# Software Requirement Specification Document

# MIU-X-PROSTHETIC Arm

Amr Hamdi, Hassan Hamdy,Philip Naguib, Hossam Badr
Supervised by Dr. Sarah Nabil and Eng. Radwa Samy

February 10, 2018

## 1 Introduction

### 1.1 Purpose of this document

This purpose of this document is to set the all functional requirements or non-functional of our project, also show an overview of the project in clear way and its interaction with other software is also stated.

### 1.2 Scope of this document

This document targets the end users like businesses that would integrate their systems with the MIU-X-Prosthetic system. It will also be beneficial and helpful for designers and developers that may work on the MIU-X-Prosthetic system in the future.

### 1.3 Overview

Real time moving and observing of virtual arm and enhancement of classification system. The MYO armband got 8 channels can detect signals throw each channel for acceleration and orientation. After reading signals it be filtered from noise and preprocessed to send to firebase cloud. Firebase cloud send the readings to processing unit to detect the movement and be classified and send the output generated to the cloud. The classification system should mainly aim to provide accurate movements to help amputees achieving their daily life tasks easily. Cloud sends detailed signals and action for arm to be rendered to in unity3d arm. If misclassification occurred the user sends feedback to the cloud. This document proposes MIU-X-Prosthetic which is a system that performs detection and identification of users EMG signals using MYO armband device, the user will get an account when he/she downloads the application account to allow him uploading his dataset files which are CSV files to make it customizable, the user will be able to send feedback if the movements where wrong through log files and he/she must also send his intent of the movement, the user can add new movements, then an alarm will be sent to the admin to check if the movement can be added so he will add it in the next update, if a new movement is add a notification will be sent to all users to make them update their application to the latest version.

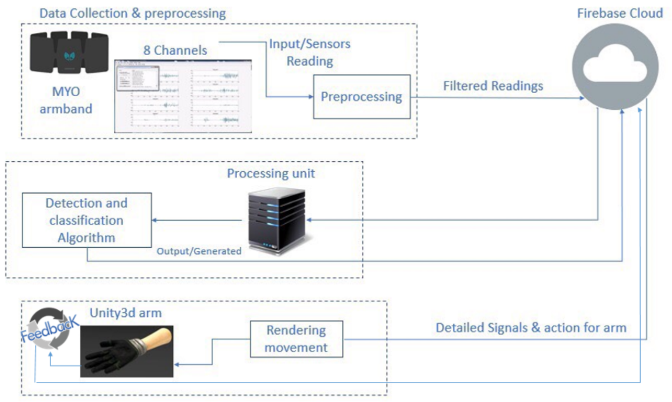

Figure 1: system overview

### 1.4 Business Context

According to a newspaper in Indonesia, one of biggest public hospitals in Jakarta named Rumah Sakit Cipto Mangunkusumo (RSCM) has statistical data about 35 % of diabetics that end with amputation. There are nearly 2 million people living with limb loss in the United States [1]. Cyberdyne is a Japanese robotics and technology company most noted for the marketing and distribution of the HAL 5 robotic exoskeleton suit. [6] Approximately 185,000 amputations occur in the United States each year [2]. In 2009, hospital costs associated with amputation totaled more than 8.3 billion dollars [3]. So, we decided to take challenge to help this people to feel normal again with an assistive limb.

## 2 General Description

### 2.1 Block Digram

User Manipulation: user can update his files remove, and delete his files.
Signal detection system: user will send his signals and it will be classified with the system to know which movement.

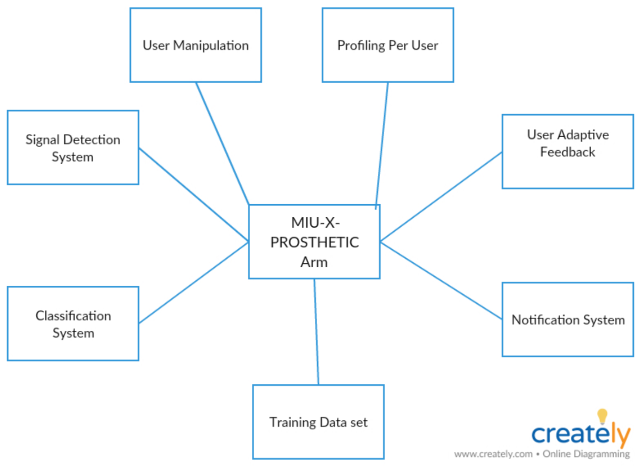

Figure 2: Block Diagram

Alarm System: the system will generate alarm if the user sends his feedback to the system.
Notification system: the system will send notification through user account on the online system. classification system: the system will classify the signals according to received data from MYO armband.

## 2.2 Product Functions

### 2.2.1 Customer (disable people)

The system allow to recognize the user's EMG signals. the system allow to send to classify the signals to thier movement class.

### 2.2.2 Admin

the system allow to add new movement to the software. the system allow to manage other user. the system allow to get monthly reports about new actions.

### 2.2.3 Notification System

send Notofication to the customer if there is any new Movement added to the system.

### 2.2.4 User Adaptive Feedback

send Alarm to the Admin if there is any new feedback from customer added to the system.

### 2.2.5 Training data set

the system allow to users to train their own data set.

## 2.3 Similar System Information

### 2.3.1 bebionic arm

As Shown in Figure 3, The bebionic Arm is a Commercial Arm .its a programmable not depend on machine learning. It has individual motors for each finger and microprocessors to control each finger. It achieves 14 Movements. The cost of this arm about 11,000 $.[4]

Figure 3: similar system 1

### 2.3.2 Toward improved control of prosthetic fingers using surface electromyogram (EMG) [5]

This paper showed that how to improve using of individual and combined fingers control. They use electrodes under skin to collect data (EMG signals) from muscles And it features 10 movements. As shown in figure 3,We choose this 10 movements on our proposed system.

### 2.3.3 MyoDrive: A new way of interacting with mobile devices

this system works in real time gesture recognition in order to control his favourite application like calls, music or navigation using MYO sensors. it is an automobile enviroment that is specified for driver in order makes him consentrate more while driving.[7]

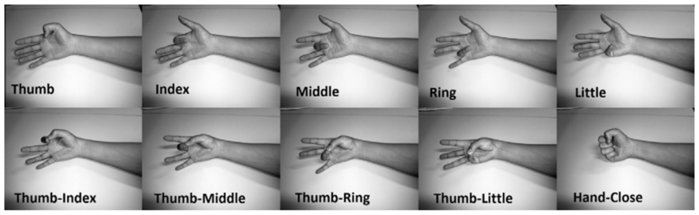

Figure 4: similar system 2

### 2.4 User Characteristics

There are types of user that will benefit from the system: people who made accident, diabetes problem or cancer that leads to get amputated.
1. amputation person
He is expected to be able to send EMG signals from his muscles(The nerves are not killed).
He can add his EMG signal dataset.
He adds new movements to system database to be updated in future.
He can get any new updated movements.
He gets notified with new updates in system.
He gives feedback if there are misclassification of movement.

### 2.5 User Problem Statement

This system will help disabled people a lot to live their lives as ordinary people. They can use their fingers to do any movement they want through the 10 movements or more on the system.

### 2.6 General Constraints

Username and password are used for the identification of users to login.
MYO armband connected to processing unit.
Bluetooth device connected to processing unit with distance 4 meters.

## 3 Functional Requirements

Code (01)
Name: CheckConnection
o Input:
o Action: System show that the MYO Arm band is connected or not to start

get the data from it.
o Requirement: USB Bluetooth.
o Output: the MYO and the PC connect or disconnected.
o Post-condition: if it was disconnected we should try again and make sure that MYO arm working.
code (02)
Name: ReadFromMyo
o Input; none .
o Action: System detect the MYO Arm band sent emg signals casting in numbers in array to the PC. o Requirement: USB Bluetooth.
o Output: : List of arrays consists of 8 indices each index horizontally refer to the sensor in myo armband.
o Post-condition: if it wasnt received to PC make sure that MYO arm and PC are connected well, if The PC received the data it will start working on it.

code (03)
Name: LPS
o Input: List of arrays consists of 8 indices every index horizontally refers to sensor in myo .
o Action: System detect the signals as 8 arrays and try to remove the noise from the exist data in array. o Requirement: none.
o Output: the signals are clear without noise.
o Post-condition: if the signals still have noise system will work on it again and check.

code (04)
Name: RMS
o Input: List of arrays consists of 8 indices every index horizontally refers to sensor in myo has a noise.
o Action: System work on signals with feature extraction algorithm to get the important data only.
o Requirement: none.
o Output: System extract the important features from the data .
o Post-condition:

code (05)
Name: Classify
o Input: the extracted features from data.
o Action: System make classification to all of our data to get a class for every movement and the data will be consists of classes.
o Requirement: use the extracted features.
o Output: classifier object.
o Post-condition: if system have any new data from MYO system will use the classifier to get the nearest movement to it.

code (06)

Name: SplitTest
o Input: CSVfile.
o Action: split dataset to make training and testing.
o Requirement: ratio of training more than the ratio of testing.
o Output: the data split to testing and training part.
o Post-condition: none.

code (07)
Name: Testing
o Input: classifier object
o Action: system test the classification to all of our data to get a class for every movement.
o Requirement: the classifier object is trained.
o Output: the dataset classified to movement classes.
o Post-condition: if system have any new data from MYO system will use the classifier to get the nearest movement to it.

code (08)
Name: Notify
o Input: none
o Action: Get notified with new movement the user add it and testing it.
o Requirement: none.
o Output: notification send to the user when the new movement is added.
o Post-condition: System updated.

code (09)
Name: SendClassifiedModel
o Input: classifier Model.
o Action: System sent the data to the arm to do it.
o Requirement: use the 3D unity SDK to do that.
o Output: the action of movement ready to fire on unity.
o Post-condition: try another movement.

code (10)
Name: DrawMove
o Input: the action of movement according to the classification model
o Action: System compare the action to other action on arm handler to know which movement must execute.
o Requirement: use the 3D unity SDK to do that. o Output: the arm will do the movement.
o Post-condition: try another movement.

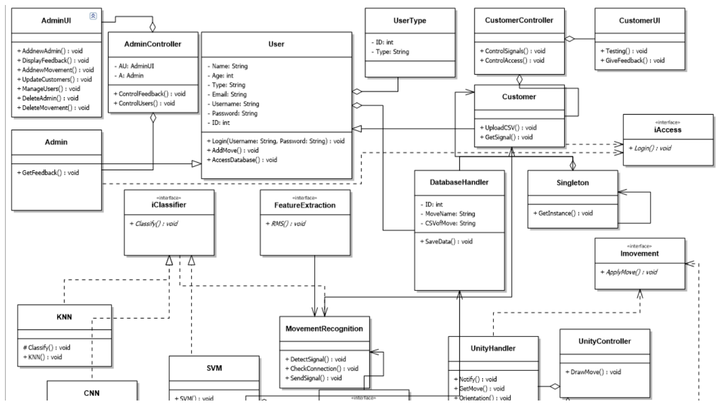

Figure 5: Class Digram

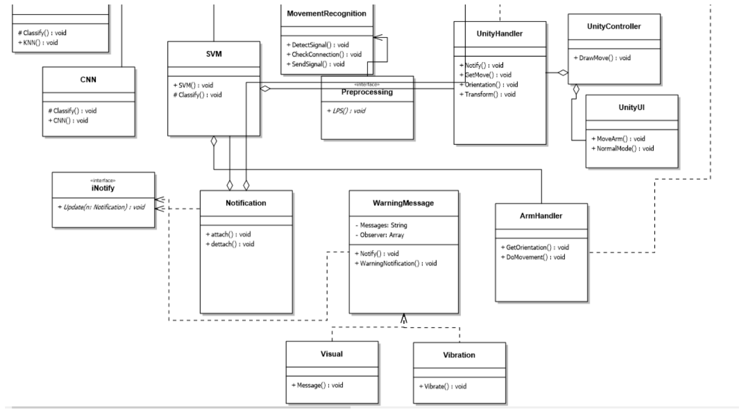

Figure 6: Class Digram

## 3.1 Customer

Code (11)
Name: Signup
o Input: name, password, age, email
o Action: Send user data to the database.
o Requirement: Connected to the server.
o Post-condition:

code (12)
Name: Login
o Input: username, password
o Action: Connect to server and retrieve user data.
o Requirement: Get user name and password from user.

o Output: User session start.
o Post-condition:

code (13)
Name: AddNewMovement
o Input: CSV file
o Action: the user add new movement according to his needs which is specific to him .
o Requirement: MYO arm band connected with the processor unit and record the movement 3 times , by using start , stop and save to control the input signal.
o Output: the movement of this new signal are not existing and the new class will create for these signals.
o Post-condition: future work deep learning.

code (14)
Name: Delete Movement
o Input: specify certain movement
o Action: remove the movement from dataset.
o Requirement: movement is found.
o Output: delete the movements.
o Post-condition: .

code (15)
Name: Sendfeedback
o Input: put subject and description
o Action: submit and send the feedback.
o Requirement: must call uploadCSV bexause it will send logs.
o Output: send the feedback.
o Post-condition: future work deep learning.

code (16)
Name: UploadCSV
o Input: Filename
o Action: Send files to the server.
o Requirement: Connect to server connect send data to it. o Output: User data stored in server.
o Post-condition: Must be logged in with his account.

code (17)
Name: ChooseMode
o Input: select the mode
o Action: the classification is set according to the user selection.
o Requirement: none.
o Post-condition: the mode selected.

## 3.2 Admin

code (18)
Name: AddAdmin
o Input: username, password, email
o Action: Send user data to the server.
o Requirement: Connect to server connect send data to it.
o Output: Admin can connect to our server to handle data set. o Post-condition: Must be logged in with admin account.

code (19)
Name: UpdateAdmin
o Input: username, password, email
o Action: Send user data to the server.
o Requirement: Connect to server connect send data to it. o Output: User data stored in server to login anytime.
o Post-condition: Must be logged in with admin account, and have an account before editing.

code (20)
Name: DeleteAdmin
o Input: Username
o Action: Deletes an admin from the database.
o Requirement: Connect to server connect send data to it. o Output: Admin is deleted.
o Post-condition: Must be logged in with admin account.

code (21)
Name: DeleteMovement
o Input: datasetID, userID.
o Action: Admin can remove records from data set of users.
o Requirement: Connect to server connect send data to it.
o Output: Dataset with specific ID is removed.
o Post-condition: Must be logged in with his account.

code (22)
Name: AverageOfCSV
o Input:
o Action: Take average from signals for individual user to get the best accuracy and take the important data only.
o Requirement: Connect to server connect send data to it. o Output: the data set will be smaller.
o Post-condition: Must be logged in with admin account.

## 4 Interface Requirements

The system interface is going to divided in two main components a Desktop application and unity3d arm which is going to control signals coming from user and the desktop application is going only to manage user files and signals by logging in our system the user will have privileges like deleting his files or uploading new one then he will be able to download his files to train the system based on his profile.

### 4.1 User Interfaces

this is our application prototype.

#### 4.1.1 1

Training window.

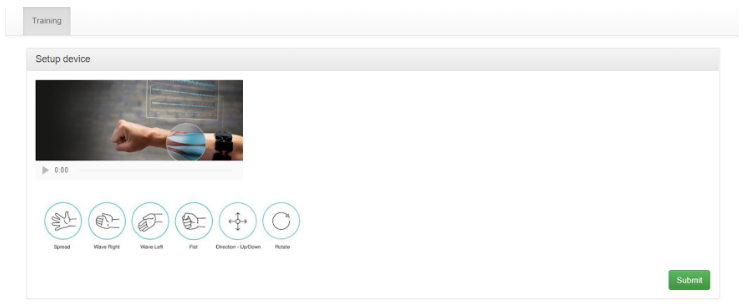

Figure 7: Training

#### 4.1.2 2

Choose mode window.

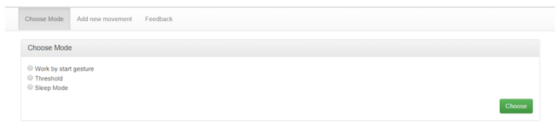

Figure 8: Choose Mode

#### 4.1.3 3

Add new movement.

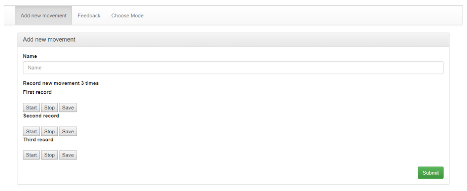

Figure 9: Add Movement

#### 4.1.4 4

Feedback window.

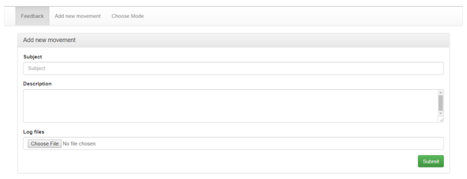

Figure 10: Add feedback

### 4.2 CLI commands:

All we needed from the command line is installing bootstrap libraries using npm install command and the installation of unity3d on Linux system finally installing python and anaconda from the command line.

### 4.3 API :

Using MYO armband API which is some SDKs available on MYO website to download.
Unity SDK.

### 4.4 ROM:

We will use MYO armband device for sending signals from the user to the PC MYO sends 5 files per reading and in each file the MYO sends new signals to each file individually every 15 MS as long as the MYO is running so it will keep on sending signals, the 1st file contain EMG signals ,2nd file contain orientation of the arm, 3rd file contain Eulrorientation, 4th file contain gyro, the 5th file contain accelerometer each file is a CSV file we will read the data then apply noise removing, feature extraction, finally classification.

### 4.5 Hardware Interfaces

### 4.6 Communications Interfaces

USB Bluetooth device.

### 4.7 Software Interfaces

Not available .

## 5 Performance Requirements

All of the computations are made on 8gb ram and i7 1.8GHz, 2.40GHz 64bit Nvidia 740 m the system will not make any other movements until the last movement is made.

### 5.1 standards compliance:

The system will run on 64bit processor.

## 6 Design Constraints

The system will need MYO to send user data for computations.

### 6.1 Application memory usage

TAG: Application Memory Usage
GIST: The amount of Operate System memory occupied by the application.
SCALE: MB.
METER: Observations done from the performance log during testing

MUST: No more than 8 GB.
PLAN: No more than GB
WISH: No more than 16 GB
Operate System: DEFINED: Windows.
MB: DEFINED: Gigabyte

# 7 Other non-functional attributes

## 7.1 Scalability

TAG: System scaling.
GIST:use firebase to Increase the number of movements.
SCALE: The arm should be able to learn new moves from signals to move the arm.
METER: Attempts to get new signals during testing, and learn the new move.

## 7.2 Reliability

TAG: System Reliability
GIST: The reliability of the system.
SCALE: The reliability that the system gives the right result of the movemrnt.
METER: Measurements obtained from 1000 searche about the movement during testing.

## 7.3 Maintainability

TITLE: Application extendibility
DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.
RAT: In order for future functions to be implemented easily to the application.
DEP: none

## 7.4 Performance

TITLE: Application performance
DESC: The application should have high percentage of accuracy.
RAT: In order to increase application performance.
DEP: none

# 8 Preliminary Object-Oriented Domain Analysis

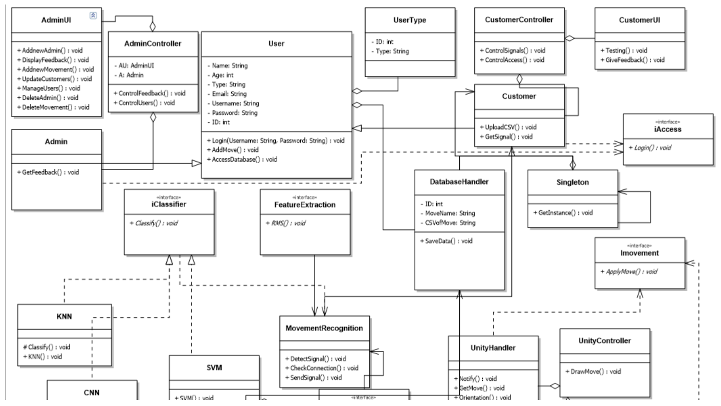

Figure 11: Class Digram

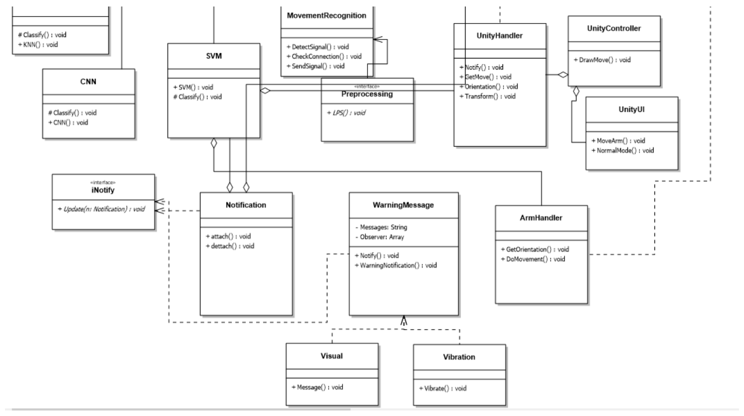

Figure 12: Class Digram

# 9 Operational Scenarios

## 9.1 Scenario for managing project:

The MYO arm band will detect the EMG signals from the User then send it to the Processing unit. the system will filter the signals with noise removal Algorithm then use feature extraction algorithms to get the important data from this signals.finally use classification algorithm to classify the signals to a movement

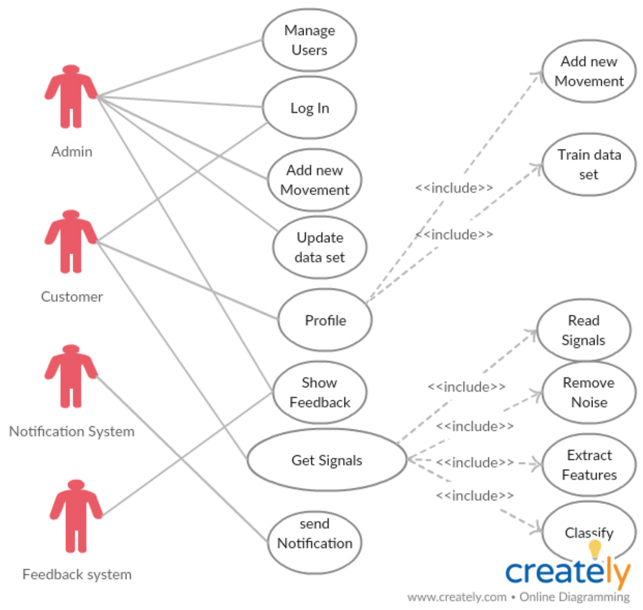

Figure 13: USE CASE

then send it to the 3D Unity to apply it to the arm.

## 10 Refrences

[1] ZieglerGraham K, MacKenzie EJ, Ephraim PL, Travison TG, Brookmeyer R. Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050. Archives of Physical Medicine and Rehabilitation2008;89(3):4229.

[2] Owings M, Kozak LJ, National Center for Health S. Ambulatory and Inpatient Procedures in the United States, 1996. Hyattsville, Md.: U.S. Dept. of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics; 1998.

[3] HCUP Nationwide Inpatient Sample (NIS). Healthcare Cost and Utilization Project (HCUP). Rockville, MD: Agency for Healthcare Research and Quality; 2009.

[4]http://bebionic.com

[5]Khushaba, R.N., Kodagoda, S., Takruri, M. and Dissanayake, G., 2012. Toward improved control of prosthetic fingers using surface electromyogram

(EMG) signals. Expert Systems with Applications, 39(12), pp.10731-10738.

[6] https://www.cyberdyne.jp/english/

[7]Rajavenkatanarayanan, Akilesh, et al. ”MyoDrive: A new way of interacting with mobile devices.” Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments. ACM, 2016.