SOFTWAREREQUIREMENT SPECIFICATION

**Motion and Proximity Tracking**

**Prepared by**

**Snehal Rajwar**

**Table of Contents**

**1. Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Project Scope

1.5 Definitions, Acronyms and Abbreviations

**2. Overall Description**

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

2.8 Flow diagram

**3. External Interface Requirements**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

**4. Other Non-Functional Requirements**

4.1 Performance Requirements

4.2 Safety and Security Requirements

4.3 Software Qulaity Attributes

**1. INTRODUCTION :**

This wearable device is used to track a user’s motion (using angular rotation and acceleration) along with position (using proximity) in order to naturalize the movement of the user and store the corresponding data to the database.

1.1 Document Purpose

The purpose of this report is to fully document the project that will be implemented by the organization. No changes in workflow will occur unless this document is modified.

1.2 Document Conventions

Bold Text - Signifies important content or keywords

1.3 Intended Audience and Reading Suggestions

This document is intended for the professors of ST.vincent pallotti college of engineering and technology and for the staffs of TPF Software Pvt Ltd. This SRS will discuss the details and implementation of the project. It is recommended for those familiar with computer science to start in section 2, while those unfamiliar should start in section 1.5 to understand some of the terms used throughout the document. Those seeking details about the implementation and how to use the product should continue onto section 3.

1.4 Project Scope

This project will create an integrated system of more than one sensor(s) and software to make it possible to track the movement of a person whoever is wearing the device on his body. The main intention of this project is to allow a user to monitor the activities of a person and calculate his in-range and off-range timings. A webpage has been developed in such a way that employees can access it to monitor the activities of their laborers. Our project provides the foundation for this significant benefit.

1.5 Definitions, Acronyms and Abbreviations

SRS - Software Requirement Specification

IDE - Integrated Development Environment

Wi-Fi - Wireless Fidelity

BLE - Bluetooth Low Energy

MQTT - Message Queuing Telemetry Transport

HTML –Hypertext Markup Language

CSS – Cascading Style Sheets

JS - JavaScript

IOT – Internet Of Things

**2. OVERALL DESCRIPTION :**

2.1 Product Perspective

* Cross platform support
* Number of users being supported by the system
* A database that stores :

|  |
| --- |
|  |
|  | Index Denotes Short-Form |
|  | 0 S.No. NA |
|  | 1 UUID UUID |
|  | 2 Session Number SN |
|  | 3 Enter Time ST |
|  |  |
|  |  |
|  | 4 Exit Time (ET) |
|  | 5 Session-AT - Active (SWT) |
|  | 6 Session-IT - Idle (SIT) |
|  | 7 Total AT - Active (TWT) |
|  | 8 Total In-Time - InRange (inrangeTime) |
|  |  |
|  | 9 Last Activity Start (LSST) |
|  | 10 Last Activity Stop (LSET) |

2.2 Product Features

The following features are in scenario’s when an aircraft either arrives to/departs from an airport:

* An employee entering the zone (range-specific to aircraft) to serve the aircraft, is said to have a Session, in which he performs various activities, such as baggage loading, Cabin Service, Catering, Re-Fueling, Field Operation Service, etc…
* Thus for every Session of an employee, the following details are tracked :
  + UUID, Session Number, Enter Time, Exit Time, Session-AT, Session-IT, Total-AT, Total In-Time, Last Activity Start, Last Activity Stop, Gyro Values: Yaw, Pitch & Roll, and Accelo Values.

2.3 User Classes and Characteristics

* It is considered that the user do have the basic knowledge of operating the internet and to have access to it. The administrator is expected to be familiar with the interface of the tech support system.

2.4 Operating Environment

Operating environment for the airline management system is as listed below.

* Client Side System
* Operating System: Windows, MacOS, Linux
* Database: SQLite database
* Platform: JavaScript

2.5 Design and Implementation Constraints

* SQL commands for above queries/applications.
* Managing various employees on the field working for 2 different aircrafts, but have an overlapping range space.
* Implement the database at least using a centralized database management system.

2.6 User Documentation

Users of the system should be able to track the following details of employees serving a flight, from the database:

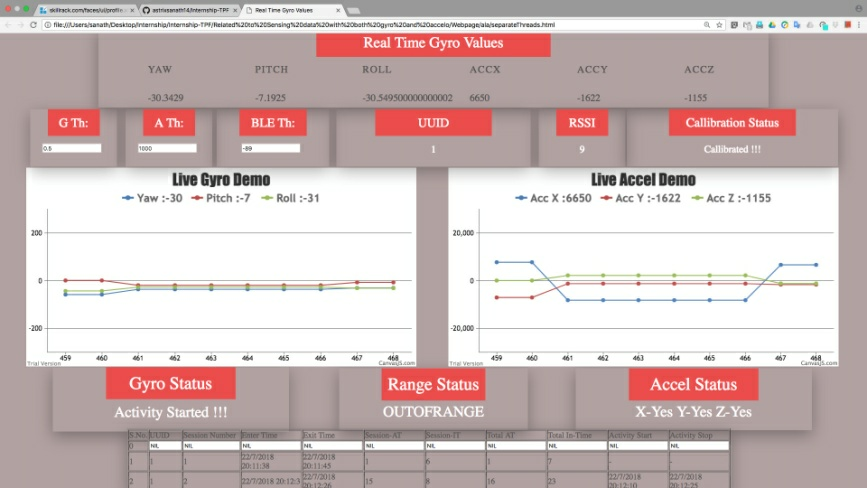
UUID - Session Number - Enter Time - Exit Time - Session-AT - Session-IT -

Total-AT – Total In-Time - Last Activity Start - Last Activity Stop

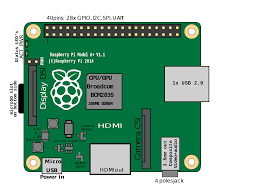
2.7 Assumptions and Dependencies

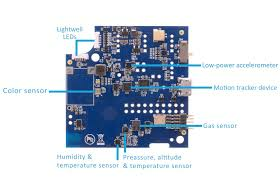
* This software highly depends on type and version of browser being installed in the system i.e. browser version should be used which have HTML5 support.

2.8 Flow Diagram :

**Webpage**

**MQTT broker (iot.ecllipse.org)** **raspberry**

**wemos (wi-fi module) MPU6050 (6degree sensor) BLE Beacon**

**3. External Interface Requirements :**

3.1 User Interfaces

* Webpage – through which user can visualize data graphically.

3.2 Hardware Interfaces

* WeMos D1 R1
* Mpu6050
* BLE Beacon
* Device should be enabled with Internet.

3.3 Software Interfaces

* Web Browser – to view the live feed on Webpage
* The user’s browser should be HTML5 compatible for a satisfactory user experience.
* NodeJS for scanning BLE devices

3.4 Communication Interfaces

* LAN – Local Area Network
* BLE Advertiser and Listener (BT channel)

**4. OTHER NON-FUNCTIONAL REQUIREMENTS :**

4.1 Performance Requirements

• The user is to stand within the range

• The system should run in Real-time

• Should be able to track more than 1 person in the Workspace

* Should be able to run on commercially available computing platforms

4.2 Safety and Security Requirements Safety

* As the webpage is designed using HTML, CSS, and JS, its source code will not be secure as it is basically Client Side scripting, and hence the data maybe vulnerable!

4.3 Software Quality Attributes

• Robustness - The system needs to be robust enough generate a realistic and accurate environment based on the locations of the sensors. The system should also be able to dynamically accommodate additional senor readings and store the values to the database.

• Reliability - The system should able to continuously run for a long duration of time (multiple hours) and not suffer from system slowdowns or crashes caused from memory leaks and zombie process.

• Portability - The software should be able to run on any Microsoft windows based platform. To set up and tear down the entire system, displays need to be set in place.

• Ease of use - Someone with little to none technical experience in the operations of electronics should be able setup and use this system by following a simple set of instructions

• Ease of Learning - The learning curve for this software should be short since the software should perform the corresponding tasks based on natural human motions.