

# Hiking Band: Software Requirements Specification

**Course:** ELEC-E8408 Embedded Systems Development, Aalto University

**Members:** Holappa, Heidi & Lundén, Jaakko-Juhani & Rislakki, Tuomas

**Group:** I

## Table of Contents

1. Introduction
  - i. Purpose
  - ii. Scope
  - iii. Definitions, acronyms, abbreviations
  - iv. References
  - v. Context Diagram and Overview
2. Overall description
  - i. Product Perspective
  - ii. Product functions
  - iii. User characteristics
  - iv. Constraints
  - v. Assumptions and dependencies
  - vi. Apportioning of requirements
3. Specific Requirements
  - i. Performance requirements
  - ii. Design constraints
  - iii. Software-system attribtues

# 1 Introduction

This chapter highlights the purpose and scope of this document, introduces the essential definitions, acronyms and abbreviations and the relevant references.

## 1.1 Purpose

The purpose of this document is to list and provide context and considerations for the Hiking Band product system. This document **SHOULD** be used within software development, integration and testing of the whole system.

The core audiences for this documents are the product development team and the course instructors, who both review this document and review the final system in based on the requirement specification of this document. The final product **MUST** match this documentation.

## 1.2 Scope

The LilyGo Hiking band system includes a LilyGO T-Watch smartwatch application for recording hiking trips and a separate application for storing trip information into persistent storage and for displaying trip information on a Web Application.

The system consists of two hardware components: the Raspberry Pi and the LilyGo watch. The system includes interactable interfaces, hardware, protocols and functionalities which are required for the system to function. This document defines the requirements in well detail to give a broad understanding of the product and its use case.

## 1.3 Definitions, acronyms and abbreviations

**1.user:** the person, or persons, who operate or interact directly with the product.

**2.hiking session:** an event during which activity data such as step count, travelled distance and average speed are recorded on the LilyGO smartwatch. The event begins when user pushes “Start” button and ends when user presses “End” button in hiking session view

**3.step:** An act or movement of putting one leg in front of the other in walking or running

**4.step count:** An approximation of taken steps computed by a step algorithm based on BMA423 accelerometer data

**5.distance:** An approximation of travelled distance that **MUST** be based either on an approximation of the length of an average step or on GPS data

**6.duration:** A count of seconds from the moment counting steps began, based on the RTC module of LilyGO

**7.average speed:** An average based on the duration of the hiking session and travelled distance

**8.burned calories:** an approximation of calories burned. The approximation **MUST** be based on count of steps and **MAY** additionally be based on average speed during the trip.

## 1.4 Overview

### 1.4.1 Abbreviations and Definitions

Abbreviation or Word	Definition
System	The whole system including Rpi, LilyGo and relevant interfaces
Hiking Band System	System
RPi	Raspberry Pi V3+
LilyGo	Lilygo T-Watch V2
LilyGo application	The firmware and applications running on the LilyGo
Web application	The Web-based application running on the RPi

### 1.4.2 Unites of measure

The step

### 1.4.3 Keywords

The specification follows the requirement level keywords defined in [RFC-2119](#):

Keyword	Description for the specification
MUST	Absolute requirement for the specification
MUST NOT	Absolute prohibition of the specification
SHOULD	Can be ignored with valid reasons
SHOULD NOT	Can be implemented with valid reasons
MAY	Optional, extra

## 1.5 References

This Software Requirement Specification shall be used in conjunction with the following publications:

[1] IEEE 830:1998 IEEE Recommended Practice for Software Requirements Specifications. <https://ieeexplore.ieee.org/document/720574>

[2] RFC-2119. Key words for use in RFCs to Indicate Requirement Levels. Published in March, 1997. Available at: <https://datatracker.ietf.org/doc/html/rfc2119>

The development team should familiarize with the following documents:

[i.] LilyGo T-Watch. LilyGo T-Watch GitHub repository. Available at: [https://github.com/Xinyuan-LilyGO/TTGO\\_TWatch\\_Library](https://github.com/Xinyuan-LilyGO/TTGO_TWatch_Library)

[ii.] ESPRESIF documentation. ESP32 Arduino Core's documentation. Available at: <https://docs.espressif.com/projects/arduino-esp32/en/latest/>

## 1.6 Context Diagram and Overview

This document follows the SRS structure detailed in IEEE 830 section 5 [1]. Section 2 of this document lays out the over all description and section 3 the specific requirements, as detailed in IEEE 830.

The image below details an overview of the system concept.

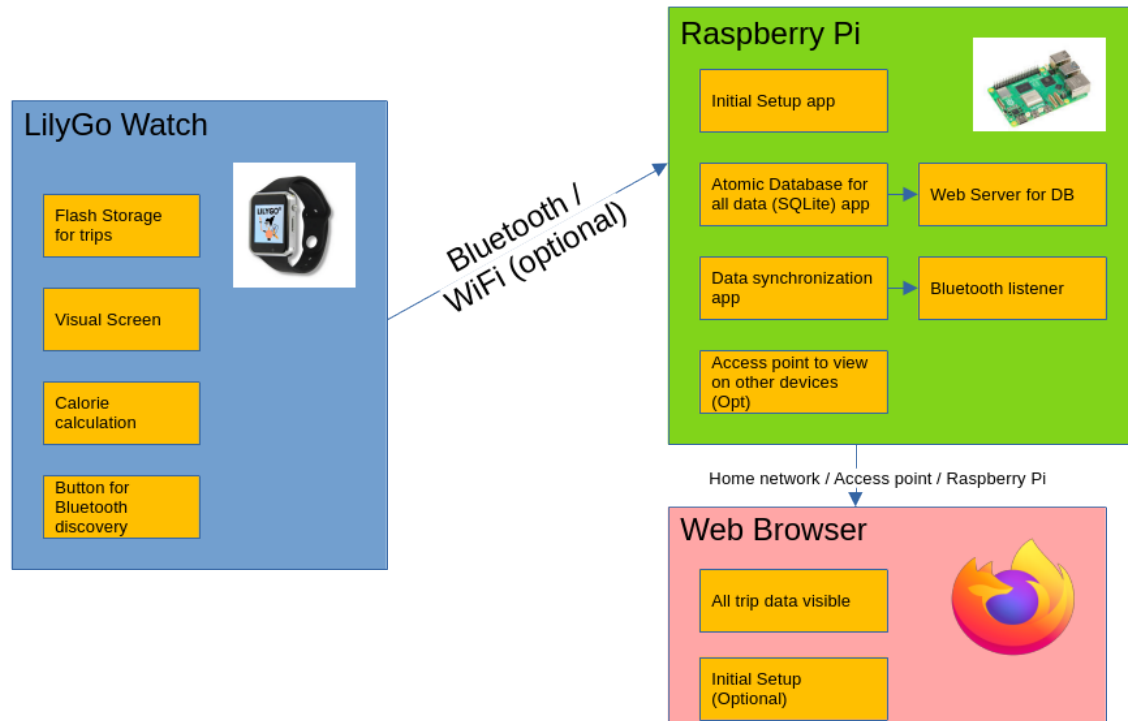


Figure 1: Initial Concept Overview

## 2 Overall description

### 2.1 Product perspective

#### 2.1.1 System interfaces

This subsection expands and defines the frontends of the Web Application and LilyGO functionalities defined in Functional Requirements.

The [User interfaces](#) section lists the characteristics between the software and the user.

The [Hardware interfaces](#) section lists the characteristics between hardware and software.

The [Software interfaces](#) section lists the characteristics between different software and applications within the system.

### 2.1.2 User interfaces

The LilyGO interface user **MUST** be able to interact with the functionalities:

- “start a hiking session”
- “stop a hiking session”
- “display step count on display”
- “display travelled distance on display”
- “record step count”
- Enable Bluetooth capabilities

The LilyGO interface **SHOULD** follow the given interface layout:

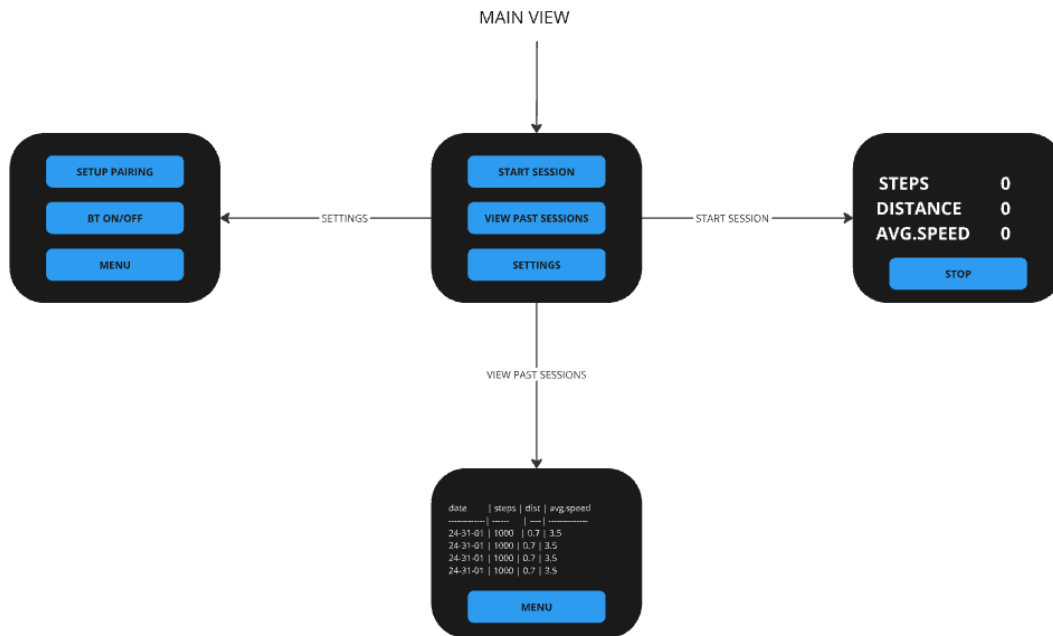


Figure 2: LilyGo Views

Note: the GUI of the final product may not be a one to one match with the early visualization draft represented here. It will however meet the requirements for functionalities.

The Web UI interface user **MUST** be able to interact with the functionalities:

- “display a list of past sessions”
- “display the last session”

The Web UI interface **SHOULD** follow the given interface layout:

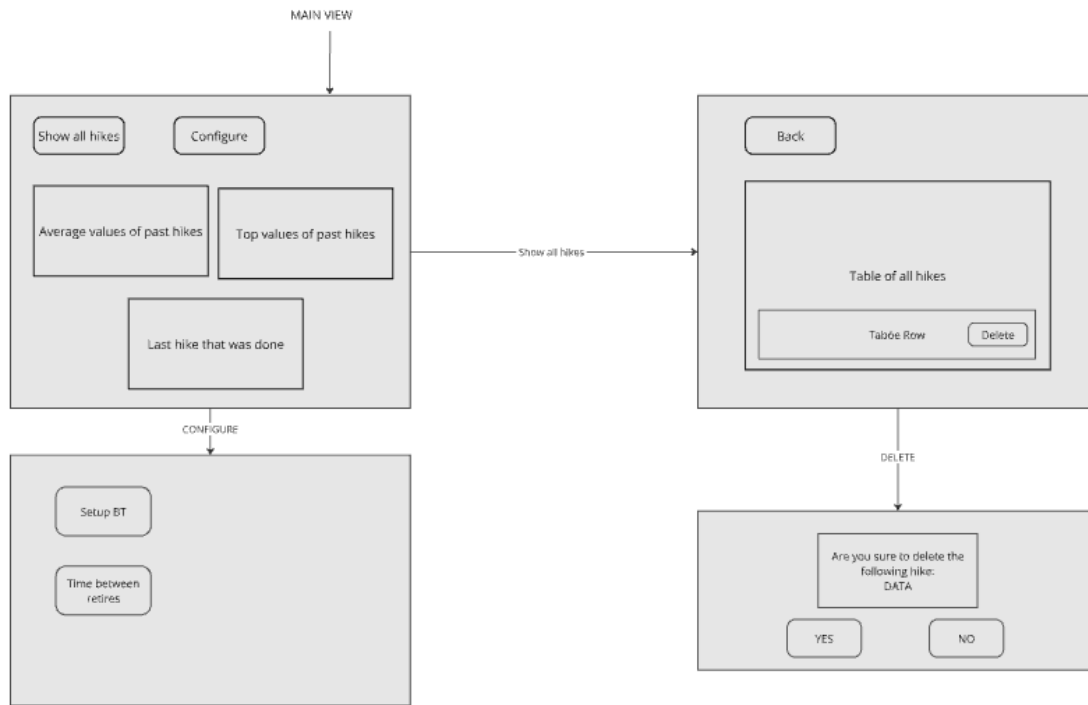


Figure 3: Web UI views

### 2.1.3 Hardware interfaces

The LilyGO application **MUST** use the touchScreen FT6336 using the pinout defined by LilyGO hardware:

- SDA: 23 pin
- SCL: 32 pin
- Interrupt: 38 pin

The LilyGO application **SHOULD** use the BMA423 using the pinout defined by LilyGO hardware:

- Interrupt: 39 pin

The WebUI HTML layout **MUST** be viewable in a web browser on Raspberry Pi. In this context viewable means that the user **MUST** be able to read all information rendered on a web page

The WebUI HTML layout **MAY** be viewable on a mobile phone screen.



#### **2.1.4 Software interfaces**

The LilyGO application SHOULD use LilyWatch maintained TTGO\_TWatch\_Library for the pin definitions and driver implementation

The Web application SHOULD use SQLite to store synchronized data locally on the RPi

The Web application SHOULD use SQLite to store its information

#### **2.1.5 Communications interfaces**

The synchronization SHOULD be able to happen only after initial setup between RPi and LilyGO

The synchronization via Bluetooth between RPi and LilyGO MUST happen via Bluetooth v4.2 or below

The WebUI MUST use http protocol within a local TCP/IP network.

The RPi MUST be able to connect to a local network OR host its own access point to view the WebUI

### **2.2 Product functions**

This subsection contains the functional requirements for the Hiking Application prototype. As the prototype consists of both LilyGo - application and the Web application to present tracking data, the functional requirements gather requirements for both of these.

#### **2.2.1 LilyGo application: Start & stop hiking sessions**

The system MUST allow user to start a hiking session

The system MUST allow user to stop a hiking session

The system MAY allow user to continue stopped hiking session

The system MAY prevent starting a new hiking session, if real-time clock of the smartwatch is compromised. In this case, the smartwatch must be first synchronized with RPi to synchronize the real-time clock.

### **2.2.2 LilyGo application: Recording multiple hiking sessions**

The system **MUST** allow user to record multiple hiking sessions to smartwatch memory

The system **MUST** store at minimum 5 last sessions in persistent memory of the smartwatch

The system **MAY** support storing more than 5 last sessions on the smartwatch

The system **MAY** warn user if memory capacity is full (max number of sessions have been stored to persistent memory)

The system **MAY** ask user to override older sessions

The system **MAY** prohibit starting new session before memory capacity is released

The system **MAY** override the oldest stored session when persistent memory is full

### **2.2.3 LilyGo application: Record steps count and convert into travelled distance during the session**

While hiking session is active, the system **MUST** record steps count

While hiking session is active, the system **MUST** convert steps count into travelled distance every time data is rendered on touchscreen

### **2.2.4 LilyGo application: Display this data on a smartwatch screen**

While hiking session is active, the system **MUST** display step count on display

While hiking session is active, the system **MUST** display travelled distance on display

While hiking session is active, the system **MAY** display average speed on display

While the hiking session is active the step count and travelled distance **MUST** update regularly

The update frequency of hiking session information **MAY** be optimized with best effort based on hardware constraints

While hiking session is not active, the touchscreen **MAY** display step count and travelled distance for last session on display

While hiking session is active, the smartwatch touchscreen **MAY** display the current date (YYYY-MM-DD)

### **2.2.5 Synchronize and store data with RPi via Bluetooth**

The smartwatch application **MUST** be capable of sending step count and travelled distance via Bluetooth to the web application on RPi

The smartwatch application **MAY** also send date of hike and average speed for each recorded hiking session

The smartwatch application **MUST** be able to connect to RPi with MAC address that is hard coded to RPi

The smartwatch application **MAY** be able to connect to RPi with LilyGo enabling Bluetooth discovery

The smartwatch application **MAY** be able connect to RPi by configured Wi-Fi connection

The smartwatch application **MAY** be able to find RPi with Wi-Fi discovery

### **2.2.6 Calculate estimated amount of calories burned during the session on RPi**

The system **MUST** calculate estimated amount of calories based on travelled distance

### **2.2.7 Web application: Initialize Web UI and show last session statistics (travelled distance, step count and burned calories)**

The system **MUST** display travelled distance, step count and burned calories for last session

The system **MUST** contain a list of past sessions

The displayed list of past sessions **MUST** contain travelled distance, step count and burned calories for each session

The displayed list of past sessions **MAY** contain date for each session

The system **MAY** provide detail view for a chosen session where additional session information is presented

The system **MAY** provide delete feature for removing past sessions from persistent memory

## **2.3 User characteristics**

This SRS details the requirements for the proof-of-concept (poc) version of the system under design (SUD). The system **SHOULD** be in a stage in which users with intermediate technical skills with Smartwatches, Bluetooth technology and Raspberry Pi product family can use the product indepentently with the help of the project documentation.

With the help of the documentation the user must be able to know how to

- clone source code from remote repository (GitHub)
- how to set up virtual environment and install dependencies
- how to flash firmware on LilyGO T-Watch

## **2.4 Constraints**

### **2.4.1 System interfaces**

### **2.4.2 User interfaces**

### **2.4.3 Hardware interfaces**

### **2.4.4 Software interfaces**

### **2.4.5 Communications interfaces**

### **2.4.6 Memory**

### **2.4.7 Operations**

### **2.4.8 Site adaptation requirements**

## **2.5 Asuptions and dependencies**

- The clock is synchronized regularly between trips

## **2.6 Apportioning of requirements**

# **3 Specific Requirements**

## **3.1 Performance Requirements**

### **3.1.1 MUST have acceptably long data synchronization**

## **3.2 Design Constraints**

### **3.2.1 MUST use atleast predefined hardware**

- The system MUST use T-Watch V2
- The system MUST use raspberry pi 2, 3 or 3+
- The system MAY use other hardware
- If the clock runs out of sync due to power loss the system SHALL synchronize its time when syncing to the RPi

## **3.3 Software-system Attributes**

### **3.3.1 Reliability**

- The system MUST log and save the current activity in periodically (15s)
- The system MUST NOT miss 2 consecutive loggings

### **3.3.2 Security**

- The LilyGo watch SHALL only send hiking data to a paired device
- The LilyGo watch SHALL only be discoverable during pairing with the RPi
- The RPi SHALL periodically check if the watch is within range, the LilyGo SHALL NOT attempt to connect to the RPi automatically