10th of February

System Requirements Specification

Hiking Tour Assistant

Nguyen Thanh Tuyet Han Miguel García González Pyry Seuranen

Group A

Table of Contents

[1. Introduction 3](#_Toc127966398)

[1.1. Purpose 3](#_Toc127966399)

[1.2. Scope 3](#_Toc127966400)

[1.3. Definitions, acronyms, and abbreviation 3](#_Toc127966401)

[1.4. References 3](#_Toc127966402)

[1.5. Overview 4](#_Toc127966403)

[2. Overall description 5](#_Toc127966404)

[2.1. Product perspective 5](#_Toc127966405)

[2.1.1. System interfaces 5](#_Toc127966406)

[2.1.2. User interfaces 5](#_Toc127966407)

[2.1.3. Hardware interfaces 5](#_Toc127966408)

[2.1.4. Software interfaces 5](#_Toc127966409)

[2.1.5. Communications interfaces 6](#_Toc127966410)

[2.1.6. Memory 6](#_Toc127966411)

[2.1.7. Operations 6](#_Toc127966412)

[2.1.8. Site adaptation requirement 6](#_Toc127966413)

[2.1.9. Context diagram 6](#_Toc127966414)

[2.2. Product functions 8](#_Toc127966415)

[2.2.1. Start Stop Hiking Section 8](#_Toc127966416)

[2.2.2. Record data related to hiking activity. 8](#_Toc127966417)

[2.2.3. Display data on smartwatch screen 8](#_Toc127966418)

[2.2.4. Synchronize and store data with Raspberry Pi 8](#_Toc127966419)

[2.2.5. Provide Web UI 9](#_Toc127966420)

[2.2.6. Basic display functions 9](#_Toc127966421)

[2.2.7. Re-calculate exact calories and distance 9](#_Toc127966422)

[2.3. User characteristics 9](#_Toc127966423)

[2.4. Constraints 9](#_Toc127966424)

[2.5. Assumptions and dependencies 9](#_Toc127966425)

[3. Specific Requirements 10](#_Toc127966426)

[3.1. External interfaces 10](#_Toc127966427)

[3.2. Functions 10](#_Toc127966428)

[3.2.1. Use cases. 10](#_Toc127966429)

[3.2.2. Use cases diagrams. 12](#_Toc127966430)

[3.2.3. Data flow diagram. 13](#_Toc127966431)

[3.3. Performance requirements 13](#_Toc127966432)

[3.4. Logical database requirements 13](#_Toc127966433)

[3.5. Design constraints 13](#_Toc127966434)

[3.6. Standards compliance 14](#_Toc127966435)

[3.7. Software system attributes 14](#_Toc127966436)

[3.7.1. Reliability 14](#_Toc127966437)

[3.7.2. Availability 14](#_Toc127966438)

[3.7.3. Security 14](#_Toc127966439)

[3.7.4. Maintainability 14](#_Toc127966440)

[3.7.5. Portability 15](#_Toc127966441)

[3.8. Additional comments 15](#_Toc127966442)

# Introduction

## Purpose

The purpose of this Software Requirements Specifications (SRS) is to define the nature, functions, requirements and constraints of a Smart Wristwatch system designed to assist hikers following the guidance of IEEE Std 830-1998 recommended practice for SRS.

The document is intended for the producers, developers, stakeholders and the customers to gain full comprehension of the device.

## Scope

The software product, to which this SRS refers to, is a Hiking Tour assistant. The product is designed to be capable of the following functions:

* Starting and stopping hiking sessions.
* Recording the number of steps taken during the session and convert it into travelled distance.
* Displaying the steps/distance/calories burned/total hiking time on the smartwatch screen.
* Synchronizing and storing the data with RB400 via Bluetooth.
* Having a web app to show the session statistics including the earlier session.

## Definitions, acronyms, and abbreviation

SRS – Software Requirements Specifications.

UI – User Interface.

RTC – Real-Time Clock

IMU - Inertial Measurement Unit.

CSV – Comma-Separated Values.

IDE – Integrated Development Environment.

LLSW – LILYGO® TTGO T-WATCH 2020 V2.

RB400 – RaspBerry Pi 400.

API – Application Programming Interface.

## References

IEEE std 830-1998 Recommended Practice for Software Requirements Specification.

Arduino IDE - <https://www.arduino.cc/en/software.>

LiLyGo Twatch + ESP32 Library - [https://github.com/Xinyuan-](https://github.com/Xinyuan-LilyGO/TTGO_TWatch_Library)LilyGO/TTGO\_TWatch\_Library

Web Bluetooth communication - <https://developer.chrome.com/articles/bluetooth/.>

Calorie burn formula - <https://www.omnicalculator.com/sports/calories-burned.>

## Overview

The rest of this SRS consists of the following in a sequence:

Section 2 gives an overall idea of the product.

Section 3 gives specific requirements of the product.

# Overall description

## Product perspective

### System interfaces

The system will use LLSW and RB400 as main interfaces. LLSW will collect all the data from the user, and RB400 will obtain LLSW data connecting via Bluetooth 4.0, displaying it on a web UI.

### User interfaces

#### Watch User

The LLSW shall provide Start/Stop button which can directly control via the screen.

#### Web UI User

The Web UI shall ask for the user to connect to its device by clicking a button, showing the hiking data in the webpage after the connection is successfully done, or showing an error message in other case.

### Hardware interfaces

The LLSW hardware shall provide interfaces from step counter, RTC, Bluetooth, and Wi-Fi connection.

For the step count function, the watch shall obtain data from IMU sensor, which is BMA432 model. This sensor contains a three-axis accelerometer which allows tracking running/ walking/still detection.

For RTC function, the watch shall obtain data from Real Time Clock, which is PCF8563.

The model of touchscreen shall be FT6336, which has a resolution of 240x240.

The LLSW shall provide V4.2+ Bluetooth connection and 802.11 Wi-Fi connection.

Raspberry PI 400 (RB400) hardware will run the web UI for the user to connect its LLSW and be able to see the data collected from a web browser.

### Software interfaces

The operating system of RB400 is Raspbian OS, which is Debian v11.

JavaScript ES6+, HTML5, and Bootstrap 4 will be used for the web UI implementation.

Web Bluetooth API will be used in the web UI to implement Bluetooth connection.

The operating system of LLSW shall be FreeRTOS.

The library used for smartwatch development shall be LiLyGo TWatch Library.

### Communications interfaces

The LLSW shall communicate with Raspberry Pi via Bluetooth 4.2+ and/or Wifi connection.

The data type shall be under text form.

The LLSW shall send all hiking data every 5 minutes to RB400 and the RB400 shall preserve the connection to LLSW until the user decides to end the connection from the web UI.

### Memory

The web UI will require a basic memory amount of 40MB of RAM to be run from RB400.

### Operations

Start and stop hiking sessions from LLSW and obtaining data via Bluetooth from the web UI running on RB400.

### Site adaptation requirement

NA

### Context diagram

At first, two parts of the system will be treated as separate. Firstly, there is LLSW, then there is RB400.

#### LLSW

LLSW – LilyGo smarwatch

BC – Bluetooth Control Module

SM – Statistics Module

WD – Watch Display Module

Diagrama

Descripción generada automáticamente

Figure 1: LLSW context diagram

#### RB400

RB400 – RaspBerry 400, stationary device.

WUI – Web UI app that will display the data.

BAPI – Web Bluetooth API, that will connect to LLSW and will retrieve the data.

Diagrama

Descripción generada automáticamente

Figure 2: RB400 context diagram

#### Hiking Tour Assistant

This is the global context diagram, where LLSW and RB400 connect and interact.

HTA – Hiking Tour Assistant, the global system

Diagrama

Descripción generada automáticamente

Figure 3: Hiking Tour Assistant context diagram

## Product functions

The system performs the following functions.

### Start Stop Hiking Section

This function allows the user to Start Hiking Section on the smartwatch. When the user starts a new section, all data relating to the old hiking section will be reset to zero. When the user stops the section, all counting and calculating activities will stop; however, all data is still stored in the watch.

### Record data related to hiking activity.

During hiking section, user steps will be recorded and transform to distance by apply this formula:

The total time of hiking section also be recorded and transform to burned calories by apply this formula:

The above formula depends on user characteristics.

### Display data on smartwatch screen

This function displays all data to smartwatch screen including total step, distance, burned calories, and total hiking time. When the watch initializes, all data will be set to zero.

### Synchronize and store data with Raspberry Pi

The smartwatch will send the latest hiking section data via Bluetooth and/or Wifi to RB400. The watch will send data every five minutes.

RB400 will receive data and display it via web UI.

### Provide Web UI

This function will display all saved data on web UI running on RB400. Moreover, the web also allows users to enter their height and weight. Those parameters will be used by the web UI running on RB400 to provide more precise data about the user.

### Basic display functions

These functions will display some basic information: Current Time, Battery (in percentage and icon) of watch.

### Re-calculate exact calories and distance

This function will re-calculate the distance and calories of users based on their height and weight when the user decides to enter them in the web UI.

## User characteristics

The users of this application are assigned following characters:

- Educational level: All.

- Technical Expertise: Not required.

- Age range: 15-80.

- Have average height around 175cm and weight 75kg.

## Constraints

Use RB400 as machine to run the web UI and connect with LLSW.

Use LLSW as smartwatch to measure and collect all the hiking data generated by the user.

## Assumptions and dependencies

RB400 has Google Chrome web browser and counts with at least 40 MB of RAM.

LLSW counts with the needed sensors to measure statistics.

# Specific Requirements

## External interfaces

The system takes input from the screen of LLSW, mouse, keyboard, and files in the memory. The system displays on the web UI and on the watch screen Hiking Data including Step, Distance, Time, Calories.

## Functions

### Use cases.

#### Start Hiking Section

**Actor(s)**: User with LLSW.

**Precondition(s):** None.

**Trigger:** Click Start.

**Procedure:**

1. User on the Watch UI click “Start”.

2. System change title of button to “Stop”.

3. System reset all the variables related to previous sections.

4. System starts recording the steps and hiking time.

5. System calculates distance and calories based on the steps and hiking time.

6. System displays all current data on screen.

**Post condition(s):** None.

**Exception(s):** The hiking section has already started.

#### Stop Hiking Section

**Actor(s):** User with LLSW.

**Precondition(s):** None.

**Trigger:** Click Stop.

**Procedure:**

1. User on the Watch UI click “Stop”.

2. System change title of button to “Start”.

3. System stop recording the steps and hiking time.

4. System still displays all current data on screen.

5. System updates the steps to the main bar.

**Post condition(s):** None.

**Exception(s):** The hiking section has already stopped.

#### Access web UI

**Actor(s):** User with LLSW and web browser

**Precondition(s):** Web browser running.

**Trigger:** introduce URL to access web UI

**Procedure:**

1. User writes the URL of the web UI in the browser.

2. Browser loads the webpage.

3. The web UI initial screen is displayed.

**Post condition(s):** the web UI is displayed and running.

**Exception(s):** The URL is not available/the web UI does not load.

#### Display hiking data.

**Actor(s):** User with LLSW and web UI

**Precondition(s):** web UI accessed from the browser.

**Trigger:** User clicks on “Connect to watch”

**Procedure:**

1. User on the web UI click “Connect to watch”.

2. The browser shows the devices available.

3. User clicks on the device to connect and confirms connection.

4. Web UI displays the hiking data sent via Bluetooth.

**Post condition(s):** web UI displaying statistics.

**Exception(s):** The connection is not successful, and no data is displayed.

### Use cases diagrams.

#### Start/Stop Hiking Section

Diagram

Description automatically generated

Figure 4: Start Stop Hiking Section

#### Display Hiking Data in Web UI

Diagram

Description automatically generated

Figure 5: Display Hiking Data in Web UI

### Data flow diagram.

Diagram

Description automatically generated

Figure 6: Data Flow Diagram

## Performance requirements

**Static numerical requirements:**

* The number of terminals shall be two, the LLSW itself and web UI used to view the data in detail.
* The product, being a smartwatch, shall support only one user when worn.
* Web UI shall support the connection of one LLSW at a time.

**Dynamic numerical requirements:**

* The data from the LLSW to the RB400 shall be transferred every five minutes.
* The connection via Bluetooth with LLSW shall be successful 90% of the time.
* The watch shall update data on screen at least once per five seconds.
* The step counter shall have over 90% accuracy.
* The time that LLSW displays shall not differ from the actual time by more than five minutes.
* The data shall be sent via Bluetooth per five minutes ± one minute.

## Logical database requirements

NA

## Design constraints

The limitations of this project are the following:

* The available hardware is RB400 and the LLSW.
* The hardware itself cannot be modified or tinkered with, outside of adding 3D-printed parts or hardware modules, both of which must be detachable.
* The operating environment is restricted around the Arduino IDE-based program of the watch and the interface of RB400.

## Standards compliance

NA

## Software system attributes

### Reliability

* + The LLSW delivers data via Bluetooth every five minutes ± one minute.
  + Web UI always admits connection with LLSW.
  + The user gives the exact weight and height variables for the precise evaluation of burned calories.
  + The dynamic performance requirements ensure that the connection and measurement accuracy are over 90 %.

### Availability

* + The software has a start/stop functions for restarting the process.
  + Web UI will be running if the stationary device platform is running (RB400 is running).
  + The LLSW data on the screen will be updated regularly according to the dynamic performance requirements.

### Security

* + The last hiking session is saved to Raspberry.
  + The data is restricted to between the LLSW and Web UI.

### Maintainability

A detailed manual for the project will be available, explaining functionalities, and assigning specific parts of the code of the project to those functionalities.

The software is well documented for the purpose of maintenance for software developers, users, and management.

### Portability

* + The programming is done with Arduino IDE.
  + The used libraries, LiLyGo and ESP32, are openly available.
  + The software is otherwise designed for LLSW only.
  + Web UI is developed with JavaScript ES6+, HTML5 and Bootstrap 4.
  + Bluetooth connection with LLSW with Web Bluetooth API.

## Additional comments

NA