MARKLAR

Software Requirements Specification

version 1.0

18.7.2016

Ing. Jan Lochman
<jan.lochman@vezas.cz>

Lead Software Engineer

Introduction

Purpose

This document is written to define specifications of software, which will be delivered together with instrument for glass plasticity and resistivity measurements by VEZAS Company.

Scope

Software will be developed as one single-running desktop application, which will be called *Marklar*, which will be designated to control one instrument only.

Marklar will automatically connect to instrument throughout TCP/IP protocol. It will perform automatic measurement procedure. Measured data will be stored so they can be browsed and displayed later. Data will be allowed to be exported either in csv format (for further processing e.g. in MS Excel) or in pdf format as measurement report.

Definitions

Experiment instance = measurement

- this represents one measurement of resistivity or plasticity on one sample.

Marklar

- software name

Specific Requirements
Marklar will be designed in a way to fulfill all requirements stated in this section.

External Interface Requirements
These requirements must be fulfilled before Marklar is installed
1. Computer with stable user-preferable OS (can be delivered by VEZAS Company)
2. Computer with Java SE version 8.64 or newer
3. MySQL server (can be installed on computer itself)
4. Instrument will be equipped with ethernet switch, connecting computer with devices in instrument throughout ethernet cable.
5. Local Area Network connecting computer with devices in instrument

Functional Requirements
These requirements are a short overview of what Marklar will be able to do

---

Sample

1. User will define sample by sample name and additional description
2. Each sample will obtain creation date and unique id number automatically
3. User can search for a sample according to its name, id and creation date

---

Experiment instance

1. Sample will contain arbitrary number of experiment instances, which represents one particular measurement
2. Concrete instrument will be assigned to experiment instance
3. User can define an operator, which performs measurement, write commentary to measurement and override default measurement settings
4. Experiment instance contains measured data and measurement results, which will be displayed in graph and int table
5. Experiment instance will change its state according to diagram shown in Appendix
6. Results will be evaluated automatically once the experiment instance is finished
7. When performing more than one measurement on a sample, statistic from partial experiment instances will be evaluated
8. Actually measured data will be displayed on a special panel

---

Settings

1. Connection settings
    - user can change device IP addresses
    - user has an overview of components used and is allowed to read/write its values
2. Experiment instance settings
    - these settings include initial temperature, read out interval and many others
    - user can set default settings (which will be applied to new experiment instance)
    - user can change these settings for each experiment instance
3. Temperature correction
    - user can define correction table for temperature sensor (for calibrated sensor)
4. Calibration
    - user can perform instrument calibration
5. Protocol settings
    - user can define settings specific for protocol

Language

1. Default and only language provided will be English
2. To add another language, approximately 500 phrases has to be translated
3. Program allows switching between different languages

Outputs

1. Measured data can be exported in csv format (for further processing e.g. in MS Excell)
2. Measurement protocol in pdf format can be created

Error handling

1. There will be a simple graphical logger inside Marklar
2. Marklar itself will create log file logging stack trace of every error. This will allow quick and simple error removal

Use Cases

The most common scenarios will be measurement performance and data export. Here are, in a nutshell, basic steps which must be taken to perform these tasks.

Measurement

1. User selects sample / creates new sample
    - sample is defined by name and description
2. User selects experiment instance / creates new experiment instance
    - user defines instrument, which will perform measurement
    - user defines initial temperature, read out interval and other required settings
3. User locks measurement (so its settings cannot be changed further)
4. User starts measurement
5. Measurement runs automatically (user interaction with program is reduced to a necessary minimum)
6. User terminates measurement and decides, if it was successful or unsuccessful
7. Measurement is evaluated automatically

Data Export

1. User selects sample
2. User selects experiment instance
3. If selected experiment instance is marked as successful, a export tab is shown
4. User chooses directory for export and desired formats (csv or pdf) and initiates export

Logical Database Requirements

1. MySQL server
2. Available disk space of 1 GB

Appendices

State Transition Diagram