10th of February

System Requirements Specification

Hiking Tour Assistant

Nguyen Thanh Tuyet Han
Miguel García González
Pyry Seuranen
Group A

Table of Contents

Document version table

Version	**Date**	**Author**	**Description**
1.0	2023-01-28	Miguel García	Creation of the document
1.1	2023-02-05	Han Nguyen	Overall description from product perspective section added
1.2	2023-02-06	Pyry Seuranen	Introduction section added
1.3	2023-02-08	Han Nguyen	Specific requirements section added
2.0	2023-02-10	Pyry Seuranen	First draft created
2.1	2023-03-15	Miguel García	More details to document added
2.2	2023-03-17	Han Nguyen	Document revision
3.0	2023-03-20	Pyry Seuranen	Final version

1. Introduction

1.1. 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.

1.2. 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.

1.3. Definitions, acronyms, and abbreviations

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
BLE	Bluetooth Low Energy
WebUI	Web User Interface as Flask web application
WD	Watch Display Module
AD	Activity Data Module
BC	Bluetooth Control Module
HTA	Hiking Tour Assistant

Table 1: Definitions and Acronyms

1.1. Communication standards

- **Communication protocol**: Bluetooth Low Energy (BLE)
- **Transmission frequency**: 2.4 GHz
- **Transmission speed**: 1 Mbps

- **Data Format**:

The data will be sent in a single string separated by a comma (",") with the following format:

steps (int32), *distance* (float), *time in seconds* (long), *calories* (int8), *id* (int32)

Each data type will have the following size:

- *steps*: 4 bytes (32 bits) - signed integer
- *distance*: 4 bytes (32 bits) - floating point number
- *time*: 4 bytes (32 bits) - signed long
- *calories*: 1 byte (8 bits) - unsigned integer
- *id*: 4 bytes (32 bits) - signed integer

The data will be sent in little-endian byte order.

If the values of the data are:

steps = 10000

distance = 2.3

time = 3600

calories = 150

id = 123456

The data will be received through a Bluetooth socket using the PyBluez library in Python. The received data will be the following string:

"10000,2.3,3600,150,123456"

1.2. 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-LilyGO/TTGO_TWatch_Library

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

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

1.3. 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.

2. Overall description

2.1. Product perspective

2.1.1. System interfaces

- LLSW starts and stops hiking sessions with a start/stop button that is always present on the display.
- The LLSW communicates with the RB400 through the Bluetooth interface using the BluetoothSerial.h library from Arduino.
- The RB400 acts as the Bluetooth central device, receiving the data sent by the LLSW through the established Bluetooth connection.
- The data received by the RB400 is processed in the Flask application that runs on the same RB400. The Flask application uses the PyBluez library for data reception through Bluetooth.
- The Flask application presents the received data on a web page that runs on the RB400, allowing the user to view the data retrieved from LLSW.
- The user interface on the LLSW presents the sensor data in real-time to the user, allowing them to track their physical activity. The LLSW's firmware has the following modules:
    - Watch Display module manages the screen display.
    - Activity Data module manages the activity data collection and calculations.
    - Bluetooth Control module manages the sending of activity data via Bluetooth.

2.1.2. User interfaces

2.1.2.1 Watch User

The LLSW shall provide a start/stop button which is always present on the screen.

2.1.2.2 Web UI User

The WebUI 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.

2.1.3. Hardware interfaces

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

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.

- Source: The LLSW and RB400 are the hardware components used in the system.
- Inputs: 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. The precision of step counting is ± 10 steps.
- Outputs: The LLSW communicates via Bluetooth with the RB400 to send the user's physical activity data. The LLSW also has an OLED display to show the user's physical activity data.
- Purpose Description: The purpose of the LLSW is to measure and collect the user's physical activity data and display it on its OLED screen. The purpose of the RB400 is to receive the data from the LLSW and display it on a webpage using a Flask application.
- Range or Accuracy: The accelerometer of the LLSW has a range of ± 16g with a precision of 0.244mg/LSB.
- Units of Measure: Steps are measured in units, distance in miles, time in seconds and calories in kilocalories.
- Timings: The accelerometer of the LLSW can update at a frequency from 12.5 to 800 Hz.
- Data Formats: The steps are stored in an int32 type variable, distance is stored in a float type variable, time in seconds is stored in a long type variable, calories are stored in an int8 type variable, and the id is stored in an int32 type variable. The data is sent via Bluetooth in a comma-separated string. The calories are calculated as 0.1225 kcal per second. The distance is calculated based on 0.0045 miles covered every 10 steps.

2.1.4. Software interfaces

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

Flask 2.2.x, HTML5, and Bootstrap 4 will be used for the WebUI implementation.

PyBluez library will be used in the WebUI to implement Bluetooth connection with LLSW

The operating system of LLSW shall be FreeRTOS.

Arduino library for Bluetooth data sending from LLSW shall be BluetoothSerial.h

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

The LLSW's firmware has the following modules:

- Watch Display module manages the screen display.
- Activity Data module manages the activity data collection and calculations.
- Bluetooth Control module manages the sending of activity data via Bluetooth.

User Interface	Inputs	Outputs	Purpose Description	Range or Accuracy	Units of measure	Timings	Formatos de Datos
LLSW screen	Phisical activity information	Screen display with activity data	Start and stop hiking session and show activity data	±10 steps	Miles, kcal, unit steps, seconds	Real-Time updates	int32, int8, float, long, string
WebUI on RB400	Phisical activity data sent by LLSW via Bluetooth	Table with activity data	Display activity data of the last session or past sessions on the web browser	±10 steps	Miles, kcal, unit steps, seconds	On-demand updates	SQLite and Python Dictionary

Table 2: Software interfaces

2.1.5. Communications interfaces

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

The data transmitted shall be a string with data divided by commas.

The LLSW shall send all hiking data every time the user requires it from the WebUI and the RB400 shall preserve the connection to LLSW until the user decides to end the connection from the WebUI.

2.1.6. Memory

LLSW has an ATSAM3X8E microcontroller with 512 KB of flash memory and 96 KB of SRAM. The firmware for LLSW is stored in the flash memory, and it includes code to handle data acquisition from sensors, Bluetooth communication with RB400, and data processing for display on the OLED screen.

The RB400 has 4GB of RAM, 16 GB of data storage given by the microSD card used in the system.

The RAM will be used mostly when the WebUI and the browser are up and running which would use an estimated amount of 250 MB.

The data storage is used for storing the Flask application code and any additional libraries that may be required, and the web browser that has to be run to access the WebUI. The microSD card is used for storing the database that collects all the session's data collected from the LLSW, including the step count, distance, calories, time, and id.

Overall, the memory usage of the system is well within the limits of the RB400 and the LLSW, and the microSD card provides ample storage space for the data collected over an extended period.

2.1.7. Operations

- Start and stop hiking sessions: the user can choose when the LLSW will collect its activity data with a start/stop session button present on the LLSW display.
- Data Collection: The LLSW device collects data on the user's activity during each session, including steps taken, distance traveled, calories burned, and time of session. The data is then sent to the RB400 device for further processing.
- Data Processing: The RB400 device receives data from the LLSW and processes it with the WebUI. The processed data is then stored on the system database to be ready to be displayed on the web page.
- Data Display: The web page displays the processed data in a user-friendly format. Users can view their physical activity data, including steps taken, distance traveled, and calories burned, in real-time.

2.1.8. Context diagram

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

2.1.8.1 LLSW

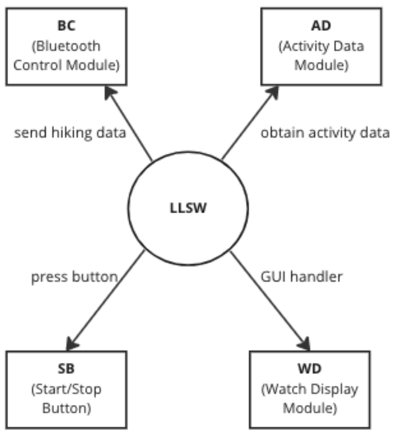

Figure 1: LLSW context diagram

The start/stop button (SB) is an input to the system that allows the user to start and stop the monitoring of physical activity. The Activity Data (AD) module is responsible for collecting the user's physical activity data, such as steps taken, distance traveled, and calories burned.

The Bluetooth Control (BC) module is responsible for wireless communication between LLSW and other devices, such as RB400 in this system. Finally, the Watch Display (WD) module is responsible for displaying the collected data on the watch display.

2.1.8.2 RB400

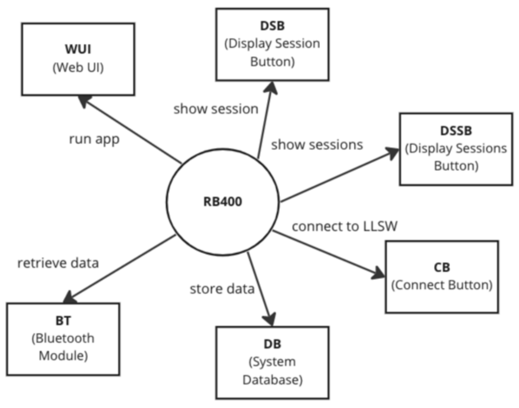

Figure 2: RB400 context diagram

The WUI component represents the user interface of the system, which is accessed through a web browser. The DSB and DSSB components represent the buttons on the

WebUI used for controlling the displaying of sessions. The CB component represents the button used to connect the RB400 device to a Bluetooth-enabled device (in this system, the LLSW). The BT component represents the Bluetooth Module of the WebUI, which provides a Python interface to the Bluetooth stack. Finally, the DB component represents the system database, which will store the information retrieved from LLSW via Bluetooth.

2.1.8.3 Hiking Tour Assistant

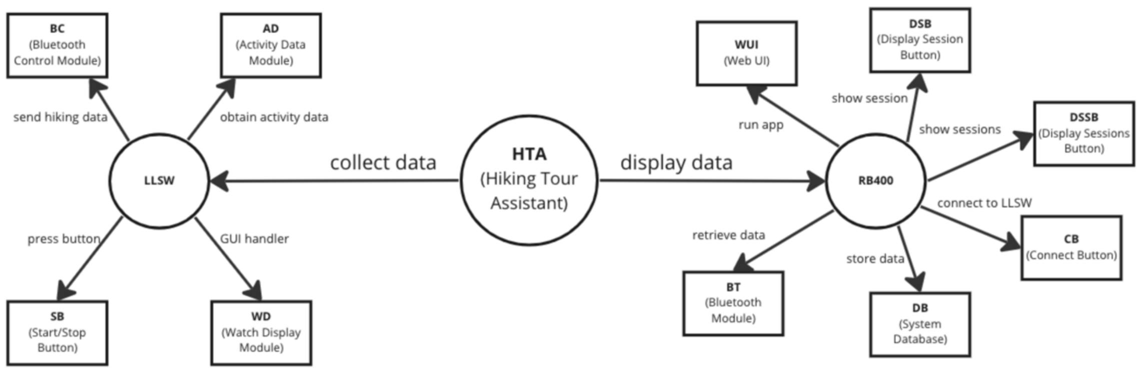

Figure 3: Hiking Tour Assistant context diagram

LLSW is the component that obtains the physical activity data from the user, and RB400 is the component that connecting to LLSW will retrieve that data and display it in the WebUI running on the web browser.

2.2. Product functions

The system will perform the following functions.

2.2.1. Start/Stop Hiking Session.

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

2.2.2. Record data related to hiking activity.

During hiking section, user steps will be recorded and transform to distance by applying this formula: $Distance = \frac{Steps}{10} * 0.0045\ miles$

The total time of hiking session will also be recorded and transform to burned calories by applying this formula: $Calories = Time\ (second) * 0.1125$

The above formula depends on user characteristics.

This function involves collecting data about physical activity, such as the number of steps taken, distance covered, and calories burned, using the activity data module (AD) on LLSW. The AD module continuously collects data applying the above mentioned formulas and sends it to the watch display module (WD) for displaying the information.

2.2.3. Display data on smartwatch screen.

This function allows the collected information by activity data module be passed to the watch display module (WD) of LLSW, so in that way the activity data is displayed. The displayed information includes the number of steps taken, distance covered, calories burned, and hiking time, which are updated in real-time.

Whenever the user starts a new session, all data resets to zero in the activity module, and these changes are reflected in the watch display module.

The smartwatch also displays actual time of the day, smartwatch battery level, and the total steps taken in previous hiking session (on the top Status bar).

2.2.4. Synchronize and store data with Raspberry Pi.

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

RB400 will receive data and display it via web UI.

This function involves establishing a Bluetooth connection between LLSW and RB400 using the Bluetooth control module (BC) on LLSW. This allows LLSW to transmit the collected activity data to RB400, which will receive this data using the Bluetooth Module and will process the data to store it in the system database using its database module.

2.2.5. Initialize Web UI

This function involves displaying the collected activity data on the WebUI in a user-friendly and intuitive format. The user can access the WebUI by opening a web browser and entering the IP address of RB400.

2.2.6. Calculate amount of calories burned in the session on WebUI

This function aims to calculate the amount of calories burned during a physical activity session. To achieve this, the following formula will be used to take into account the time of the session, the weight of the user and the intensity of the activity (the MET is a factor that changes depending on the exercise that has been made). The results will be displayed in the WebUI user interface so that users can get an idea of how many calories they have burned during the session.

$$Calories = Time\ (second) * MET * 3.5 * \frac{Weight(kg)}{(200 * 60)}$$

2.2.7. Show activity data.

With this feature, users can view the physical activity data of the last session recorded in the system. This includes information such as session duration, number of calories burned, distance traveled, and number of steps taken. The information will be displayed on the WebUI user interface when the user presses the corresponding buttons, which is the Display Session Button (DSB) to display last session data.

2.3. 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.

2.4. Constraints

- The system is designed to store a limited amount of data on LLSW and will require regular data transfer to the WebUI to prevent data loss.

- The system is designed to work with a specific set of hardware components, including LLSW and RB400, and may not function properly with other hardware.
- Bluetooth compatibility: The RB400 and the LLSW device must be compatible with Bluetooth Low Energy (BLE) version 4.2 or higher to enable wireless communication between the devices.
- User safety: The LLSW device should be designed to minimize any potential safety hazards during physical activity, such as skin irritation or allergic reactions to materials used in the device.
- Data accuracy: The activity data collected by the LLSW device should be accurate and reliable to ensure that the data is useful for analysis and tracking physical activity progress.
- Security: The data collected by the LLSW device and transmitted to the WebUI should be secure and protected from unauthorized access.
- Compatibility with WebUI: The LLSW device must be compatible with the WebUI, and the data transmitted from the device must be in a format that is usable by the WebUI.
- Usability: The LLSW device should be designed with a user-friendly interface that is easy to navigate and understand, to ensure that users can quickly start and stop physical activity tracking and access their activity data. The WebUI should be also designed with user-friendly interface easy to navigate and understand, to ensure that users can quickly connect to the watch and see their hiking sessions.
- Size and weight: The LLSW device should be compact and lightweight to enable comfortable use during physical activity without causing any discomfort or hindrance.

2.5.Assumptions and dependencies

Assumptions:

- The LLSW and RB400 devices will be compatible with the firmware and software requirements specified in this document.
- A web browser will be installed on RB400 to access the WebUI.
- The user will have a basic understanding of how to use the LLSW and RB400 devices.
- The LLSW and RB400 devices will be used for their intended purpose, which is tracking physical activity.

Dependencies:

- The successful implementation of this software depends on the proper functioning and integration of both the LLSW and RB400 devices.
- The software relies on the BluetoothSerial.h and PyBluez libraries to establish a Bluetooth connection between the LLSW and RB400 devices.
- The software depends on the availability and functionality of the WebUI to transmit and display activity data.
- The software assumes that the user will interact with the system as specified in the Product Functions section of this document.
- The software is dependent on the accuracy of the activity data collected by the LLSW device.

3. *Specific Requirements*

3.1. *External interfaces*

Name of Item	**Description of Purpose**	**Source of Input or Destination of Output**	**Valid Range, Accuracy, and/or Tolerance**	**Units of Measure**	**Timing**	**Relationships to Other Inputs/Outputs**	**Data Formats**
Watch Screen	Displays information about physical activity on LLSW	User	N/A	Inches	Real-time	WatchDisplay module manages the display functionality	N/A
WebUI screen	Displays information about physical activity on WebUI	User	N/A	Inches	On-demand	Bluetooth Control module sends the data showed in screen	N/A
Start/Stop Button	Initiates and terminates physical activity measurement	User	Pressed-Not Pressed	N/A	On-demand	Watch Display module manages the interaction with the button	N/A
Display Session Button	Displays information about previous sessions	User	Pressed-Not Pressed	N/A	On-demand	Database system provides data when user presses button	N/A
Display Sessions Button	Selects a session to display information about	User	Pressed-Not Pressed	N/A	On-demand	Database Module provides data when user presses button	N/A
Connect Button	Initiates the connection between RB400 and WebUI	User	Pressed-Not Pressed	N/A	On-demand	Bluetooth Control module starts managing the connection when button pressed	N/A
Bluetooth Control Module	Establishes a connection between LLSW and RB400	LLSW	N/A	N/A	On-demand	WebUI screen receives the data retreived	string

Watch Display Module	Manages the displaying information about physical activity on the watch display	Watch screen	N/A	N/A	Real-time	Watch screen gets served from this module functionality	float, int32, long
WebUI	Displays physical activity data in a web browser	WebUI screen	N/A	N/A	On-demand	Database system provides data and WebUI screen shows data to user	python dictionary
Database Module	Stores data received from LLSW and retrieves data from system database	Bluetooth Control	N/A	N/A	On-demand	WebUI uses the data that the database module manages	integer, real, timestamp
BMA sensor	Retrieves the physical activity data in real time from the user	Watch Screen	12.5-800 Hz	step units	Real-time	Watch Display module uses the data collected by the sensor	units

Table 3: External interfaces

3.2. Functions

3.2.1. Use cases.

3.2.1.1. 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.

3.2.1.2. 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.

3.2.1.3. 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.

3.2.1.4. 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.

3.2.2. Use cases diagrams.

3.2.2.1. Start/Stop Hiking Section

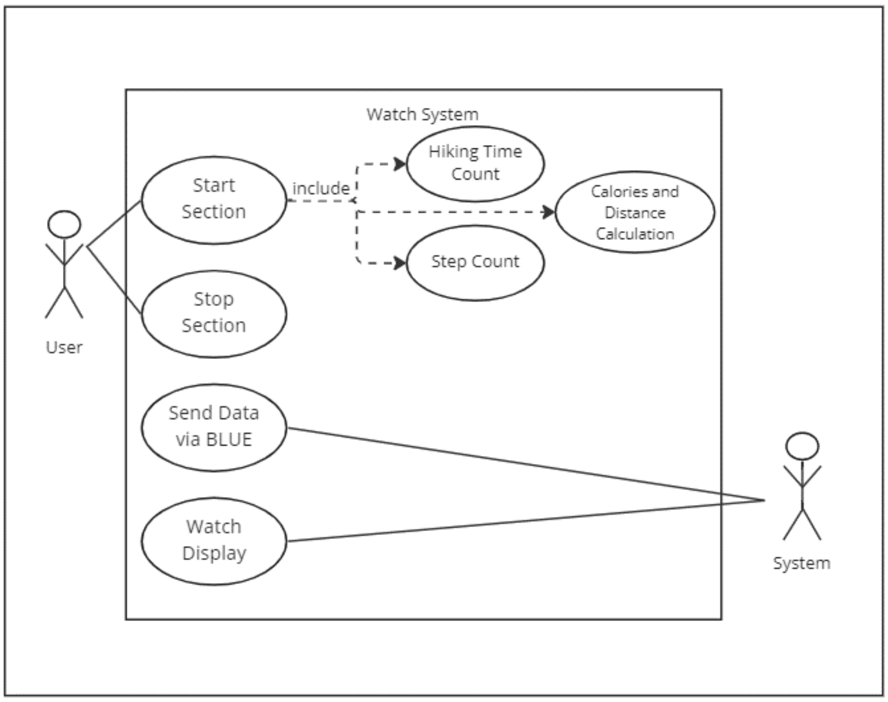

Figure 4: Start Stop Hiking Section

3.2.2.2. Display Hiking Data in Web UI

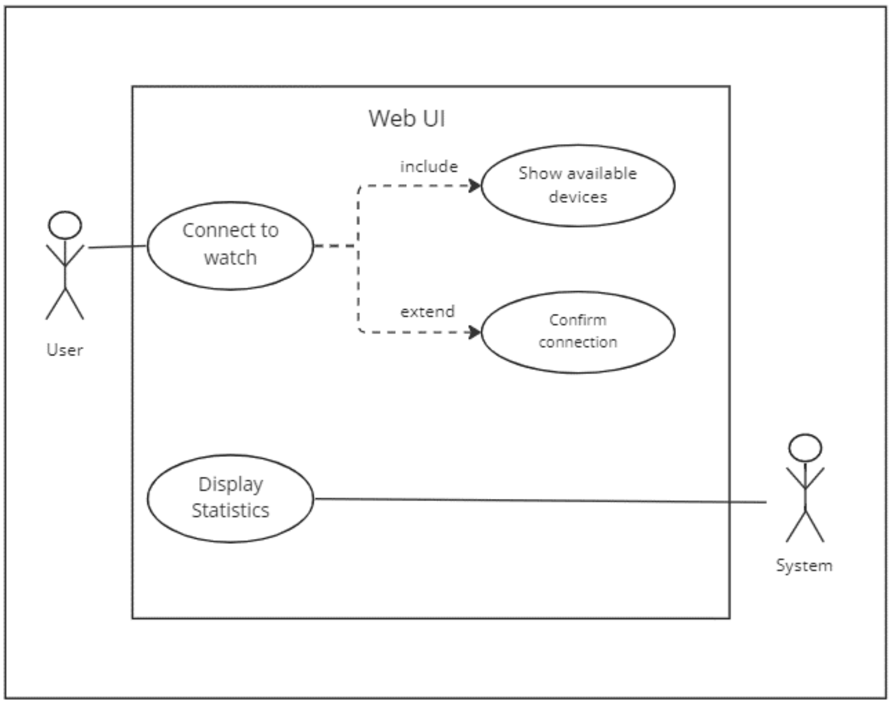

Figure 5: Display Hiking Data in Web UI

3.2.3. Data flow diagram.

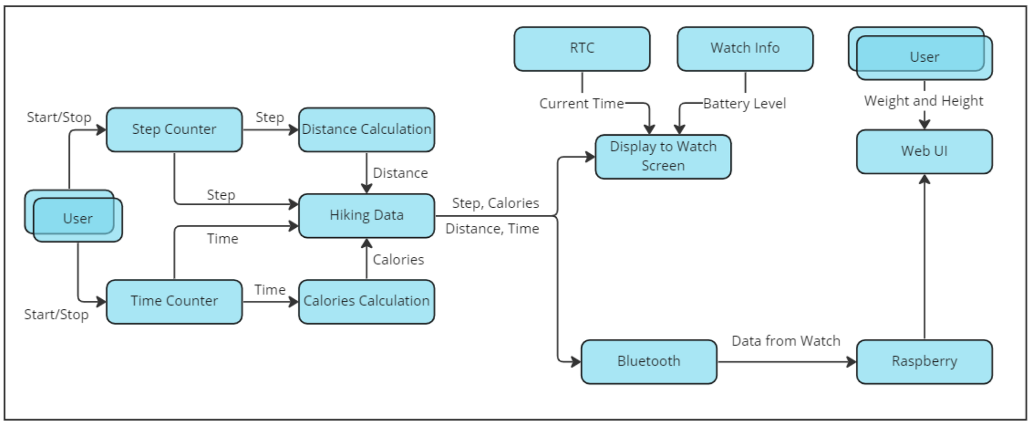

Figure 6: Data Flow Diagram

3.3. Performance requirements

Requirement name	**Requirement description**	**Objective value**	**Measurement unit**	**Tolerance**	**Relation with other requirements**	**Update of targets**
Response time	The maximum time the system should take to respond to a user request.	Less than 2 seconds	seconds	± 1 second	N/A	Based on project progress and user feedback.
Concurrent connections	The maximum number of simultaneous connections that the system must support.	1 connection	connections	0	N/A	Based on project progress and user feedback.
Concurrent users	The maximum number of users that can use the system simultaneously without affecting performance.	1 user	users	Al menos 25	N/A	Based on project progress and user feedback.
System uptime	The minimum time the system must be in operation without interruption.	99% of uptime	percentage	Less than 1% downtime	N/A	Based on project progress and user feedback.
System security	The system must assure that the form´s data has been introduced by the user	Use Flask secret key and csrf token for forms	N/A	N/A	N/A	Based on project progress and user feedback.

Table 4: Performance Requirements

3.4. Logical database requirements

Requirement Name	Description	Data Type	Field Size	Primary Key
Session ID	Unique identifier for each session	Integer	4 bytes	Yes
Steps	Number of steps taken during the session	Integer	4 bytes	No
Time	Duration of the session	Integer	4 bytes	No
Calories	Calories burned during the session	Real	8 bytes	No
Timestamp	Date and time when the session started	Timestamp	8 bytes	No

Table 5: Logical Database Requirements

3.5. Design constraints

Constraint Type	Constraint	Description
Hardware	LLSW	ESP32-D0WDQ6-V3 Xtensa LX6 microprocessor, 8 MB of PSRAM, Wi-Fi: 802.11 b/g/n, Bluetooth: V4.2+ BLE
Hardware	RB400	Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz 4GB LPDDR4-3200 Dual-band (2.4GHz and 5.0GHz) IEEE 802.11b/g/n/ac wireless LAN Bluetooth 5.0, BLE
Technology	Python	Python3.8 used for WebUI implementation
Technology	Arduino	Arduino IDE 1.8.19 will be used for watch's firmware implementation
Technology	SQLite	SQLite3 will run the system's database
Framework	Flask	Flask 2.2.X web application framework will be used for the WebUI implementation
Platform	Debian OS	The WebUI will run on RB400, with Debian OS
Technology	PyBluez	PyBluez 0.23 python library will manage Bluetooth connection from WebUI
Technology	BluetoothSerial	BluetoothSerial.h library will manage the Bluetooth connection from LLSW

Table 6: Design Constraints

3.6. *Software system attributes*

3.6.1. Reliability

1.1. The LLSW delivers data via Bluetooth every five minutes ± one minute.

1.2. Web UI always admits connection with LLSW.

1.3. The user gives the exact weight and height variables for the precise evaluation of burned calories.

1.4. The dynamic performance requirements ensure that the connection and measurement accuracy are over 90 %.

3.6.2. 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.

3.6.3. Security

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

3.6.4. 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.

3.6.5. Portability

- The watch's firmware 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 Flask 2.2.x, HTML5 and Bootstrap 4.
- Bluetooth data sending from LLSW with BluetoothSerial.h library.
- Bluetooth connection from WebUI with LLSW with PyBluez library.