M4 - Feature Rich Prototype

(Technical Milestone)

PEI



**Rodrigo Rocha**, 71731

**Rodrigo Relvas**, 71860

**João Pedro**, 80098

**Fábio Nunes**, 80139

**Diana Silva**, 80239

**Susana Dias**, 80410

Mestrado Integrado em Engenharia de Computadores e Telemática

Index

[**1 Client area**](#_lj2pljjii4b9) **3**

[1.1 Prototype](#_72hmkl6cco0h) 3

[1.1.1 API:](#_ifuquuvu1sqi) 3

[1.1.2 BLE:](#_ajg9q6f8f3bv) 3

[1.1.2 Data Analysis:](#_dtwghbisru24) 4

[1.2 Schedule of features/ functionalities release](#_izl2agvd52nt) 5

[**2 Personas and goals**](#_87zk4ki3hrim) **7**

[**3 Specification area (2st demo focus)**](#_s26z15djtyt5) **8**

[**Regardless of the convention used, our system will do:**](#_rl0g42wuflfw) **8**

[**4 Current architecture**](#_6x8qpx3hgxt) **8**

[**5 Tests and validation**](#_5diykff622cr) **9**

[**6 Developer area**](#_t85lr6vfdrqe) **10**

# 

# 

# 1 Client area

## 1.1 Prototype

### 1.1.1 API:

1. Login
2. Open note
3. Delete note
4. Show note
5. Create note
6. List notes
7. List process
8. Process detail
9. Stop Process

Link to the videos:

* <https://drive.google.com/open?id=1-EiE3eOr91cLdfk3VyLVa9PivTdYWwez>
* <https://drive.google.com/open?id=1jL6dPXQb8OvqCj0E5yYfednfLZuLhTMg>

### 1.1.2 BLE:

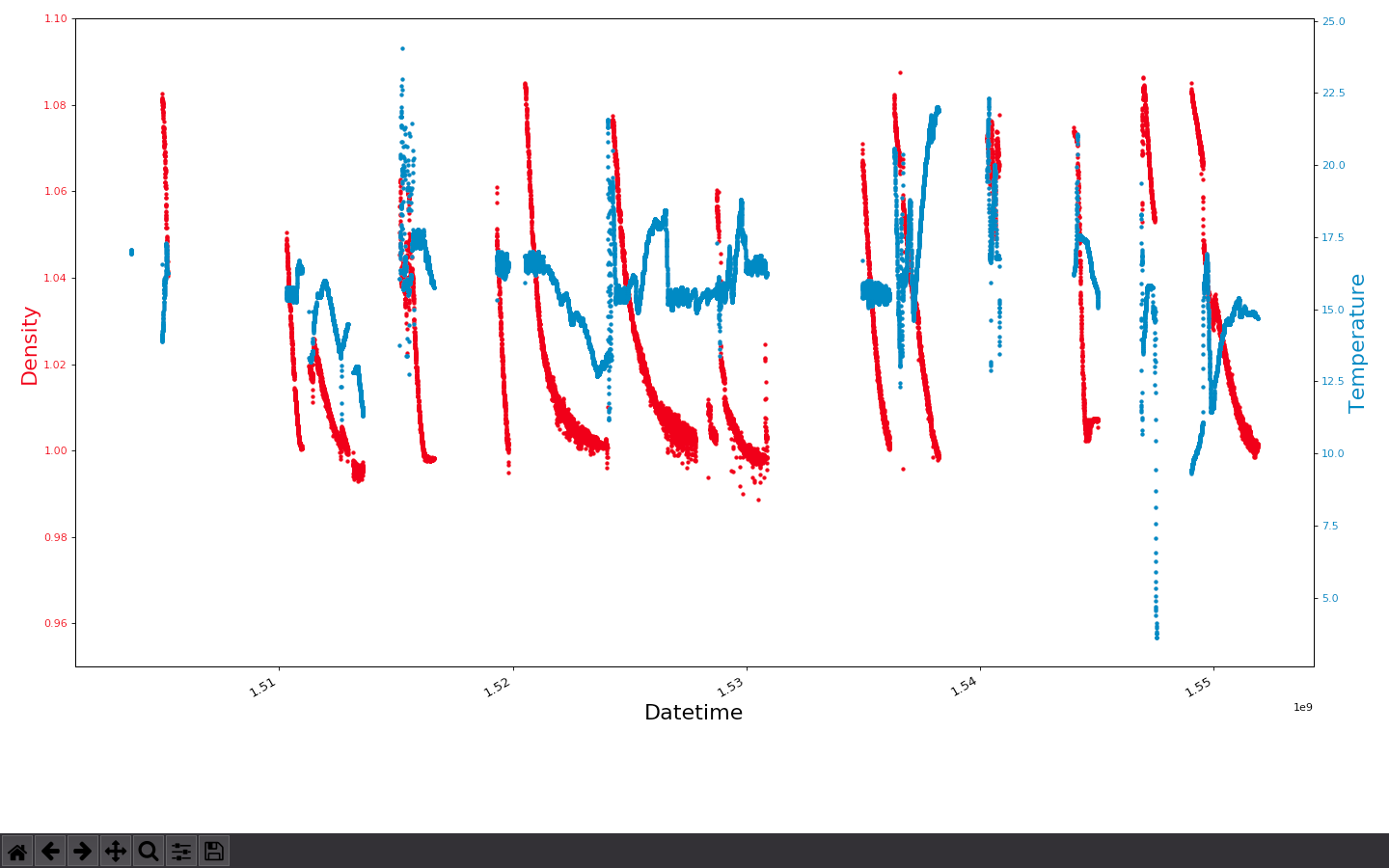
1. Open serial COM to communicate with DevBoard (ESP32);
2. Dev Board reset (ESP32);
3. Open app;
4. Scan Barcode and QR code;
5. Dev Board connect;
6. Verify app and serial COM communication status;
7. Verify received data from Dev Board;
8. Improved security with diffie-hellman algorithm

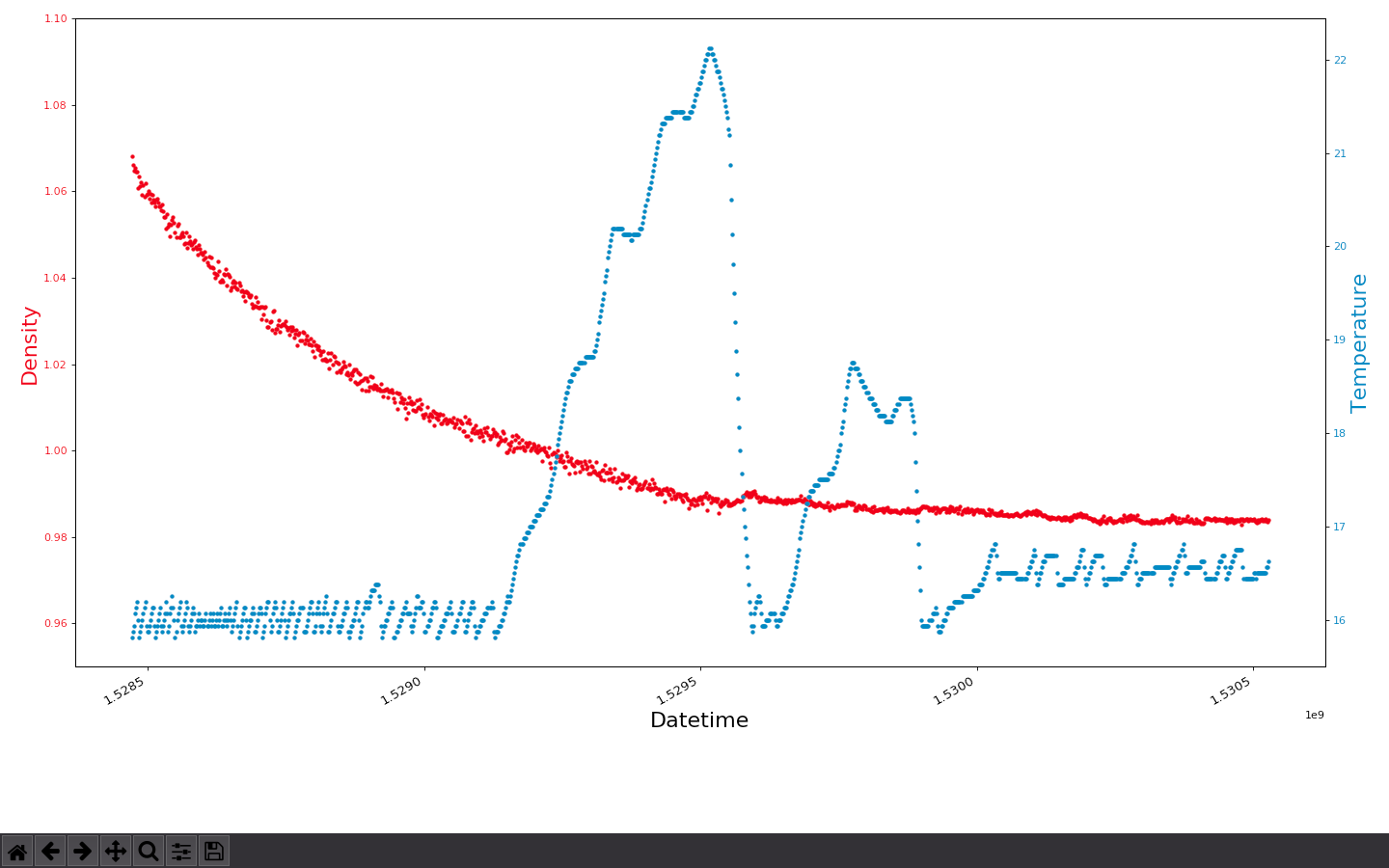
Link to the video:

* <https://drive.google.com/open?id=12QwdGgOWLTUkWM1crpgN-qDbyWmKJBeh>

### 1.1.2 Data Analysis:

1. Process detection while analysing the curves





## 1.2 Schedule of features/ functionalities release

|  |  |  |
| --- | --- | --- |
| **Week** | **Date** | **Notes** (What we propose to do) |
| 1 | 2019/2/13 | **-** |
| 2 | 2019/2/20 | Project selection |
| 3 | 2019/2/27 | Project research about BLE, Data Processing techniques and Application level design and implementation |
| 4 | 2019/3/06 | Continuous project research |
| 5 | 2019/3/13 | Further research about all technologies used |
| 6 | 2019/3/20 | Application Prototype Ready |
| 7 | 2019/3/27 | Minimal BLE system implemented |
| 8 | 2019/4/03 | Integrate Application Prototype with BLE System |
| 9 | 2019/4/10 |  |
| 10 | 2019/4/17 | Improve Application |
| 11 | 2019/4/24 | Improve BLE system protocol usage and security concerns |
| 12 | 2019/5/01 | Implement data processing techniques on the data recovered from the sensors |
| 13 | 2019/5/8 | Integrate all systems and data processing techniques in one unique prototype |
| 14 | 2019/5/15 | Rectify bugs that might be found and improve all integrated systems |
| 15 | 2019/5/22 |  |
| 16 | 2019/5/29 | Rectify bugs that might be found and improve all integrated systems |
| 17 | 2019/6/4 |  |
| 18 | 2019/6/12 | Students@DETI ( Specification + Code) |

## 

## 

# 2 Personas and goals

Liliana Costa, operator of “Adega de Borba”



Liliana Costa, born on 4th December, 1980, in Benfica, Portugal, lives alone. She’s a very active person, she likes to hang out with her friends, go to parties, wine appreciator and she’s a volunteer at the municipal shelter.

Liliana has a problem in her hip since birth, that causes her uncomfortable pain.

She works at the *Adega de Borba* and her work consists in evaluate the vats, that is a time consumption work and painful for her hip.

Most of the time spent is due to the information she has to write on paper of the vats and then transcribe it to *Excel*.

She needs an application that allows her to save time and her hip.

Scenarios

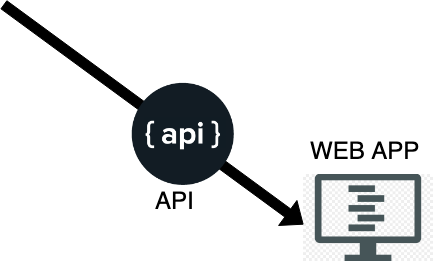
* Liliana needs to walk around the building to see the changes of the vats with the help of the application she can open the dashboard and see them.
* Liliana needs to control the time to make changes at some vats, she can open the application and set an alarm to know when she needs to go to the vat and make a change at the wine;
* Liliana has to take paper notes of changes watched she can open the application and take notes of the changes she has observed.

# 3 Specification area (2st demo focus)

# Regardless of the convention used, our system will do:

* Create a note through our app to the the web app;
* Create a process through our app to the the web app;
* Have a BLE server running on a micro controller and connect it to our app;
* Show some graphics where we can detect processes while analysing the curves.
* Login of our app with the data collected from the API;
* Read QR/barcode;
* Improve security with Diffie-Hellman algorithm;
* Receive data from microcontroller and show it in our app through BLE;

# 4 Current architecture



We have a smartphone running our app. Our app will communicate to the webapp using the API. We also have a microcontroller running a BLE server. The app will connect to the microcontroller via QR/Barcode, read data and show it in the mobile app.

# 5 Tests and validation

**We have implemented**:

* A BLE Server running on the **ESP32** microcontroller:
  + Using **ESP32** Arduino framework;
  + 1 Service with 4 characteristics used to implement a real time data transfer protocol[**Client** writes to “*characteristic 4*” a KEY, **Server** reads the sensor data and notifies the **Client**, **Client** reads the data];
  + CallBacks checking device connection status, updating data accordingly;
  + Some additional security introduced with Diffie-Hellman algorithm;
* A QRCode/BarCode Decoder on the web App to decode the BLE **Server** name:
  + The connection will be made through that name collected;
* A BLE **Client** running on the web App that connects to the BLE **Server**:
  + FlutterBlue framework used;
  + **Client** tries to connect to the board that has the localName calculated with the QRCode/BarCode Decoder;
  + *If connected* the **Client** implements the real time data transfer protocol mentioned above and updates the Home Page with the given values;
  + End user can reload the values tapping a refresh button;
  + We give the end user the option to see value in seconds intervals if they want;
* A Mobile App
  + An API driven Login:
    - HTTP Request which uses a POST method where a payload is given with the username and password;
    - HTTP Response that returns a JSON Web Token and an API Key that allows to authenticate the user;
  + A method to add/delete notes and processes through the given API in the web App:
    - HTTP Requests;
    - POST to create;
    - GET to list;
    - PATCH to edit;
    - DELETE to remove;

# 

# 6 Developer area

Until now for our work, we have implemented BLE server running in a microcontroller, a QRCode/Barcode decoder, a Mobile App and some Data Analysis.

The BLE server is running on a microcontroller called ESP32, we can connect to a BLE device via QR code and Barcode and send data from 2 characteristics (density and temperature). We can also check if the connection is up and close it when we want to. This connection security was improved by implementation of a Diffie-Hellman algorithm.

For the Mobile App we have made everything that was stipulated to this date, we could implement the method of creating, delete and edit notes and processes in the app and through the API we can visualize them in the dashboard that’s on the web app.

For the Data Analysis we are on a good path to finish the requirements in the next milestone (M5), we have already studied the data we collected and we can already filter all the things that don’t matter and create an accurate graphic and recognise processes.

## 