VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Systems Requirement Specifications

Wearable Smart Device

Version 1.1

Last Updated: 12/3/2019 11:07 AM

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Bachelor of engineering (IT)

ABSTRACT

Author	Nguyen Duy Thanh	e1601122
	Thanh Van	e1601139
	Quynh Le	e1601110
	Quynh Luong	e1601128
	Huy Nguyen	e1601121
	Aapo Koskela	e1401146
	Viet Tran	e160
Title	**Systems Requirement Specifications** for *Wearable Smart Device*	
Year	2019	
Language	English	
Pages	17	
Name of Supervisor	Jukka Matila / Johan Dams	

This project is a part of the course Embedded Systems Programming and Project at Vaasa University of Applied Sciences. The idea was to get the students familiar with the development process of embedded systems and enable them to develop a small full-scale device.

The aim of this project is to design and implement a wearable smart device that calculates and help end user monitor their corresponding health data using sensors, a communication module, an UDOO Neo board as well as an LCD display and user app. The device should be powered by battery and embedded into a comfortable piece of clothing/accessory to accommodate daily-use purposes while still ensuring high data security solutions to protect the end users' personal/sensitive information.

At the end of the project, the system should be completed and satisfy at least 80% of the systems requirements.

Table of Contents

Index of Tables

Index of Figures

Revision History

Revision No.	*Date*	*Description*	*Rationale*
0.1	17/10/2019	Initial Release	Initial Draft of Document
1.0	21/10/2019	Version 1.0	Updated Draft 1
1.1	23/10/2019	Version 1.1	Updated Draft 2

Table 1. Revision History

1. Introduction

This section provides the purpose of this SRS document as well as giving out a scope description and overview of everything regarding the project. Other specifications such as definitions, acronyms, and abbreviations are also included.

1.1 Purpose

The purpose of this systems requirement specifications (v1.0) document is to give a detailed description of the requirements of the user app and establish functional requirements for the **Wearable Smart Device (WSD)** project. It will illustrate the purpose and usage of stated software and hardware device as well as explain system constrains, interface and interactions (between device and app).
This SRS document is primarily served as an interactive online tool to aid the development team in managing and compiling their work and a proposal to a customer for its approval.

1.2 Scope

The *Wearable Smart Device* is, as the name suggests, a piece of wearable accessory that gives end user the ability to monitor and control their health data. The system is embedded with a pedometer sensor to measure count steps, calories burned, distance travelled, speed, temperature and humidity, given user inputs such as height, weight and age. The system is controlled by a UDOO Neo board and accompanied with a free-to-download user-friendly application that collects, sums up and display measured data. Such application should be available on a mobile phone application store.

As stated above, the user can provide personal information such as height, weight and age using the web-portal. An admin can manage user information and monitor app/web-portal functionality.

The software needs Internet connection to fetch and display the measurements.

1.3 Definitions, acronyms and abbreviations

Term	*Definition*
User	A person who owns the device as well as interacts with the application
Admin	The system administrator who is given specific permission for managing and controlling the system
Web-portal	A web application which presents measured data
Application store	An installed application on mobile phone which helps user to find new compatible applications with mobile phone platform and download them from Internet
Pedometer	A sensor that measures count steps, distance travelled, speed, etc.
Sensor	A device, module, machine, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics such as a computer processor

Temperature Sensor	A sensor that measures temperature of environment
Body Temperature Sensor	A sensor that measures temperature of human's body
Humidity Sensor	A sensor that measures humidity of environment
LCD	A screen that display information, which is attached on smart suit
Android App	An application on mobile that use for showing and interact with data

Table 2. Definitions, acronyms and abbreviations

1.4 References

[1] LMT86 2.2-V, SC70/TO-92/TO-92S, Analog Temperature Sensors
[2] Pulse-output Interface Embedded 3D Pedometer Module STP100M
[3] Full-Featured Pedometer Design Realized with 3-Axis Digital Accelerometer

1.5 Overview

From here on, this document contains four chapters including: Overall description, Specific requirements, Prioritization and Release Plan, and lastly the Appendix respectively.

Chapter 2 prioritizes in providing a broad over-view of the system, it's functionality, and interactions with other aspects of the program. Furthermore, this chapter also mentions the constraints of the system and other assumptions regarding the product.

Chapter 3 is the main purpose of this document. It provides the requirements specification in detail as well as the systems' interface(s). Different techniques will be in use to specify each requirement more accurately for a broad range of audience.

Chapter 4 goes in depth into the prioritized requirements. Including explanation behind such chosen methods and discussions regarding alternatives.

Lastly, chapter 5 is the appendix, where the document is concluded, and a release plan is to be worked out.

2. Overall description

This section will give an overview of the entire system. Every detail regarding the project such as interactions between device and application, basic functionalities, constraints and assumptions for the system will be presented.

2.1 Product perspective

This system consists of two part: UDOO micro-processor and Web server. The UDOO micro-processor will receive data from 3 main sensors: Pedometer, Temperature and Humidity, which are attached on the smart suit. The sensors will collect data from the surrounding environment and send it to the UDOO. After calculations are done using specified algorithms, the micro-processor will return the desired information to the user through an LCD, located on the wrist of smart suit. User can easily select between different information /data through means of a physical button on the LCD.

Within UDOO phase, data will be stored inside the memory chip of the board and send to the database server when the UDOO is connected to internet. With UDOO web server, user can add and modify data in the database while the Android App mainly act as a vessel to present the calculated/collected information/data with the addition of several interactive functions that'll most likely be developed in future versions.

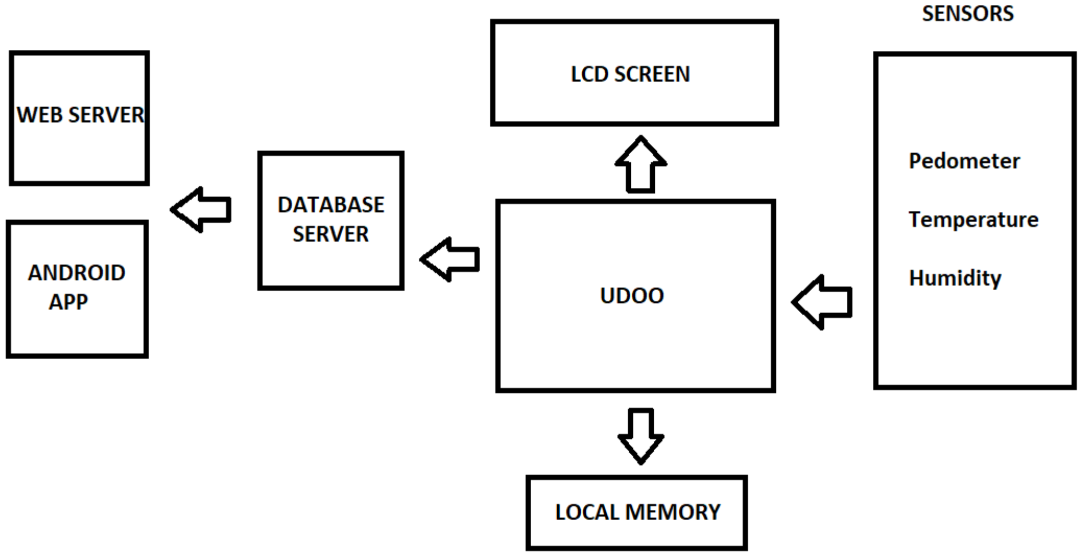

Figure 1. Block diagram

2.2 Product functions

With the sensor's data, UDOO will make required calculations to produce desired outputs like calories burned, distance traveled, and average speed based on the provided input by the user and the step count calculated by the processor. Then give recommendations on workout routines and monitor the process through means of the Android App.

User can access the information in many ways: using built-in LCD, Android App or access web server. While the LCD can only present user information, the Android App provides more control in the sense of managing data. With data retrieved from the server, the App gives user advice about workout lessons, predictions of user habit and monitor the whole process. Webserver provides full access and control over user data and should be able to synchronize with other 3rd party services such as Google Health. It will also provide information regarding the system, E.g. show when there is a new update or special bug fixes, hot patch, etc.

2.3 User Characteristics

There are two types of users that interact with the system: owner of the product and administrator. Each of these two types of users has different use of the system so each of them has their own requirements.

The owners of the product (the "customer") can access all three platforms: built-in LCD, mobile app and the webserver. Each platform with distinct advantage(s):

- LCD: ease of access, show daily data.
- Mobile app: interactive, has more functionalities, can hold data up to a whole week.
- Webserver: total control over data, system notification, logs every and all data recorded over a long period of time.

The administrators, on the other hand, only interact with the webserver. They manage the whole system, making sure everything is working in order and is responsible of making changes wherever/whenever needed.

2.4 Constraints

The calculations done by the micro-processor is heavily constrained by the user input. As there is no easy way to precisely measure aspects of the user body's characteristics (for example 2 people with the same "weight" but one comprises of mainly fat and the other muscle), the system can only produce on-average predictions.

The Internet connection is one of the biggest constraints for connection between mobile application and UDOO through database server. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

Both the webserver and the mobile application will be constrained by the capacity of the database. Since the database is shared between both application it may be forced to queue incoming requests and therefore increase the time it takes to fetch data

2.5 Assumptions and dependencies

One assumption about the product is that every customer can use the product through different platforms. Even though there will be instructions in the manual, there will be different people from different backgrounds who might find it difficult to fully utilize every feature of the product and therefore have a less great experience.

Another assumption is that every user's mobile phone is compatible with the mobile app. Since the project is still in the making, there are many constraints regarding it's compatibility that cannot be dealt with in the current release. And even when the app is compatible with the users' phone, there is no guarantee that the mobile device has enough performance. If these devices do not have enough hardware resources to accommodate the application, there might be cases where the app under performs or maybe even not work at all.

2.6 Apportioning of requirements

In the case that the project is delayed, there are some requirements that could be transferred to the next version of the application. Those requirements are to be developed in the second release.

3. Specific requirements

This section contains the functional and quality requirements of the system. It gives a detailed description of the system and all its features accompanied with visual illustrations.

3.1 External interface Requirements

This section provides a detailed description of the systems inputs and outputs. It also gives a description of the hardware, software, communication interface and basic prototypes for the user interface.

3.1.1 User interfaces

A first-time user of the mobile application should see the log-in page when they open the application as shown in figure 2. After logging in for the first time, the user must give their information as input for later calculations (figure 3). For the main feature of the application, the calculated result done by the UDOO should be shown as in figure 4. User can interact with their information, trade or simply forward data with/to 3rd party services such as Google Health, get information about weather from Weather, predicting user habits, generating tips and advice for exercise routines.

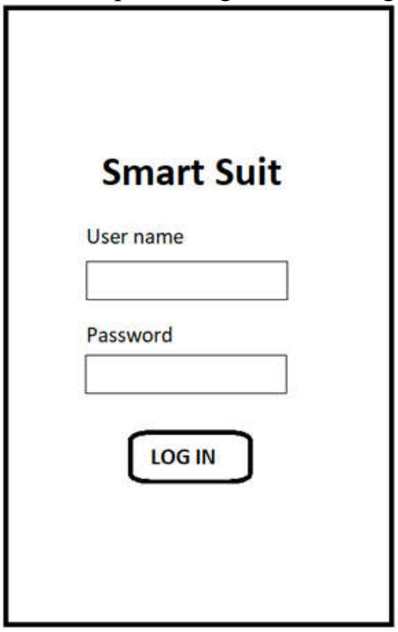

Figure 2. Log-in page

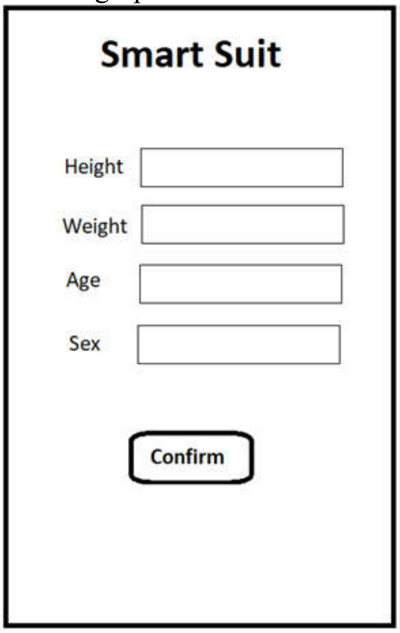

Figure 3. User input

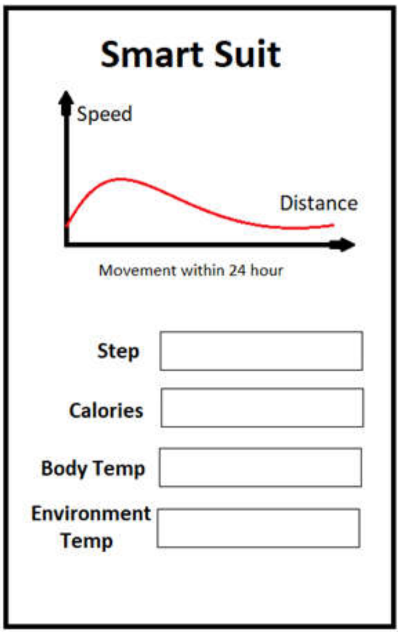

Figure 4. App feature

For web server, not only the information that has been shown but every information recorded in database server must be available so user can access their data with ease while still having full management over those data.

And of course, user can also get information through the built-in LCD. The screen will show each information/data separately as demonstrated in the figure below; different information/data can be switched between each other using the physical button on the LCD.

Figure 5. LCD screen showing the temperature

3.1.2 Hardware interfaces

The whole hardware system is embedded onto the smart suits that was designed to be both trendy and comfortable. The hardware consists of 3 main parts: input components, output components and the UDOO board.

Input components

- STP100M 3D pedometer: a functional module which has adopted the pulse-type interface, with the 3D MEMS sensor (G sensor) and high precision of 3D pedometer algorithm, it can give a precisely pedometer in any direction.

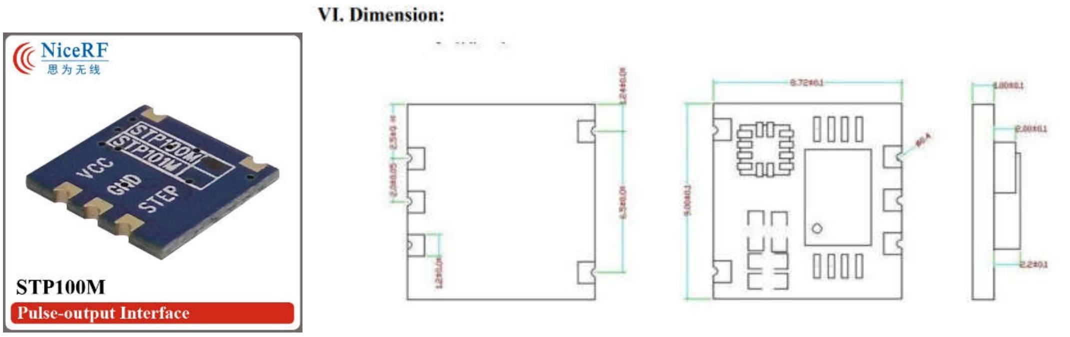

Figure 6. STP100M pedometer

- LMT86: a precision analog output CMOS integrated-circuit temperature sensor that operates at a supply voltage as low as 2.2 V, the wide temperature range of 50°C to 150°C.

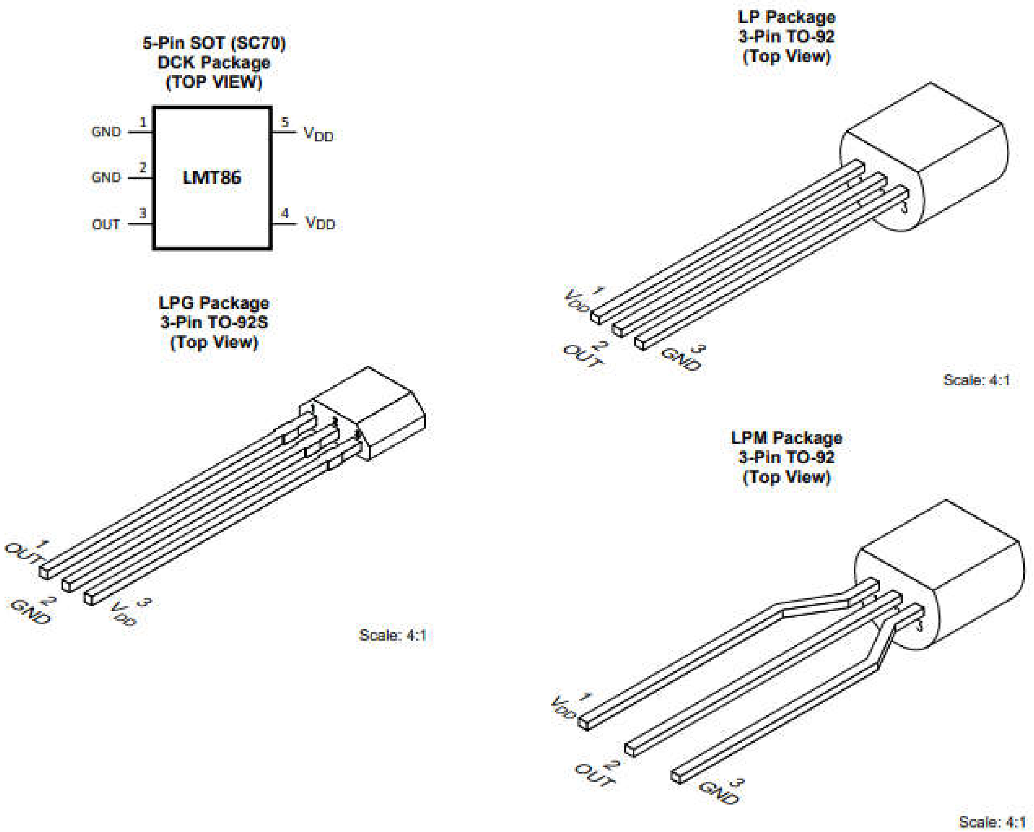

Figure 7. LMT86

- Humidity sensor:

Output components

- LCD 1602: a kind of dot matrix module to show letters, numbers, and characters and so on. It's composed of 5x7 or 5x11 dot matrix positions; each position can display one character. There's a dot pitch between two characters and a space between lines, thus separating characters and lines. The model 1602 means it displays 2 lines of 16 characters.

Figure 8. LCD 1602

- REES52 I2C module: a module which helps LCD displays with I2C Interface, which means that fewer pins are necessary to use this product than would be needed with a regular 16x2 LCD Display: VCC, GND, SDA and SCL. Each I2C bus consists of two signals: SCL and SDA. SCL is the clock signal, and SDA is the data signal. The clock signal is always generated by the current bus master; some slave devices may force the clock low at times to delay the master sending more data (or to require more time to prepare data before the master attempts to clock it out).

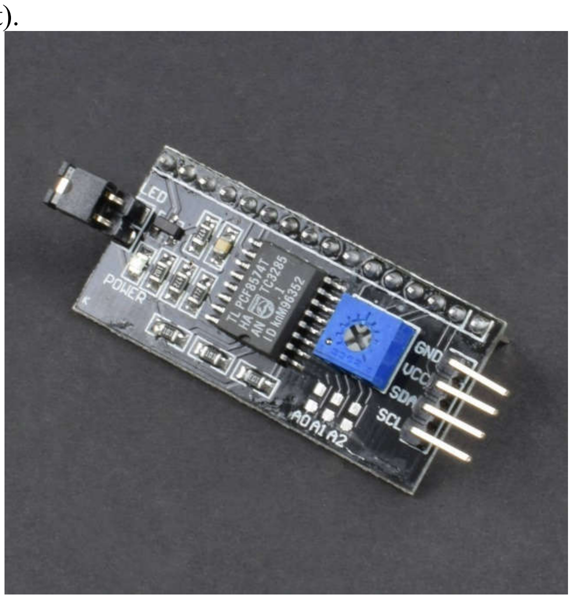

Figure 9. REES52 I2C module

Mainboard

- UDOO Neo: a powerful all-in-one open hardware low-cost computer equipped with a NXP™ i.MX 6SoloX applications processor for Android and Linux. UDOO NEO embeds two cores on the same processor: a powerful 1GHz ARM® Cortex-A9, and an ARM Cortex-M4 I/O real-time co-processor that can run up to 200Mhz.

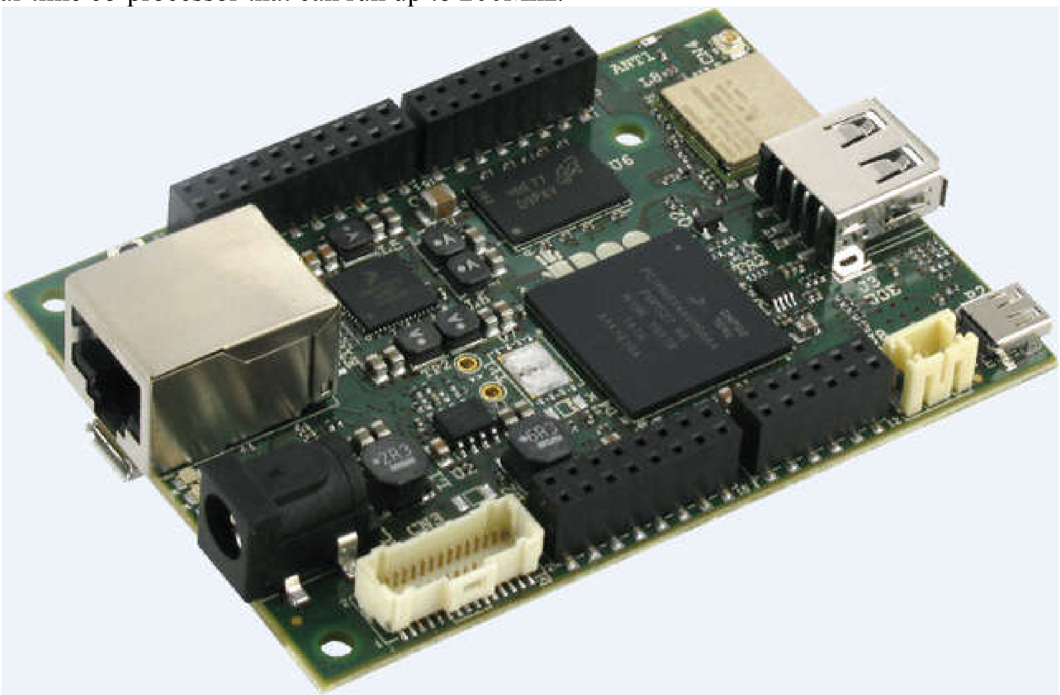

Figure 10. UDOO Neo board

3.1.3 Software interfaces

The software application will be used as an interactive method for end user during the time they wear the smart device. The purpose of this software application development is to display end users' data in real time therefore helping them better monitor and keep track of those information.

The execution process can be divided into 3 steps.

- Step 1: First, the software application will establish a connection between the UDOO and the sensors -> read/retrieve corresponding data.
- Step 2: Then, the data will be stored inside a database (in this case we will use MySQL) and categorized in different columns for ease of management.
- Step 3: After data has been sorted and stored inside MySQL database, we will start implementing the web server and deliver these data to end user.

3.1.4 Communications interfaces

The communication between hardware and software application is Wi-Fi. The reason for using Wi-Fi as a communication protocol is that it allows user to access network resource conveniently, easy to be expanded and to be deployed.

3.2 Functional requirements

This section includes the requirements that specify all the fundamental actions of the software part of the system.

ID: FR1
TITLE: Initial Setting - User personal information

DESC: At the first time of using, user is requested to enter some information related to his/her gender, his/her weight, height and age. This information is helpful for calculating how many calories he/she has burned after his/her journey.
RAT: In order to calculate the number of calories burned.
DEP: <**Null**>

ID: FR2
TITLE: View trip Information
DESC: After user completes his trip – button is pressed and the device runs in Stop mode – he is able to know information about his trip, including how many steps he traveled, how many calories has been burned, how far he walked and how fast of his speed. They all are displayed in LCD display or there is a possibility to check that in webpage
RAT: For user to read all information related to his/her trip
DEP: <**Null**>

ID: FR3
TITLE: Check body information
DESC: A user should be able to check his body temperature and humidity information. There will be six temperature sensors and one humidity sensor to collect those and give him a nice view of his body information.
RAT: For user to check his body temperature and humidity information
DEP: <**Null**>

ID: FR4
TITLE: Information display
DESC: There are two possibilities for user to read data from device: LCD screen and web page. LCD is a convenient for him if he would need check information during his trip. When a trip is finished and a network connection is established, data can be sent and displayed on a web page.
RAT: For user to read data during and after a trip
DEP: <**Null**>

ID: FR5
TITLE: Start-up system
DESC: User can start system up by pressing button. Once the system is on, all trackers for time, step counter, temperature and humidity begin to work.
RAT: In order to start device
DEP: <**Null**>

ID: FR6
TITLE: Stop system
DESC: User stops the system and put it into Sleep mode by pressing a button. All information linked to his body information and his day trip are saved into device's memory. If there is network connection available, those data are sent to web server and displayed on a web page.
RAT: In order to stop device
DEP: <**Null**>

ID: FR7
TITLE: Built in button on LCD screen
DESC: Users can switch between data value and choose working mode for the smart wearable system by using build in button on the screen.
RAT: In order to give user an interface for choosing function and viewing data in catalog
DEP:<**Null**>

ID: FR8

TITLE: Change display screen
DESC: Display separately function and data for specific purpose
RAT: In order to choose and switch between functions
DEP: <**Null**>

ID: FR9
TITLE: Step Counting
DESC: The amounts of step a user has walked is counted since the button is pressed (“Start” mode) to a moment the button is pressed again (“Stop” mode). The system is able to catch user’s movement at maximum rate of three steps in one second.
RAT: In order to count steps of user’s movement
DEP: (Button function)

ID: FR10
TITLE: Get Distance info
DESC: The whole distance a user has traveled might be calculated by the number of user’s steps multiplies by an average distance of one user’s step, which can be defined by user at initial device’s settings
RAT: In order to get distance information of user’s journey
DEP: (Step Counting function)

ID: FR11
TITLE: Get Calories info
DESC: The amounts of calories are burned in complete user’s journey will be calculated based on how many steps he has walked at all.
RAT: In order to get information of calories-burnt amount
DEP: (Step Counting function)

ID: FR12
TITLE: Get Speed info
DESC: The user should be known his average speed in a whole trip. A time tracker is trigged from his starting point to an end point and is divided by total distance of his trip.
RAT: In order to calculate average speed of user
DEP: (Step Counting function)

ID: FR13
TITLE: Human body temperature and humidity measurement
DESC: User’s body temperature and humidity is measured by placing sensors in six various places in user body. It helps to collect temperature and humidity of different points at same time, and then give average values at the result.
RAT: In order to return user body temperature and humidity
DEP: (Button function)

3.3 Performance requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

3.3.1 Built-in LCD

ID: PR1
TITLE: Clear display
DESC: The screen should be able to show the data clearly
RAT: So the user can easily track their data while using the device
DEP: <**Null**>

ID: PR2
TITLE: Functional button
DESC: The physical button located on the LCD should be functional
RAT: So the user can easily switch data display
DEP: <**Null**>

ID: PR3
TITLE: Water resistant
DESC: The LCD should be water resistant in order to withstand the harsh conditions while the user is exercising.
RAT: So the product can last longer
DEP: <**Null**>

3.3.2 Mobile Application

ID: PR4
TITLE: Working log-in page
DESC: The log-in page should function properly for the user to access the features of the app
RAT: So the user can use the app properly
DEP: <**Null**>

ID: PR5
TITLE: Working user input page
DESC: The user input page should function properly for the user to access the features of the app
RAT: So the user can use the app properly
DEP: <**Null**>

ID: PR6
TITLE: Working app features
DESC: features of the application should work properly
RAT: So the user can use the app properly
DEP: <**Null**>

3.3.3 Web server

ID: PR7
TITLE: Data storage
DESC: The web server should be able to log all measured data for the user to fully utilize and manage their data
RAT: So the user can manage their data properly
DEP: <**Null**>

ID: PR7
TITLE: System notification
DESC: The web server should be able to show notifications regarding the product such as new updates etc.
RAT: So the user can manage their data properly
DEP: <**Null**>

ID: PR7
TITLE: Security
DESC: The web server should be secured
RAT: So the user can protect their data
DEP: <**Null**>

ID: PR8
TITLE: Accessibility
DESC: The web server should be easily accessible, especially by the admins so they can monitor and manage the system.
RAT: So the admin(s) can manage the system
DEP: <**Null**>

3.4 Design constraints

Because the product is expected to be used by anyone, there are many scenarios that may pose as a design constraint. For the product to be made, certain constrains had to be overlooked, but only to a moderate amount so that it can cover a large range of user.

The biggest constraint right now is that the product is an integrated system permanently attached to a specific piece of clothing. Of course, this is only for now. In future versions, the product should be able to integrate to any existing piece of clothing, making it more convenient for the users. Though with the current version, this constraint poses a huge problem because even with unisex design, there are still too many different sizes to cover from XS to XXL (in EU measurements). And because everybody is bit different, many users might not find the product that perfectly fits their body type, therefore significantly dropping the value of the product. Furthermore, the weight of the product might also be a major design constraint as some might find the integrated batteries too heavy but reducing the size of the batteries might lower the performance of the product overall.

Another big constraint is the need for WLAN internet at home or common living place. This might not seem too big of a problem, but it still prevents some e.g. older customers from buying the product because they would need to invest extra on the installation of Internet. The user interface for server-side use is not dependent on operating system because it will be web/browser based, and therefore can be used in almost all cases.

Finally, is the problem of mobile application. There are dozens of operating systems for phones, and we can't support all of them. Thus, we decided to choose Android as our operating system. This time and age, Android has 76% of market share in mobile OS market. There were also compelling reasons to choose iOS, even though it only has 22% of market share. The balance between these are closer to each other when we look at in the current market.

3.5 Software system attributes

For the whole system to run properly it is crucial for the software must reach certain criteria, such as:

- Reliability: the system is expected to return the right measurements at least 98% of the time.
- Availability: every feature of the product that's available to the end user should be up and running at least 98% of the time; every feature regarding the admin's rights should be running at least 99% of the time (in order for them to manage the system properly).
- Security: the system should always be secured to protect the users' data/information.
- Maintainability: the software should be written in such ways that would support implementations of new functions.

4. Prioritization and Release Plan

For any project to be successful, there needs to be a working plan. Because this project is part of the University's course, progress must keep up with the predetermined schedule made by the supervisor. There are three main deadlines: first prototype, second prototype and lastly final product launch.

The free and open source, web-based project management and issue tracking tool Redmine was used to aid team members in managing tasks/events, sharing and compiling work. Different importance level can also be set for each task. Redmine is a tool provided by VAMK.

4.1 Prioritization method

Redmine's prioritization levels were used when creating new task/event. That way the more important ones can be dealt with first. Of course, there were cases where changes had to be made to the prioritization level after but over-all it is a convenient, simple and effective tool.

Levels of priority (high to low): Immediate -> High -> Normal.

Immediate: This level was set for one thing only: documentation. This is due to the not-so-appealing nature of writing reports dissuading members from taking it seriously. By setting it's importance level higher, members of the development team was more inclined to get the document done.

High: Any task/event that had to do with the core idea of the product was set to high priority. Because without having all these tasks done, the product simply can't work.

Normal: Any task/event that only act as an update/upgrade to the already working product was set to normal priority since they're not directly responsible for the make-or-break of the whole system.

4.2 Release plan

Release plan will be in accordance with the supervisors' set dates and requirements: 2 prototypes and 1 final design. These prototypes should consist of every function made final by the whole team during the planning phase.

First release is the prototype 1, which should be the working device, only lacking certain features or sensors. It is also not integrated with any clothing and is just the barebone version. Basic functions can be clearly seen/demonstrated, and the final product should be developed on the baseline of this device.

Second release is the prototype 2, which should have more exciting features/functions that would up the price value of the product. In this release, the system should be integrated onto some clothing. All the functions/features in this design should be final and only some necessary adjustments are required to reach the quality of the final product.

5. Appendix

<NULL>

<to be updated in future release>