Customer re quirements research document

UWB positioning area 51



**Table of contents**

[**Research goal**](#_yeci5kc2tvv1) **2**

[**Research questions**](#_sywlz8tsuman) **2**

[Main question](#_4c1wrf24rxsq) 2

[Sub questions](#_14q8bgszemkt) 2

[**Research methods**](#_o25pccex9cs) **2**

[**Preface**](#_uklr5ncm2e7h) **2**

[**What prototypes are there and how do they work?**](#_zevbwc72c73n) **3**

[Proof of concept website](#_rbu1nb40kxbz) 3

[Mqtt local logger](#_7fis2vo56yw8) 4

[Mqtt sensor logger](#_qnkm6grkr0d2) 4

[Proof of concept data visualizers](#_tj7m8ab6d0kw) 4

[SMC\_arno](#_36z0oilupl8) 4

[Conclusion](#_48c9lijio49w) 4

[**What requirements are connected to the entire project and which ones are only for this sub-project?**](#_a59k9zvlsfxx) **5**

[Requirements](#_2jvxf17lpua7) 5

[Conclusion](#_76dljt1lh4ld) 6

[**Based on the requirements which prototype is reusable and how is this going to continue developing?**](#_b6wqvcycylmx) **7**

[**Conclusion**](#_lp4n90s3dxr6) **8**

# Research goal

The goal of this research is finding out what the customer exactly wants in the final product.

# Research questions

## Main question

Based on different prototypes, how can this project contribute to building a software solution that improves BMX sport by tracking the movement and/or other parameters of athletes?

## Sub questions

What prototypes are there and how do they work?

What requirements are connected to the entire project and which ones are only for this sub-project?

Based on the requirements which prototype is reusable and how is this going to continue developing?

# Research methods

1.1 Available product analysis, document analysis

1.2 problem analysis, explore user requirements. interview.

1.3 problem analysis, user requirements

# Preface

This research

# 

# 

# What prototypes are there and how do they work?

Given the provided one drive repository we found the following software solutions/ programs.

|  |  |
| --- | --- |
| **Project name** | **link** |
| Proof of concept website | [Bmx-in-motion](https://innosportslab-my.sharepoint.com/personal/info_innosportlabsportenbeweeg_nl/_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9pbm5vc3BvcnRzbGFiLW15LnNoYXJlcG9pbnQuY29tLzpmOi9nL3BlcnNvbmFsL2luZm9faW5ub3Nwb3J0bGFic3BvcnRlbmJld2VlZ19ubC9Fb1JsWk9OSGVlbEVzenZoTzFvMTVaMEJidXlDZG5MWW1JZERfNzNoTzZHTnp3P3J0aW1lPVlrM084X0RlMkVn&id=%2Fpersonal%2Finfo%5Finnosportlabsportenbeweeg%5Fnl%2FDocuments%2FMedewerkers%20InnoSportLab%2FUSPC%20%2D%20general%2F3%2E%20BMX%2FSMC%20Sensor%2FBMX%20In%20Motion%20%28Arno%20Borgers%29) |
| Mqtt local logger | [Programma-Jesse](https://innosportslab-my.sharepoint.com/personal/info_innosportlabsportenbeweeg_nl/_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9pbm5vc3BvcnRzbGFiLW15LnNoYXJlcG9pbnQuY29tLzpmOi9nL3BlcnNvbmFsL2luZm9faW5ub3Nwb3J0bGFic3BvcnRlbmJld2VlZ19ubC9Fb1JsWk9OSGVlbEVzenZoTzFvMTVaMEJidXlDZG5MWW1JZERfNzNoTzZHTnp3P3J0aW1lPWZ3TkZBX0hlMkVn&id=%2Fpersonal%2Finfo%5Finnosportlabsportenbeweeg%5Fnl%2FDocuments%2FMedewerkers%20InnoSportLab%2FUSPC%20%2D%20general%2F3%2E%20BMX%2FSMC%20Sensor%2FProgramma%5FJesse%2Fapplication%2Ewindows64) |
| Mqtt sensor Logger | [“Python”](https://innosportslab-my.sharepoint.com/personal/info_innosportlabsportenbeweeg_nl/_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9pbm5vc3BvcnRzbGFiLW15LnNoYXJlcG9pbnQuY29tLzpmOi9nL3BlcnNvbmFsL2luZm9faW5ub3Nwb3J0bGFic3BvcnRlbmJld2VlZ19ubC9Fb1JsWk9OSGVlbEVzenZoTzFvMTVaMEJidXlDZG5MWW1JZERfNzNoTzZHTnp3P3J0aW1lPWZ3TkZBX0hlMkVn&id=%2Fpersonal%2Finfo%5Finnosportlabsportenbeweeg%5Fnl%2FDocuments%2FMedewerkers%20InnoSportLab%2FUSPC%20%2D%20general%2F3%2E%20BMX%2FSMC%20Sensor%2Fpython) |
| Proof of concept data visualizers | [SMC-final-zonderlogo](https://innosportslab-my.sharepoint.com/personal/info_innosportlabsportenbeweeg_nl/_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9pbm5vc3BvcnRzbGFiLW15LnNoYXJlcG9pbnQuY29tLzpmOi9nL3BlcnNvbmFsL2luZm9faW5ub3Nwb3J0bGFic3BvcnRlbmJld2VlZ19ubC9Fb1JsWk9OSGVlbEVzenZoTzFvMTVaMEJidXlDZG5MWW1JZERfNzNoTzZHTnp3P3J0aW1lPWZ3TkZBX0hlMkVn&id=%2Fpersonal%2Finfo%5Finnosportlabsportenbeweeg%5Fnl%2FDocuments%2FMedewerkers%20InnoSportLab%2FUSPC%20%2D%20general%2F3%2E%20BMX%2FSMC%20Sensor%2FSMC%5Ffinal%5Fzonderlogo) & [Smc-sensor-programma](https://innosportslab-my.sharepoint.com/personal/info_innosportlabsportenbeweeg_nl/_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9pbm5vc3BvcnRzbGFiLW15LnNoYXJlcG9pbnQuY29tLzpmOi9nL3BlcnNvbmFsL2luZm9faW5ub3Nwb3J0bGFic3BvcnRlbmJld2VlZ19ubC9Fb1JsWk9OSGVlbEVzenZoTzFvMTVaMEJidXlDZG5MWW1JZERfNzNoTzZHTnp3P3J0aW1lPWZ3TkZBX0hlMkVn&id=%2Fpersonal%2Finfo%5Finnosportlabsportenbeweeg%5Fnl%2FDocuments%2FMedewerkers%20InnoSportLab%2FUSPC%20%2D%20general%2F3%2E%20BMX%2FSMC%20Sensor%2FSmc%5Fsensor%5Fprogramma) |
| SMC\_arno | [SMC-arno](https://innosportslab-my.sharepoint.com/personal/info_innosportlabsportenbeweeg_nl/_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9pbm5vc3BvcnRzbGFiLW15LnNoYXJlcG9pbnQuY29tLzpmOi9nL3BlcnNvbmFsL2luZm9faW5ub3Nwb3J0bGFic3BvcnRlbmJld2VlZ19ubC9Fb1JsWk9OSGVlbEVzenZoTzFvMTVaMEJidXlDZG5MWW1JZERfNzNoTzZHTnp3P3J0aW1lPWZ3TkZBX0hlMkVn&id=%2Fpersonal%2Finfo%5Finnosportlabsportenbeweeg%5Fnl%2FDocuments%2FMedewerkers%20InnoSportLab%2FUSPC%20%2D%20general%2F3%2E%20BMX%2FSMC%5Farno) |
| Unity data logger and visualizer | [SMC-V1.0](https://innosportslab-my.sharepoint.com/personal/info_innosportlabsportenbeweeg_nl/_layouts/15/onedrive.aspx?originalPath=aHR0cHM6Ly9pbm5vc3BvcnRzbGFiLW15LnNoYXJlcG9pbnQuY29tLzpmOi9nL3BlcnNvbmFsL2luZm9faW5ub3Nwb3J0bGFic3BvcnRlbmJld2VlZ19ubC9Fb1JsWk9OSGVlbEVzenZoTzFvMTVaMEJidXlDZG5MWW1JZERfNzNoTzZHTnp3P3J0aW1lPWZ3TkZBX0hlMkVn&id=%2Fpersonal%2Finfo%5Finnosportlabsportenbeweeg%5Fnl%2FDocuments%2FMedewerkers%20InnoSportLab%2FUSPC%20%2D%20general%2F3%2E%20BMX%2FSMC%5FV1%2E0) |

The following chapters will describe these programs on a per-program basis.

## **Proof of concept website**

This is a website where riders can login with their account and have an overview of their profile and qualities. It is only a proof of concept so it’s not 100% a finished product.

## 

## **Mqtt local logger**

This project writes sensor data to a local file. It connects to a pre-programmed address and writes all data directly to a CSV file. It should be noted that this program is functionally equivalent to the following project:

## **Mqtt sensor logger**

This project consists of 2 python files. The “filter.py” file writes sensor data to a file and the “logger.py” prints this data on screen.

## **Proof of concept data visualizers**

This project includes 2 slightly different versions of the same project. The only difference between these versions is that one contains a logo, while the other version does not. As such, these 2 projects shall both be considered in this chapter.

The main goal of this proof of concept is to capture and display data from either a remote or local device.

## **SMC\_arno**

This project does not contain a program (nor source code), but does contain two videos where measurement data is displayed on screen with the ability to track the user. It is not clear how these videos were generated, but it appears to be automatically generated.

**Unity data logger and visualizer**

This project contains two main features: reading Tag data from MQTT, and reading tag data from CSV log files. This data gets visualized in the 3d environment in Unity,

## Conclusion

In conclusion we are going to put our focus on the Unity data logger. We are going to do this by using parts of the mqtt sensor logger code. We want to make this application more object oriënted and use the SOLID principle. The rest of the prototypes are not going to be used.

# 

# What requirements are connected to the entire project and which ones are only for this sub-project?

To answer this question we first had to know what the program was going to be used for. This question was not quite clear to the customer either. After a week he came to us with his answer. The program is going to be used as a training tool for the BMX riders.

## Requirements

When we knew this we could set up and talk about some requirements. Here are the requirements we came up with and the customer agreed on:

* See the speed/Acceleration/Height/etc. (i.e relevant available data) at current timestamp
* Be able to connect 1 MQTT server for receiving pozyx data.
* Developers should be able to easily add more protocols and/or file formats for importing/exporting (live or regular) data
* for the SD6 course
  + Automatic unit testing
  + Automatically published (working) executables for windows
  + Automatically published (working) executables for android
* Export data to the following formats:
  + JSON and CSV[[1]](#footnote-0)
* Import data from the following formats/protocols:
  + JSON, CSV and MQTT
* Record measurement data for 1 user[[2]](#footnote-1) per recording device
* Offset the measurement data in time
* Offset the (positional) measurement data in space
* Be able to connect to more than 1 sensor system for receiving data
* Measure the average speed over a period of time
* User can select a timestamp and the system provides data at that timestamp (e.g. speed, position)
* Save and load user settings
* Finalized user interface
* Data synchronisation between program instances.
* ability to connect to more than 1 sensor system (though the possibility to do so should be in the design)
* The new 3d view of the park
* Video support

## Conclusion

In conclusion the above requirements are the ones for the entire project. We do not have the time in our prof assignment to finish all of these so we're going to specify what we want to do during our prof assignment in the future.

# 

# Based on the requirements which prototype is reusable and how is this going to continue developing?

|  |  |
| --- | --- |
| **Project name** | **Conclusion** |
| Proof of concept website | It looks like it could be very useful to keep track of riders and their progress but our area of expertise is not web development and therefore we cannot accurately verify this project. |
| Mqtt local logger | Not usable for our project. |
| Mqtt sensor Logger | The code is incomplete. There is also an explanation document where the code isn’t explained. In conclusion, this code is unusable for us. |
| Proof of concept data visualizers | While the code works, it is not exactly reusable. The parts of the code that handle drawing the UI and handling interaction contain hard-coded positions of which button is where, instead of using (f.e.) a more object-oriented approach (where each button keeps track of its position and size, and provides (f.e.) a callback to the main UI in case it is clicked.  We want to improve this code by basically starting over but make a more object-oriented environment where it is much easier to expand the code if necessary. It is also just a better quality code and future programmers will be able to understand the code easier. |
| SMC\_arno | Because there is no code available, this project shall not be considered for re-use. |
| Unity data logger and visualizer | Currently reading MQTT data is only partially implemented (It can only connect to tags, but not get any data from them)  Reading CSV files is fully implemented.  We are not going to reuse this code. We are however going to remake it and make it more object-oriented and maybe use parts of the existing code. We are also reusing the Unity Project itself, so we don’t have to create our own UI. |

# 

# Conclusion

In conclusion, the customer wants a training application where the bmxer can see incoming data like location, speed, height, ect. We are going to do this by remaking the data visualizer and making it more object oriented. This also includes the data handling (calculations, importing and exporting) via MQTT, CSV and JSON. This way we can make it easily expandable if you want to add more sensors for example. It also makes it easier to understand for future groups.

1. JSON data can contain multiple sensors per file; CSV will only be able to handle 1 sensor (e.g. pozyx position sensor) at a time. [↑](#footnote-ref-0)
2. A user can consist of multiple sensors (e.g. positional data combined with heart rate data). These sensors are all connected to the same user (or BMX). [↑](#footnote-ref-1)