A logo with blue swirls

Description automatically generated

**Vision Document**

Team Oriented Project

Team D  
Data Acquisition and Development Tool

Date:

**October 2024**

**Cooperation with Institute for Mechatronics and Medical Engineering**

**Technische Hochschule Ulm**Piro, Neltje EmmaMoosbauer, Sebastian

Graf, Philipp

**Introduction**

We are planning to develop a Data Acquisition and Development Tool that will build upon the current state of the project. Our customer is currently using sensors for Photoplethysmography (PPG), a technique to measure a patient’s pulse and oxygen saturation using LEDs and photodiodes.

* These sensors will be placed on the pacifier, which is typically in motion, and an Inertial Measurement Unit (IMU) will be placed alongside the PPG sensor to reduce motion artifacts in our measurements.
* The two sensors will be controlled using an ESP32, which can communicate via Bluetooth and WLAN, though it has certain protocol limitations (which can be discussed during the kick-off). Currently, the platform will transmit the following parameters:
  + Timestamp (not synchronized with other devices)
  + One to three floating-point values for PPG measurements
  + Six to nine floating-point values for motion measurements (three axes for acceleration, rotation rate, and magnetic field)
  + Optional: a checksum (still under consideration)

The extended plan is to test the sensor setup on different parts of the body (e.g., lips, wrist, arms, legs, ears) with various individuals. This will lead to multiple platforms sending data simultaneously. Therefore, the tool will need to monitor multiple platforms at once, and our team must determine the maximum number of platforms that can be monitored concurrently

**Objectives**

**Initial User Cases**

* **Use Case 1 - Data Acquisition**
  + Using the HMI, the plan is to configure the existing smart pacifiers (Some of the   
    configuration is not available right now).
  + The plan is to also monitor the data that is received from the pacifiers in real-time (Expected sampling rate is planned around to be 500 kHz).
  + Another objective is to save the pacifiers data while respecting the data privacy regulations
    - Enable anonymization
    - Encrypted data transmission
    - More data privacy regulations topic can be added over the period of the project’s development
* **Use Case 2 - Development**
  + As it was mentioned in the project description, the tool is also meant to support our customer during development.
    - Therefore, it needs to be capable of live algorithm execution data conversion (Perhaps even providing a query API for a database?)
    - Offline mode" uses saved data instead of live data for development / tests
  + The plan is also to create a functionality in the tool where the users are able to read/filter the information that is coming from the pacifier for debugging purposes

**Further "Soft Requirements," also known as customer wishes:**

* As the project is executed at the Institute for Mechatronics and Medical Engineering (IMM), it would be beneficial to keep in mind that future students working on this tool are likely not students of informatics and thus have fewer software development skills. Therefore, maintenance and easily understandable code are prioritized higher than performance!
* It's very likely that future projects (e.g., modules, algorithms, etc.) will be developed using Python. So, compatibility with that language is preferred! This does not mean that your project must be done using Python.
* **Meeting notes after the first meeting with the customer:**
  + Agreed to use .NET Framework, while using Python for the backend algorithms for live data conversion.
  + Depending on the progress of the project, a mobile application for the same tool may be developed.

**Stakeholders**

**The Customer/Management**

1. **Prof. Dr.-Ing. Philipp Graf**Dean of the Faculty of Computer Science
2. **Moosbauer, Sebastian**-----------N/A-------------
3. **Piro, Neltje Emma**Professor

**Software Development Team:**

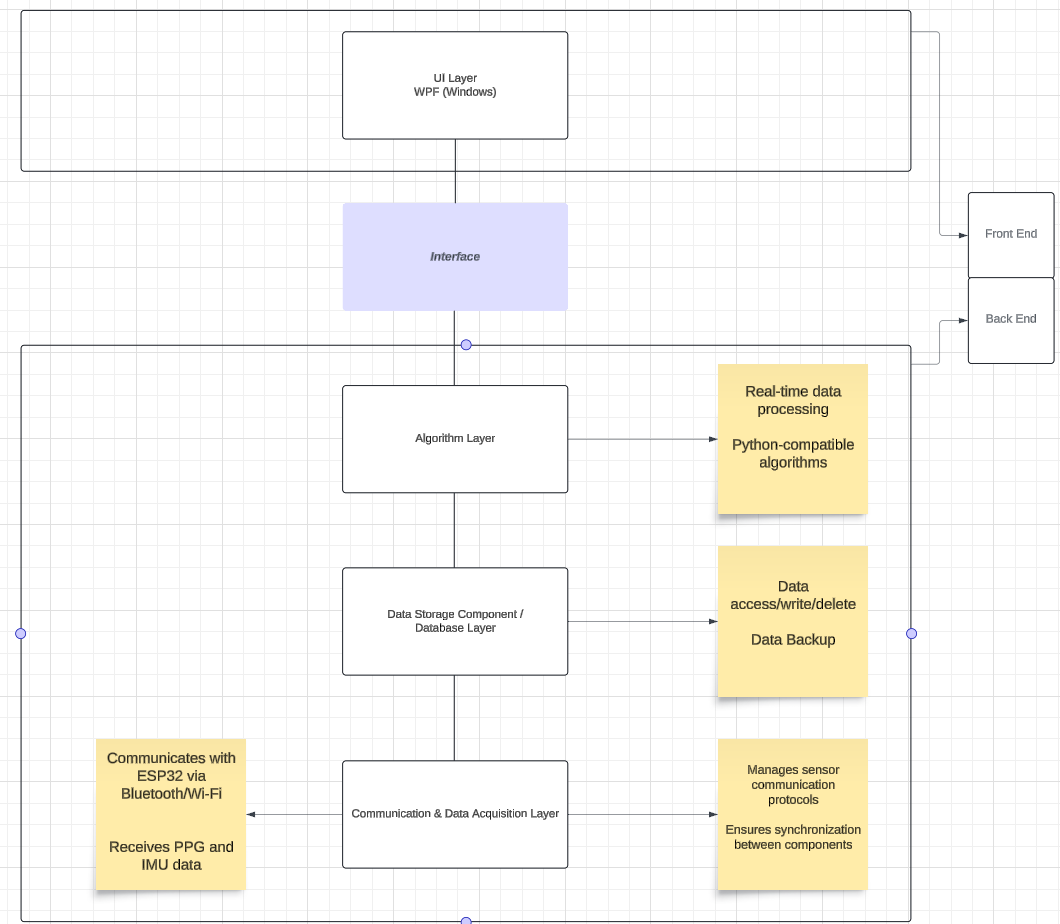
1. **Muazzam Bin Aqeel:**Main: Scrum Master & Software Architect   
   Second priority: Development in Full-Stack
2. **Aida Zhumiyeva:**Main: Product Owner & Quality Assurance  
   Second priority: Development in Automation Testing (Front-End)
3. **Simon Engel:**Main: Developer & Tester (Back-End)  
   Second priority: Software-Design
4. **Cosmin Constantin Andronescu:**Main: Developer (Front-End)  
   Second priority: Tester (Front-End)
5. **Nusret Duygu Yilmazer Seidenstücker:**Main: Integration & Development in Full-Stack Testing   
   Second priority: Quality Assurance
6. **Vincent Striegel:**Main: Developer (Front-End)  
   Second priority: Tester (Full-Stack)

**Application Requirements**

1. **Software Requirements**:  
   Operating System: Windows 10 or later (64-bit)
2. **Hardware Requirements**:  
   Processor: Intel i5 or equivalent (Quad-Core) or better  
   RAM: 8 GB (minimum) for development and testing  
   Storage: Minimum of 256 GB SSD for quick data handling and sufficient storage for large sensor data sets  
   Display: 1080p (Full HD) monitor for optimal display of UI and diagrams  
   Bluetooth/Wi-Fi: Required for ESP32 communication

**Required Tools - Development Process**

* **Front-End:**Windows Presentation Foundation (WPF) is a UI framework
* **Back-End:**   
  .NET,   
  Python
* **Integration:**Python
* **Testing:**
* **Sensor Protocols:**   
  MQTT Protocol? - TBD
* **Version Control:**Git - Repository hosted on GitHub
* **Software Process:**Jira - Scrum
* **Additional Tools:**Docker  
  Teams  
  draw.io - Diagram creation

**High-Level Architecture Component-Based Layer Diagram**

* **WPF (Desktop)** represents the front-end user interfaces.
* **The UI Interface** is shared between the two standalone points, allowing extensibility of the more frameworks in the future **.**
* **Data Acquisition Component** connects to the sensors (e.g., via Bluetooth/Wi-Fi) and gathers real-time data from the pacifier sensors.
* **Algorithm Layer** processes that data (e.g., removing motion artifacts) and is modular so that Python-based algorithms can also be plugged in.
* **Data Storage** handles both local and cloud storage of the processed data, ensuring it is anonymized and encrypted.
* **Communication & Integration Layer:** is responsible for handling all communication between sensors and data storage, ensuring seamless interaction across components and platforms.

**Documentation and Future-Plans**

As our architecture supports future scalability, including if required mobile and multi-platform development. We combine Scrum with Rational Unified Process artifacts to ensure both agility and thorough documentation. Here's a breakdown of key documents we will produce:

* **Use Case Diagrams**: Visual representations of user interactions with the system.
* **System Architecture Diagram**: A detailed overview of the software components and their relationships.
* **User Manual**: Step-by-step instructions for end-users to interact with the tool.
* **Technical Documentation**: Covers system architecture, data flow, and design decisions for developers.
* **Test Plan**: Documentation outlining the testing strategy, cases, and expected results.
* **Optional - Maintenance Guide**: Instructions for future teams on how to extend or update the system.
* **Software development / Scrum Artefacts**:
  + Sprint Review Notes
  + Sprint Retrospective Notes
  + Lessons Learned
  + Dailys
  + Sub Team Meeting - (To Solve blockers & implement Hotfixes)
  + Coding Standards
  + Design Pattern explanation