Biodex Data Management System

Semester: 4-5

### Group Members:

|  |  |  |
| --- | --- | --- |
| First Name | Last Name | Email |
| Paul | Luley | be18b096@technikum-wien.at |
| Tobias | Gehrer | be18b095@technikum-wien.at |
| Rishad Arnab | Howlader | be18b120@technikum-wien.at |

### Group Spokesperson:

|  |  |  |
| --- | --- | --- |
| First Name | Last Name | Email |
| Jonathan | Gehmayr | be18b019@technikum-wien.at |

### Project Supervisor:

|  |  |  |
| --- | --- | --- |
| First Name | Last Name | Email |
| Stefan | Sauermann | stefan.sauermann@technikum-wien.at |

### Version of the Document:

|  |  |  |  |
| --- | --- | --- | --- |
| .v# | Name of Author | Date | Changes performed |
| .v1 | Jonathan Gehmayr | 05.04.2020 | Added Project Goals, User requirements, Naming conventions and Workpackages |
| .v2 | Paul Luley | 06.04.2020 | Minor adjustments of naming standards |
| … |  |  |  |

Table of Contents

[Group Members: 1](#_Toc18923250)

[Group Spokesperson: 1](#_Toc18923251)

[Project Supervisor: 1](#_Toc18923252)

[Version of the Document: 2](#_Toc18923253)

[1. Introduction 3](#_Toc18923254)

[Vision 3](#_Toc18923255)

[Goal of the project 3](#_Toc18923256)

[State of the art 3](#_Toc18923257)

[Analysis 3](#_Toc18923258)

[2. Materials and Methods 3](#_Toc18923259)

[Project structure, Workpackages 3](#_Toc18923260)

[Technical Documentation for software projects 3](#_Toc18923261)

[Technical Documentation for hardware projects 3](#_Toc18923262)

[Risk management 4](#_Toc18923263)

[3. Results 4](#_Toc18923264)

[4. Discussion 4](#_Toc18923265)

[5. References 4](#_Toc18923266)

# 1. Introduction

## Vision

Text

## Goals of the project

The overall goal of the project is to implement an application that can handle the documented user requirements. To fulfil that goal the project was subdivided in minor aims that describe which parts are needed to be implemented.

**Database:**

1. The first goal is to implement a database that is able to save the data that is produced in measurement activities.

**GUI:**

To conncect all components of the application a GUI is used. Subgoals of the application are following:

1. Use the current MATLAB GUI as base for the new GUI:

The current MATLAB GUI has been used for various measurements and can analyse data and save it. Problems of the software are among others that there is no way to save the data to a database system. Moreover, the software does not work correctly. According to the *Biodex\_Guidelines* document the software is not running stable and the laptops of students are often overcharged by the workload of the current program. To fix these problems the GUI has to be revised, or completely new implemented. As implementing a new GUI seems to be the better solution the current GUI is used as base for the new one, which will be written in MATLAB or as C# .NET application.

1. Usability in daily use in laboratory:

As mentioned above the GUI does not work appropriately. The work in the laboratory shall not be interrupted by having issues using the Biodex data management system. In order to grant frequently use of the Biodex the produced software has to be tested carefully by multiple test procedures.

1. Live plotting of muscle force, velocity and angle:

This function is already implemented in the old GUI and will also be implemented in the new one. The old function will be used as base for the new one. To ensure an improvement the new function will run more fluently than the old one. This is also listed as a problem in the document *Biodex\_Guidelines*.

1. Data export to database:

The new GUI shall include the function of exporting the measured data to the database.

1. Start and Stop of measurement:

To ensure that the user can decide the start and stop time of measurement, the GUI shall include a start and stop button.

1. Reading of data sent by microcontroller:

The GUI shall read the data sent by microcontroller. The function shall be connected to the start and stop button. This means the data stream will be started when *start* is pressed and stopped when *stop* is pressed.

1. Inserting patient data:

The GUI shall include the function to insert patient data. Patient data contains for example data like sex, height or weight. The data that will be inserted is referenced to *HL7 Implementation Guide for CDA® R2: Allgemeiner Implementierungsleitfaden für ELGA CDA Dokumente.*

1. Individual adaptation of plots in the GUI:

In the old GUI this function is implemented by using checkboxes for each plot that shall be displayed. The new GUI shall include this function as well. According to document Biodex\_Guidelines it was not possible to view each plot independently. The new GUI shall fix that issue.

1. Observation of microcontroller:

The new GUI shall include the function of observing the microcontroller. This means it shall be displayed if the microcontroller is connected to the serial port, which serial port is used, if data is available and the current working status of the microcontroller.

1. Correctly processed data:

The collected data shall be processed correctly, and that process has to be documented. A current problem of the old GUI is that there is no information on how the gathered data is transformed.

**Extensibility:**

Another goal of this project is to implement an application that can easily be extended by other developers to add further features:

1. Application documentation:

To ensure that people understand what happens within the application, a manual has to be written besides the project documentation. The manual shall include how to use the application, which functionalities are included, how the data is processed and which problems can occur.

1. Code annotation:

For developers it is easier to extend an application if the code is commented properly. Thus, an appropriate annotation of code is necessary to implement this application.

**Optional-Goals:**

1. Comparison of user data:

With a tool providing a function of comparing user data it would be possible to differentiate current measurements with data of previous measurements. This enables users of the application to determine whether their measurement results are similar or if measurements were performed under different adjustments.

1. E-card authentication system:

By extending the application with an e-card authentication system, the process of data acquisition and saving in a database can be more secure, because a user can only export his data into the database. Besides of that, the patient data could be loaded automatically and would not have to be inserted manually.

**Non-Goals:**

1. Microcontroller system shall not be adapted:

Purpose of this project is to implement a data management system for the Biodex System. Therefore, the microcontroller and the whole embedded system shall not be adapted. Tough the microcontroller has to be revised referencing *Biodex\_Guidelines*, which says that there are occurring problems dealing with the performance of the microcontroller. Adjusting the microcontroller would need additional skills that would go beyond the scope.

## State of the art

* products available on the market
* technologies in use
* research reports, recent findings
* future trends, visions

## Analysis

## User requirements

The work environment of the Biodex data management system will be in student laboratories for measurement of diverse physical exercises. As the Biodex System can be used in a very broad field of activities, the data management system shall be able to handle various functions of the Biodex System. Therefore, the produced data has to be saved in a database. To be able to observe the current data acquisition the ongoing test must be graphical monitored. Observed values on the GUI will be the muscle force, velocity and angle. Some of the tasks the Biodex data management system has to deal with are maximum force measurements, EMG data collection, Ergonomics and everyday measurement tasks as well as EMG force data analysis and evaluation.

Scenarious of use

* scenarios of use (probably not in the following order): narrative, basic scenario, variants, details

# 2. Materials and Methods

## Project structure, Workpackages

Workpackages:

The project members meet every week to discuss the progress of the project. The meetings include checking the development of the workpackages, defining new workpackages as well as discussing general ideas. The discussions are documented in meeting minutes.

**Paul Luley:**

**General workpackage:**

Database

Database GUI interface

**Specified tasks:**

Work package 30.3.:

Setting up GitHub

Setting up individual Timetable

ER-diagram for database

**Tobias Gehrer**:

**General workpackage:**

GUI

Data Acquisition

**Specified tasks:**

Work package 30.3.:

Setting up GitHub

Setting up individual Timetable

Data identification

**Rishad Arnab Howlader:**

**General workpackage:**

Database

Database GUI interface

**Specified tasks:**

Work package 30.3.:

Setting up GitHub

Setting up individual Timetable

ER-diagram for database

**Jonathan Gehmayr:**

**General workpackage:**

Organisation

GUI

Data Acquisition

**Specified tasks:**

Work package 30.3.:

Setting up GitHub

Setting up Wiki

Setting up individual Timetable

Writing first version of group paper

Email exchange about unclarities with Iris Nemec

Data identification

## Naming standards:

For a better overview of the whole program some common naming conventions are used. The rules are easy to understand and help developers to identify the data type of a variable. In general, the first letter of a word is written capital. In Table 1 the naming conventions of variables and a few examples are listed.

|  |  |  |
| --- | --- | --- |
| Datatype | Convention | Example |
| Number | Prefix 'n' | nMyNumber |
| String | Prefix 's' | sMyString |
| Boolean | Prefix 'b' | bMyBoolean |
| List | Prefix 'a' + Suffix ‘List’ | aMyObjectList |
| Array | Prefix 'a' + Suffix ‘Array’ | aMyObjectArray |
| Constant | Capital letters with underscores | MY\_CONSTANT |

Table 1: Naming conventions for data types

Methods:

Methods need a comment block above that describes the function of the method. Moreover, the content of the comment block needs to contain, which parameters the method is working with, how they are used and which values get returned. The name of the method needs to describe the function of the method. To detect methods in code blocks a method has the Prefix 'm'.

Documents:

Database:

* detailed descriptions of tasks, to-do's, responsibilities (who does what by when)
* expected results, milestones

## Technical Documentation for software projects

* system and detail architecture (possibly the "UML"-like part)
* information models
* system architecture (devices, servers, etc involved)
* timing diagrams(?) (what happens in what order)
* transaction diagrams(what happens between "actors")
* ...
* descriptions of blocks
* description of test procedures for each block and for the complete system

## Technical Documentation for hardware projects

* Block circuit diagram
* Circuit diagram, description, specifications, description of tests for each block and for the complete system
* Layout, part list and part placement diagram
* References to all relevant component datasheets, datasheets shall be handed in as separate pdf files together with the report

## Risk management

* what types of risks are identified
* how shall they be handled

# 3. Results

* Results of tests as described in “materials and methods”
* Time and costs spent on the project

# 4. Discussion

* Summarises achieved results
* relates them to requirements / analysis
* highlights what has been successfully achieved: successes and failures
* possible future steps

# 5. References

* Clear and concise and consistent citation and referencing