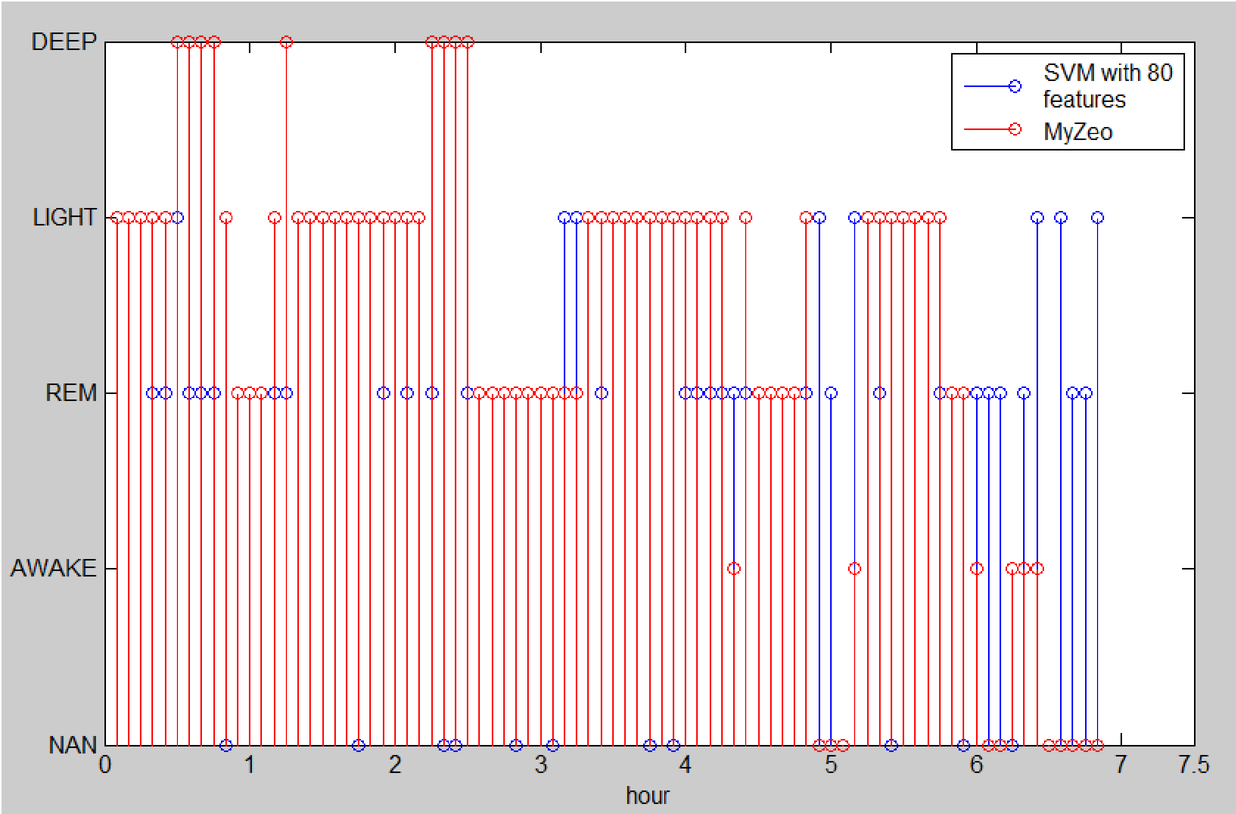
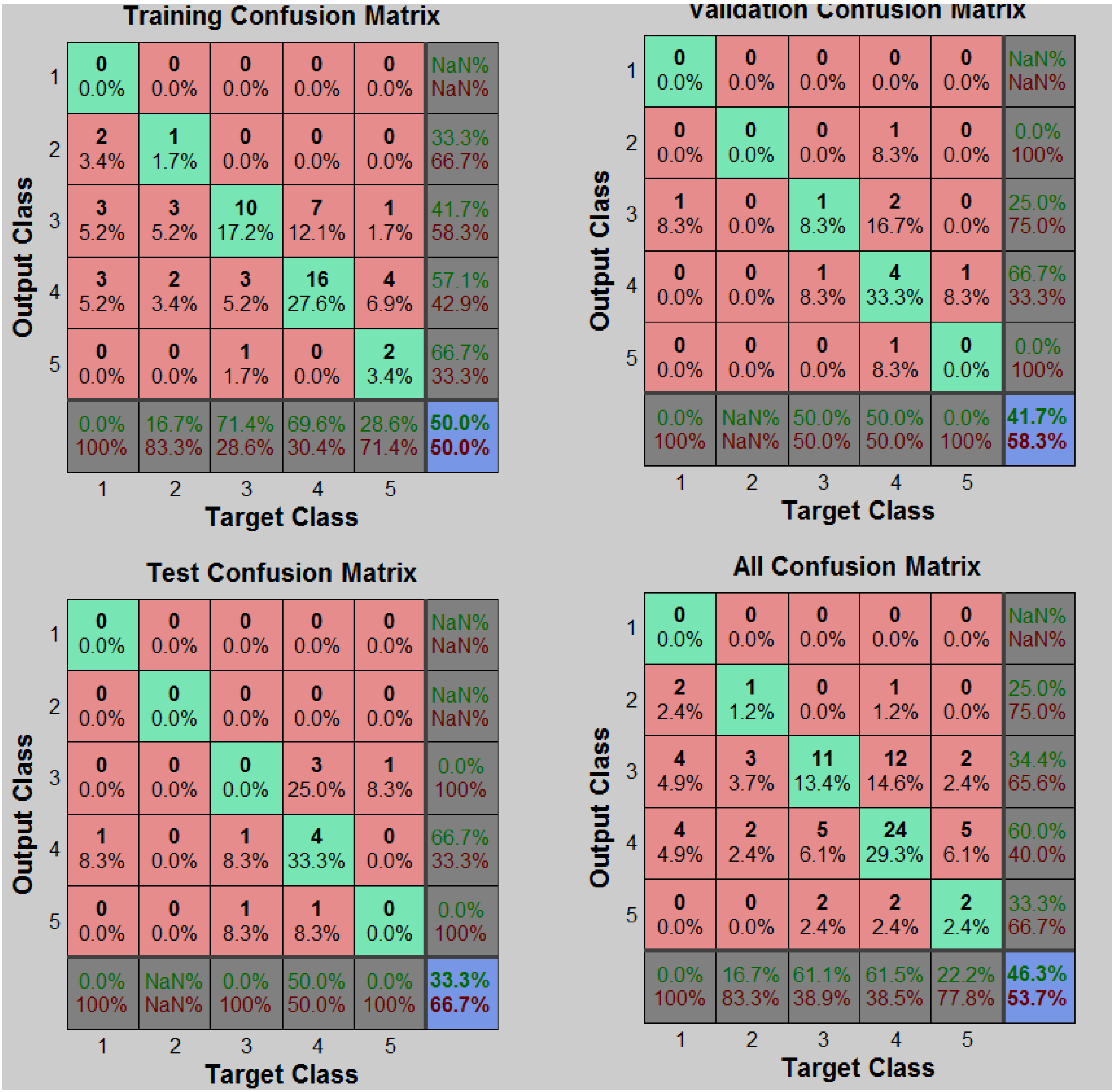


Figure 12 5 SVM with 80 features

Finally we tried neural networks algorithm. For our neural network we used 10 hidden layers. Also we used 70% of data for training 15% for Validation and 15 percent for Testing (58 samples, 12 samples, 12 samples respectively) Confusion matrix is given below



As we can see even the neural networks we couldn't get satisfactory results

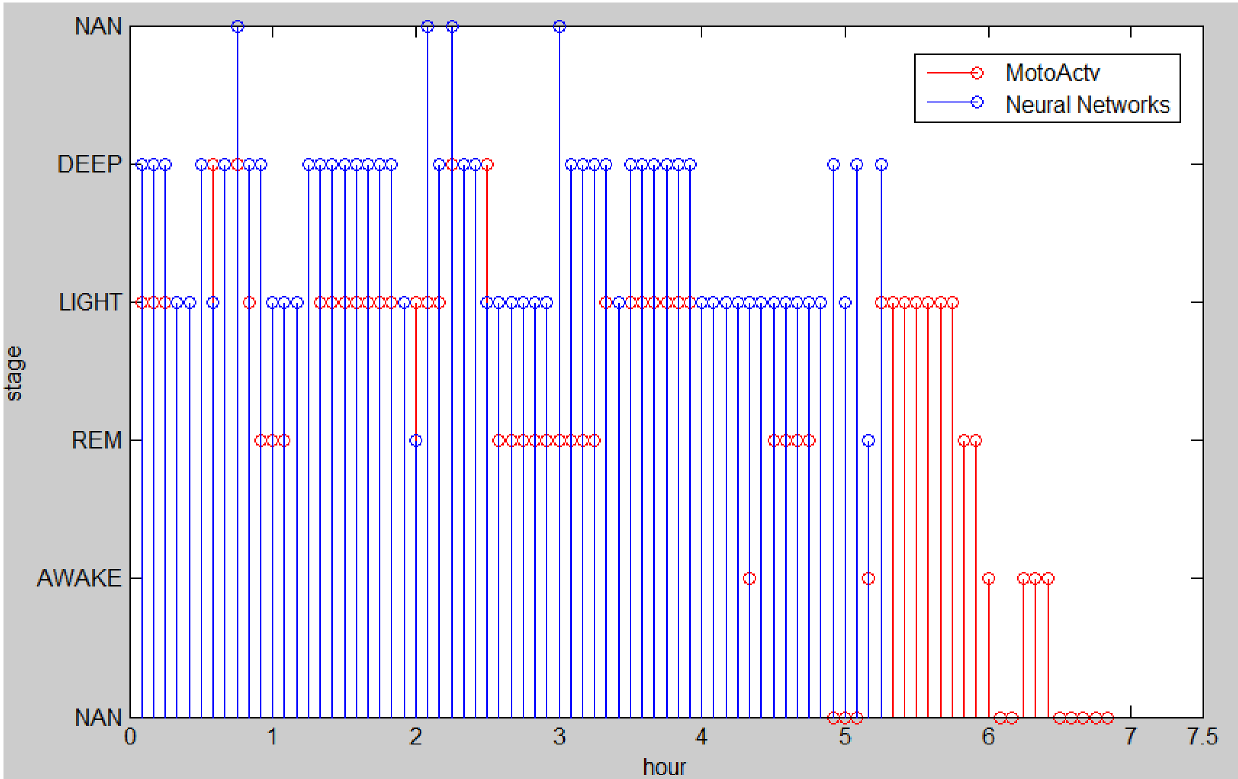


Figure 12 6 With Neural Networks

Thus we deduced the difference in MOTOACTV results and MY ZEO RESULTS could be attributed to the following reasons:

1. The signal sources are different EEG v/s heart rate and the transformation to compare and eeg sourced signal with a heart rate sourced signal we are using (in this case classification may be unmatched)

2. My zeo may be inaccurate

3. The algorithms we based our detection and classification may not translate to a corresponding algorithm in MyZeo

13 HISTORY OF WORK, CURRENT STATUS AND FUTURE WORK

In this section, we would like mention about different steps of our development process and compare them how we planned them, how it actually occurred. In our previous reports, we have already mentioned about our plans and deadlines, now it is time to compare them with reality. Following items are key stages :

Interface Design:

In our previous reports, we planned Interface design a part of all milestones of the project. We believed Interface must evolve with use-cases and reflect their functionality dynamically. Our envision about interface was right and we first designed every user screen to reflect the requirements of the functionality that screen serves for. After completion of the individual stages, we decided to reform our user interface for the sake of uniformity across screens and simplicity. For this reason, our heart rate screen, sleep pattern screen and suggestion screens are really similar to each other. In each of those screens we provided

- combo box control (*Every screen)* : That allows user to choose a specific session and show it in the screen
- An import button *(Heart rate & Sleep Pattern Screen)* : That allows user to import heart rate /sleep patterns from MotoActv website/MyZeo file.
- A graph (*Every Screen) :* To show current sessions data to user.
- A button *(Heart rate & Sleep Pattern Screen) :* To take user to next stage such as to create sleep patterns from heart rate data or suggestions from sleep patterns
- A button to Main menu *(Every screen) :* That allows user to navigate easily between screens

Interfacing Heart Rate :

Heart rate monitoring is the uttermost important piece of the our program. Obviously our program wouldn't have even the half of its functionality without it. If we cannot get the heart rate data from MotoActv we wouldn't be able to show heart rate data, create sleep patterns from heart rate data and compare it with the sleep patterns from MyZeo. Because of this priority, importing MotoActv is planned as one of the first milestones. Thanks to our hard work, reality conformed with our plans and we were able to finish Heart rate import tool first. Our tool is even better than our expectations, with a single click we download from the MotoActv's website seemlessly. On the other hand, downloading user heart rates from database takes a while.

Database Structure Design:

In our project planning, we also accepted database design as a fundamental module since it serves to all other modules. Therefore we assumed that it needs to be implemented first to serve other blocks. When we are developing our project, we saw our assumption about database was actually wrong. Database design require a good understanding of the data that we are going to use which is something that we don't have in the early beginning of our project. We observed we gain more information about data as we develop our project and database also needs to be evolve as the project. As a consequence, database design improved in every part of the project although we planned it as 4 week design stage

Algorithm Design , Test & Debugging :

We tried to stick our schedules as much as we can in terms of Algorithm Design, test and debugging. For the most of the project we were really successful in that sense. After we realized, matching ratio of sleep patterns from MotoActv and MyZeo is really low, we decided to try different algorithms and techniques. This unexpected event led to longer Algorithm design and test stages.

Other stages : Since the other parts of the project are reports and demos which have strict deadlines, our plans and reality had to match more strictly and did so

IN TABLE 13.1 we summarize our planned time-table for individual stages and what we happened in reality

Week #	41	42	43	44	45	46	47	48	49	50
Interface Design										
Interfacing Heart Rate										
Database Structure										
Algorithm Design										
Algorithm Test										
Debugging										
System Test										
Report Draft										
Report 1										
Demo 1										
Report 2										
Report 3										
Demo 2										
	Planned Schedule					Reality				

Table 13 1 Schedule of work

13.2 Key Acheivements

Following list summarizes our key achievements in a nutshell

- Creation of database to provide a backbone to our project
- Implementation of user account management system.
- A module to download heart rate data from motoactv website and import it to our local database.
- A module to create sleep patterns from heart rate data to classify rem, light and deep sleep stages.
- A module to import sleep patterns from MyZeo app.
- A module to evaluate user sleep patterns and gives scores and suggestions to the user
- Converting ECG data from MIT BIH database and creating heart rate data out of ECG data to train our sleep-apnea-detection module
- Our SVM-based sleep-apnea-detection module

13.3 Future work

We believe there is a place for improvement in creation of sleep stages and sleep suggestions. In our experiments, we have observed our sleep patterns created with the algorithms from heart rates of MotoActv doesn't match with sleep patterns from MyZeo. Therefore we first need to decide which sensor (Motoactv or MyZeo) we can take as a ground truth and improve the results from the other one. Also, creation of sleep-suggestions should be improved.

Sleep patterns from MyZeo cannot be taken as ground truth for sure, because there are not many scientific documents about reliability of MyZeo and also MyZeo has no support, since the producer went off . Obviously we cannot rely on sleep patterns from MotoActv neither, since the algorithms developed are relatively new and not tested. We believe first step in further improvement of the project should be comparing the sleep patterns both from MyZeo and MotoActv with some other reliable method.

In our project we want to get the best sleep patterns independent of which sensor is being used. After we decided which sensor outperforms, we should improve the results from the other device. Such a improvement will require incorporation of more sensors, such as accelerometers, temperature sensors, humidity etc. Actually, we have already started data collection for this purpose. We used Texas Instrument's SensorTag[22] device which has 6 sensors; namely accelerometer, gyroscope, magnetometer, temperature, humidity and pressure sensors. This device is also known for low power requirements, thanks to Bluetooth smart, and can operate with a single cell battery for years.

The final item in future work section is creating better suggestions for users, our current system uses a star-rating system that we developed based on our own intuition. We believe this suggestion system should be more scientific and must base on sleep science papers in the literature.

REFERENCES

[1] [11]. Software Engineering Book. Ivan Marsic.
http://eceweb1.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf
[2] Heart rate change from awake stage to sleep stage
URL: http://www.livestrong.com/article/105256-normal-heart-rate-sleeping/
[3] Ghosh P. Effect of Sleep on Heart Rate Variability[J].
[4] Masao Yaso, Atsuo Nuruki, Sei-ichi Tsujimura, and Kazutomo Yunokuchi, Detection of REM sleep by heart rate, Proceedings of The First International Workshop on Kansei
[5]. Test case. Wikipedia. March 3, 2012. http://en.wikipedia.org/wiki/Test_case
[6] Elsenbruch S, Harnish M J, Orr W C. Heart rate variability during waking and sleep in healthy males and females[J]. Sleep, 1999, 22(8): 1067-1071.
[7]Berlad, I., et al. "Power spectrum analysis and heart rate variability in Stage 4 and REM sleep: evidence for state-specific changes in autonomic dominance." Journal of sleep research 2.2 (1993): 88-90.
[8]Bunde, Armin, et al. "Correlated and uncorrelated regions in heart-rate fluctuations during sleep." Physical Review Letters 85.17 (2000): 3736.
[9]. What Does Usability Mean:Looking Beyond 'Ease of Use'. WQusability. March 7, 2012. http://www.wqusability.com/articles/more-than-ease-of-use.html
[10]. How to write effective Test cases, procedures and definitions. March 7, 2012
http://www.softwaretestinghelp.com/how-to-write-effective-test-cases-test-cases-procedures-and-definitions/
[11]. Unified Modeling Language. WikiPedia.
http://en.wikipedia.org/wiki/Unified_Modeling_Language#Interaction_diagrams
[12]. A. Rechtschaffen, A. Kales, A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects, Public Health Service, U.S. Government Printing Office, Washington, DC, 1968.
[13]. P. Achermann, A.A. Borbely, Simulation of human sleep: ultradian dynamics of electroencephalographic slow-wave activity, J. Biol. Rhythms 5 (2) (1990) 141 – 157.
[14]T. Penzel, H. Becker, J. Peter, P. Bunde, Detrended fluctuation analysis and spectral analysis of heart rate variability for sleep stage and sleep apnea identification, Comput. Cardiol. 30 (2003) 307 – 310.
[15]T. Penzel, J. Kantelhardt, L. Grote, J.H. Peter, P. Bunde, Comparison of detrended fluctuations analysis and spectral analysis for heart rate variability in sleep and sleep apnea, IEEE Trans. Biomed. Eng. 50 (10) (2003) 1143 – 1151.
[16]L. Toscani, P.F. Gangemi, R. Silipo, P. Ragghianti, E. Sirabella, M. Morrelli, L. Bangoli, R. Vergassola, G. Zaccara, Human heart rate variability and sleep stages, Ital. J. Neurol. Sci. 17 (6) (1996).

[17]P. Busek, J. Vankova, J. Opavsky, J. Salinger, S. Nevsimalova, Spectral analysis of heart rate variability in sleep, Physiol. Res. 54 (2005) 369 – 376.
[18] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages;http://circ.ahajournals.org/cgi/content/full/101/23/e215]; 2000 (June 13).
[19] Netzer NC, Stoohs RA, Netzer CM, et al. Using the Berlin Questionnaire to identify patients at risk for the sleep apnea syndrome. Ann Intern Med 1999 Oct;131(7):485-491.
[20] MIT-BIH Polysomnographic Database, http://www.physionet.org/physiobank/database/slpdb/
[21] Pan J. and Tompkins WJ., A real-time QRS detection algorithm, IEEE Transactions on Biomedical Engineering, 32:230-236, 1985.
[22] Texas Instruments Sensor Tag, http://www.ti.com/sensortag