

System Architecture

Specification

Template

Huttunen Pekka

20 January 2025

**Kajaani University of Applied Sciences**

PL 52, Ketunpolku 1  
87101 KAJAANI  
kajaanin.amk@kamk.fi

Copyright 2024 KAMK. All rights reserved.

Contents:

[1 Introduction 4](#_Toc188280720)

[1.1 System Overview 4](#_Toc188280721)

[1.2 System users 4](#_Toc188280722)

[2 System Requirements 5](#_Toc188280723)

[3 System Architecture 6](#_Toc188280724)

[3.1 Components 6](#_Toc188280725)

[4 System information model 7](#_Toc188280726)

[4.1 Measurement database 7](#_Toc188280727)

[4.2 Configuration database 8](#_Toc188280728)

[4.3 Model repository 8](#_Toc188280729)

[5 System interfaces 9](#_Toc188280730)

[5.1 Measurement API 9](#_Toc188280731)

[5.2 database interface 9](#_Toc188280732)

[5.3 Model repository interface 9](#_Toc188280733)

[5.4 WebUI status interface 9](#_Toc188280734)

[5.5 WebUI control interface 9](#_Toc188280735)

[5.6 Graptical User Interface (webUI) 9](#_Toc188280736)

[6 System Functional Description 11](#_Toc188280737)

[6.1 Use Case Diagram 11](#_Toc188280738)

[6.2 Sequence Diagrams 11](#_Toc188280739)

[6.2.1 Use Case 1 12](#_Toc188280740)

[6.2.2 Use Case 2 12](#_Toc188280741)

[6.2.3 Use Case 3 12](#_Toc188280742)

[6.2.4 UC4, Send Measurement Data to Database 12](#_Toc188280743)

[7 System testing 13](#_Toc188280744)

[7.1 Integration test cases 13](#_Toc188280745)

[7.1.1 Integration Test Case 1 13](#_Toc188280746)

[7.1.2 Integration Test Case 2 13](#_Toc188280747)

[7.2 System test cases 13](#_Toc188280748)

[7.2.1 System Test Case 1 13](#_Toc188280749)

[7.2.2 System Test Case 2 13](#_Toc188280750)

[7.3 Acceptance test cases 14](#_Toc188280751)

[7.3.1 Acceptance Test Case 1 14](#_Toc188280752)

[7.3.2 Acceptance Test Case 2 14](#_Toc188280753)

[References 15](#_Toc188280754)

[Appendix 1 16](#_Toc188280755)

# Introduction

This is a template of System Architecture Specification. It is very simple, so that most important things will be added to the document. As you see, it includes several pictures.

## System Overview

Add here short description about the system and high level illustration picture.

A white square with black text

Description automatically generated

Figure . Use Case Diagram example.

## System users

Name and describe here all system user roles.

Table . System users

|  |  |
| --- | --- |
| **User role** | **Description** |
| Water Quality Manager | Person who uses system every day to control water quality |
| Sensor Installer | Person (usually subcontractor) that makes the sensor installation |
| Administrator | System admin, makes installations / system upgrades and manages user rights |

# System Requirements

Refer here to System Requirement Specification document [1]. Or any other place, where requirements can be found, e.g. Requirement Management tool.

If there are requirements that affects to the system architecture, insert references for those here, e.g. security requirements or performance requirements.

# System Architecture

Figure 2 shows an example picture about system architecture.

Diagram of a diagram

Description automatically generated

Figure . System Architecture example.

## Components

Name and describe here all system components that are presented in system architecture picture above.

Table . System Architecture Components

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| sensors | Sensor TypeA1234567 | Water quality sensor with LTE interface. |
| measurementServerSW | Linux server SW | Python code to handle measurement data. |
| measurementDB | MariaDB | Water quality measurement database  (> 1k sensors and > 1M measurements/sensor) |
| configurationDB | MariaDB | Water quality sensor configuration database (supports > 1k sensors) |
| modelsRepository | Gitlab | AI models with version info. |
| webGUI | NGINX server map view based on  https://leafletjs.com/ | Map view based management GUI for sensors |

# System information model

Describe System information model here or refer to System information model document [2]. Or any other place, where information model can be found, e.g. relevant tool.

## Measurement database

Table . Measurements table

A screenshot of a computer

Description automatically generated

Table . Sensors table

A screenshot of a computer

Description automatically generated

## Configuration database

## Model repository

# System interfaces

Introduce here all system interfaces.

## Measurement API

System interface to save measurement data provided by water quality sensors.

See Measurement API Interface Specification [3] more detailed description.

## database interface

SQL database interface. Accessed with library https://www.sqlalchemy.org.

See database specification documents for used tables and parameters.

## Model repository interface

Gitlab repository. Models can be downloaded with model url.

## WebUI status interface

<status.csv> file to store latest status information.

See status.csv file format specification [4] more detailed description.

## WebUI control interface

<control.csv> file to store latest control parameters.

See control.csv file format specification [5] more detailed description.

## Graptical User Interface (webUI)

See figure 3 for system graphical user interface example.

A screenshot of a computer

Description automatically generated

Figure . GUI example.

# System Functional Description

Use Unified Modeling Language (UML) to describe system functionality. For short introduction to UML see [6].

## Use Case Diagram

Figure 4 shows an example Use Case diagram.

A screenshot of a device

Description automatically generated

Figure . Use Case Diagram example.

## Sequence Diagrams

Add at least one Sequence Diagram for all use cases. If needed you can separate happy case and unhappy cases to different diagrams.

### Use Case 1

### Use Case 2

### Use Case 3

### UC4, Send Measurement Data to Database

Figure 5 shows an example Sequence diagram.

A screenshot of a black screen

Description automatically generated

Figure . Sequence Diagram example.

# System testing

Description how system will be tested in module levels, integration level and system level. Tools and interfaces should be included in chapters 3.1 Components and 5 System interfaces.

## Integration test cases

### Integration Test Case 1

|  |  |
| --- | --- |
| Test case ID | *ITC#1* |
| Name | *Sensor – server integration* |
| Description | Send water quality report from Sensor to Server over LTE. |
| Test environment | 1 sensor  1 server  LTE connection |
| Test procedure | 1. Start server 2. Start sensor 3. Send 10 quality reports from sensor 4. Read server water quality DB. |
| Acceptance criteria | 1. Sensor – server connection status is: OK 2. All 10 messages in water quality DB are correct. |

### Integration Test Case 2

## System test cases

### System Test Case 1

### System Test Case 2

## Acceptance test cases

### Acceptance Test Case 1

|  |  |
| --- | --- |
| Test case ID | *ATC#1* |
| Requirement | *REQ23 - System must support 100 simultaneous sensors* |
| Name | *Simultaneous sensors* |
| Description | Test that N simultaneous water quality sensors can send water quality reports simultaneously to server. |
| Test environment | 10 sensor  1 server  LTE connection |
| Test procedure | 1. Start server 2. Start sensors (all at the same time) 3. Check that all sensor connections’ status is OK 4. Wait 10 min. 5. Read all server water quality DB 6. Check water quality from server dashboard |
| Acceptance criteria | 1. All sensor – server connections’ status is: OK 2. Missed packet rate < 0.1% 3. Quality levels in dashboard are correct. |

### Acceptance Test Case 2

# References

1. KAMK, Water Measurement System Requirements, P.Huttunen. version 2.0.0.
2. KAMK, Water Measurement System Information Model, P.Huttunen. version 2.0.0.
3. KAMK, Measurement API Interface Specification, P.Huttunen. version 1.6.0.
4. KAMK, status.csv file format Specification, P.Huttunen. version 1.6.0.
5. KAMK, control.csv file format Specification, P.Huttunen. version 1.6.0.
6. Wikipedia, unified modeling language (UML). <https://en.wikipedia.org/wiki/Unified_Modeling_Language>, version 2020.08.31.

# Appendix 1

Add here