**Software Requirements**

**Specification**

**for counting cars on lot**

**Version 1.0**

**Prepared by Joanne Helen Mana,**

**student ID: 20093755**

**Date: December 2022**

***This document based on srs\_template-ieee.doc***

Table of Contents

[Introduction and Purpose 3](#_Toc121246384)

[Product description 4](#_Toc121246385)

[Interface requirements 5](#_Toc121246386)

[Software Features 6](#_Toc121246387)

[Non-functional requirements 7](#_Toc121246388)

# Introduction and Purpose

The purpose of ‘Counting cars on lot’ (the product) is to provide information about the number of available parking spots in a parking lot in near real time, together with the temperature and time of the reading. This information must be published every time a car enters or exits the parking lot in question, and the number of available parking spots, temperature, and time must be shown on a display.

This Software Requirements Specification covers counting cars entering and exiting a parking lot and publishing the number of available spaces, time, and temperature to an MQTT server/broker.

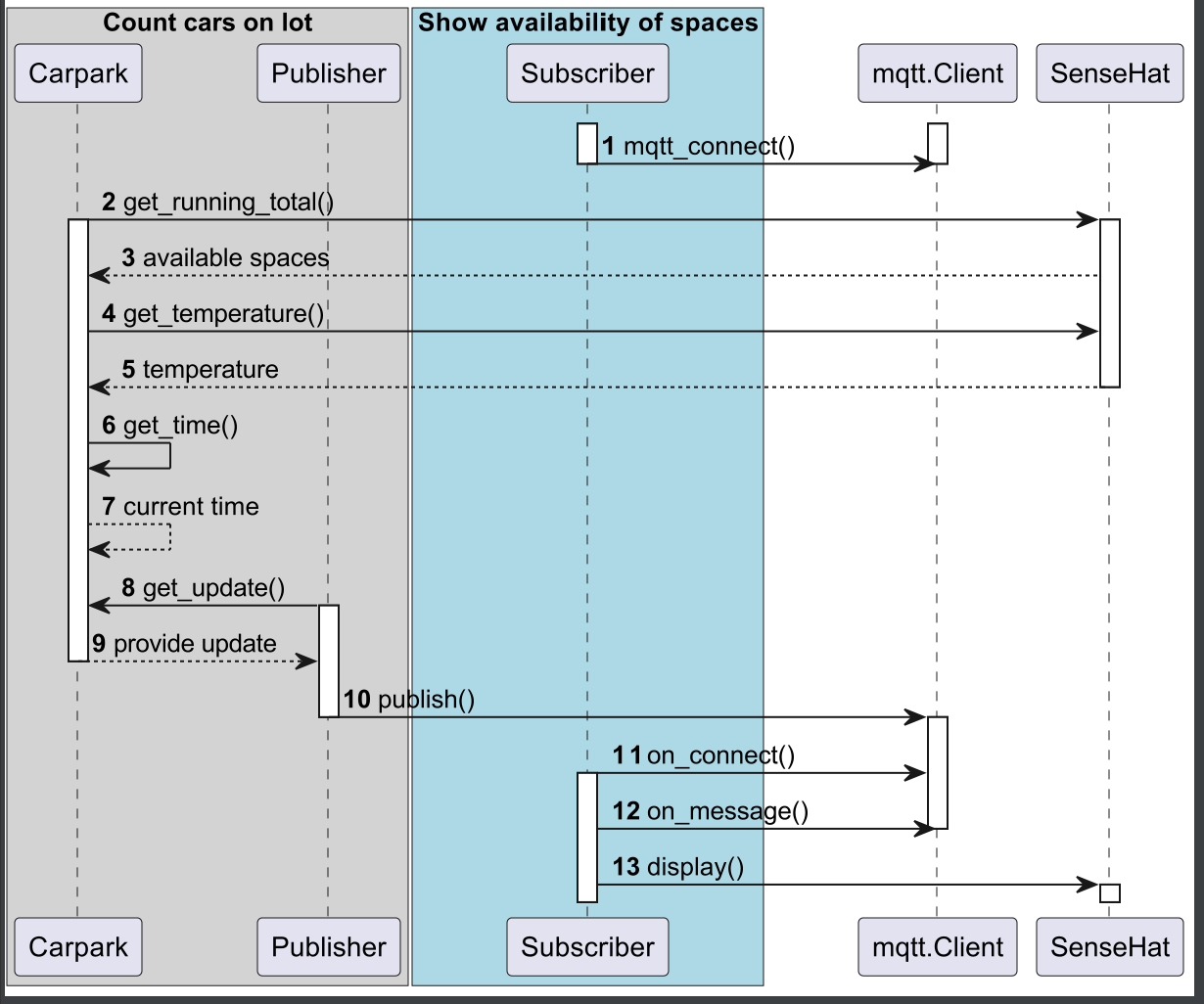
# Product description

The product will be developed as part of a project by the Department of Transport of the City of Joondalup to upgrade various public parking spaces in said district.

This product will be one of the two main components of a larger system, where the first one will keep a running total of available spaces in a parking lot at all times and publish these updates, together with time and temperature at the time of the reading, to a MQTT server. The second component of the system will consist of a program that subscribes to the MQTT server, continuously receives its updates, and shows the information on a display.

The product must use sensors to count the cars entering and exiting the parking lot, and to measure the temperature at the time of the reading.

The following diagram describes the behaviour between the two components of the system:



# Interface requirements

The product must feature the following interfaces:

**User interfaces:**

* Command line interface
* Graphical user interface

**Hardware interfaces:**

* Raspberry Pi
* SenseHat
* PC

**Software interfaces:**

* Python

**Communications interfaces:**

* Wireless internet
* MQTT server

# Software Features

List your features (the functional requirements) in the table below. Try to group features in the “Software Feature x” headings.

|  |  |
| --- | --- |
| 1. Keep a running total of cars in the parking lot | |
| 1.1 | The starting point of the program should be an initial number of available spaces and a known number of parking spaces in the parking lot. |
| 1.2 | The program must count every time a car enters or exits the parking lot, and keep a running total at all times accordingly. |
| 1.3 | The program should consider that more cars may be on the parking lot than there are parking spots – those cars will either leave the parking lot or take someone else’s spot who then leaves the lot. |
| 1. Retrieve additional data points to the number of available spaces | |
| 2.1 | The program should retrieve additional information to the reading of available spaces, such as temperature and time. |
| 1. Publish updates to an MQTT server | |
| 3.1 | The program must publish an update to an MQTT server every time there is an updated number of available spaces in the parking lot. |
| 3.1 | The update must include the number of available spaces, and the time and temperature of the reading. |

# Non-functional requirements

Non-functional requirements of the product include:

* Security Requirements
  + Channels of communication within the components of the system should be secure in a way that external parties cannot access or modify the information sent to and received by the MQTT server.
* Software Quality Attributes
  + The program should be available and functioning at all times.
  + The information being retrieved by the program and sent to the MQTT server should be available to be accessed by authorised parties.
  + A remote dashboard could be developed in order to access the information retrieved and sent by the program.