TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MIIKKA HAURINEN

LABORATORY INFORMATION MANAGEMENT SYSTEM MODULE FOR INTEGRATION INTO MANUFACTURING OPERATIONS MANAGEMENT SYSTEM

Master of Science thesis

Examiners: Prof. Matti Vilkko,
D.Sc. David Hästbacka
Examiners and topic approved by the
Faculty Council of the Faculty of
Engineering Sciences
on 29th March 2017

ABSTRACT

MIIKKA HAURINEN: Laboratory information management system module for integration into manufacturing operations management system
Tampere University of Technology
Master of Science thesis, 78 pages, 12 Appendix pages
February 2018
Master's Degree Programme in Automation Technology
Major: Process Automation
Examiners: Prof. Matti Vilkko, D.Sc. David Hästbacka
Keywords: LIMS, Information System, Systems Engineering, Microservices

Laboratory information management systems (LIMS) enhance the productivity of the laboratories as the data throughput and the market demand increase. Integrating the laboratory with a manufacturing operations management system (MOMS) is the key to improve the productivity in the whole production chain. In the field of the mining and metal industry the laboratory can be located at a different location. Thus a software module functioning as an integrator transferring and scheduling the data between the laboratory and the process is a justifiable investment. This Master's Thesis documents the feature design and specification of a LIMS software module by following the method of the system development life cycle.

This Master's Thesis consists of the following parts: the theory of the method, the overview of LIMS systems, the designing of the LIMS module and finally documenting the specifications and realization of the features and the requirements. A comprehensive description of a complex information system like LIMS requires use of different models describing the system from several perspectives. The models used in this Master's Thesis are the user journey map, the requirement specification list, the database diagram and the system interface diagram.

The research problem stated, that various process plants with a laboratory have some LIMS functionalities that are yet not uniform. The question was about finding methods and results of the set of features and architecture for a platform independent, uniform LIMS module that can be used as the actual LIMS itself as well. The feature and requirement specification was designed by studying the functionality of the existing systems and by interviewing the key users and the stakeholders. The platform independency was solved with a microservice architecture, that is a set of coherent smaller services, that can be added seamlessly into the LIMS module and connected to any external system.

TIIVISTELMÄ

MIIKKA HAURINEN: Laboratorion tiedonhallintajärjestelmäpalvelun integraatio tuotannon toimintojen hallintajärjestelmään
Tampereen teknillinen yliopisto
Diplomityö, 78 sivua, 12 liitesivua
Helmikuu 2018
Automaatiotekniikan koulutusohjelma
Pääaine: Prosessiautomaatio
Tarkastajat: Prof. Matti Vilkko, TkT. David Hästbacka
Avainsanat: LIMS, Informaatiojärjestelmä, Järjestelmäsuunnittelu, Mikropalvelu

Laboratorion tiedonhallintajärjestelmät (eng. laboratory information management system, LIMS) tehostavat laboratorioiden tuottavuutta, samalla kun tiedon läpikulku ja markkinoiden vaatimukset kasvavat. Laboratorion liittäminen tuotannon toimintojen hallintajärjestelmään (eng. manufacturing operations management system, MOMS) on ratkaisu tuottavuuden tehostamiseen koko tuotantoketjussa. Kaivosteollisuudessa laboratorio voi sijaita eri paikassa kuin tehdas. Täten ohjelmistopalveluun investointi on perusteltua tehokkaaseen tiedon siirtoon ja aikatauluttamiseen. Tämä diplomityö kuvaa LIMS-ohjelmiston ominaisuuksien suunnittelun ja määrittelyn seuraten ohjelmistokehityksen elinkaaren menetelmää.

Tämä työ sisältää seuraavat osat: käytetyn menetelmän kuvaus, yleiskuvaus LIMS-järjestelmistä, LIMS-palvelun suunnittelu ja lopuksi määrittelyn toteutuksen dokumentointi. Kattava kuvaus LIMS:n kaltaisesta monimutkaisesta tietojärjestelmästä vaatii monien mallien käyttöä järjestelmän kuvaukseen eri näkökulmista. Mallit joita tässä diplomityössä on käytetty ovat: käyttäjän polun kartta, vaatimusmäärittelylista, tietokantakaavio ja järjestelmärajapintakaavio.

Tutkimusongelma perustuu ilmiöön, että monilla tuotantolaitoksilla on käytössään joitakin tiettyjä mutta toisistaan eroavia LIMS:n ominaisuuksia. Tutkimus edellytti sellaisten ominaisuuksien ja ohjelmistorakenteen tutkimisen menetelmien selvittämistä, että saadaan kehitettyä järjestelmäalustariippumaton ja yhdenmukainen LIMS-palvelu, jota voisi käyttää tarvittaessa myös varsinaisena LIMS:na. Vaatimusmäärittelyn suunnittelu tehtiin tutkimalla olemassa olevan järjestelmän toimintaa ja haastattelemalla käyttäjiä ja sidosryhmiä. Järjestelmäalustariippumattomuus ratkaistiin käyttämällä mikropalvelurakennetta, joka koostuu pienistä yhtenäisen kokonaisuuden toteuttavista pienemmistä palveluista, joita voi lisätä LIMS-palveluun ja yhdistää mihin vain ulkoiseen järjestelmään.

PREFACE

This Master's Thesis was made for a system supplier organization in Espoo, Finland. The organization also came out with the real project need and inspiration for the research. The development will continue on the following projects with a need for the LIMS module like the one developed as the result of the research of this Thesis. I'm looking forward to the following challenges related to the topic of this Thesis during my continuing career.

First, I would like to thank Ahti Rossi and Aki Kössilä for providing the possibility to work in this interesting project, for the valuable professional advice and for helping me whenever I had problems during this project. I would like to thank my examiners Prof. Matti Vilkko and D.Sc. David Hästbacka for the guidance and advice for writing and producing scientific text.

Special thanks to my friends for making the years in the Tampere University of Technology unforgettable, you know who you are.

Finally, I want to thank my family for supporting and believing in me through all these years. Especially I want to thank my mom and my grandparents for the financial and moral support through all of my studies, without you this would not have been possible.

In Espoo, Finland, on 23. January 2018

Miikka Haurinen

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

API	Application programming interface
CASE	Computer aided system engineering
CSV	Comma separated values
DCS	Distributed control system
ELN	Electronic laboratory notebook
ERP	Enterprise resource planning
FTP	File transfer protocol
HTTPS	Secure hypertext transport protocol
IDE	Integrated development environment
I/O	Input and output
IPSec	Internet protocol security
LIMS	Laboratory information management system
MES	Manufacturing execution system
MIS	Management information system
MOMS	Manufacturing operations management system
ODBC	Open database connectivity
OPC	Open platform communications
SDLC	Systems development life cycle
SMTP	Simple mail transfer protocol
SOA	Service-oriented architecture
SQL	Structured query language
SSL	Secure sockets layer
TCO	Total cost of ownership
TCP/IP	Transmission control protocol / Internet protocol
TLS	Transport layer security
UI	User interface
UP	Unified Process
UX	User experience
XML	Extensible markup language

1. INTRODUCTION

Manufacturing organizations need to adapt to conditions on the market and make the decisions that enhance the profitability. New technologies increase quality and especially quantity of an information. A management of the data is proved to be needed to match the demand on the market and to overcome significant challenges as the volume of the data in laboratories is increasing exponentially. An information system is a combination of interactions between the people, processes and operations, information technology and the data. A management system ensures that an organization can reach its objectives with an efficient data management and adequate processes and procedures. [25, p. 187–188]

An information need can begin within the organization when daily batch-reports are required to manage the daily tasks. While the daily reports represent what has happened, the demand on the reason that why something has happened and the prediction on what will be happening, are indisputably the priority for the organizations. A real-time information and analysis will fulfill aforedescribed needs. Moreover, predetermined events can be required to trigger based on the real-time information. Manufacturing operations management system (MOMS) provides the framework and the platform to integrate the required events. A comprehensive solution to enhance the manufacturing operations performance is therefore obtained by carefully determining the events with the MOMS platform [12].

A definition for a platform describes it being a structure where multiple software products can be built within the same technical framework, and be connected and integrated together [5]. Thus a software is a block to be added to the technical framework of the platform to enable the functionality.

The hierarchical organization layers are standardized with an international standard ISA-95 by International Society of Automation for standardizing the development of an automated interface for data and information exchange between enterprise and control systems. Figure 1.1 represents the ISA-95 standard for the automation layers.

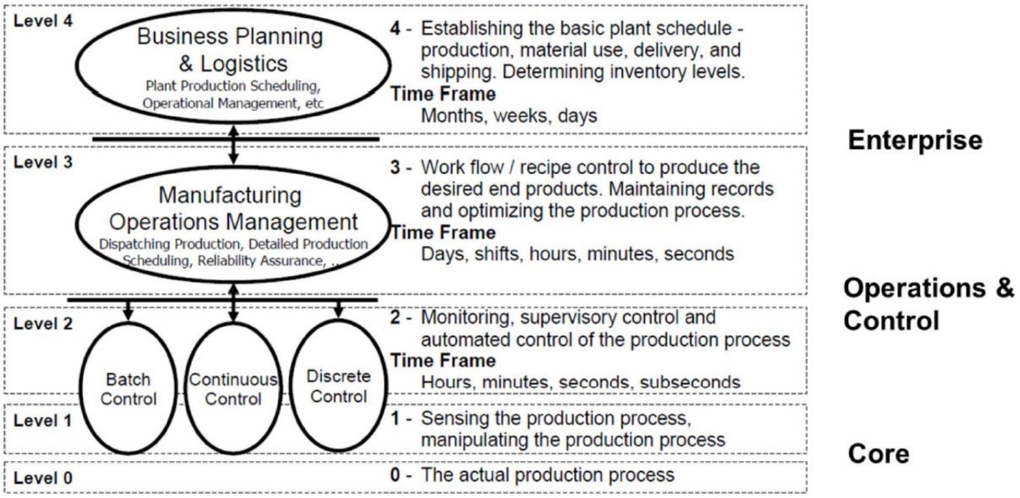


Figure 1.1 *The ISA-95 represents the standard hierarchical automation layers [20].*

A laboratory information management system (LIMS) is an information system software that manages and transfers data in a laboratory to enhance the laboratory productivity by increasing automation. The data the LIMS manages and transfers is mostly data from samples taken from e.g. an industry process, including sample types, chemical elements that the sample type consists of, and test results for each element that are done in a laboratory. Referring the Figure 1.1, the LIMS integrated into the MOMS forms an information management system in the ISA-95 level 3 to optimize the communication between the laboratory automation devices in the level 2 and the organization administration and business planning system in the level 4, including e.g. enterprise resource planning (ERP). The intelligence and a collaborative communications between these systems at hierarchical organization layers enhance a comprehensive information exchange. [13, p. 842]

LIMS in collaboration with the automation produces strategic assets to an organization since the efficiency increases along with the increasing management control resulted by the laboratory automation [25, p. 190]. LIMS solutions are designed to fulfill the demand on faster and more transparent research and more efficient data transfer and management. The data will be more accessible and more adequate as well, i.e. the data errors are significantly decreased and consequently the data quality is increased [34, p. 19], [28, p. 2022]. A flexible communication protocol design for the system integration is a requirement to ensure a bidirectional and synchronizable communication. The protocols have to be designed for scanning process results, importing and transferring the data, and finally a real-time visualization of the data [13, p. 842].

The most of the LIMS solutions has a basic set of features of workflow scheduling, data and result management and visualization, and controlling the data accessability. The LIMS will be evolved as the laboratory requirements change and technological progress keeps increasing. Thus new features and characteristics are introduced regularly. The most of the LIMS solutions also have a few unique functionalities for solving and fulfilling specific problems that do not frequently arise. Researching and developing the right set of features and requirements according to the specified process needs will strengthen the LIMS to match the market demands. The right process specific set of features enhances the productivity of the process as well.

The background of this Master's Thesis is that the system supplier the Thesis is made for has several individual MOMS software products at several customer sites in the field of mining and metal industry. The MOMS platforms have a few LIMS functionalities, but they are not uniform. Spreadsheets are in demand to be replaced with an automated laboratory management system. The LIMS module will cover the basic features of the LIMS, like scheduling, management of the results and an access to the data for the operators and administrators. The problem is that the right set of LIMS features are uncertain. The LIMS module is required to be generic in a way that it can be used at different sites regardless of the MOMS platform in use, or to be a standalone solution if needed. The question to be researched is that how the most efficient set of features and requirements can be determined and realized for a platform independent module to transfer data between MOMS and LIMS? The set of features and requirements are determined by studying existing LIMS systems and open-source LIMS implementations, and validated by user interviews. The designed solution enabling independent deployment and seamless integration of new services or other applications to the module without a need to modify the running service is discussed. The designed software and database architecture is discussed in addition.

The method used in this Master's Thesis to solve the research question is the Systems Engineering with the Unified Process. Following the systems development life cycle (SDLC) the steps and activities required to complete the system development project are covered. The Unified Process SDLC is an iterative, incremental and adaptive approach. Unified Process (UP) life cycle consists of four iterative phases; inception, where the system and the business model vision is developed and refined, elaboration, where the requirements and the core architecture are defined, construction, where the design and implementation continues, and finally transition, where the system is executed into an operational mode. Each of the iterations consists of a full cycle of requirement, design, implementation and testing disciplines [29, p. 45–47].

The UP methodology consists of the aforedescribed phases. The tools to create models and components are software like for example integrated development environments (IDE), code generators and computer-aided system engineering (CASE). Techniques used in this Master's Thesis are use case modeling, system interface designing, database designing and user-interviewing technique. The techniques are guidelines to model the activities of the disciplines [29, p. 47–50].

The interview is done as a scientific theme-interview, which is based on a theory with a framework including the essential concepts and characteristics of the system. Therefore the analysis of the interview-material is more straightforward. The topic of the interview is practically interesting for the users and the purpose of the interview is to study and enhance the user experience. Therefore interviewees are the key users and stakeholders of the laboratory management systems. [36]

Several scientific publications about the LIMS approaches have been done and can be found within the literature, e.g. Göde et. al. represents in a scientific article a laboratory information management system approach as an integration platform for application in life sciences [13]. Another scientific publication is a development of a laboratory information system for cancer collaboration projects by Quo, Wu and Wang [26]. The systems engineering method used in this Thesis is strongly based on a book by Satzinger, Jackson and Burd [29].

This Master's Thesis represents the theory, case studies and the description of the design and the implemention of the LIMS system. Chapter 2 describes the method that is used to resolve the research question. Chapter 3 is reserved for the background and the theory behind the work about LIMS module. The generic features of the LIMS solutions are presented. Chapter 4 describes designing the functionalities, the interfaces, the architecture and the use-cases using the requirement specification, and how they are supposed to be embedded together to have as generic and comprehensive solution possible to be used at every site needed. Chapter 5 presents the results of the Thesis; a user journey map of the LIMS module, the results of the theme-interview analysis, the final feature requirement specification, and realizing the required features in the LIMS module. Finally, chapter 6 gathers up the conclusions of the Thesis and the results.

2. INFORMATION SYSTEM DEVELOPMENT ENGINEERING

Information systems enable the success of modern business organizations since the information technology can accomplish a significant impact on productivity and profits with an increasing need of information [29, p. 3]. An information need can begin within the organization when daily batch-reports are required to manage the daily tasks. In addition to the daily reports representing what happened, ad hoc reporting to represent the reason that why it happened and a prediction-analysis to determine the future behaviour of the processes are also frequent demands among the companies. A real-time information and analysis will fulfill aforedescribed needs. Moreover, predetermined events can be required to trigger based on the real-time information. These information needs are the basis and the data management processes are the methods for a laboratory automation strategy to adapt to conditions on the market and to make the decisions that enhance the profitability. [12]

The laboratory automation strategy is frequently based on the performance abilities of an information management system including a laboratory information management system (LIMS) together with the databases [33, p. 164]. LIMS is a software based laboratory and information management system to implement features to support operations of a modern laboratory [18]. An adaptive LIMS with an adequate integration enhances functionality and flexibility of a high throughput laboratory automation. Therefore, a reliable and bi-directional comprehensive communication between the LIMS and the laboratory automation is the key to manage the laboratory data in a secure and efficient way. [13, p. 841]

Information systems are constantly under development to enhance the competitiveness of the businesses. The key to a successful system development engineering is a comprehensive analysis and design of the system to identify the business requirements that the information system needs to accomplish [29, p. 4]. This chapter covers the analysis and design methods to solve the problems for matching the requirements.

2.1 Systems development life cycle

The system development engineering project is an organized, a goal-oriented and a properly planned process that produces a new information system. For the information system to be robust, efficient and reliable, the system development project has to include a methodical sequence of steps and activities. The systems development life cycle (SDLC) is a fundamental concept to describe the methodical sequence of the information system engineering project. The system life cycle starts with designing based on a need and an idea, followed by the implementation and deployment. Finally, the information system is published for the production. However, the system engineering is a dynamic process that is updated and modified during each phase, later going back to an earlier phases for update as well. [29, p. 38]

Various different approaches of SDLCs are still currently in use. A single comprehensive classification might be irrelevant and nearly impossible to nominate for the SDLC approaches. Nevertheless, one applicable technique is to classify the approaches for more predictive and more adaptive ones. Practically almost all the projects have both predictive and adaptive characteristics. [29, p. 39]

2.1.1 The predictive systems development life cycle

The traditional, predictive SDLC approaches are based on an assumption of well understood requirements and definition of the development project, which is also entirely planned and organized in advance. Further, there is a low technical risk on the predictive SDLCs. The predictive SDLCs are advantageous e.g. for a case, where a company wants to update an old mainframe system to a new networked client/server system. The project can be designed and organized thoroughly and implemented according to the specifications since the system key users already understands the process and its requirements. [29, p. 39]

The first step according to the predictive SDLC is to plan the resources and the budget, and to develop a project schedule. In order to understand and solve the business problem and the requirements, a careful analysis must be done. Defining the right activities to cover the market demands is essential as well. The architecture and algorithms of the system are needed to design according to the system requirement analysis. Finally the solution system according to the system design have to be implemented. The implementation has to include building, testing and installing the system. Further, a support phase follows the system deployment to keep the system productive. [29, p. 40]

The four steps, project planning, analysis, design and implementation forms the predictive SDLC. The steps are completed sequentially and can be modelled with a traditional waterfall approach, as represented in Figure 2.1 [29, p. 40].

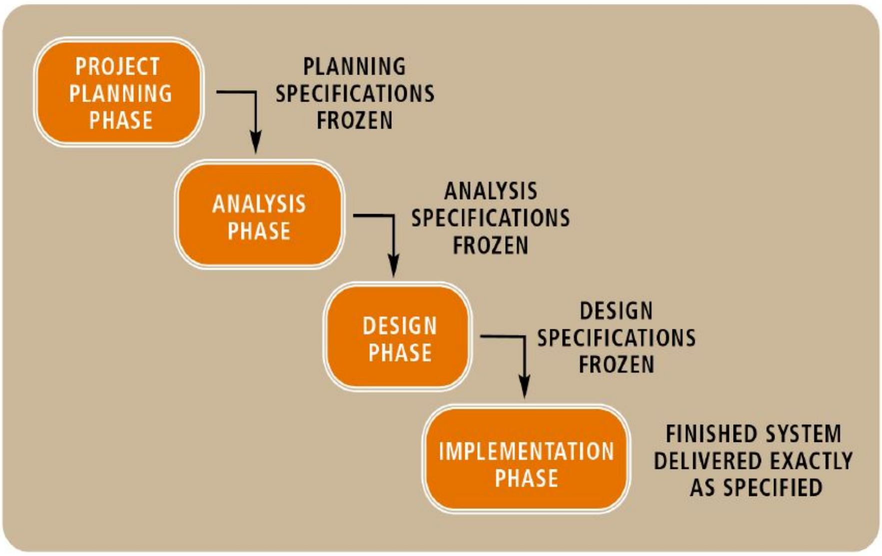


Figure 2.1 *The traditional waterfall model for the predictive SDLC [29, p. 41].*

The predictive SDLC provides a fundamental basis for the systems development engineering. Every step according to the predictive SDLC needs to be included in every systems development. However, the sequence of the predictive SDLC is rigid, i.e. once one phase is completed there is no coming back, as can be seen from Figure 2.1. After every phase the completed phase is frozen. Therefore the predictive SDLC leaves rooms for humane mistakes and important features might be left out. Adding features later is quite time and money consuming. Although the predictive SDLC presents the basis of the systems engineering, the steps must depend on each other and have an adaptive and flexible character. However, the sequence needs to remain, since some project steps depend naturally on each other. The steps need to be in a defined order. [29, p. 41–42]

2.1.2 The adaptive systems development life cycle

Characteristic for the adaptive SDLC approach is that the system can not be throughly planned in advance, due to lack of a comprehensive understanding of the system. The reason might be that some requirements are yet to be defined, and therefore the project must be modified with the progress. [29, p. 39]

The most fundamental description of the adaptive SDLC can be represented with an iterative spiral model, as in Figure 2.2. The starting point, the initial planning phase, is in the center of the spiral life cycle model. The initial planning phase consists of a feasibility study, a comprehensive user requirement survey, and a choice of the implementation strategy and the overall design. The requirement of the initial planning phase is to generate an adequate process knowledge basis to begin the development of the initial prototype. Around the spiral center, the initial planning phase, are cycles that each represents a prototype of the information system. Each prototype represents a preliminary working model, a smaller, more easily manageable subsystem of the final system. Each prototype cycle begins with an analysis and design of the prototype. Followed by sequential steps of testing and integration to the previous prototype, and planning for the next prototype, iteratively moving on to the next cycle until the system is complete, as represented in Figure 2.2. The adaptive spiral cycle model can have any number of prototypes, i.e. more than four, although the Figure 2.2 model includes only four cycles. [29, p. 42–43]

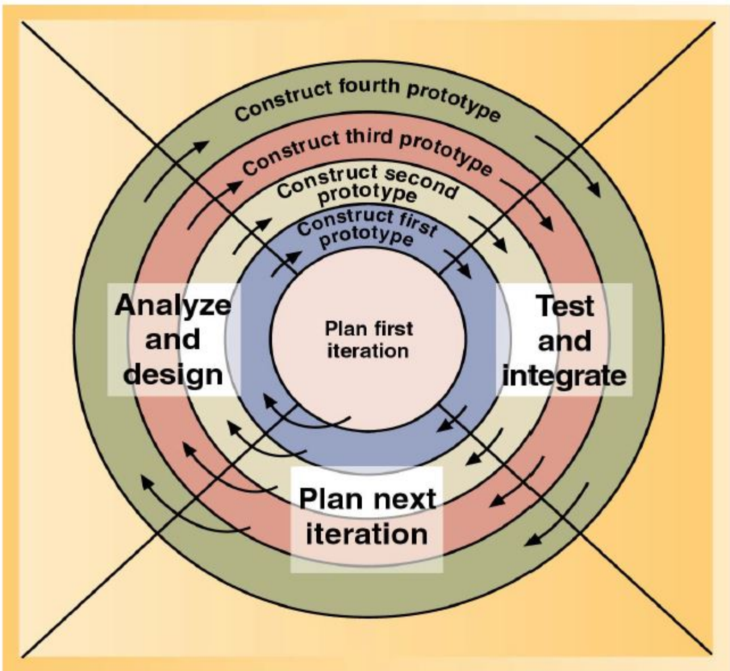


Figure 2.2 *The iterative spiral life cycle model for the adaptive SDLC [29, p. 43].*

A risk management is a key concept in the spiral life cycle model considering the priority of the definitions of the functionalities per each iteration. The risk factors have to be identified and the most relevant risk factors have to be taken under deeper analysis as part of the system design. The most adaptive approaches suggest to consider the most difficult problems with the highest risk in the first iteration. There are cases where the highest risk might be the feasibility of the whole new technology instead of one subsystem, prototype or set of functionalities. If the feasibility is the highest risk, the first prototype should be built to prove that the technology works as designed. The highest risk might be the user acceptance of the new features as well. In that case the first prototype has to prove the users how they benefit of the new system and that their job will be more reasonable. The fourth case for the first prototype is the most unusual - that there would be no significant risks. In that case the first prototype should cover the most important functionalities of the system, and continuing the iterations including lower priority features until the last prototype includes the "would-be-nice-to-have"features. [29, p. 43–44]

2.1.3 The Unified Process life cycle

A continuous improvement is an acknowledged requirement for the successful businesses and therefore new system development life cycles are also developed to model the system development more easier and efficient way. The Unified Process (UP) life cycle contains the equal main principle activities as the predictive SDLC - requirements, analysis, design and implementation, which are all included in every adaptive and iterative phase. Each phase can have one or more iterations of the main principle activities, and each iteration is completed with a required feature or model for testing. The phases of the UP SDLC are inception, elaboration, construction and transition, as is represented in Figure 2.3. In addition supplementary requirements might be requested to be analyzed during each iteration. The supplementary requirements being observed are metrics for performance, reliability, usability and maintainability. [29, p. 45], [1, p. 29]

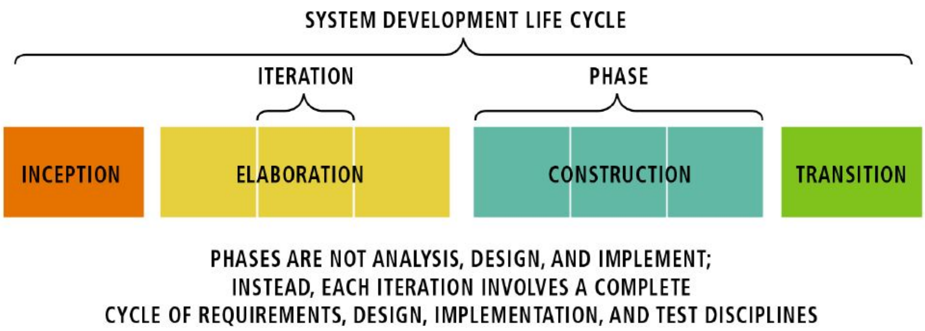


***Figure* 2.3** *The Unified Process life cycle model [29, p. 45].*

The inception phase is for developing an approximate vision of the information system by the project manager. The vision describes how the new system will enhance the operations efficiency and profitability of the organization. The business case have to be built including a cost and benefit analysis to define how the benefits of the new system will outweigh the development costs. The scope of the system must be defined unambiquously so that it is clear what features are going to be included and which key requirements the information system will accomplish. [29, p. 46] The risk management in UP life cycle is defined in the inception phase as well, using the same principles as in the adaptive SDLC. The risk factors have to be identified along with the highest risk. The highest risk is recommended to be taken into account first in the design. The rate of effort required to finish the project is defined not until the highest risk is eliminated. [29, p. 46]

In the elaboration phase typically several iterations of analysis, design and implementation are gone through. Definitions the system requirements are usually finalized as the result of the early elaboration iterations. However, since the UP life cycle is an adaptive development approach, the system requirements can be expected to be altered and grown on demand. The core architecture of the system is one relevant result from the elaboration phase iterations. The elaboration phase should culminate on having more realistic estimation of the costs and schedule of the system development project. Having the realistic estimations enables the business case of the project to be confirmed. [29, p. 46], [8, p. 135]

The construction phase involves typically several iterations of the main principle activities as well. During the construction phase the details of the information system are implemented and finalized. The data have to be validated and a fine-tuning of the system interface have to be done according to a preferred feedback by the system users. The user preference functions have to be finalized as well. An initial plan for the system deployment to the production environment has to be done as well. [29, p. 46–47]

Finally in the transition phase the system development project culminates on the user acceptance test, followed by the final fine-tunings and fixes to prepare the information system ready for the production. The customer will likely appreciate product-support once the commissioning of the system is completed. The support includes possibly required maintaining operations to keep the information system productive, as well as possible updates on demand through the lifetime of the system. [29, p. 47]

The UP life cycle is a roadmap for a comprehensive system development methodology, the Unified Process, which is described extensively for the LIMS in this chapter.

2.2 The inception phase

According to the UP life cycle, the inception phase begins the information system development project with a goal of defining an initial business vision according to the business needs. The scope of the information system to be integrated to laboratory automation has to be defined. An initial schedule has to be planned and finally a business case has to be build for the project. The business case consists of a risk management assessment and a feasibility study. The feasibility study includes organizational and cultural aspects, as well as technological, schedule- and resource-based feasibility [29, p. 86].

2.2.1 Business modeling

The business modeling has to be done to ensure that the business needs, environment, processes and strategy are thoroughly understood. Further, during the business modeling the methods to fulfill the business needs are defined and the expectations of the information system users as well as additional interested stakeholders are taken under consideration. [29, p. 86]

The purpose of the information system development project is to supply a solution for a business problem. If that is not clear from the beginning, the new provided solution might be wrong and irrelevant. Therefore defining the business problem and documenting it is the first step in business modeling. [29, p. 87] Manufacturing organization need to adapt to conditions on the market and make the decisions that enhance the profitability and matches the market demand [25, p. 187].

The next step in understanding the business environment is a study of interoperability. The need of interfaces to existing systems and determining the software architecture has to be initially examined. The developed information system will be part of an integrated processing environment in most development projects. A bidirectional data exchange with the existing system is a frequent requirement and have to be taken under consideration. The new system have to fit into the existing strategic architectural plan as well; the existing laboratory system may consist of systems by various vendors. The interface and architecture examinations has to be documented as well. [29, p. 87]

A major analysis in the business modeling is done by considering the stakeholders. The stakeholders are a group of the system users, the sponsors or the investors and the technical support staff. That is to say, a group of people who have some interest in the new system development. The ownership of the information system and information systems in general, should be on the production side. The stakeholder analysis consist of identifying all the stakeholders, determining the system access levels for all the system users, determining the information the support staff needs and what information is needed from the support staff, and finally considering the desires and need from the sponsors. [29, p. 88–89]

2.2.2 Information system vision

The information system vision is determined to ensure a clear definition and understanding of how the new system will solve the business need and how the system contributes to the organization's strategy. [29, p. 91]

The information system benefits have to be analyzed and defined according to the business point of view. One important metric for the business benefit would be the return on investment (ROI), that indicates how the information system pays back the investment and further improves its value. The business benefits should be identified by the project investors and sponsors, i.e. the responsibles of the investment decision. However, the project development engineers have to determine the system capabilities to achieve the objectives based on the business benefit analysis. The system capabilities are the tools to produce the business benefits. The system capabilities will also establish the information system project scope. [29, p. 91–92]

2.2.3 Project and system scope

The objective of the finalizing the scope is to ensure that the project and system scope are well defined. The system scope defines the system capabilities and features, i.e. everything the project includes, and the project scope defines the methods used to accomplish the information system successfully. [29, p. 96]

The scope defines the system features considering the budget and schedule constraints. A prioritizing can be also determined to maximize the business benefits inside the scope. A frequent problem with the information system development projects is a scope creep. A scope creep refers to changes to the project that increases the workload with additional tasks, leading to a continuous or uncontrollable growth in the scope. [17, p. 40] Especially the adaptive character of the UP life cycle is vulnerable for the scope creep since the scope is more flexible.

An initial use case model can be done as well to finalize the scope. A use case describes a functionality of the information system to complete a business event [29, p. 97].

A comprehensive schedule should be done as well according to an effective project management practises, but this Master's Thesis does not emphasis on the technique on the project management scheduling, so it is kept short in this case. A project schedule defines a sequential list of every task the project has in the correct order. In addition for every task has to be estimated the effort it takes as timely manner. [29, p. 100]

2.2.4 Risk management

Analyzing a project feasibility determines whether the information system has a reasonable chance for successful implementation. An essential part of the feasibility analysis is identifying all the risks considering the project. [29, p. 106]

Risk management is a process for a systematic risk identification and determination, including an analysis and reacting preventively to the risk as well. The purpose of the risk management is to maximize the probability of the desired events and to minimize the probability of adverse events. [17, p. 21–22] The information system experience is frequently the best asset in determining the risks.

The potential risk can be considering the organizational culture. Since every organization has its own culture, a new information system might not be effortlessly adapted, especially if the new system differs significantly from the current company policies. The operators and other employees might not be overly familiar with the computer systems or there might be a fear of change considering job responsibilities or employment due to greater ratio of automation. [29, p. 107–108]

The technology might not be feasible, e.g. the integration to older existing systems may cause a risk, or the expertise within the company might not be enough even for the existing systems. Further, the requirements of the complexity and the level of technology by the customer can be too high for a successful implementation. In the end, the number one priority should be to deliver a working, reliable information system. [29, p. 108]

The project schedule can be a risk for successful information system implementation as well. Since various approximations and assumptions has to be made when determining the schedule, the project schedule might be too tight for implementing all the features. The pressure for schedule can originate from the customer management as well. The scope creep can be a risk for the schedule in addition, since the UP life cycle has an adaptive characteristic, new additions might occur during the way of development and might cause a risk of slipping of the schedule. [29, p. 108–109]

2.3 Determination of the requirements

The information system requirements and the system architecture are the areas that are the most comprehensively concentrated in the elaboration phase according to the UP life cycle. The key for determining the information system requirements effectively is to first understand the business processes and daily operations. The information system must meet the business needs. [29, p. 126], [8, p. 134] For example the requirements for the laboratory data management systems vary significantly from project to project. Many laboratories manage their data using only spreadsheets [28, p. 2022].

2.3.1 Functional and nonfunctional requirements

Systems requirements are the features and functionalities that the new system needs to include and accomplish. The system requirements can be categorized into two parts: functional requirements and nonfunctional requirements. [8, p. 136]

The functional requirements consists of the activities that the system is required to execute to fulfill the business needs and to run the business. The functional requirement activities are related directly to the use cases. However, modeling and describing all the functional use cases can be very challenging since the functionality and the relationships between the use cases can be very complex. [8, p. 136]

The nonfunctional requirements describes the systems required characteristics. Technical requirements determines the software, the hardware and the architectural requisites. Performance requirements specifies further operational characteristics considering the capacity, such as throughput and response time. Finally usability requirements perfects the operational characteristics with requirements of the user interface, user related workflows, online support and documentation. [29, p. 130], [8, p. 136] Furthermore, reliability characteristics are frequently determined with the requirements of dependability, such as detecting the problems and recovering from them. Lastly, security requirements has to be determined considering the system access security for every user group, e.g. to exclude certain system users from certain system output analytics. In addition the security requirements for a system communications network and data storage can be applied. [29, p. 130], [8, p. 136] Defining the new information system completely requires the both functional and nonfunctional requirements.

2.3.2 Techniques for gathering requirements

Figure 2.4 represents the best information gathering techniques to understand the business processes for determining the requirements.

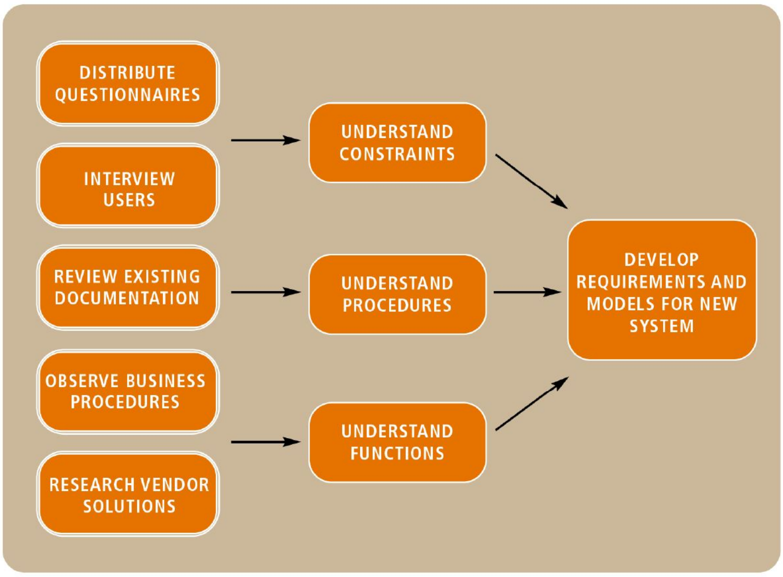


Figure *2.4 A map from understanding the business process to determining the requirements [29, p. 137].*

A user interview is one of the keys to understand the business operations of the process environment effectively. The requirements covers the methods for achieving the business objectives and the needs for the information system architecture to accomplish the objectives and the user interview validates and completes the system requirements [29, p. 126].

In case there is an existing information system that is going to be updated or replaced, the existing documentation have to be studied carefully. [29, p. 126], [8, p. 137] In addition, interviewing the users who will be using the system is an essential part of requirement definition as well. As described in Figure 2.4 the supplementary system information consists of researching the solution on market and observing the business workflows. The business process understanding consists of understanding the business and technical constraints and functionalities as well.

When the old information system is replaced with a new system with a similar functionality, the users tend to be sure about the needed features and a desirable interface. However, the new automated functionalities that replace previously manually performed tasks cause frequently uncertainty among the users. The users do not always really know that which features are desired and which are achievable.

An interface is a simple and reliable model to illustrate the functionality and the appearance of the system, instead of the use cases and activity diagrams that might be abstruse in the user point of view. The interface represents the information system for the most of the users. Therefore, delivering a prototype of the interface to the users before the implementation would be a good practice. [29, p. 128–129], [8, p. 137–138]

2.3.3 Modeling requirements and types of models

Various different types of models are used to simplify describing the requirements and the features of the information system. Models are built from different perspectives and describe aspects from different levels of detail and abstraction including high-level overviews as well as detailed descriptions of only a few certain aspects of the information system [29, p. 131], [8, p. 137–138].

The models in the information system development can be classified into three types: mathematical models, descriptive models and graphical models. Mathematical models describes the system functionalities with mathematical formulas. The most detailed aspects of the information system can be represented with formulas and mathematical notations. Mathematical models are most appropriate for modeling technical requirements, such as equation for expressing response time for database query or representing requirements for network throughput [29, p. 133], [8, p. 137–138].

Descriptive models are notes, lists, reports, narrative memos or any textual description of an aspect of the system. Descriptive models can be rather informal, as long as the information is clearly and descriptively presented. Recording information in written form is one of the most fundamental types of descriptive models. Users can describe their role and job with a narrative descriptive model, e.g. "At first I select these samples from the list and then I mark them as collected and ready to be sent forward". Notes of an interview or a report of a meeting can be transcribed to form narrative models. Descriptive model can also be list of features, users, inputs and outputs. Lists are self-descriptive and simple, but also specific and effective description of aspects of the system. Descriptive models can be converted to graphical models when compiling the information of the system requirement and feature research [29, p. 133], [8, p. 137–138].

Finally probably the most effective and useful models are graphical models, that are modelled by diagrams, schemas, charts etc. visualized representations of the information system. Graphical models simplifies complex relationships and makes complex verbal descriptions easier to follow and understand. Graphical models visualize causations that are much easier to understand than mathematical, textual or verbal descriptions. Graphical models can utilize also symbols to represent abstract aspects of the system, such as data structures, data objects, connections and the relationships between the functions of the program. Graphical models can also be used for e.g. user interface layouts, a plant floor layout and other very visual models. Moreover, graphical models can be most effectively utilized for modeling the information system from multiple points of view and from different levels of details and abstraction [29, p. 134], [8, p. 139].

2.4 Interview technique

A user-centric approach is one of the keys in determining the requirement information. The key users have the experience of how the business operations are most efficiently managed and possibly a vision of how to improve the data management. Thus interviewing the users who will be using the system is an essential part of requirement definition. However, the requests from the users have to be prioritized to exclude the non-essential features from the fundamental functionality. The absolutely needed features has to be known to avoid the scope creep, the uncontrollable growth of the requirements. [29, p. 127–128]

The interview is a broadly utilized technique for information gathering. Both the gathering technique and analysis have to be systematic to be part of the requirement research. The interview can be a structured interview with explicit questions or a theme interview with a frame and guideline of questions. The theme interview is more appropriate for the information system requirement research since the amount of the interviewees doesn't have to be large, but the information is valuable and deep, i.e. there will be more efficient results. [36, p. 2]

The framework and concept of the interview should be taken from the theory and business processes instead of intuitive baseline. Therefore the familiarization to the process is guaranteed and the uninteresting points can be excluded from the interesting ones. In addition the interview should describe the requirements from different points of view and multiple ways of working to be comprehensive. [36, p. 4–5]

The most important step prior to the interview is to establish the objective what is wanted to accomplish with the interview. A research problem has to be set to help accomplishing the objective. Another essential step is to determine the interviewees. The information system users should be definitely involved and in addition system stakeholders if needed. These two steps leads everything else considering the interview. [29, p. 140]

The scientific norms determine that the interview has to be the universal, the interview has to able to be public within the organization, it has to be impartial and finally the essential part of the interview is a systematic criticality. [36, p. 5] During the interview, it would be important to critically question the workflows, requirements and approaches. Opportunities to ask "what if"should be benefited, since the exceptions fulfill the requirement definition and makes it comprehensive. Especially if the users or other stakeholders does not have a answer for some exception case, the results should be taken seriously under consideration. [29, p. 141]

Figure 2.5 determines the fundamental themes and questions that builds a firm base in information system requirement definition. Understanding what are the business operations and processes leads to a question that what the users exactly do. The description of the information system functionality by the system users is founded typically on the current system and what the users do with the current system. [29, p. 136]

THEME	QUESTIONS TO USERS
What are the business operations and processes?	What do you do?
How should those operations be performed?	How do you do it? What steps do you follow?
What information is needed to perform those operations?	What information do you use? What forms or reports do you use?

Figure 2.5 *A fundamental themes and questions in defining the information system requirements [29, p. 137].*

The methods how the information system functionality should implement the business operations moves the discussion from the current system to the new system. The subject has to be on how the new information system should support the daily operations and the job of the users instead of what kind of functionality is supporting the tasks with the current system. [29, p. 137]

The requirements of the information needs to be provided finalizes the business requirement fundamentals. The information the users use to do their jobs is one essential key and another one is that with which analyzing and reporting tools the users benefit from the information. However, all the required parts of the information is typically difficult to identify. [29, p. 137–138]

2.5 Construction and transition phases

The construction phase carries on the implementation by programming the information system features and requirements designed at the elaboration phase. The core architecture and the basic features along with taking account the highest information system risks are finished at the end of the elaboration phase. The focus on continuing and deepening the design and realization of the information system features is the purpose of the construction phase. Multiple iterations of design and implementation are typical during the construction phase. The iterations include validating the functions by testing and validating the data that the information system delivers. The system and user interfaces are fine tuned and the fundamental, routine data processing and maintenance functions are completed. The helping and the "nice-to-have"functions are designed, implemented and finished as well. Finally the design of the deployment and the transition of the information system has to be started at the construction phase [29, p. 46–47].

Constructing and implementing an information system software is a complex process of programming, assembling and integrating software components. The resources have to be used efficiently regardless that every information system construction process is different due to unique characteristics of each target project it is developed to. A software component is defined to be a section or a client of software that is completely assembled, prepared ready for using and its interfaces are defined properly to ensure an adequate integration to other clients or assemblies of the software [29, p. 532]. The implementation starts with choosing a component interaction standard to ensure the functionality of the different software components. The component is constructed of classes that actually execute each feature. The component can be either acquired from an existing system with modifying the component to fit and interact the information system under development, or the component can be built from scratch or by unifying ready functions and classes. Finally the components including the classes are constructed into executable units according to the determined standard. The executable units are installed seamlessly according to the designed hardware and software architecture [29, p. 532–533].

Testing the information system is a process of exercising the operations of the software components to validate the functionality of the system or to catch and identify possible defects in the system. To ensure locating the source of the defect, the interaction standards between the software components have to be well defined. In case of the exercising the system or the results indicates an error, iterations of re-design, building, exercise and testing have to be implemented until the error or errors does not occur anymore [29, p. 533]. The testing of each component begins with a definition of a test case. The test case consist of an initial state, the events that the information system have to respond to and the result state with an expected response. A test data and the result data with expected deviations compared to the test data represents the input and the output of the test case, respectively. A defect in functioning can be investigated if the result data is something else than expected or if the software itself throws an error, when a catching error functions have built in the program. The defect can result e.g. from a wrong data type passed to another component, a unexpected variable value or a run-time exception resulted e.g. from that a file is already in use or running out of memory. The test case can be either a unit test that monitors a performance of an individual function or unit isolated from the other units or components, an integration test that evaluates the performance of the unit when operating with other components, a usability test that determines how the performance of the component, a subsystem or the system meets the user requirements or finally the user acceptance test determines the information system performance against the system requirements, ideally in the real production environment [29, p. 534–539].

The final phase, the transition phase, focuses on finalizing the information system and deploying the ready system for production operations. The hardware and the system software have to be prepared and acquired, e.g. the database has be created and transferred to the production environment. The information system executable have to installed as well. A user training can be kept when deploying a complex information system [29, p. 547–552]. The transition phase consists of at least one final iteration, involving a user acceptance and a pilot test in co-operation with the customer before ensuring the information system is ready for the production use. A technical support and maintenance have to be offered as well for the customer to co-operate keeping the information system functional and updated [29, p. 47].

3. LABORATORY INFORMATION MANAGEMENT SYSTEM OVERVIEW

Laboratories keep growing with the market demand, and therefore the increasing amount of data is continuously more difficult to process accurately and yet in minimum time. To face the market demand and to keep ahead of the competitors requires enhancing the laboratory throughput and increasing the profitability by taking advantage of the state-of-the-art systems. [22, p. 38] Two essential techniques for gathering information to understand the business processes are observing other similar business processes and researching the publications and manuals. Researching general features of LIMS solutions ensures that information is gathered to understand the processes and to determine the right set of features for the information system under the development. Information systems development life cycle has to be followed to assure an inclusive combination of business vision, requirements analysis, information system design and implementation. LIMS is the information system whom the Unified Process is applied in this Master's Thesis. Choosing the LIMS obliges decisions based on the performance requirements, such as selecting an adequate database engine and LIMS software and utilization of the existing LIMS solutions for the customization purposes. [22, p. 38] This chapter describes the general features of LIMS solutions by researching publications and open-source LIMS specifications to construct an overview from LIMS systems.

3.1 Business vision and value of laboratory information management system

LIMS integrated to the MOMS forms an information management system to provide a communication bus between the laboratory automation devices and the organization administration system, ERP. The intelligence and a collaborative communications between these systems at hierarchical organization layers enhances a comprehensive information exchange. [13, p. 842] LIMS enables the administrators to register work requests, print and monitor the laboratory sample order data, and finally communicate and schedule the workflows [25, p. 189]. To justify the LIMS investment, a comprehensive study of the business processes and practises is required both internally and externally of the laboratory [25, p. 190].

The system supplier the Thesis is made for has several individual MOMS software products at several customer sites in the field of mining and metal industry. The MOMS platforms have a few LIMS functionalities, but they are not uniform. Spreadsheets are in demand to be replaced with an automated laboratory management system. The LIMS module will cover the basic features of the LIMS, like scheduling, management of the results and an access to the data for the operators and administrators. The problem is that the right set of LIMS features are uncertain. The LIMS module is required to be generic in a way that it can be used at different sites regardless of the MOMS platform in use, or to be a standalone solution if needed.

LIMS provides functionalities for the laboratory data management. LIMS reduces costs and saves time since the automated data management controls, organizes, transfers, stores and reports the laboratory research data in an efficient way [34, p. 19], [28, p. 2022]. Furthermore, the data quality and accessability within the departments are secured [28, p. 2022]. LIMS in collaboration with the automation produces strategic assets to an organization since the production operating efficiency increases along with the increasing management control resulted by the automation. [25, p. 190] Moreover, there are corporate reasons for a request of available analytic laboratory data to external groups for further data manipulation or to back up decision making. [33, p. 165]

Probably the most frequent reasoning for investing to the LIMS consist of decrease in throughput time, enhance the analyses and reporting and overall information quality improvement. The sample logging is automated and the gathered data is easily available in the database. Therefore the information does not need to be re-entered every time and the backlogging and production reports can be generated easily. [22, p. 41]

A measure of the laboratory performance is the productivity, which is affected by the number of the analyzed data samples and an error-free reports in a given time period [33, p. 164]. LIMS improves accuracy and availability of the sample data inflow. The real value of the LIMS is the result gained from the capability to maximize the data sample throughput and to minimize labor costs. With a good knowledge of the process, the laboratory supervisors and operators are efficiently able to minimize the data entry errors by manual correction. [25, p. 189], [22, p. 42] The data search time can be significantly decreased and data validation, instrument integration and automatic calculations can be applied to greatly increase the productivity, since the operators do not have to manually enter the results from each instrument into the spreadsheets. Therefore the transcription errors are decreased as well. [22, p. 42] The laboratory personnel are in addition able to maximize the sample batch flow as well as minimize the downtimes by the data management scheduling. Security of the laboratory data and traceability of the sample storage can be efficiently controlled by the data management. [25, p. 189]

3.2 Requirements for efficient data management

Laboratories are utilized in information business. Market leaders are those laboratories that are first to deliver the quality and performance information to the clients, before the competitors. Selecting the system that fulfills the laboratory data management needs and requirements is important to be able to enhance the laboratory profitability. Prior to selecting the LIMS it is essential to gain a solid understanding of the laboratory data management requirements, in addition to the expected benefits considering the LIMS and laboratory automation. One part of the research question is to study how the most efficient set of features and requirements can be determined. According to the method for gathering the information, researching the market solutions is one of the key techniques. This section fulfills the research need in the most exact level. Frequently minor modifications are needed to complete the LIMS that fits the laboratory business model. First the process knowledge has to be acquired comprehensively before customizing and finetuning the LIMS adequately. [22, p. 38, 41]

3.2.1 Features for functional requirements

The functional requirements consist of the activities that the system is required to execute to fulfill the business needs and to run the business. The LIMS generally have a basic set of features and functionalities that defines LIMS. Therefore being aware of certain fundamental LIMS features is the key for understanding the business processes. Functionalities of the LIMS includes first a sample tracking. The samples are tracked through different separate laboratory departments. The samples are identified with a unique id. [22, p. 38] The sample registration process may involve producing a barcode attached to the sample container. Audit logging all the changes to LIMS data is the most adequate method for sample tracking in the field of mining and metal industry [18].

The laboratory operators frequently have to be able to register results into LIMS with manual data entries [22, p. 38]. The manual entry is fundamental especially for laboratories receiving analysis data from multiple sources, including non-digital records registered to paper and records by instruments that are not able to be connected to the LIMS. In addition, manual data entries enables correcting data entry errors and adding comments and notes for the individual samples. [18]

The sample scheduling is an essential functionality of the LIMS. The samples are automatically logged in, transferred to the laboratory and assigned to laboratory technicians according to the analysis schedule. The electronic and automated data transfer increases productivity by decreasing the potential of transcription errors. [22, p. 38] Production of the reports can be automated with the scheduling in addition. The scheduling enables the development of the laboratory workflows and events to enhance the productivity. [18]

The workflows can be assigned to run batches as well. Batching is grouping samples that have similarity with respect to sampling procedures, leading to the samples being processed as a unit. The batching enhances flexibility in laboratories that have a high throughput, since the multiple similar samples can be also managed as batches. [18] Alternatively or additionally an internal file and data linking-feature can be implemented to enable linking reports, procedures and results to strengthen the data structure. E.g. a test process can be linked to a particular customer, a report can be linked to a sample batch or lab results can be linked with a correct test method. [18]

The operators have to be able to generate charts and view trends of the laboratory analysis graphs to visually represent the data. The charts can include or correspondingly exclude blank samples, duplicates, peaks, standards etc. [22, p. 38] The modern LIMS solutions can also offer multiple additional options for visualizing the data, including data gradients, vectors, reports and even 3-D structuring for complex cases. [18]

Finally, the database administrator has to have access to the maintenance operations. The database can be managed and the administrator can keep track of client lists, analyses, employees, configurations, permissions, priorities etc. [22, p. 38]

Figure 3.1 represents the sample workflow through a typical laboratory. The workflow begins with the sample request generation and accordingly the collection of the requested sample. The sample data management consists of the workflow scheduling, distributing the samples and managing the sample storage and retrieval. [22, p. 41]

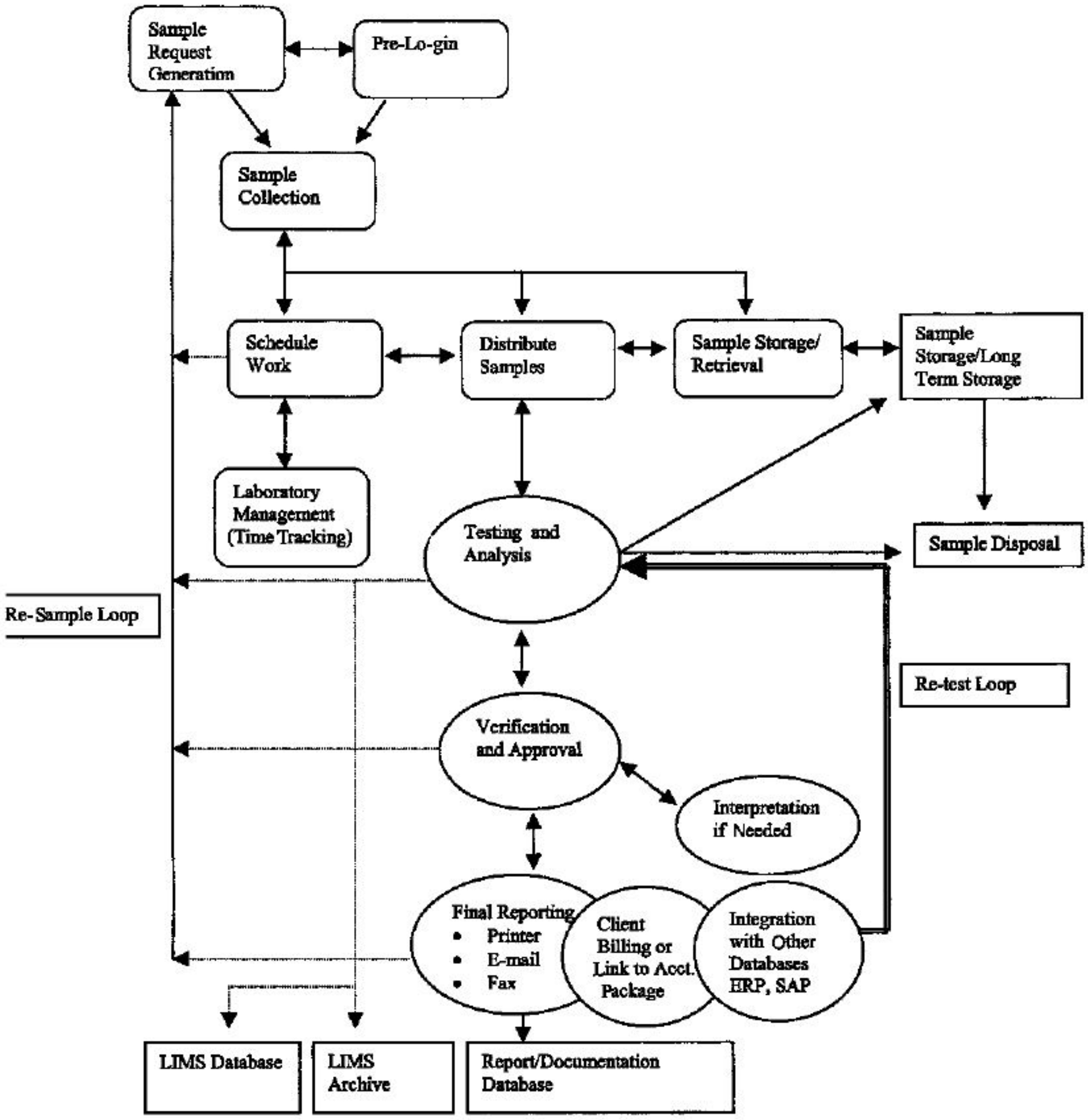


Figure 3.1 *Block diagram representing a typical LIMS workflow [22, p. 41].*

The samples are distributed for testing and analysis, followed by the verification and approval with a possible interpretation. Finally the tests and analyses are registered, reported and archived. Testing and analysis, as well as verification are iterative, i.e. can be implemented with several loops depending on the complexity of the data management. All the phases and steps of the workflow are integrated tightly to the LIMS database as well. [22, p. 41]

In addition to the data storage and registering samples, a modern LIMS solutions offers a feature of sharing the analysis results, reports and equipment data, etc. Sharing the sample data to enhance the efficiency of both internal and external collaboration of the laboratory is made easier by modern LIMS. In addition to test result data, charts, graphs, reports, policies and other documents can be shared, and most importantly only to the users who have been granted access to such documentation. [18]

There are LIMS features that covers the entire LIMS that cannot be categorized under any certain step or phase. One general example of a such feature would be localized multilingual support, that allows the users to interact with the LIMS in several languages. [18]

In addition to the key functions of the LIMS there are supplementary operations that the LIMS can manage. These functionalities include a barcode management, which enables reading and extracting information from the barcode, and also generally processing and converting data into certain formats to manage the data distribution. Important maintenance of the laboratory instruments can be scheduled and recorded and a calibration of the instruments can be implemented. A laboratory inventory of the laboratory equipment and possible vital supplies can be managed. Finally the workflow and analysis processes can be tracked considering the spent time, and based on the data calculations can be done to optimize and predict the laboratory processes. The additional functional features are not limited to aforedescribed ones, i.e. new features can be added on demand. However, the scope for each project is defined to prevent the demands of growing out of hands. [18]

3.2.2 Features for nonfunctional requirements

Laboratory personnel, i.e. users, operators and supervisors own the processes that accomplish the laboratory functionality. The process owners have an interest in verifying that the process quality is as high as possible. Verifying and enhancing the process quality requires measuring and parametrizing the performance of the processes. [33, p. 166]

The best practise for enhancing the process quality is to have an information strategy that is used to design the LIMS architecture. The information strategy is embedded to the business processes to form the business rules to determine the integration of the processes and to deliver the inputs into the outputs. Hence the functionality of the LIMS evolves into a workflow application to be able to flexibly confront the changing business requirements. [33, p. 167]

Further requirements for the functionality of the LIMS can turn up from the end-user after the commissioning while data management requirements increases as well. A modular design, i.e. the ability to be able to respond quickly to required software changes is a feature of successful business practice. A thorough modern database configuration and LIMS system design ensures that changes in requirements can be managed, considering new data structures or reports for the management. However, requirement for e.g. changes in the business rules or processes are much more complex and built deeper to the system, and therefore changes in all cases cannot necessarily be managed. [33, p. 165]

The modern LIMS functionality has spread continuously wider with frequently integrating features like sample data management, data mining and analysis, and electronic laboratory notebooks (ELN) to several LIMS solutions. The purpose of the ELN is to replace handwritten paper laboratory notebooks for registering samples and analyses. ELN enhances the data security, supports the collaboration among the laboratory staff by information sharing and has a natural advantage of being more efficient and easy to search with keywords from. Therefore the reusability of the documentation will become more effortless. [18]

3.2.3 Audit trail

A chain of custody stores information about traceable records providing unbroken chain of documents, raw data and samples with extensive information about initially collecting the sample, ordering and finally disposing the sample. Manual documents including entering the sample locations and timestamps are replaced with automated electronic records created and maintained by LIMS. The technology and security development with establishment of regulatory standards have enabled enhancements required to implement a protected electronic tracking system with a maintainable chain of custody records. [37, p. 1]

The requirements of security and traceability depends essentially on the industry the LIMS is applied to. In the food and drug industries it is vital and legally required to keep records of exact locations and users who have possessed the samples at each phase [37, p. 1]. In mining and metal industry the chain of custody recording do not have to be detailed and secured by the legal demands. Instead of the nonessential chain of custody, an audit trail is recorded to enable manual editing of the samples at required phases of the laboratory process.

The audit trail includes a documentation of the sequential steps of the activities that have been implemented. The included audit trail information consists typically of an operator id, a timestamp, a location and destination, a case number, a transaction type, a quantity and quality information prior the modification and possible additional user comments. [18]

3.3 Information systems strategic planning

Laboratory processes are owned by the laboratory operators and supervisors who interact with the LIMS and have impact on the processes they own, producing analyses and reports as the output of the LIMS. An information strategy assists in determining the software applications required to facilitate the demanded functionality of the laboratory automation [33, p. 166].

The laboratory data management functionalities are enabled by an intelligent laboratory informatics architecture that enhances the management efficiency of the information and data produced by the laboratory. Hence the laboratory performance is being enhanced as well. The management system effectiveness depends essentially on the laboratory architecture. [25, p. 189]

The laboratory informatics produces a specialized information through a platform consisting of software, hardware like instruments and other laboratory devices, and data management tools. The data management tools allow scientific data to be gathered, transferred, processed and interpreted for instant utilization. The data is stored to the database, managed and delivered to information for research, development and testing. Therefore, the laboratory performance is enhanced as a consequence. The laboratory informatics focuses on the technology associated with the information management processes. [18]

Information systems strategic planning strives to define the information system technology and applications to support the strategic plan and business needs of the organization. The information systems strategic planning definition sometimes involves the whole organization and according to Satzinger et.al. [29] is an organization-level concept. However, the partition describes accurately and systematically the features and components needed for a development of a single information system as well. Therefore studying and researching the information strategic planning deepens the understanding of the business processes. Thus determining the most effective set of features and requirements is based on even more comprehensive information principal. Figure 3.2 represents the components of the information systems strategic planning.

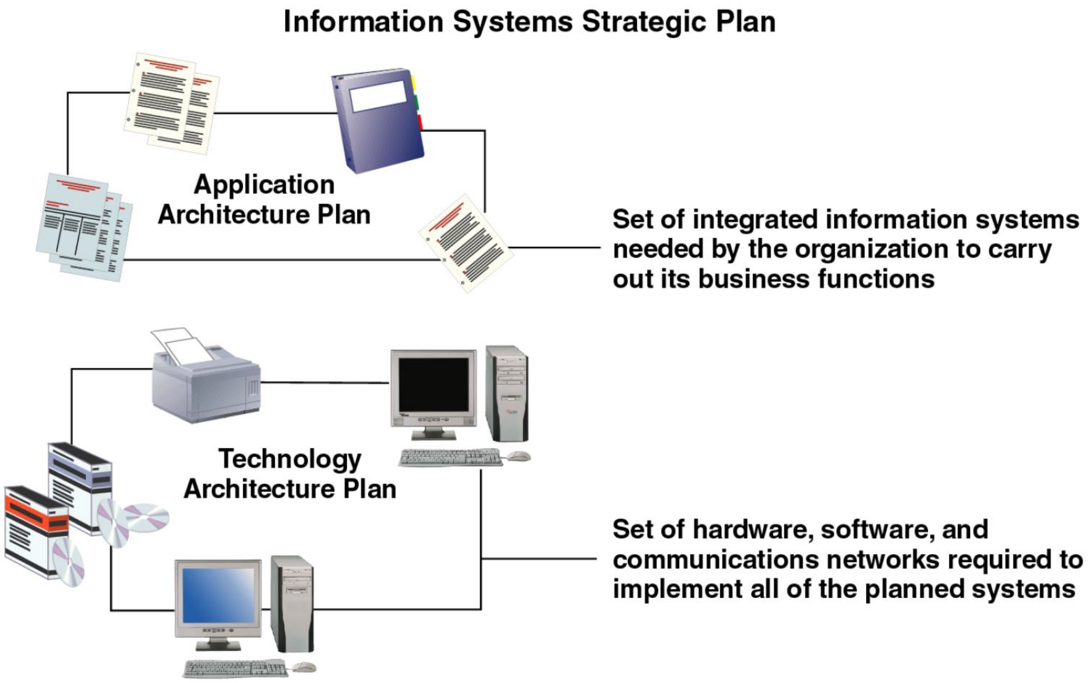


Figure 3.2 *Partition of an information systems strategic plan [29, p. 17].*

As represented in Figure 3.2, the information systems strategic planning can be divided to the application architecture plan and technology architecture plan. Application architecture plan determines the integrated systems needed to fulfill the business needs [29, p. 16]. Accordingly a data-centric LIMS model determines business rules, that defines how the collaborative process linking should be implemented to meet the business requirements [33, p. 167].

Technology architecture plan determines the hardware, software and communications networks required to form the defined LIMS and to implement the defined applications processes [29, p. 17]. Accordingly a process-centric approach identifies the software applications needed to form the automated process to improve process quality [33, p. 166].

3.3.1 Application architecture plan

Plans tend to follow a model to form methodical sequence of modelling actions. An application architecture model determines the integrations of information systems to fulfill business needs with managing the information. An emphasis on analytical data and reports (i.e. information) leads to applying a data-centric LIMS model, where the primary objective is to determine the data management automation to bind the integrated systems tightly together with bidirectional data transfer instead of emphasizing to the data producing automated processes. The data-centric model is used to apply business rules, which are in a relevant role determining how the integrated systems form the LIMS architecture according to the business needs. [33, p. 166–167]

Distinctive for the business rules of the data-centric model is to log the data management status of the laboratory entities, i.e. sample, sample type, order, result. States are descriptions of the data management process, e.g. available, in testing, rejected. The data-centric model can be used to monitor the workflows, that visualizes the management actions. The most value of the model is gained when it is used to determine what will be next happening instead of the current status of the entities. [33, p. 167]

The data-centric model can be applied to determine LIMS business rules, which can be e.g. that an already rejected data sample cannot be approved. However, new, more complex business rules are difficult or impossible to be implemented without customization since the rules are embedded into the LIMS architecture. An example of an aforedescribed rule could be "in certain circumstances a rejected result can be approved". [33, p. 167]

Since the rules will be tightly embedded into the LIMS architecture, applying an application architecture plan as a part of the information strategy on the early stages of LIMS design is essential. The data-centric model of the plan enables defining the business rules to determine the linking of the business processes. By determining the linking of the data management processes as workflows, the LIMS will be designed to be an application that is able to respond also to changing business requirements more flexibly [33, p. 167].

A workflow application characteristics opens possibilities for deeper analyses and understanding of the data produced by the laboratory processes. For example the laboratory activities can have deadlines and time limits for accomplishing each task. The actual spent times can be automatically recorded and afterwards analyzed to be able to finetune the scheduling and therefore to enhance the productivity. Further, availabilities and workflow bottlenecks can be predicted with adequate calculations and analyses. Warnings or alerts could be delivered against critical workflow tasks if there is a problem at the production. [33, p. 167]

The workflow based LIMS will arise an interesting possibility of the system being able to learn. If the LIMS system would enable the laboratory analyst to design and build the workflow for an ad hoc sample on the fly, the best successful approach data could be stored to be learned. The next case processing a similar sample could utilize the best worklow applied before. [33, p. 167]

Researching and applying the application architecture plan will deepen the understanding of the business processes. The application architecture plan not only works as a part of the information gathering for developing a new information system, but actually also can guide on the problem that how the features should be realized to form the information system. Building a well designed linked network of systems according to the information strategy will make the produced information much more powerful to increase the business profitability with the right information [33, p. 168].

3.3.2 Technology architecture plan

The laboratory informatics architecture is based on an information strategy. Identifying the required software applications to achieve the goals of the strategy improves the process quality. The LIMS will become more receptive to changes in business practices as a result of an aforedescribed process-centric approach. [33, p. 166]

The reasoning behind deploying the LIMS in the process technology point of view is to simplify the process while increasing the effectiveness, the efficiency, the transparency and the rate of the data transfer. The most versatile and flexible LIMS will be approached by utilizing the high end systems advantages and the migration from the systems on the market to the customized solution. [25, p. 190]

The efficient data management and an electronic data exchange requires an adequate database engine. Open Database Connectivity (ODBC) is a standard for the database integration that enables integrating the LIMS seamlessly with the other relevant databases in the organization, such as an ERP database. One of the most important aspects in selecting the database engine is to choose a software-independent database engine that is as straightforward as possible to migrate with the evolving technology. There are three basic types of database engines: proprietary databases, that are not software-independent and not ODBC compatible, indexed sequential array method (ISAM) databases, e.g. Microsoft Corp. Access and Microsoft FoxPro, and Paradox in addition, and finally SQL databases, including the Microsoft SQL Server, Oracle and Sybase. [22, p. 38] Most of the laboratories rely on the database market leaders, Microsoft and Oracle [22, p. 40].

The spreadsheets are frequently the primary technologies tracking the samples and the orders. When the amount of data increases, the interaction between the database and the spreadsheets becomes continuously more challenging. Therefore a relational database is frequently the solution for the management of huge amounts of data. The LIMS should be able to accommodate with the existing laboratory sample flow and the database classes. Moreover, a flexibility in case of upgrades and possible new technologies is a usual requirement as well. [22, p. 38]

3.3.3 Current communications used for laboratory information management systems

A communication system consists of communication nodes which couples automation components and offers service-oriented application protocols. The communication protocols between communication nodes are client-server based to enable a flexible system integration, as can be seen from Figure 3.3. Bidirectional communication relationships between LIMS and laboratory automation components behind laboratory process control system and laboratory instrument control are represented in the Figure 3.3 as well. The data flow and operation layer is represented in Figure 3.3 as integration of LIMS and electronic laboratory notebooks (ELN), that maps and resources the workflow-operations within the communication nodes. [13, p. 842]

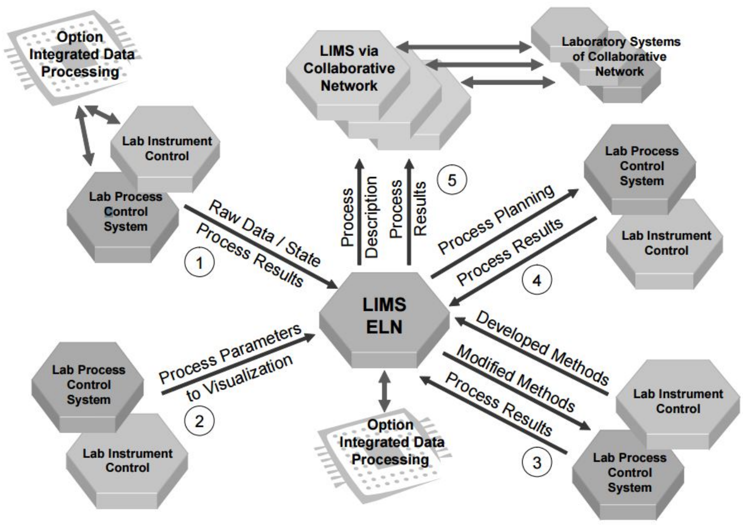


Figure 3.3 Communication relationships of the data flow and operations layer [13, p. 842].

ELN is a software application designed to replace the traditional manual paper notebooks in the laboratories. The ELN is used by laboratory technicians to document, register, save, retrieve and share the laboratory records within the organization fulfilling legal, scientific and technical regulations of documentations, especially in the medical and pharmaceutical industry. The advantages of ELN compared to the paper notebooks are that the search feature can be utilized, the research data is structured, sharing and collaborations with other users is enabled, and the security is at higher level. Links can be created between records at the other notebooks by structured data identification as well, and a scheduling can be designed for routine procedures. In practise the ELNs can be divided to two categories; a specific ELN is designed to support specific applications, data types and scientific instrumentation, and a generic ELN that's responsibility is to support access for all the data and information that is required to be recorded in the ELN [18].

The service-oriented communication protocols can be implemented with e.g. the network standard TCP/IP and tunneling protocols to ensure a distributed redundant communication nodes. Thus the requirement of efficiency, reliability, data capacity and real-time data transfer can be achieved. A fast and efficient real-time data exchange within the communication network can be achieved by web service architectures to monitor data for interfaces and relational databases to offer LIMS new possibilities for data analysis and visualization. [13, p. 842]

With relational databases a flexible cross linking for laboratory data and mapping administrative-oriented workflows are achieved by management of relationships within the laboratory information classes. A workflow mapping framework sharpens collaborative system integration to its best being responsible of scheduling of communication tasks with synchronizing sequences and managing synchronization errors. Managing the message structures for cooperative and dynamic process mapping exchange for evoking communication operations will enhance the performance as well. [13, p. 843]

3.3.4 Laboratory information management system architectures

Several LIMS architectures have been utilized during the years along with the technology development. The following architectures have been used over the years [18].

Thick-client architecture is a traditional client-server based LIMS architecture, where some of the system functionality is located at the clients and the rest is on the server. The LIMS software and the data processing are located at the clients and primary task of the server is to store the data. Hence the upgrades and updates are implemented directly to the client machines. The advantages are higher processing speeds and security, since the processing and access are directly related to the client machines. The disadvantages are the lack of required robustness, the updates require more time than the web based solutions and the web based functionality itself is a disadvantage as well. [18]

Thin-client architecture is a modern approach with an access to a full application functionality through a web browser. The LIMS software and the data and information processing are located at the server, enabling all the upgrades, developments and modifications being done at once and only from the server side LIMS software. The clients need only a web browser to access the LIMS functionality. Thin-client architecture lowers significantly the cost of ownership and client side maintenance and upgrade costs. However, the architecture requires a real time server access and can provide slightly less functionality since a increased network throughput is required. A few LIMS vendors utilizes the thin-client architecture as software as service (SaaS) solution. The ability to configure and customize the solutions are more limited than on premise solutions. Therefore the SaaS solutions are implemented to laboratories with less throughput and users than on average laboratory. [18]

Web-enabled architecture is basically the thick-client architecture with an implemented web browser add-on. Therefore the mobility integration is enabled and the users can access the LIMS through any device's web browser. However, the functionality is limited to only certain features in the web clients. A cross platform functionality is required as well since both the server and clients include functionalities of the LIMS solution. [18]

Web-based architecture is the most complex of the used LIMS architectures, with combining features and characteristics from both the thick- and thin-client architectures. The web browser includes most of the client side functionality. However, e.g. Microsoft's .NET Framework technology or other framework is required to be installed on the client devices. There is a support for the mobile device platforms as well with the full functionality [18].

3.4 Solutions available on the market

LIMS began to generalize on the market over twenty years ago and has been established since then. According to recent market surveys a demand for customized solutions holds still an important segment due to a great volume of customer-specific requirements [13, p. 841].

Open-source software is a common option for development since many applications are freely available on the Internet to be at least partially adapted to the system under development. Moreover, a distinct reason is the open-source community where application knowledge can be shared and transferred with the fellow developers. However, open-source licenses are frequently restrictive of the authorization of the usability or reciprocally the licenses demands to share all the related and integrated systems and products as well. [26, p. 2859]

Calabria, et. al. evaluated LIMS solutions available on the market [6, p. 4]. The evaluation consisted of commercial LIMS solutions to open source solutions. The findings were that the commercial or stand-alone LIMS solutions are frequently rather expensive. The disadvantages are that the LIMS solutions lack the flexibility and scalability considering the sample data management and customizing the ad-hoc procedures and analyses. [6, p. 4] In addition the research group studied open source software implemented specially for biological industry laboratories, such as Bika LIMS [3], LabKey [16] and Galaxy [30]. The research perspective was the potential for customizing the LIMS and utilizing the built-in functionalities such as workflow editing and configuration. However, none of the open source LIMS software passed the requirements of the research group due to lack of the fundamental activities such as exporting the laboratory data and reporting service. [6, p. 4]

4. DESIGNING THE MODULE FOR METAL AND MINING INDUSTRY

One of the most powerful and descriptive method of designing an information system is to build models to represent aspects of the system. Models describe the features and the requirements of the information system in a more simplified way, helping the realization of the the most efficient set of determined features during the design phase. Information systems are ever so complex, and thus models are created to describe a few aspects of the system at a time to simplify the system structure [29, p. 131]. It will be advantageous to build multiple different models to cover as many aspects of the system as possible. Different models can be used to describe the system from various points of view, and different levels of abstraction to provide the most comprehensive overall picture [29, p. 131]. This chapter describes how the UP is applied for designing and modeling the LIMS module to enhance the laboratory business processes in the field of the mining and metal industry.

4.1 The sampling process

A thorough familiarization with the sampling and the analyzing processes is required with understanding the creation of the samples and producing as well as interpreting the orders. A lack of understanding the process will cause a risk for LIMS solution to be unsuccessful in addition to the technology being inadequate. A proper study of the process will reduce problems in the subsequent stages of the development, including especially modeling the system that naturally reflects to the implementation. [25, p. 191] Further, the knowledge of the process is the base for tailoring the LIMS to fulfill the individual requirements.

The mining and metal industry is mainly a continuous process. Therefore the throughput is voluminous and the sensors and on-line analyzers have an essential role. Figure 4.1 represents a sampling process diagram from one customer site that was visited for getting know to the mining process at the plant. This sampling process is only one individual sampling process, but the basic technique is the same [21].

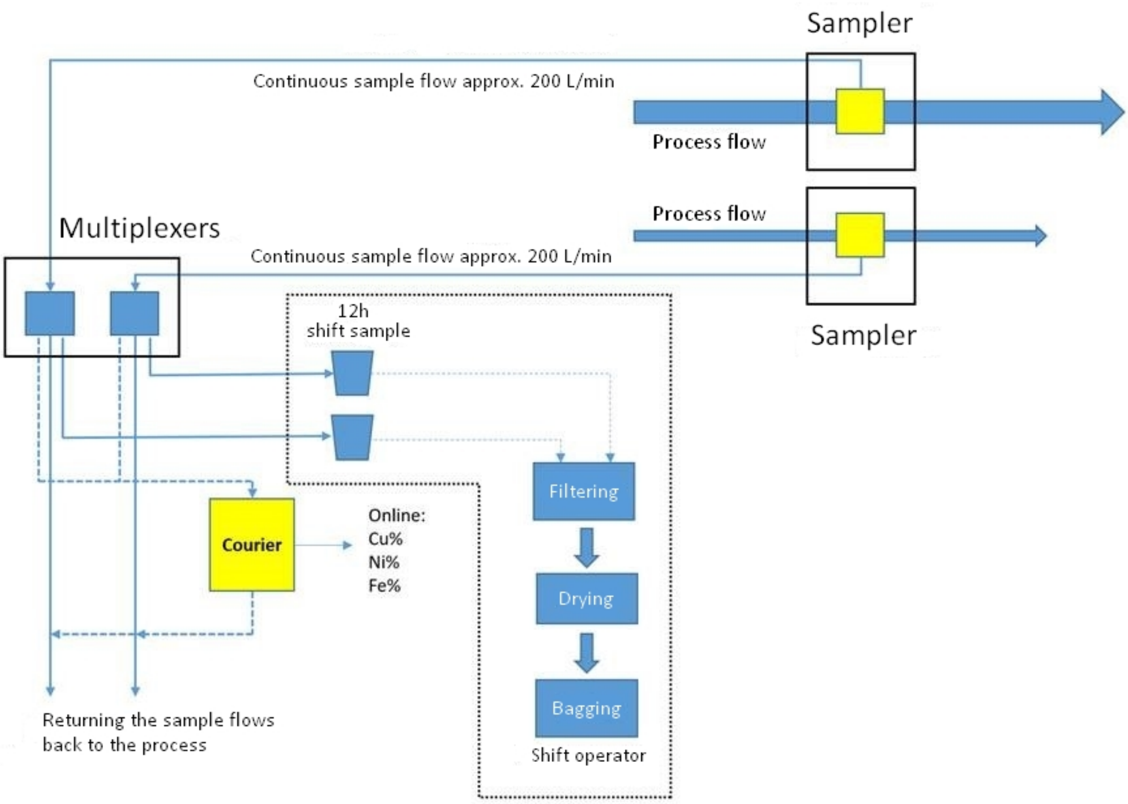


Figure 4.1 *Sampling diagram representing one sampling process from customer site [15].*

As represented in Figure 4.1 a sampler takes a small portion of samples from the whole process flows. The process flows are slurry flows, that are products from the mineral processing grinding and flotation machines. The process flows are set forth from slurry pools. The slurry process flow is a continuous flow. After going through the sampler a sample flow goes through a multiplexer, while rest of the slurry flow is returned to the slurry pools. The multiplexer demerges the sample flow: one part-flow goes through an on-stream element analyzer (Courier). The measurement principle is based on a laser-induced spectroscopy or x-ray fluorescence (XRF). The laser-induced spectroscopy and XRF are powerful methods for measuring metal elements from the slurry. The other part of the sample flow is further divided to form shift- or daily samples into respective sample buckets. The shift samples are collected continuously during each shift, i.e. typically three times a day, and the daily samples are collected once a day. The rest of the sample flow is returned to the slurry pools. The collected samples in the buckets go then to a pressurized filtering to remove the water, then to an oven for drying and finally divided to bags with a unique sample id to identify every individual sample. The sample id can consists e.g. from a creation date, a sample type and a shift id. The coded sample bags are sent to laboratory for further analyses. So far the sample id's have been manually written to a paper, to be attached to the sample bags. The sample id is an unique id, formed by a timestamp, a shift id, a sample code and an additional running number depending on a sample category, to ensure unique sample id's. The sample code contains an information of the elements need to be analyzed and the methods that have to be used.

4.2 Identifying use cases

Use cases model activities that the user requests the information system, LIMS in this case, to implement. The use cases describe the process from the user point of view as well. Several techniques to identify and model use cases have become established. Identifying the use cases by listing different types of users and user roles, as well as mapping the objectives of their job is an effective systematic approach that can go into the details if needed. Another technique of constructing a narrative description of user roles, usually by interviewing the users of their job tasks, is preferably used for constructing the list of the use cases, since the list includes only the key features and cases without narrative content [29, p. 166], [8, p. 139–140]. The third technique, event composition technique, is the most comprehensive guideline for modeling the use cases. The event composition technique identifies all the events the system must respond to, and accordingly listing the ways how the system will respond. In addition the user roles are listed in the model.

However, in this Master's Thesis the use case list is the most appropriate way of identifying the use cases, since the research question is about platform independent generic system, and too detailed description of use cases is inappropriate as every process have its own unique characteristics and special use cases. In addition to the use case list, a user journey map is implemented to form an overall picture of main functionalities using the LIMS module from the user point of view. The use case list for this Master's Thesis has three columns, the first pointing the user role (Who), the second for the activity (What) and the third for the reasoning behind the activity (Why), to gain more comprehensive understanding of the process and the actions of the user roles relating to the process. The use cases were combined tentatively from the use cases of the existing system that had to be developed. Predicted development requirements were added according to requests from the customer along the years of use of the system. The use case list was then validated and finalized during and after the user interviews. Table 4.1 represents the tentative use case list for identifying the use cases for this Master's Thesis.

Table 4.1 *The list of the use cases for the LIMS module.*

Who	What	Why
Customer LIMS admin	Define methods (analysis methods), sample types, scheduling	Not need to be dependent on vendor
Process/Plant Operator	Sample collection from process, task acknowledgment (collected sample id's that are incoming to laboratory)	To document sample collection
Process/Plant Operator	View the latest LIMS results in PCS/DCS screen	To see the newest results/status
Process/Plant Operator	Event triggered sample creation (e.g. sample from outgoing truck)	To verify that all the outgoing material is analyzed
Metallurgist	View and analyze sample reports	To verify the process quality and functionality (to perform process quality)
Metallurgist	Manual sample corrections (daily activity)	To ensure the in and out flow balance (reconsiliation)
Metallurgist	Receive and acknowledge alarms related to missing/invalid analysis results	To act on problems and/or deviations
Metallurgist	Manual sample corrections based on alarms (not necessarily (not hoped) to be daily activity)	To perform fix to problems based on alarms
Metallurgist	Manual correction due to incorrect manual input	To fix incorrect input
Metallurgist	Approve the sample analysis results	Prior sending sample to calculations to prevent the use of incorrect or unapproved results (Nice to have)
Metallurgist	Event triggered sample creation for calibrating online analyzer	To check/calibrate online analyzers against the laboratory
Metallurgist	Create campaigne samples manually (define methods, sample types, scheduling)	There are "campaignes"to analyze certain themes
Metallurgist	Export data to Excel	To do further analyses
Laboratory Technician	Enter test results manually for related sample (+ validation)	Possibility to provide enter data offline/manual analyzing, to save results to system
Analyzer Device	Send result data (e.g. .csv/XML) to LIMS system (+ validation)	To save result and send to right place
Operation Support Engineer	Combine the data with other production (etc) data	To be able to produce reports and calculations based on LIMS analysis result

Table 4.1 describes the use cases for five user roles that require human interaction and one user role for an automatic interaction through an interface. The human interaction roles are a customer LIMS administrator, a process plant operator, a metallurgist, a laboratory technician and an operation support engineer. The automatic analyzer device refers to a laboratory analyzer device that stores the test result data.

The LIMS administrator is named for the customer side also for configuration purposes, for that the customer does not need to be dependent on vendor or system developers. The customer LIMS admin designs and configures LIMS workflow scheduling regularly, and defines analysis methods, sample types and elements when needed.

The process plant operator collects and prepares the samples from the process according to the process description in the previous section "The sampling process". In addition there can be so called event samples triggered by an event, such as truck dispatches heading to a dock and a ship for further logistics. These event samples are collected from the load from the trucks. The operator have to verify that all loads from the trucks are sampled and sent to analysis by creating the samples in the LIMS module UI. Finally the operator is able to see the latest LIMS results from the distributed control system (DCS) screen, and can then compare the results for the online analyzer results and make automation control set point changes if needed.

The metallurgist is responsible for analyzing, validating and controlling the samples, the sample results, and the process quality. The metallurgist views and analyzes the sample reports, and implements sample result corrections manually based on his/her observations if needed. The metallurgist receives alarms related to possibly missing or invalid sample result values, acknowledges the alarms, and implements sample result corrections or corrective actions manually according to the alarms. Manual sample result corrections can be needed also related to manual input, i.e. if sample results are entered manually to the system, by the metallurgist as well. Finally the metallurgist approves the sample results after the validations. The metallurgist is responsible for creating calibration samples for calibrating the online analyzer, triggered by reports comparing the laboratory sample results and the measurement results from the online analyzer from the process. The difference in the comparison can be the most efficiently visualized by drawing lines graphs from the results into the same graph. Thus the difference on the the lines diverging from each other can be instantly established to result from the difference in the sample result values. The metallurgist have to have an option for exporting the sample result data to spreadsheet, e.g. Microsoft Excel, to do further analyses. Spreadsheets enable easily and quickly creating graphs.

The laboratory technician may need to enter test results manually in offline conditions, or if there is manual analyzing performed. The laboratory technician is the one who implements the required tests from the samples, but have a little to do relating to the LIMS module. That is due to increased automation made possible by the laboratory analyzer device. The analyzer device is used for taking the test and analyses from the samples but also sending the test results as e.g. a csv-file to the LIMS module automatically.

The operation support engineer represents a process engineer that is responsible for the whole production quality, and who combines the whole production data for further analyses and calculations.

4.3 User journey map

A user journey map provides a comprehensive overall picture of using the information system from the user point of view with a sequential visualization of the process and the experience the user goes through using the system to reach the specified goals [32]. The user journey map helps to understand and visualize comprehensively the user and the customer needs, requirements and thoughts using the system, all leading to detect development opportunities [32]. Combining storytelling and visualization, the user journey map visualizes the whole process of work tasks and pain points in using the system [32]. Therefore the user journey map provides a visualization of the workflow of the process in the user point of view as well. The user journey map requires a pre-study or an initial use case list in order to create it comprehensively. User interviews can be used to add content to the user journey map, especially for describing the user experience and the thoughts and the pain points the users go through. In this Master's Thesis an initial list of use cases and the user interviews were utilized for designing and implementing the user journey map, to build an overall picture of using the LIMS module.

The user journey map consists of certain key elements. The actor(s) of the story are determined to determine the point of view the user journey map is made for. The scenario of the map is determined to describe the process more detailed way. The most significant part of the user journey map is describing the user actions and the experience through mindsets and emotions [32]. Finally developed functionalities are listed as an output if the user journey map for current developed system is made being based on an existing system. Alternatively the output can describe the business goal the map supports or the pain points discovered.

4.4 Requirement specification

The requirements of the features that the LIMS is required to execute to fulfill the business needs were based on several sources. First the features of an existing solution were studied by reading to the documentation and the code, and studying the database structure in addition. The fundamental approach for system development with an existing system is to getting to know the old system properly. There can be seen bottlenecks and problems quite easily with recognizing the essential basic functionalities that can be still functioning outstandingly. Studying the documentation and the implementation of the existing system efficiently asks for taking the role of the system user or administrator to see the practical advantages and problems from the system instead of the pure functionality and performance to handle the daily tasks. The better performance follows more efficient and easier working framework build by system development.

The uses cases of the LIMS module were developed and utilized to specify the executors of the system requirements for the LIMS module of this Master's Thesis. The user roles are not necessarily added or modified, but the new development areas are mapped for corresponding user roles and groups. Moreover, the system developments are intended to reduce the workload of the users and to ease their daily job. However, the use cases have to be determined to model the important features from the users point of view and to clarify the basic functionality that is the most important task of the LIMS module and what is most strongly expected.

The user interviews were analyzed and utilized for validating the requirement specification. The process know-how gained by the users through several years of experience was thoroughly considered and utilized. The know-how of the users and the stakeholders was received in addition to the interview as well, along the SDLC, by making specifying queries to the users and the stakeholders of the required functionalities of the LIMS module in detail when needed. The specifying queries were made unofficially by calling or writing email, thus no official documentation were made in addition to notes for system development purposes. Moreover, since the interviews were carried out during the development and due to the iterative and adaptive structure of the UP SDLC, some features were considered and added during the later phases of the elaboration phase and in the construction phase.

Further, additional change requests from the customer for the existing system were considered as "nice to have-features. Finally, the LIMS feature theory of this Master's Thesis was an essential source for validating the features and requirement referring to the theory and open source solutions, especially based on Figure 3.1. Moreover, "nice to have-features were included referring to the modern features that are yet to be largely tested for various processes.

4.5 Architectural design

Modern information systems requires a complex deployment environment based on composition of computer hardware, communication networks and system software to operate effectively. The hardware consists of servers, client workstations and possible additional related equipment such as printers. The communication networks consists of local and remote communications and the communication hardware related, like routers and firewalls. Finally, the system software consists of the operating systems with the external databases, the database management systems, the system interfaces, the service software, and the security service software. In case the organization have complex existing architecture, the integration with the new information system could point out as a significant challenge. Moreover, the reliability, security, throughput and synchronization have to be comprehensively analyzed and designed. [29, p. 264]

4.5.1 Microservices as an architectural model

The deployment environment consisting of the computer hardware, the communication networks and the software can be built around a single service or a multiple microservice architecture depending on the information system complexity and the workload. The microservices can be deployed to physical computers or virtual machines within an organization network. Figure 4.2 represents the microservice architectural model for a modern deployment environment. [4, p. 159]

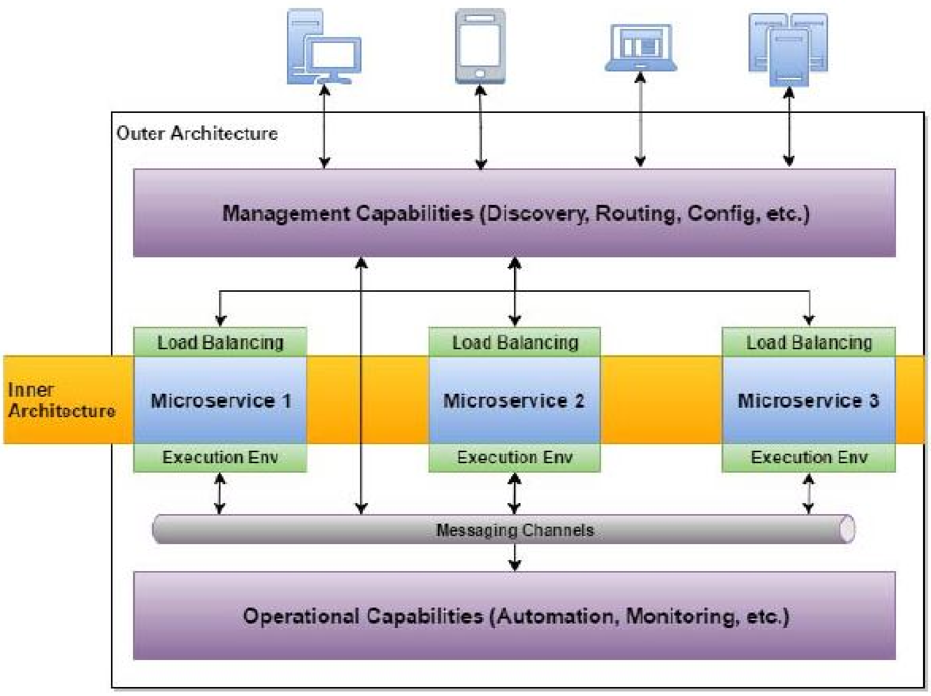


Figure *4.2 Architectural model for microservices [4, p. 159].*

One efficient solution for handling complex information system integrations is to distribute the service into several separate microservices. Microservices refer to distributed architecture model of service-oriented architecture (SOA) by applying an approach for combining a set of independent smaller services into a single application. In Figure 4.2 the application is divided into three microservices that are connected via messaging channels into the process and operational capabilities, and into the clients via routings. Each of the microservices runs its own process and are independent of each other, thus the microservices have their own inner architectures. Microservices can communicate with lightweight protocols such as TCP/IP or other HTTP resource API [4, p. 158].

Each distributed microservice represents a single functional unit, a bounded context, that is as small, simple and coherent as possible. Therefore each microservice unit can be built with different programming language and each microservice can use different data storage technology if needed, i.e. the most favourable technology for each bounded context unit can be used. Moreover, microservices provide stronger software resilience compared to the single service approach, i.e. if one microservice fails it does not affect to the other microservices. Finally, a new microservice can be deployed into the application faster and easier, possibly avoiding modifying the core service and the other microservices at all if the microservice architecture is properly designed making the information system agile, flexible, scalable, easily maintainable and independent of technique and other software [4, p. 159], [35, p. 155].

Although each microservice is less complex than a single service application, the overall complexity of the system consisting of microservices is not decreased. As can be seen in Figure 4.2 the complexity of the system is transferred from the inner architecture to the outer architecture of the services, since the multidirectional communication with possibly multiple simultaneous request-response messages increases the overall complexity. Moreover, in case all or most of the microservices are built with different techniques and programmming languages, the technological diversity can increase to an overwhelming level [4, p. 159].

A client-server architecture further divides the software into a server and clients. A server manages the information system resources and provides a well-defined, bidirectional service. A client communicates with the server requesting resources or services, and the server responds to the requests. The typical arrangement is to place the server on a separate computer or virtual machine, and distribute the clients to every site so that the users can use the information system. [29, p. 277] Figure 4.3 represents the bidirectional communications between one separated server and the distributed client machines.

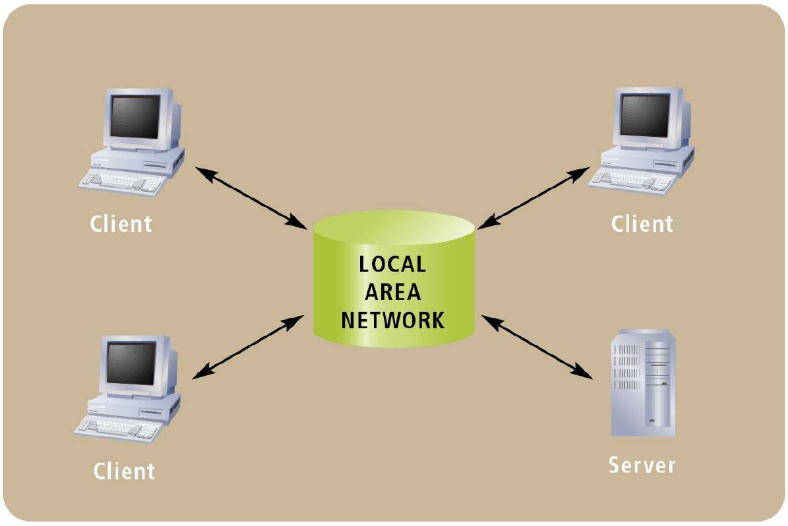


Figure 4.3 *Bidirectional communications within a client/server software architecture [29, p. 279].*

The client-server based architecture enables the laboratory to utilize wide area network, allowing the data to be accessible collaboratively among different laboratory departments. Moreover, the information sharing is significantly easier. In the client-server architecture the database and the tables are located in the server and the user interfaces are distributed along the client machines. Therefore the data processing can be handled only at the server, decreasing the load at the client machines. [22, p. 42]

The advantages of client-server architecture are the location flexibility, scalability and maintainability. The disadvantages are that the complexity increases, the client-server can potentially have poor performance, and there can be security and reliability issues [29, p. 279].

4.5.2 Software architecture and database models

Structural composition of the information system forms the software architecture. The database is one of the most essential components or systems forming the information system. The database is an integrated data and information store for the whole information system. The database is managed and controlled by a database management system. The data in the database is structured within class attributes, e.g. sample id, sample type and category, as well as can be associated with other classes, e.g. associations with a category determines all the samples belonging to the same category. The data values can also be restricted if needed and with security settings the user access rights can be determined [29, p. 398]. Arguments in the favour of choosing the databases and the database management systems for storing the data for the modern information systems are related to the ability of accessing the data. The data can be accessed simultaneously by several users or systems and without necessarily having to write data requesting queries. Many database management system are well standardized and uniform through different system versions as well [29, p. 399].

Along the history of using the database management systems four database models to manage and access the data store have become established due to widely spread use. The hierarchical database model is the oldest database model that structures the data in sets of hierarchically organized records. The network model is another old database model structuring the data in sets of records as well, but enabling more flexible network data structures in addition. However, the hierarchical and the network models have not been developed for last few decades and are relatively fractionally used nowadays [29, p. 399–400].

The object-oriented database model is one of the two widely used and developed database models. The object-oriented model divides the database model into classes, describing the objects, variables and the data structure, that will be managed by the software. The corresponding classes can be divided across the architecture locations and interact bidirectionally with a database management system via communication protocols. The database design depends and especially refers efficiently with the service class design, since the classes use the database tables and data; the object-oriented class design includes a direct support for a class-inheritance, object linking and a programmer-defined data objects [29, p. 400]. Object-oriented database models are expected to override gradually relational database management system for more traditional business processes as well. Commercial object-oriented database management systems are provided e.g. by GemStone, ObjectStore and Objectivity.

The relational database model is the second widely used and developed database model. The relational database model stores the data in database tables, which consists two-dimensionally of columns and rows. However, a row is called in relational database terminology as a row, a record or a tuple. A column is called an attribute or a field, and finally a cell in a table is called an attribute value, field value or data element [29, p. 407]. In the relational database model, each table must have a unique key that is an attribute or set of attributes, which values exist only once per table. If there is only one unique attribute or set of attributes which every attribute value are unique in a table, the key is then called a primary key. The primary key can be natural or designed. For example the atomic weight of a chemical element is always unique for every element [29, p. 407]. The keys are utilized for representing relationships between different tables, thus they are an essential part of the relational database design. For example one product in a table may have a relationship to another table, that includes several rows of additional information relating to only that specific product. The attribute values in the relating table referring to the primary key in the main table are called foreign keys [29, p. 407]. Figure 4.4 represents the relationship between two tables.

sample_table_id	sample_id	type_id	sample_type_code	category_id	status_id
11135	ABC201505233	1785	NCT DEV	5	6
11134	MFF201505233	1784	MFF DEV	5	6
11133	SFC201505233	1782	SFC DEV	5	6
11132	CFC201505233	1783	CFC DEV	5	6
11131	NFC201505233	1787	NFC2 DEV	5	6
11130	ABC201505231	1785	NCT DEV	5	6
11129	MFF201505231	1784	MFF DEV	5	6
11128	SFC201505231	1782	SFC DEV	5	6
11127	CFC201505231	1783	CFC DEV	5	6
11126	NFC201505231	1787	NFC2 DEV	5	6

sample_result_id	sample_table_id	sample_id	link_id	order_id	element_id	value
2696	11135	ABC201505233	1123	NULL	294	10.28
2695	11135	ABC201505233	1114	NULL	293	0.00
2694	11135	ABC201505233	1105	NULL	292	0.00
2693	11135	ABC201505233	1096	NULL	291	1.18
2692	11135	ABC201505233	1087	NULL	290	0.71
2691	11135	ABC201505233	1078	NULL	289	3.57
2690	11135	ABC201505233	1069	NULL	288	0.00
2689	11135	ABC201505233	1060	NULL	287	3.82
2688	11135	ABC201505233	1051	NULL	286	0.00
2687	11135	ABC201505233	1042	NULL	285	37.00
2686	11135	ABC201505233	1033	NULL	284	2.38
2685	11135	ABC201505233	1024	NULL	283	0.00
2684	11135	ABC201505233	1015	NULL	282	3.12
2683	11135	ABC201505233	1006	NULL	281	0.00
2682	11135	ABC201505233	997	NULL	280	25.06
2681	11135	ABC201505233	988	NULL	279	4.75
2680	11135	ABC201505233	979	NULL	278	0.58
2679	11135	ABC201505233	970	NULL	277	2.39
2678	11135	ABC201505233	961	NULL	276	0.75
2677	11135	ABC201505233	952	NULL	275	0.91
2676	11135	ABC201505233	943	NULL	274	0.00
2675	11135	ABC201505233	934	NULL	273	5.38

Figure 4.4 *Relationship between the data in two tables in the relational database.*

In Figure 4.4 the table below has a foreign key *sample_table_id*, which refers to the corresponding unique primary key in the table above. The attribute values *type_id*, *sample_table_code*, *category_id* and *status_id* for the primary key applies to the records in the table below as well, that refers to the primary key. There can be also thousands of records for other *sample_table_id*'s in the table below. The naming conventions of the table attributes should correspond to the data objects in the information system project. A widely utilized relational database software environment is based on Microsoft Windows .NET, and SQL Server as the data store with SQL Server Management Studio managing the data as a relational database management system. There are other deployment environments which forms very different structural capabilities and characteristics. An example of other database environment is Oracle [29, p. 265].

The key for the entire information systems architectural design is to make decisions about database servers, database management systems and database distribution. The database architecture has an influence also on network design, the information system security, architecture of Web services and finally services of the rest of the components [29, p. 431]. Decision of the database distribution is about storing the data across many different databases, since it is uncommon for an organization to store all the data in a single database. The reason can be existing databases remaining from the earlier system development, or an existing database that have to be integrated to the information system. Finally it would be advantageous to have the applications and the data integrated physically near each other in order to improve the system performance [29, p. 423].

The key for determining the database model for the information system is whether there is existing databases to interact with or not. The designed database must adapt to the existing database(s) and therefore accommodate the constraints included in the existing databases. The database architectural design and the database adaptation combined forms one of the highest risks in the information system development through both financial and operational risk point of view. The financial costs can be increased unexpectedly without a proper design and with a possible disruption of the use of existing systems. The operational risk forms of the risk of the system adaptations and integrations [29, p. 432–433].

The most widely spread database model in use is relational database model [29, p. 407]. Therefore the integration and the adaptation for the existing databases and systems and the other separate systems under development in order to interact together is the most seamlessly and effectively handled by determining the relational database as the database model for the information system.

4.5.3 Naming the database objects

Designing the database structure and data elements requires applying appropriate standards. The structure is required to accommodate to the laboratory workflows, e.g. certain sample type has certain sampling methods, that includes certain elements. Further, the sample type has to be linked to additional data, sample results and orders. The data elements have to be designed according to a standard, to ensure uniform naming conventions for technique independence and user satisfaction. ISO-11179 was used to standardize the metadata by describing rules for naming the data elements [7, p. 13].

Naming for database columns, tables, stored procedures etc. should always avoid special characters and the names should start and end with a letter. Underscore is a best practise for indicating a space relating to a normal text, e.g. naming a column or a variable "sample_id". The underscore enhances the readability as well compared to a camel case style, e.g. naming the same column as "sampleId" [7, p. 15].

Styles for database tables have slightly diverse guideline. According to ISO-11179-4, the tables, sets and other collections should be named with a name that represents a collection, class or to be in a plural form. Moreover, the standards of the industry has to be used as a base for naming the data elements. Collective, class or plural table names awakes a mental image of the object consisting of a set of elements. E.g. "SampleTypes"creates a more descriptive name especially for a table than a name "SampleType". The names of the tables are more convenient to name with a camel case style, since the table names should always differ from the column- or variable names [7, p. 17–18].

4.5.4 System security

Nowadays information systems are deployed in networked environments, thus the security matters are significantly crucial components to protect the organizational assets. An access to the interfaces and the databases has to be granted only for authorized users or user groups with the interface and the database security settings. Further, the communications through the networks has to be protected in case of malwares or spywares [29, p. 266]. The most of organizations have gateways installed to prevent security threats against the internal systems through Internet. Preventing and controlling unauthorized access is one of two main focuses in the system security. The other one is protecting the transactions and communications through the network using data protective techniques during the transition from the source to the destination [29, p. 513].

The security access limits access to determined components or functions within the information system, limits access to the data for authorized users only, and finally restricts the access to the hardware. The system security access control functionality can be embedded in the information system software with an adequate design. Therefore a uniform security controls can be distributed for every resource in the information system [29, p. 514].

The development lifecycle of the transaction security has established a few standard protocols through the years to control the authentications, privacy and integrity. The secure sockets layer (SSL) is one of the original secure transaction protocols, which was adopted as an Internet standard being developed to transport layer security (TLS). Creating a secure TLS connection through the Internet creates identity certificates for both sender and receiver. The sender and the receiver verifies the identities for each other by exchanging the public keys, i.e. the identity certificates, to enable secure message transactions. Internet protocol security (IPSec) is a more modern Internet standard for secure message transactions. IPSec can operate with higher speeds due to it operates at lower network protocol stack layers. IPSec supports more secure encryptions than SSL as well, and can be utilized to replace SSL and TLS [29, p. 521]. Finally secure hypertext transport protocol (HTTPS) is an Internet standard for Web-based security. Every modern Web browser and server supports HTTPS. HTTPS enables transmitting Web pages securely.

4.5.5 System interfaces

System interfaces manage frequently extensive input and output (I/O) for the information system. System interfaces execute data transfer and sending and receiving requests, events and messages within the information system from component or unit to another. System interfaces handle operations between automated systems, having a minimal or no human interaction, and the I/O requirements for the system interfaces form a fundamental for the information system to operate independently, i.e. operate as a service [29, p. 487]. Identifying and taking advantage of opportunities for automated system interfaces have a significant impact on return on investment (ROI) as well. Examples of standard outputs from the system interfaces include reports, printed forms and data objects sent to other automated systems. Correspondingly standard automated inputs, or inputs that come from non-user devices include barcode readers, automated scanners, pattern recognition devices, file handlers observing specified folder for incoming files and other systems part of system interfaces [29, p. 488].

Modern highly integrated operating information systems require real time input processing, interacting within components and distributing outputs over the system interfaces. Including as many automated system interfaces in the information system as possible increases the system performance and return on investment (ROI) since the performance requirements for processing I/O grows increasingly concerning performance speed, efficiency and accuracy, and being able to process I/O 24 hours a day, 7 days a week [29, p. 488].

Inputs to the system interfaces are received from other information systems or components as network messages or events. The input message or event triggers the actual processing in the information system, e.g. incoming file in the specified folder is an event that triggers parsing the file into specified structured data object, following sending a message to another system to send the data object forward or writing the variables in the data object into an external database. In addition to inputs, also internal messages can be sent within the information system between different components. The difference between inputs and internal messages is that inputs are received via system interfaces. An example of capturing inputs would be a barcode scanner that could be used to trigger printing labels according to the barcode or sending acknowledgement of receiving the barcode at corresponding location. Input can be received from external databases as well. For example an expiration date can be specified for a record, and when the record threshold time has been exceeded, the database triggers an event input for the information system [29, p. 489–490].

Outputs from the system interfaces are processed in the information systems. The system can produce large amounts of data and storing the data into an external database would be frequently the best practise. For example reports are produced from a database data that is produced from system interface outputs, and finally the reports are distributed further to clients [29, p. 490].

An example of a widespread system to system communication message language is extensible markup language (XML). XML is formed by data structures that is also in a human friendly and self explanatory format. The data can be handled more fluently due to well structured format than older communication methods, e.g. hypertext markup language (HTML). Comma separated values (CSV) is a powerful format for sending result values or other lists through interfaces, since parsing and forming csv files are widely used techniques in the industry resulting in various open source solutions. Furthermore, spreadsheets are in wide use in the industry anyway, therefore the csv-files are applicable as they are as well. When the data structures are defined at every system in question, the assembling, processing and dissembling the transactions can be done seamlessly [29, p. 491].

4.5.6 Communication network within the architecture

The wireless network is widely applied with LAN networks, and the standard network protocol within computer networks is TCP/IP providing the communication services. As the information management system, LIMS requires standard protocols and software for storing the data in addition, such as Structured Query Language (SQL) via an Open Database Connectivity (ODBC) [29, p. 278–279].

A web-based software architecture is a complex approach of the client-server architecture favourable especially for complex information systems. Web-service standards like WebSocket enables a web-oriented client-server architecture. The WebSocket protocol provides an efficient real-time data exchange and communication from the server to the clients over the network. The WebSocket communication is also faster than HTTP polling according to the research of Pimentel and Nickerson [23, p. 45]. The WebSocket protocol enables a bidirectional communication between the server and the client over a single TCP socket. The WebSocket consists of two parts: the handshake that is a response from the server to a request message from the client and the data transfer over the network [23, p. 46–47], [24, p. 8].

A commonly known service-oriented architecture (SOA) is a communication model between the processes and services. The SOA packages application software into a server software, which enables connectivity and access from any Web browser to manage the data and information to implement the business processes. The SOA is particularly business-oriented architecture, thus the service requests and responds can be transmitted in extensible markup language (XML) documents over the Internet. XML is standardized language especially for business use [29, p. 281–282]. The key for using the SOA is flexibility, since the accessibility is straightforward and flexible, the communications are not complex nor expensive, and finally the standards are widely used. However, the security and reliability risks and potentially lower throughput are disadvantages of the SOA approach [29, p. 283].

Microservices is based on the SOA model extending the functionality, efficiency and flexibility beyond the concept of how the information system is distributed. Microservices are very similar to the SOA with services being independent and autonomous with explicit system boundaries. Additional important features brought with microservices are lightweight communication across the components via WebSockets and the deployment of the microservices can be done independently without even touching to the core service or the other microservices [31, p. 2–3].

4.5.7 Microservices and the Cloud

Limitations and challenges in virtualization have been gradually reduced by recent advancements in container technology. Thus capability and prevalence using containers in the cloud and cloud-hosted software have taken a potential step towards realizing deployment of the microservice architecture into the cloud, also lowering infrastructure and maintenance costs at the same time [9, p. 10]. Updating and deploying the software in the cloud would be time and cost efficient due to dynamic cloud infrastructure with optimized resources. Using microservice-based software distributes the load and enables even more efficient resourcing and independent and durable operation, as can be seen from Figure.

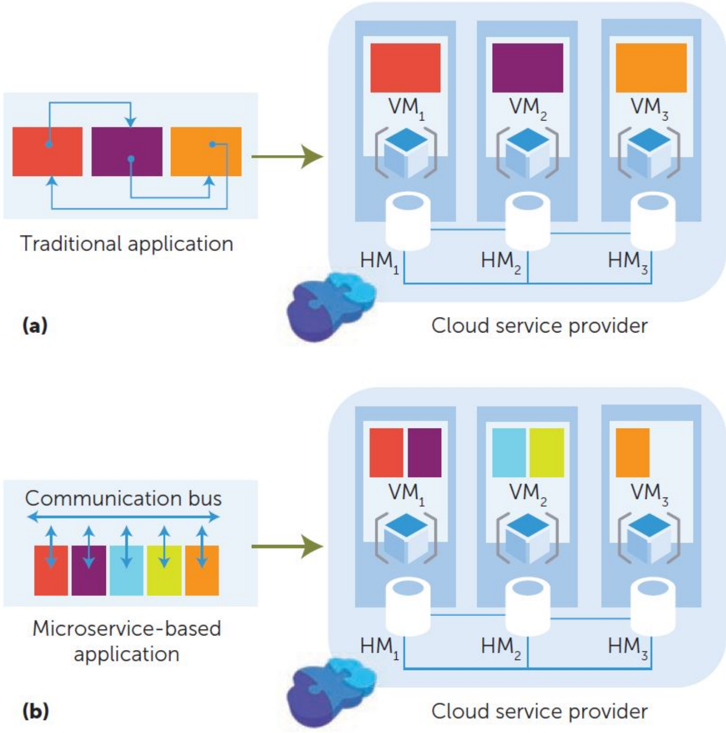


Figure 4.5 *Deploying a)traditional b)microservice-based application into service provided in the cloud [9, p. 11].*

Microservices are intrinsically cloud-native since they can run their own processes without being dependent on others. The data storing can be segmented into smaller partitions as well, with respect to the centralized data storage. The cloud provided microservices avoids also a technology lock-in since the development and deployment can be handled in that efficient and fast way. Also reduced time-to-market and allocated development teams per their own speciality would be strong advantages of cloud-hosted microservices [2, p. 42–43].

However, distributions and deployments in the cloud results in vulnerability to breaches and cybersecurity threats [9, p. 12]. There are no standardized communication protocols or data formats for communication between microservices. Cloud-hosted microservices are under various networking issues as well, although there are researches of software defined networking and network function virtualization to resolve the networking problems. Finally a standard large-scale scheduling platform with optimized resources is yet to be realized slowing down becoming more common technique for deploying the services [10, p. 81–83, 88].

Deploying microservices in the cloud would be a interesting idea since the microservices are already platform and location independent. At this point an on-premise solution is still more reliable and ready for production use, thus the cloud provided platform has to be considered already in near future. At least it is an advantage for business vision to recognize the cloud possibility. Moreover, deploying the module as microservices from the start makes it easier to migrate the application into the cloud more seamlessly and already in an efficient and flexible architecture. Sources [10] and [2] include especially high quality further reading of the topic.

5. FEATURE AND REQUIREMENT SPECIFICATION AND REALIZATION

Specification of the most efficient set of features and requirements is represented with multiple different models from different points of views and from different levels of abstraction, as described in the earlier chapter. Realizing the feature and requirement specification means constructing the information system, LIMS, according to the well designed specification. The designed system functionality with relationships between the database model, the system interface description and the use cases are documented with models, as visualizing represents complex functionalities and relationships in the most understandable form. The LIMS module was requested to be platform independent, i.e. independent of which other systems are requesting actions and messages, and to which systems the LIMS module is going to write the data. Answering to the last part of the research question as well, the platform independency is shown in this chapter.

5.1 User interviews

A knowledge of the process and functionalities related to the laboratory and the infrastructure of the laboratory is fundamental in terms of the effectiveness of the management system development. [25, p. 189] A general LIMS cannot fulfill all the requirements for managing the data of every individual laboratory. Thus, a thorough familiarization with the sampling and the laboratory processes is required with understanding the creation of the samples and producing as well as interpreting the orders. A lack of understanding the process will cause a risk for LIMS solution to be unsuccessful in addition to the technology being inadequate [25, p. 191]. Further, the knowledge of the process is the base for tailoring the LIMS to fulfill the individual requirements. The process knowledge and the LIMS requirement information were acquired via user interviews utilizing the process experience of the users. The interview results were carefully studied and analyzed.

5.1.1 The preparations and implementation

The user interview was implemented as a theme interview, with the frame and question guideline designed based on the LIMS theory knowledge acquired writing this Master's Thesis. The objective of the interview was established based on part of the research question, researching what features have to be included in the LIMS module. The feature requirement specification was validated with the results of the user interviews as well. In addition, some chosen problems that other LIMS-developers had faced and published in the scientific articles were considered designing the interview questions. The interview questions were divided under four themes. The four themes were the laboratory operations and processes, the methods to perform the operations, information needed to perform the operations and finally LIMS development requirements and stating the missing features of the current LIMS module. Appendix A includes a template for the interview questions classified under the interview themes. One of the fundamental characteristics of the scientific theme interview based on the theory of the topic is that the interviewer presents complementary questions based on the discussion, and based on when interesting information comes up regarding to the desired outcome of the interview. These complementary questions could not be listed in the Appendix A since the complementary questions are solely presented on occasion and per interviewee.

Another essential step was to determine the interviewees based on the definition of the user groups during the early user journey map design. Turned out, that the LIMS module users had to be interviewed on-site locally, since the users, the operators, the laboratory technicians and the metallurgists are very busy and do not even have a Skype or Lync interview possibility. Therefore two travels were carried out, first to Pori to visit a research laboratory and second travelling to Sodankylä to visit a mineral processing mining plant. Travelling on site made possible interviewing the operators, the metallurgists and the laboratory technicians locally. The stakeholders, i.e. product owners, laboratory managers, and software engineers were able to be interviewed via Skype. Appendix B contains the introduction email for the interviewees that was sent prior to the actual interview. The introduction email includes a description of the objective of the interview, i.e. what is wanted to accomplish with the interview. The interviews and the travel visits were scheduled and reserved via same email-chain as well. Finally eleven interviews were implemented in the entirety. The interviewees consisted of one operator, three metallurgists, two laboratory technicians, two laboratory managers and three product managers.

5.1.2 The results and analysis

As stated in the introduction email, the results were treated in confidence and anonymously so that the answers of the individual interviewees could not be identified. There were total of eleven users and stakeholders interviewed. The duration of the interviews were in the one hour to one and a half hour time-frame. The interview material was carefully read and translated into English to have a uniform interview material. Most of the interviews were carried out in Finnish, yet this Thesis is written in English. The important sentences, points, requirements and ideas answering the interview objective were clearly marked and written in a separate document to gather the important details. The most important material was about the current situation of the process to understand it thoroughly, development requirements and ideas, specific development ideas of user interfaces, system interfaces and possible risks, and finally "nice to have-ideas. Following dividing the interview questions into four themes, in addition to the overall picture, the interview results were divided into four result groups as well, in respect of the most important material. The four groups were user interface, system interface, possible risks and nice to have.

The current situation of the processes enabled forming a detailed understanding of the process and needed functionalities and pointing out clear problems and pain points in the functionalities, but also helped to form an overall picture as well as a very detailed picture of the LIMS. Understanding the process was utilized in describing the sampling process and building the base for the user journey map. The required and expected features and functionalities of the LIMS module forming the overall picture are:

- More automation needed (e.g. barcodes, printing)
- More (timestamp) information
- Taking the laboratory into account in the development phase
- Reliable, fast and again reliable functionality

Requirement for increased automation includes generally automatic sample creations, sending automatic emails, sending automatically result files from the analyzer etc. automation that pre-empts and removes possibility for manual errors, since manual errors are the most frequent source of errors according to ten out of eleven interviewees.

Moreover, adding barcode-support into the LIMS increases automation while there could be automatic acknowledgements sent to the system when the barcode is read.

Six out of eleven interviewees require barcodes to increase the automation level. Barcodes could include all the needed information as well, e.g. timestamps for creating, sending and receiving samples and an expiration timestamp for the samples are requested to streamline the daily job by increasing the offered information for those who work with the samples. Five interviewees demand more information, especially the timestamps. Barcodes could be also automatically printed to be attached to the sample containers. Previously the sample id's were handwritten to the containers, causing a clear risk for errors.

According to the experience of three out of eleven interviewees, one of the most frequent risk for comprehensively efficient functionality is to neglect the laboratory in the development phase. It has been surprisingly generally the case that the laboratory has not been taken into account, causing them additional work and problems, as an assumption of the interviewer was that the laboratory has been taken into account every time. While the manufacturing process itself is the more important process, the laboratory should be taken into account as well, listening for their suggestions and ideas to ensure the best possible information system.

Lastly by far the most important expected feature is reliability, that is so important for the users and stakeholders that it had to be mentioned twice in the list. Five out of eleven interviewees mentioned the reliability as one of the most important features for the LIMS. However, the emphasis was really strong for the reliability and the user role of the interviewee affects to the distribution of the results. The functionality has to be reliable in a continuous process and hectic laboratory environment. In addition to reliability the LIMS module should function fast as well and provide the information quickly.

First of the detailed result groups is the most apparent interface for the users, the user interface (UI). The users see the whole system through the UI, and therefore they form their whole experience of the system by using the UI. Findings regarding the UI were as follows:

- Excel-integration (Both export and import)
- Option to Excel: UI to be as flexible as Excel
- Adequate filtering options for the data
- Manual work causes the most errors
- The look to be calm and peaceful, for the user to notice alerts

The UI is basically based on data tables e.g. for workflow scheduling and validating the sample results. Navigating and moving between the data cells can be quite heavy operations at least compared to Excel spreadsheets. Giving a fact that most of the current users are used to Excel, it is a very hard task to try to replace Excel. Three out of eleven interviewees required the Excel integration to be included to maintain the option to use it to working. Interview findings show, that creating new groups of samples are also easy with Excel; once an row has been written, it is fast to copy by drag and dropping ten other samples where one letter, number or date per row is changed. Therefore adding Excel export and import in the UI would be a solution to keep the users using the UI. It is important for them to use the UI, since only that way the essential data goes to the database. It is extremely risky to keep the data in spreadsheets in some users or stakeholders private "my computer"files. Thus the data wouldn't be shared to others and might be lost. Another option would be to make the UI as flexible than Excel in navigating, copying and drag and dropping, which is quite a challenge itself.

One desired feature from five out of eleven users would be better filtering options for the data. The better filtering leads to finding the desired data faster, increasing the efficiency and profitability. Such filtering options include e.g. filtering by sample type, timestamp and shift. The developed filtering is a good example of how small features from information system development point of view makes great results in a long run following how efficient and sensible the daily work for the user is.

According to four out of eleven interviewees, the UI should look calm and peaceful to prevent missing important alerts. The alerts are usually marked with red color for the same reason as well.

There were only a few findings regarding the system interfaces since it requires some understanding of the information system design and architecture. However, some stakeholders were able to list following features:

- Interface between analyzer devices and LIMS is required for data transfer
- More reliable csv-parsing (to avoid empty columns)
- Avoid transferring the same data into many different systems

The need for a proper interface between online- or laboratory analyzer and LIMS was recognized for an efficient data transfer by three out of eleven interviewees. The ideal solution would be automated file transfer; the modern analyzers are able to deliver a .csv, .xml, .qan or .txt based sample result files, that could be automatically sent to the LIMS module, that does the parsing and storing the data in the database. Also the parsing should be developed to be reliable and time- and computing-efficient.

In the learning and development point of view the most valuable information is the recognized system development and functionality risks. According to the interview material, there are following system development risks:

- Operators underestimate the criticality of an information system
- Operators might be "afraid"being replaced by a new system
- The LIMS is too complex (if the use requires training)
- Increasing the daily workload
- User authentication and security settings
- Internet access in the plants
- Not having a user experience inquiry

The stakeholders recognized very important risks according to the interview material. Two out of eleven interviewees recognized that some of the operators may underestimate the criticality of the LIMS causing a risk for an incorrect of the system. Also some of the users may be afraid of being replaced by a new automated system that decreases the workload of the users according to one interviewee. These points are not actually reflecting to the system development, but are important to be recognized.

One significant risk in the information system development is that the new system is too complex. The complexity can be shown if the use of the system requires user training or if it takes too much time to implement the basic functions. Four out of eleven interviewees stated that the simplicity is one the most important keys in the risk management. Moreover, if the new information system increases the daily workload of the users, the system is too complex.

User authentication, security settings, and Internet connectivity can be also risks regarding the system security and communications. The database and the user interface user accesses and security settings are an essential task in the system development point of view, but are a remarkable risk if neglected. Five out of eleven interviewees emphasized the essentiality of the security settings and the authorization. According to the experience of one interviewee, it is important to recognize a risk of Internet connectivity problems in the process plants, especially if the LIMS system is built on a client-server type of architecture, resulting in the server being in a location A and the client being at the other side of the world.

One of the risks regarding the user experience and the system efficiency is neglecting the user experience inquiry after the deployment. Organizing an inquiry awakes a professional and an excellent customer-oriented image of the developer team. In addition the trust between the customer and the developers will increase with an ease of communication. Two out of eleven interviewees stated that the user experience inquiries and the trust between the customer and the developers are very important to have a successful development project according to their experience.

Finally the last result group is formed by the "nice to have-findings suggested by the users and the stakeholders as follows:

- Possibility for ERP integration
- Possibility for smartphone use
- Automatic barcode acknowledgement
- Possibility for side orders in addition to actual orders

SAP is one of the most famous and used ERP-platforms, making an integration to SAP really tempting. Two out of eleven interviewees required the ERP integration. The integration would be a doable feature in the future, but is not in the highest priority list. The same category feature is a smart phone integration that would interest especially the stakeholders due to their mobile work days. Three out of eleven interviewees required the possibility for the smartphone use. The automatic barcode acknowledgements and possibility for side orders are purely system development features that are not in the original scope, but are nevertheless stored for future discussions. Again, six out of eleven interviewees required the barcode feature, making the development need obvious for it.

5.2 Finalized use cases with user journey map

The use cases were combined tentatively from the use cases of the existing system that had to be developed. Predicted development requirements were added according to requests from the customer along the years of use of the system. The use case list was then validated and finalized during and after the user interviews. Based on the tentative use case list and the validation through the user interviews, the user experiences along the use cases could be determined. The collected information was combined into a visualization through the user journey map. Appendix C contains the user journey maps for the process plant and laboratory users and use cases separately.

The first user journey map in the Appendix C describes the user journey from the mining and metal industrial process point of view. The map has two actors: the metallurgist and the plant operator. The user actions are combined mainly from the tentative use case list, completed with details acquired from the user interview results. The user experience with the mindsets are in its entirety combined from the user interviews. The scenario of the user journey map for the mining process consists of four phases: sampling, preparing samples for the laboratory, sending the samples and receiving the results, and finally validating the sample results. The sampling and the validation phases contains actions from both of the actors.

The user actions are numbered and mapped correspondingly to the line representing the user journey handling the samples. For the sampling phase, there is not much added compared to the tentative use case list, see Table 4.1. Details are added for presenting different categories of separately created samples by metallurgist (campaign, special and calibration samples) and categories of the collected samples from the process by plant operator (shift, event and special samples). Experience and emotions of the metallurgist are related to the usability and the flexibility of the user interface; creating samples manually is often the easiest using the traditional spreadsheet, since creating ten to twenty samples is based on copying sample IDs and changing only e.g. date or running number. Thus spreadsheet would be the fastest tool for quick and flexible sample creation, and would not require learning the use for new user interface. Therefore one of the biggest challenges from user interface point of view is to replace the spreadsheet with at least as flexible and easy user interface. The experience and emotions by the operator are that as the shift samples are collected from the process flow automatically to buckets in respect to the sample type, some of the buckets might be sometimes empty or the samples could be thrown away by accident. Therefore the user interface and the backend logic should be flexible with registering only the collected samples, since typically shift samples contain always the same set of the sample types.

Sample preparations are responsibilities of plant operators, and are combined from user interview material in its entirety. Following sample collection and registering the samples, the operator takes the samples for pressurized filtering to remove the water from the samples. The subsequent step is to dry the samples in an oven to make sure all the water is gone. Finally the dried samples are bagged into sample bags by the sample type, i.e. each sample bag containing only the one specified sample type. The bags are labelled with the sample id, containing the sample type code as well. The sample id's were labelled before by handwriting, followed by requirements for logic for printing automatically generated sample id label. The barcode, including sample id, timestamps, status, category etc. additional information about the sample is printed into the label as well. The user experience and emotions are related to the sample labels. Automation generally decreases manual typing errors, and decreases errors in this case as well. However, a risk for human error of bagging the samples in a wrong bag still remains.

The metallurgist is responsible for sending and receiving the samples, with a strong support from automation. The most of the samples are scheduled samples, taken daily or per every shift, or event samples, that are taken e.g. from every dispatching truck. Since the scheduled samples and the event samples are created automatically, the orders of the samples are sent automatically as well to the laboratory. The metallurgist sends the orders of the other types of samples manually, e.g. calibration samples or other unregular special samples, such as campaign samples. The sample test results are received in a csv file sent automatically into a specified folder. Hence the LIMS module service is able to parse the file to write and save the result data to the database. The metallurgist receives alarms in case some of the sample results are missing or have invalid values so he/she can make actions. The alarms are also sent to the laboratory by mail. The metallurgist has also an option to export and import the results from the UI to the spreadsheet. The user experience and the user interviews point out as creating the special samples manually is easier and quicker using spreadsheets, it has been rather easy also to just attach the spreadsheet file in the email to be sent to the laboratory. Although the system has been designed to form the email attachment automatically from the beginning. The reason behind the preference for the manual actions is once again in the UI, since combining the manually created samples for the email has been also way too troublesome. Therefore the emphasis on the system development is found to be on the UI side. Although that is rather logical, since the UI is the only thing the system users see. One unpleasant case of potential risks from the metallurgist experience point of view is that the file names can be sometimes invalid when received from the laboratory. All the result files are stored and archived, and therefore the user accesses should be carefully configured.

Sample validation is the final phase of the process scenario. The actual sample result validation is the responsibility of the metallurgist. The validation is done by viewing and analyzing reports and graphs automatically created from the result data. Thus metallurgists can notice clear differences immediately from the graphs. Part of the validation is implementing correcting actions as well. The metallurgist makes manual corrections of invalid sample results based on his/her own observations or based on an alarm from the LIMS module. The plant operator performs validating process quality as well, by observing only the latest sample results from distributed control system (DCS) screen in the control room.

The graphs made from the report data makes possible performing online analyzer calibration as well. Metallurgist can order online analyzer calibrations when there is divergence between the line of the laboratory results and the line of the online analyzer results.

The second user journey map is made from the laboratory point of view and based on the laboratory user and stakeholder interviews. The laboratory has only a little to do relating to the LIMS module, but one of the risks in LIMS development is to neglect the laboratory and their point of view. From the LIMS module user point of view, the only user is actually a non-human laboratory analyzer device, that creates and forms the result csv-file and delivers into the specified file via FTP protocol. The only user experience case regarding the use cases is rather laboratory device specific. It might be the case that the file names created at the LIMS module automatically can be too long for the laboratory device to handle, and thus the laboratory technician might need to modify the file names. That is the reason also for a need for metallurgist to validate the file names for archiving and storing. The rest of the laboratory user journey map is support for the user interview results and for the whole process description.

5.3 Requirement aggregation

The functional requirement specification for the LIMS module was done by classifying a list to a spreadsheet of LIMS features for three classes: "must have", "nice to have"and "out of scope". The set of the possible LIMS features was composed from product information on several available LIMS vendor websites [19], studying the functionalities of the existing system and based on LIMS overview material of this Master's Thesis. The "must have"requirements were validated along with the user interviews, and "nice to have"functionalities were added as well according to the points of view from the users and the stakeholders. The final LIMS module requirements were therefore formed according to the "must have-column.

Appendix D contains the full LIMS feature requirement specification designed for this Master's Thesis. The features were classified under four groups. Based on the Appendix D the four classes of the final requirements are represented in Tables 5.1–5.4. The first group of features was "Sample, inventory and data management", consisting of e.g. logging in samples, a general sample management, a task and event scheduling, an option for manual data entry, data and equipment sharing and internal file or data linking. Table 5.1 combines the final requirements of the first group.

Table 5.1 *The list of the final requirements regarding data and sample management.*

Task and event scheduling
Sample registration and management
Manual result entry
Data viewing methods
Data and equipment sharing
Customizable fields and/or interface
Query capability
Internal file or data linking
Export to spreadsheet
Raw data management

The second group was "Quality, security and compliance", consisting of audit trail, user and security configurations, data encrypting and normalization, and finally automatic data backup. Table 5.2 combines the final requirements of the second group.

Table 5.2 *The list of the final requirements regarding security, quality and conformance.*

Audit trail
Configurable roles and security
Data normalization
Data encryption
Automatic data backup

The third group "Reporting, barcoding and printing"included report printing and barcode support, exporting to the basic file types, i.e. PDF, Word, HTML and XML, and finally email integration. Table 5.3 combines the final requirements of the third group.

Table 5.3 *The list of the final requirements regarding reporting, printing and barcoding.*

Printing reports
PDF Export
Email integration

Finally, the fourth group was "Base functionality", including e.g. administration management, modularity, alarms and alerts, multilinquality, and network and web client capability. Table 5.4 combines the final requirements of the fourth group.

Table 5.4 *The list of the final requirements regarding basic functionality.*

Modular
Alarms and/or alerts
Instrument interfacing and management
Administrator management
Multilingual
Network-capable
Web client or portal

The "nice to have"features are saved for further designing of the LIMS module as possible new features to be developed. The following sections contains documentation of the realization of the LIMS features and requirements.

5.4 The database model

The database architecture used to store the data of the LIMS module was designed according to the relational database model. The database model was determined according to the key question in the database modeling technique; is there an existing databases to interact with or not? The designed database must be able to be adapted to an existing database. Since there was an existing database containing essential service settings and logging settings in MS SQL it was a straightforward decision to settle upon the relational database model. The LIMS module service is implemented with .NET C# coding language, and therefore integrating the MS SQL database is naturally seamless and straightforward, since both the database and the service platform are Microsoft products.

The laboratory sample data is naturally structured on relations; e.g. mining and metal samples are divided in categories, orders, statuses, results and value limits, with everything related to the each sample. In the field of mining and metals each sample is unique depending on the day and the shift when the sample is taken from the slurry flow. Therefore the database for the LIMS module is designed according to the relational database model; each table has a unique primary key making each of the table rows unique and linking the row to the data in the other tables needed. The data is linked through foreign keys that refers to the primary key in the destination table.

The database containing the LIMS module data was named LIMSDB, that is designed according to the relational database model. The existing database containing the service and the logging settings as independent database tables is called IntegritySettings. The complete LIMSDB with the database table relations visualized as a database diagram is presented in Appendix E. This section also divides the LIMSDB in two parts after the two main cores the database is built on: the sample results and the actual sample data.

The sample results is the most dynamic and the most updating data group of the LIMS module. In the table *local_ LIMS_ SampleResults* occurs the highest input flow as well. Each sample result is addressed to a combination of sample type, method and element. Each sample is of one of the specified sample types. Method refers to the test method and/or location where the test is taken. Element is simply the metal element that is measured from the sample. The test result can be e.g. concentration of the metal element in percentage or in milligrams per kilogram. Each sample type can include one or more methods and each of the sample type and method combination can include one or more elements. Each sample type-method-element combination represents one *tag*. Each *sample* therefore has several *tags* which each contains test results. Also each *tag* has numerous results coming from several samples. Each tag has maximum of one row in the results table, i.e. if there is a re-entered result for one sample tag, the sample result row is updated in the table *local_ LIMS_ SampleResults*. The tables containing the relevant data about the sample results and that are linked to the sample results table are presented in Figure 5.1.

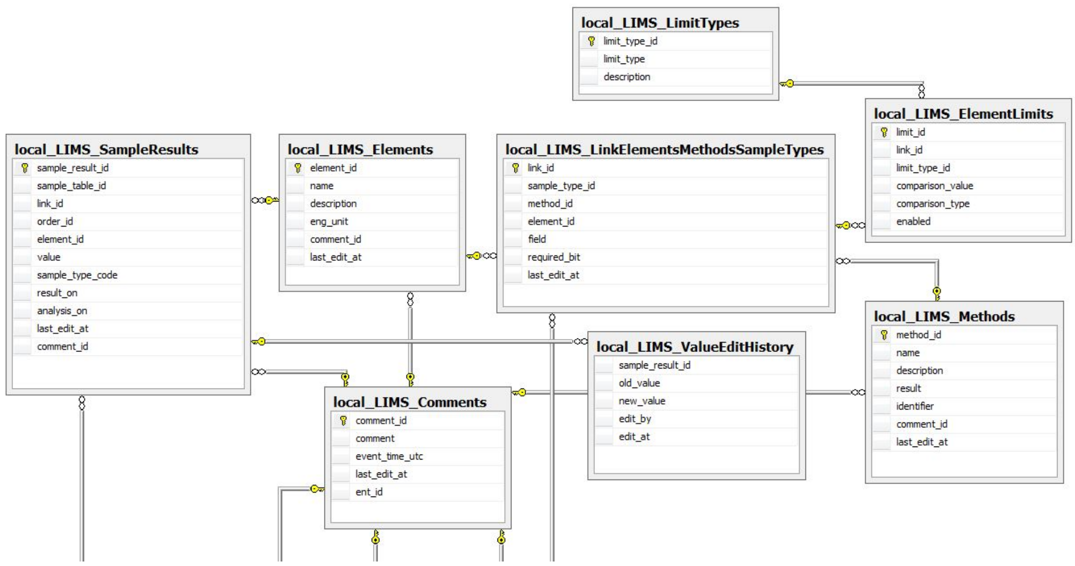


Figure 5.1 *The part of the database architecture of sample results and the most important linked tables.*

Referring to Figure 5.1 the table *local_ LIMS_ LinkElementsMethodsSampleTypes* lists all the available tags where the sample types, methods and elements are linked to each tag as foreign keys. The primary key representing each tag is *link_ id*. There is a designed possibility to configure result limit for each of the tag, i.e. for each of the element. The edit history of the result values is also stored, i.e. if the results for tags is modified after they have been once sent, then the old value and the new value with adequate timestamp and sender information is stored into the history table. Linking the table in respect to elements, methods and sample types was a result of development to form clear tags, since every sample result is anyway a combination of element, method and sample type. Moreover, the old system had two separate link tables, linking sample types and methods, and methods and elements separately. Having two separate linking tables increased the workload significantly, as every time these two tables had to be joined when modifying the configurations or when searching information about the linkings. Therefore, having a single linking table to form tags was significantly more efficient solution in the database architecture point of view.

The sample data is quite static compared to the sample results, although every created sample is added to the table *local_LIMS_Samples*. The samples are created e.g. in the beginning of each shift, which are shift samples, and at the beginning of the first shift of the day as daily samples. As said before, numerous results per sample is expected making the result table much more active. Figure 5.2 represents the tables included mainly to the samples.

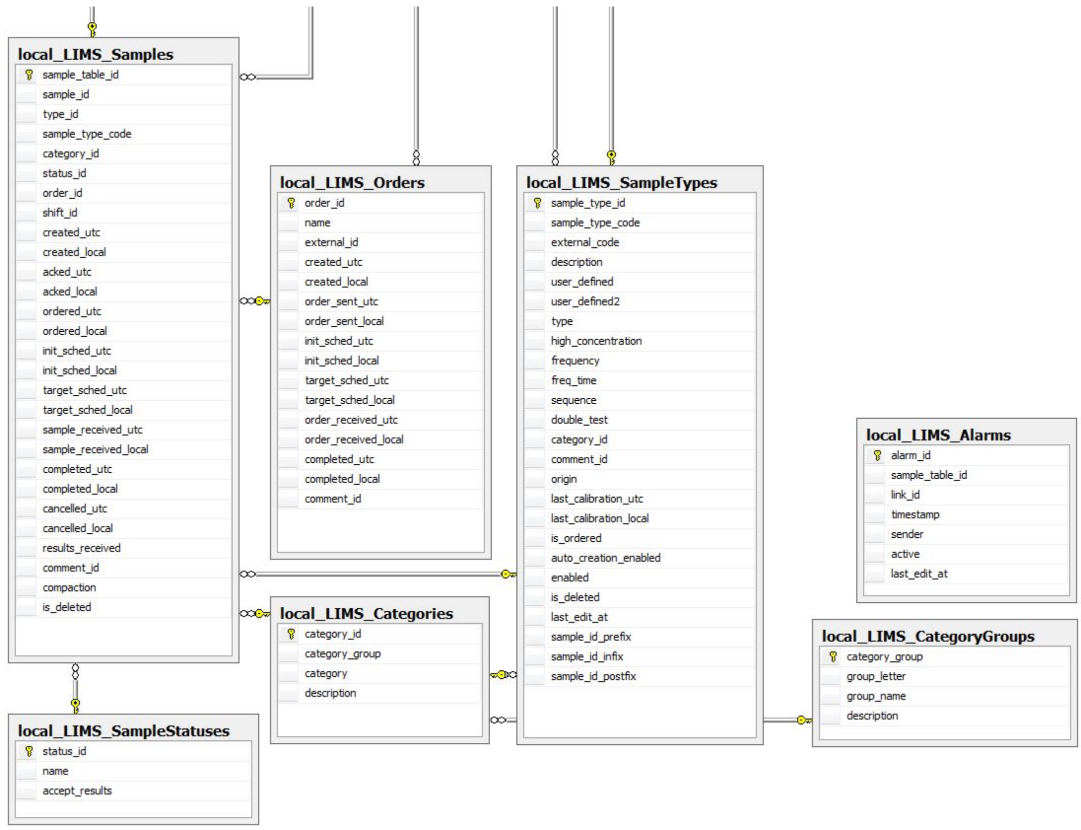


Figure 5.2 *The part of the database architecture of sample data and the most important linked tables.*

Samples can be linked to orders to emphasize certain samples belonging to certain group, e.g. shift samples. The orders are designed for more flexible system functionality referring especially to sending emails. Each sample has a category, e.g. being shift sample, daily sample, special sample, calibration sample etc. Each of the samples have also a status. Status can be e.g. created, ordered, receiving results, completed, incomplete etc. Statuses have a determined workflow, e.g. created sample cannot be receiving results without being ordered and completed sample cannot receive anymore results. Workflow expects e.g. that completed sample has to be marked first incomplete and then it can receive more results if can be definitely needed.

Samples are also linked to sample types, and for each sample type there can be configured several settings that applies for all the samples belonging to that certain sample type. E.g. high concentration sample types can be configured, frequency can be set and whole sample type group can be temporarily or permanently disabled. Finally LIMS alarms have to be listed in a separate table per sample. Alarm can be about samples that are created but are not ordered inside a determined time window or the required results have not been received in determined time. Also if the result values are not defined value limits, an alarm is raised. Alarming system sends an email to defined address(es).

5.5 The system interface architecture

The information system architecture consists of bidirectional communication between the MES software and the LIMS module. The communication between the LIMS and MES is implemented by a file transfer, that consists of sending one type of files for each direction. The LIMS module sends orders to the LIMS as an email attachment. The orders are sent when samples are taken, each order including one or more samples, recognized by a unique sample id. The LIMS sends the results for the orders to the MES via file transfer protocol (FTP). The order results are for each sampling method, as soon as the test for the requested method is completed. [27, p. 10]

The research question states that the module has to be a global platform independent module to transfer data between MES and LIMS. The previous system architecture and the functionality were strictly designed for only the specific mineral processing plant, transferring data between only one specific MES. The LIMS module designed in this Master's Thesis is communicating with n number of microservices, that are configurable in respect of the type of the service and its TCP/IP addresses and ports. The configurations are stored in IntegritySettings database. Each microservice is independent of each other and of the core service. Each of the microservices can thus be connected to whatever client, e.g. one microservice is sending results to OPC Client and another microservice is connected to any MES of choice via a database connection.

The most of the system interfaces are implemented as IrisMQ clients or IrisMQ services. IrisMQ is an internal communication bus architecture used at the system supplier organization this Thesis is made for. The messaging library selected for IrisMQ is ZeroMQ, that makes possible for IrisMQ to provide communications between software applications independently of operating system, programming language and application data. The application level data is handled with defined high level envelope messages containing serialized protocol buffers with JSON and binary data [14]. Figure 5.3 represents the system interface architecture of the LIMS module.

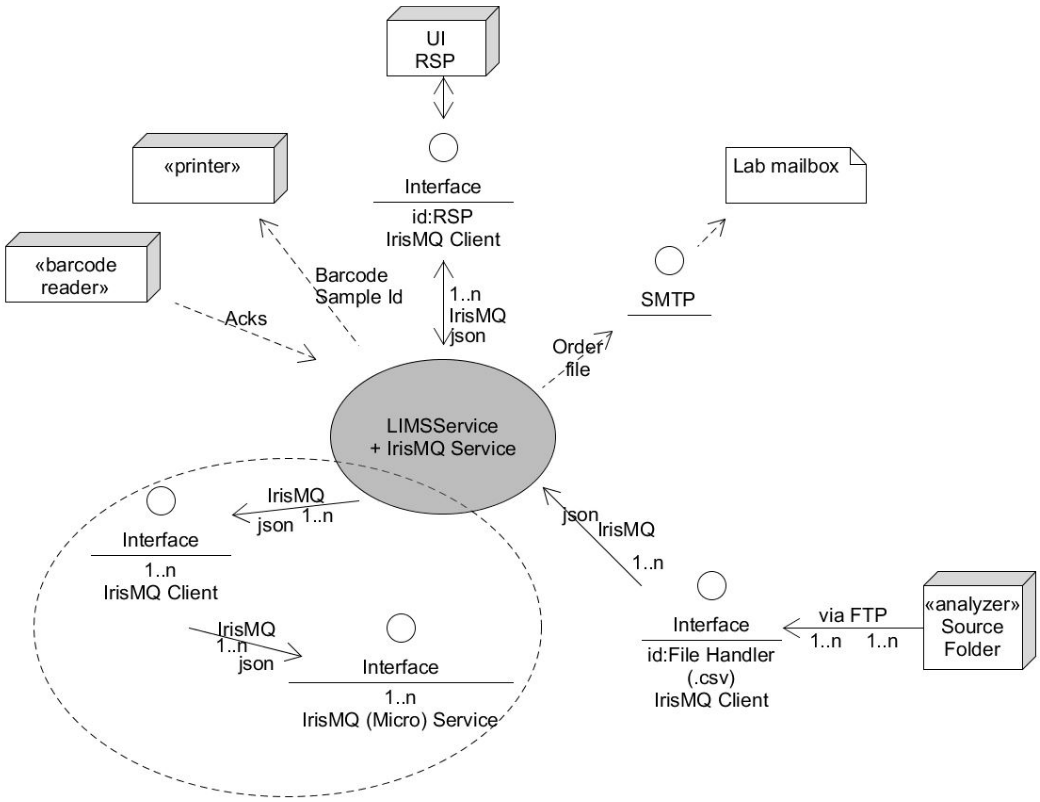


Figure 5.3 *The system interface architecture between LIMS module and microservices and devices.*

In Figure 5.3 the dashed circle represents the one or more microservice(s) that are required to handle their own tasks. The possibility to use one to n microservices is designed basically as follows. LIMSService includes certain standard data objects built with certain attributes that include all the required data of the sample results that are going to be sent forward e.g. to OPC or MES. The sample data is stored in the standard data objects, that is sent to all registered microservices. The destination where each data object is sent, is defined in the data object with a service destination id, that is defined in the database for each tag. Thus the database table containing the configurations can include multiple rows for configurations for the same tag, if that certain tag is required to be sent to several destinations. The data object is serialized as JSON data, that can be sent with IrisMQ connection from LIMSService to each microservice. Each microservice, that is an IrisMQ client, has also defined the same data object, that can be deserialized in the client, being able to be sent in whatever form forward. Since the configurations of the microservices is located in the database and the microservices is implemented with standard data objects, new microservices can be deployed to the LIMS module by only adding a row to the database and adding a new project in the service code, *without having to modify the core service code at all.*

In addition to the microservices, sending and receiving sample result files are essential system interfaces. Sending order files consisting of sample id's, e.g. shift samples, is implemented via simple mail transfer protocol (SMTP) to the specified laboratory email address. Sending emails are done automatically for scheduled samples, e.g. daily samples and shift samples, as well as certain special samples, e.g. truck dispatch samples that are taken from a load of every dispatching truck. Sending email can be done manually from the UI as well, e.g. for the calibration samples or other manual samples. The sample result files are stored in a specified folder where the certain laboratory user have access. The file is then transferred automatically via FTP to the File Handler Service that is an IrisMQ client. The File Handler Service parses the result file, that has to be in comma separated values (csv) format. Csv format is chosen since it is the most simple format of a text based result file. The csv file can be opened in human readable form with the spreadsheet software, e.g. with Excel as well. The parsing stores the csv-file in a form of standard data object, which is sent via IrisMQ as a JSON message to the LIMSService, where again a corresponding data object is defined, and where the results are then stored in the LIMS database.

The user interface (UI) of the LIMS module will be implemented with the remote service portal (RSP) that is a UI platform used at the system supplier organization this Thesis is made for. The interface between LIMS module and the RSP is yet again implemented as an IrisMQ client. The UI was left out from the scope of this Master's Thesis since the RSP platform is under development as well, but will be chosen as the UI for the LIMS module since it has support for IrisMQ already. The interfaces for automatic printing and reading barcodes are designed, but not implemented yet as well. The printing and reading of barcodes were not in the original scope and are therefore implemented in later development phases.

Microservices 1...n chosen and implemented for this Master's Thesis are presented in Figure 5.4. The chosen microservices are: OPC client service, OPC server service and OSIsoft service referring to OSIsoft MES product as an example of MES platform. Although the MES platform could be any platform as stated before.

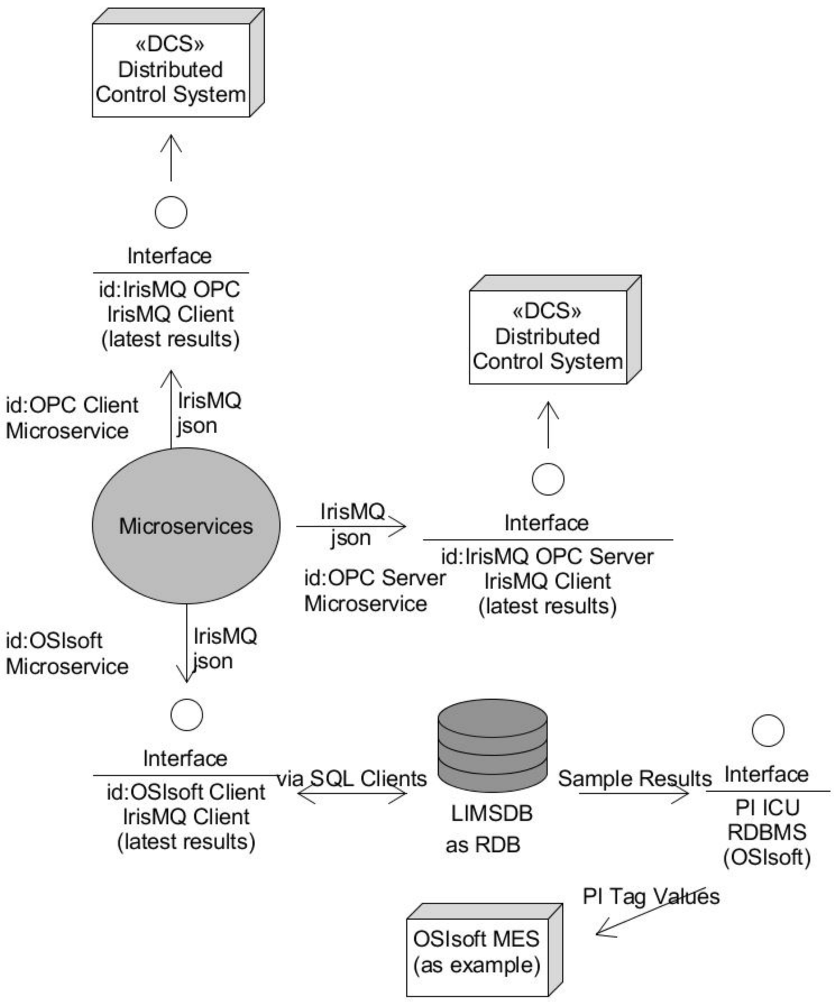


Figure 5.4 *The system interface architecture of IrisMQ microservices used in this project.*

The OPC client and the OPC server services are very similar services, that send only the latest results of the tags for OPC client and OPC server respectively. Handling the sample results is handled as standard data objects. The latest results will be finally delivered to the distributed control system (DCS) that is monitored in the operator control rooms.

The interface between the LIMS module and MES-system(s) is implemented as a

SQL procedure interface. The sample results are sent from the microservice as a standard data object as an input for a specified SQL procedure. The corresponding data object is determined as a user-defined table type at the database end. The specified procedure can be then easily modified as required and send the data to the MES database whose location is irrelevant as long as the specified procedure at LIMSDB has an access to the MES database configured by an administrator. The SQL procedure interface is a very low level integration solution from the system architecture point of view. Sending the sample result data to the SQL procedure instead of sending a JSON data object straight to each MES-system can also increase amount of work and errors when configuring a new system, since the procedure has to be written every time for every MES. However, the SQL procedure was chosen since it was the most straightforward and simple solution to be implemented in the given project schedule. Moreover, the SQL procedure works as long as the SQL procedure is carefully implemented.

Finally, the IrisMQ based system interface architecture was not the only option in the beginning. For example Web-API and REST interfaces were considered in addition. The IrisMQ was chosen since the technique was already available, and supported seamlessly the standard data object option to have a generalized data transfer. The IrisMQ is also preferred naturally for the system supplier organization, since using the own technique is advantageous in the continuous development point of view and to have a uniform use of the own products within the organization.

6. CONCLUSIONS

This Master's Thesis consisted of research of LIMS systems, design of the LIMS module service for data transfer between the LIMS and MOMS system according to the SDLC model presented in Chapter 2, and finally presentation of the requirement and feature specification of the LIMS module. The prototype of the LIMS module is constructed according to the requirement specification and the database and system interface architectures designed in this Master's Thesis.

According to the research question the techniques for gathering the information for the most efficient set of features were described and realized. Two superior techniques were utilized; the first one was to study the existing system and the commercial LIMS systems. Interviewing the users and the stakeholders was the other important technique for gathering the information. Determining the features and the requirements were done by modelling the LIMS module with the use case list, the user journey map, the requirement specification list, the database diagram and finally the system interface models to accomplish a comprehensive model of the system. The most essential features were the software independent system architecture, the barcode feature, the more efficient database architecture and finally the modularity of the LIMS module.

The system interfaces were implemented mostly as IrisMQ clients and services. The IrisMQ turned out to be a really reliable technique for transferring the standard data objects. The database was designed so that modifying and extending the database is simple due to a flexible use of primary keys.

The solution for solving the research question was fulfilled well. An efficient set of features was found with a generic system architecture, making the use of the LIMS module possible at any site. Using the microservices makes the LIMS module independent of the software. Also new microservices can be added independent of the other microservices or the LIMS module service.

The further development will continue with integrating the UI and the automatic printing and barcode features.

BIBLIOGRAPHY

[1] T. Bahill, J. Daniels, Using Objected-Oriented and UML Tools for Hardware Design: A Case Study, Journal of Systems and Industrial Engineering, Vol. 6, No. 1, University of Arizona, Tucson, USA, 2003, pp. 28–48.

[2] A. Balalaie, A. Heydarnoori, Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture, Journal of IEEE Software, Vol. 33, No. 3, May–June 2016, pp. 42–52.

[3] Bika: Open Source laboratory information management systems, website. Available (accessed on 7.4.2017): `http://www.bikalabs.com`.

[4] J. Bogner, A. Zimmermann, Towards Integrating Microservices with Adaptable Enterprise Architecture, Enterprise Distributed Object Computing Workshop (EDOCW), 2016 IEEE 20th International Conference, Vienna, Austria, September 5–9, 2016, pp. 158–163.

[5] A. Bridgwater, What's The Difference Between A Software Product And A Platform?, Forbes, Forbes Inc., Mar. 17, 2015. Available (accessed on 24.8.2017): `https://www.forbes.com/sites/adrianbridgwater/2015/03/17/whats-the-difference-between-a-software-product-and-a-platform/#759917df56a6`.

[6] A. Calabria, G. Spinozzi, F. Benedicenti, E. Tenderini, E. Montini, adLIMS: a customized open source software that allows bridging clinical and basic molecular research studies, Proceedings of the 11th Annual Meeting of the Bioinformatics Italian Society, Rome, Italy, February 26–28, 2014, pp. 1–11.

[7] J. Celko, SQL Programming Style, The Morgan Kaufmann Series in Data Management Systems, Texas, USA, 2005, 272 p.

[8] B. Dathan, S. Ramnath, Object-Oriented Analysis and Design, Springer, Universities Press, India, 2011, 485 p.

[9] C. Esposito, A. Castiglione, K. K. R. Choo, Challenges in Delivering Software in the Cloud as Microservices, Journal of IEEE Cloud Computing, Vol. 3, No. 5, Sept.–Oct. 2016, pp. 10–14.

[10] M. Fazio, A. Celesti, R. Ranjan, C. Liu, L. Chen, M. Villari, Open Issues in Scheduling Microservices in the Cloud, Journal of IEEE Cloud Computing, Vol. 3, No. 5, Sept.–Oct. 2016, pp. 81–88.

[11] M. Fish, Building a Business Case for LIMS Implementations, Scientific Computing, Vol. 10, 2005. Available (accessed on 21.2.2017): http://www.scientificcomputing.com/article/2005/10/building-business-case-lims-implementations.

[12] G. Greeff, MES, MOM, EMI and MIS, Journal of Technology and business for development, EE Publishers, website. Available (accessed on 15.3.2017): http://www.ee.co.za/article/bytes-02-mes-mom-eim-and-mis-do-you-need-them-all.html.

[13] B. Göde, S. Holzmüller-Laue, K. Rimane, M.-Y. Chow, N. Stoll, Laboratory Information Management Systems - An Approach as an Integration Platform within Flexible Laboratory Automation for Application in Life Sciences, Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering, Scottsdale, Arizona, USA, September 22–25, 2007, pp. 841–845.

[14] J. Jääskeläinen, Internal bus (IrisMQ), Internal document, Outotec Oyj. Not available.

[15] Kevitsa - Sampling process documentation, Boliden Kevitsa, Sodankylä. Not available.

[16] LabKey Open Source laboratory information management systems, website. Available (accessed on 7.4.2017): https://labkey.com.

[17] J. P. Lewis, Fundamentals of Project Management, 3rd ed., Amacom, American Management Association, New York, USA, 2007, 164 p.

[18] LiMSwiki: The laboratory, health, and science informatics encyclopedia, website. Available (accessed on 26.4.2017): https://www.limswiki.org/index.php/Laboratory_information_management_system.

[19] LiMSwiki: The laboratory, health, and science informatics encyclopedia - LIMS features, website. Available (accessed on 26.4.2017): https://www.limswiki.org/index.php/LIMS_feature.

[20] The MESA Global Education Program slides, Certificate of Awareness program, MESA International, Helsinki, 2017.

[21] L. D. Michaud, Metallurgical Plant Sampling Techniques, 911 Metallurgist, website. Available (accessed on 30.6.2017): https://www.911metallurgist.com/blog/metallurgical-sampling

[22] C. Paszko, C. Pugsley, Considerations in selecting a laboratory information management system (LIMS), Journal of American Laboratory, Vol. 32, Iss. 18, 2000, pp. 38–43.

[23] V. Pimentel, B. G. Nickerson, Communicating and Displaying Real-Time Data with WebSocket, Journal of IEEE Internet Computing, Vol. 16, Iss. 4, July–Aug. 2012, pp. 45–53.

[24] V. Pimentel, B. G. Nickerson, Web Display of Real-Time Wind Sensor Data, tech. report TR11-214, Faculty of Computer Science, University of New Brunswick, Dec. 2011, 32 p.

[25] P. J. Prasad, Trends in laboratory information management system, Journal of Chemometrics and Intelligent Laboratory Systems, Vol. 118, 2012, pp. 187–192.

[26] C. F. Quo, B. Wu, M. D. Wang, Development of a Laboratory Information System for Cancer Collaboration Projects, Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, September 1–4, 2005, pp. 2859–2862.

[27] A. Rossi, Interface Specification LIMS, DQ-110116-03 CC-Integrity Project, FQM Kevitsa Mining Oy, 2015, 30 p. Not available.

[28] H. Sanchez-Villeda, S. Schroeder, M. Polacco, M. McMullen, S. Havermann, G. Davis, I. Vroh-Bi, K. Cone, N. Sharopova, Y. Yim, L. Schultz, N. Duru, T. Musket, K. Houchins, Z. Fang, J. Gardiner, E. Coe, Development of an integrated laboratory information management system for the maize mapping project, International Society for Computational Biology (iSCB), Vol. 19, Iss. 16, Oxford University Press, USA, 2003, pp. 2022–2030. Available (accessed on 1.3.2017): `https://doi.org/10.1093/bioinformatics/btg274`.

[29] J. W. Satzinger, R. B. Jackson, S. D. Burd, Object-Oriented Analysis and Design with the Unified Process, Cengage Learning, New Delhi, India, 2009, 626 p.

[30] J. Scholtalbers, J. Rößler, P. Sorn, J. de Graaf, V. Boisquérin, J. Castle, U. Sahin, Galaxy LIMS for next-generation sequencing, Journal of Bioinformatics, Vol. 29, No. 9, 2013, pp. 1233–1234.

[31] S. Schuller, Gigaom: Are microservices just SOA redux?, Newstex Trade & Industry Blogs, Chatham, Dec. 21, 2015. Available (accessed on 11.10.2017): `https://search-proquest-com.libproxy.tut.fi/docview/1752233290?accountid=27303`.

[32] T. Somma, User Journey Mapping - internal document, Outotec Oyj, 5 p. Not available.

[33] J. E. H. Stafford, LIMS: An automating or informating technology?, Journal of Laboratory Automation and Information Management, Vol. 33, 1998, pp. 163–168.

[34] C. Stage, Why use LIMS and related software?, Pharmaceutical Technology Europe, Vol. 18, Iss. 5, 2006, Cleveland, USA, pp. 19–21.

[35] L. Sun, Y. Li, R. A. Memon, An Open IoT Framework Based on Microservices Architecture, Journal of China Communications, Vol. 14, No. 2, 2017, pp. 154–162.

[36] T. Tiainen, Haastattelu tietojenkäsittelytieteen tutkimuksessa, Informaatiotieteiden yksikön raportteja, Iss. 25, Tampere 2014, 42 p. Available: `http://www.uta.fi/sis/reports/index/R25_2014.pdf`

[37] J. J. Tomlinson, W. Elliot-Smith, T. Radosta, Laboratory Information Management System Chain of Custody: Reliability and Security, Journal of Automated Methods and Management in Chemistry, Vol. 2006, Article ID 74907, 2006, pp. 1–4.

APPENDIX A. THE USER INTERVIEW QUESTION TEMPLATE

The objective of the interview

The objectives of this interview are to determine the current status of the workflows and activities, and to find out if there is something to develop with the current workflows and with the daily job.

One objective is to determine what features are possibly desired, what features would be nice to have to enhance the production profitability and to increase the user satisfaction. Identifying the missing features is an essential objective of this interview.

Additional goal would be to possibly identify different requirements of the different product lines (departments).

In the report the interviewees cannot be individualized or cannot be traced back to you.

Themes and questions

1) What are the laboratory (business) operations and processes?

a. What do you do?

b. What tasks/activities are done in the laboratory?

c. What do you do related to laboratory?

d. Why do you do them?

d1. How the operations are linked to the laboratory processes?

e. What kind of background you have about LIMS systems?

2) How should those operations be performed?

a. How do you do your job (related to LIMS)?

a1. What tools/Software do you use (also in addition to LIMS)?

b. How do you enter manual samples? How do you edit history values?

c. What steps do you follow (related to each activity / task)?

d. What should be added to enhance the operations profitability?

e. What kind of issues do you have with the operations / tasks?

(Present current list of use cases to the interviewee)
f. In addition possibly more accurate question about some new use case?

3) What information is needed/required to perform those operations?

a. What information do you use? What information are of the customer?s interest?

b. What forms or reports do you use?

c. What forms/reports/features are of the customers interest?

d. How the search conditions to get the information are implemented and how them should be developed?

d1. What kind of ad hoc reports/analyses are possibly needed?

e. What kind of reports could/should be added?

f. How the interface could be developed to make the information even more feasible?

4) How the LIMS should be developed? What features are missing in the current system?

a. How the operations could be developed further to ease your/their job/increase "satisfaction"?

b. What kind of risks (e.g. some user resistance) do you reckon of a new information system?

c. What features are missing in the current system?

APPENDIX B. THE USER INTERVIEW INTRODUCTION EMAIL

Dear recipient,

I contact you considering the LIMS module user and stakeholder interview for my Master's Thesis. The interview method is based on a scientific theme interview model (Tiainen, 2014).

The background and objectives of the interview for the Master's Thesis

The objectives of this interview are to determine the current status of the laboratory data workflows and activities, and to find out what develop suggestions considering the daily job will arise. One objective is to determine what features and functionalities are desired and required, and which features would be nice to have to enhance the production profitability and to increase the user satisfaction. Identifying the currently missing features is an essential objective of this interview. Additional goal would be to possibly identify different requirements of the different product lines.

The content of the interview and what is expected

The aim of the interviews is to elicit user requirements for the new integrated information system which are based on actual lived experiences of the real users. The participating plants will have a possibility to reflect their current tools and operating practices and the development needs of the technical tools will be discussed and explored.

The language of the interviews is Finnish if not otherwise decided. The interviewees do not need to do any preparations for the interviews, in addition to familiarization of the question themes. The duration of the interviews are estimated to be 1h–2h. The results of the interviews are treated in confidence and they are also anonymised so that no answers of individual interviewees can be identified.

Themes are:
1) What are the laboratory (business) operations and processes?
2) How should those operations be performed?
3) What information is needed/required to perform those operations?
4) How the LIMS should be developed? What features are missing in the current system?

As a reply from you, I would kindly ask suggestion of a 1-2 hour time for the interview via Skype or in person, that is suitable for you. The interviews are scheduled to be done in during next two week time (3.5.-17.5.), but as quickly as possible.

Best regards,

Miikka Haurinen

Hyvä vastaanottaja,

Otan yhteyttä koskien LIMS modulin käyttäjä- ja sidosryhmähaastattelua diplomityötäni varten. Haastattelu perustuu tieteellisen tietojenkäsittelytutkimuksen haastattelumetodimalliin (Tiainen, 2014).

Diplomityön haastattelun tausta ja tavoitteet

Haastattelun päämääränä on selvittää laboratorion työtehtävät ja -menetelmät nykyiseen LIMS järjestelmään liittyen, sekä selvittää mitä kehitysehdotuksia ja ideoita nykytilanteeseen nähden ilmenesi. Yksi tavoite on selvittää, että mitä ominaisuuksia ja toimintoja haluttaisiin ja vaadittaisiin kehitettävään järjestelmään, ja mitä toimintoja olisi mahdollisesti hyvä saada tuotannon tuottavuuden ja työmukavuuden lisäämiseksi. Oleellisin tavoite on löytää nykyjärjestelmästä puuttuvat ja/tai puutteelliset toiminnot. Lisäksi olisi mahdollisesti hyvä tunnistaa toisistaan eroavia tuotanto/tehdaskohtaisia vaatimuksia ja ominaisuuksia.

Haastattelun sisältö ja mitä odotetaan haastattelua varten

Haastattelussa halutaan hyödyntää käyttäjien kokemuksia LIMS modulista tuotantoympäristössä. Haastateltavilla on mahdollisuus vaikuttaa nykyjärjestelmän kehitykseen.

Haastattelukielenä on suomi, ellei erikseen ole toista sovittu. Haastateltavien ei tarvitse tehdä valmisteluja haastattelua varten, kysymysteemoihin tutustumisen lisäksi. Haastattelun kesto on 1–2 tuntia. Haastattelun tulokset käsitellään luottamuksella, eikä haastattelun tuloksia voi yksilöidä yksittäiseen haastateltavaan.

Teemat ovat:
1) Mitkä ovat laboratorion operaatiot ja prosessit?
2) Kuinka nuo operaatiot tulisi suorittaa?
3) Minkälaista informaatiota vaaditaan työtehtävien suorittamiseen?

4) Kuinka LIMS- modulia tulisi kehittää? Mitä ominaisuuksia puuttuu nykyjärjestelmästä?

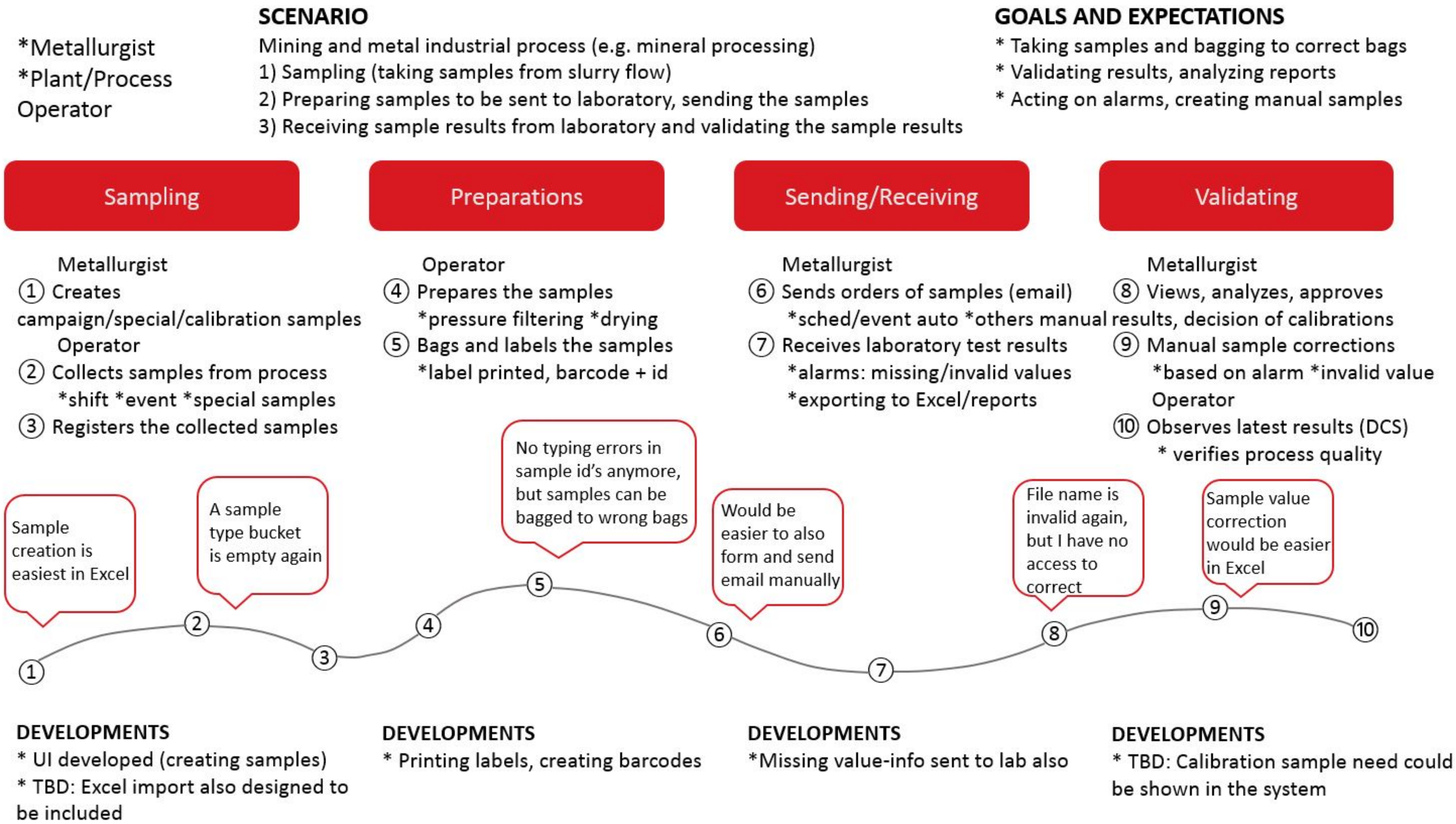
Vastauksena teiltä, pyytäisin teiltä ystävällisesti ehdotusta Teille sopivasta 1-2 tunnin ajan varauksesta kalenteristanne haastattelua varten Skypen välityksellä, tai henkilökohtaisesti. Haastattelut on suunniteltu käytävän läpi seuraavan kahden viikon aikana (3.5.-17.5.) mutta niin pian kuin mahdollista.

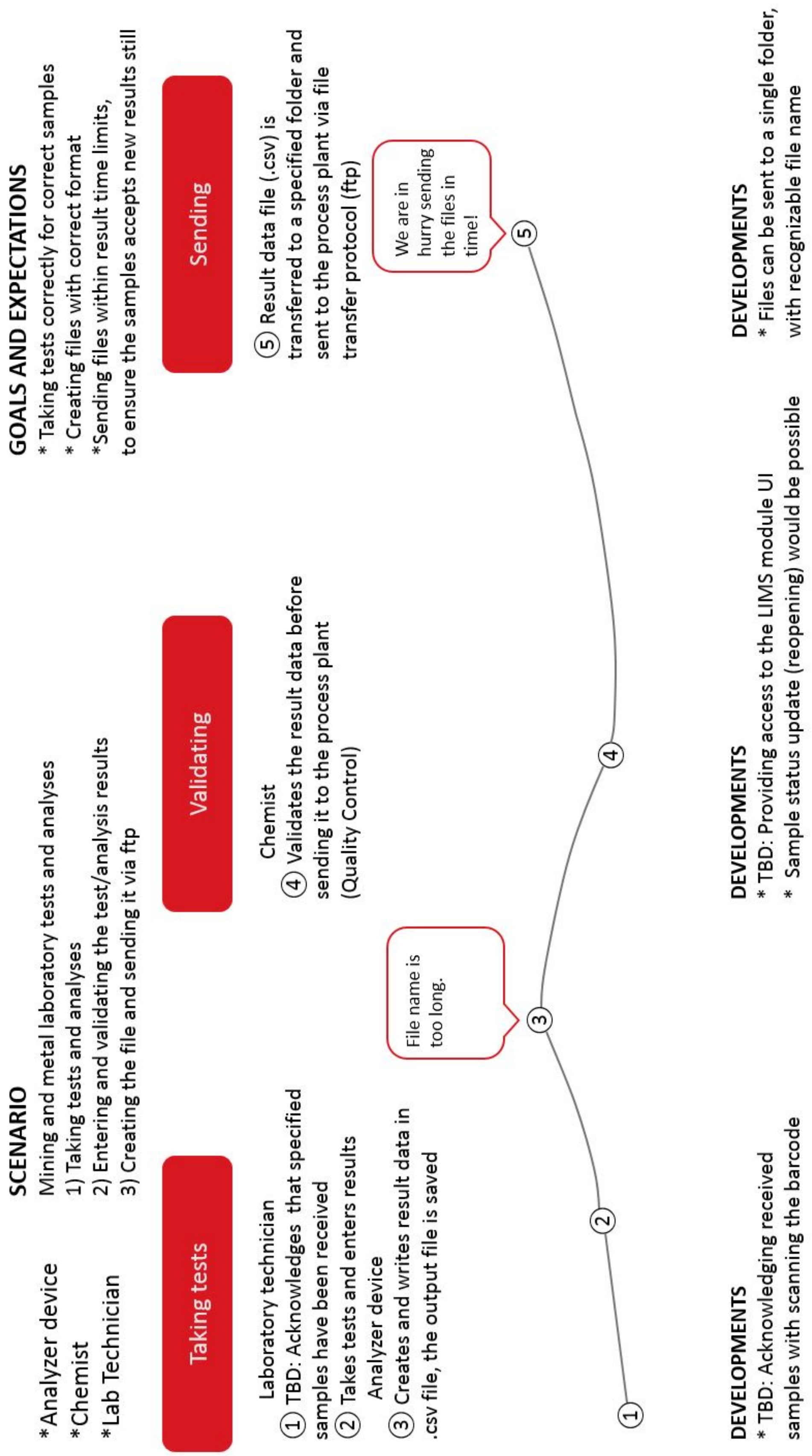
Ystävällisin terveisin,

Miikka Haurinen

APPENDIX C. USER JOURNEY MAP

This appendix contains two different user journey maps in the following pages, so that the maps can be presented in the largest form possible. The maps are located sideways due to that same reason. The first presented user journey map is made from the process plant point of view. Correspondingly the second map is made from the laboratory point of view. Presenting both processes produces the best description of the journey and experience of all the users operating the LIMS module, yet keeping the sampling process and the laboratory test processes separated to emphasize the user experiences only in these different processes each separately.


*Metallurgist
*Plant/Process
Operator
SCENARIO
Mining and metal industrial process (e.g. mineral processing)
1) Sampling (taking samples from slurry flow)
2) Preparing samples to be sent to laboratory, sending the samples
3) Receiving sample results from laboratory and validating the sample results
GOALS AND EXPECTATIONS
* Taking samples and bagging to correct bags
* Validating results, analyzing reports
* Acting on alarms, creating manual samples
Sampling
Preparations
Sending/Receiving
Validating
Metallurgist
1 Creates campaign/special/calibration samples
Operator
2 Collects samples from process
*shift *event *special samples
3 Registers the collected samples
Operator
4 Prepares the samples
*pressure filtering *drying
5 Bags and labels the samples
*label printed, barcode + id
Metallurgist
6 Sends orders of samples (email)
*sched/event auto *others manual
7 Receives laboratory test results
*alarms: missing/invalid values
*exporting to Excel/reports
Metallurgist
8 Views, analyzes, approves results, decision of calibrations
9 Manual sample corrections
*based on alarm *invalid value
Operator
10 Observes latest results (DCS)
* verifies process quality
Sample creation is easiest in Excel
A sample type bucket is empty again
No typing errors in sample id's anymore, but samples can be bagged to wrong bags
Would be easier to also form and send email manually
File name is invalid again, but I have no access to correct
Sample value correction would be easier in Excel
1
2
3
4
5
6
7
8
9
10
DEVELOPMENTS
* UI developed (creating samples)
* TBD: Excel import also designed to be included
DEVELOPMENTS
* Printing labels, creating barcodes
DEVELOPMENTS
*Missing value-info sent to lab also
DEVELOPMENTS
* TBD: Calibration sample need could be shown in the system


*Analyzer device
*Chemist
*Lab Technician
SCENARIO
Mining and metal laboratory tests and analyses
1) Taking tests and analyses
2) Entering and validating the test/analysis results
3) Creating the file and sending it via ftp
GOALS AND EXPECTATIONS
* Taking tests correctly for correct samples
* Creating files with correct format
*Sending files within result time limits,
to ensure the samples accepts new results still
Taking tests
Validating
Sending
Laboratory technician
1 TBD: Acknowledges that specified samples have been received
2 Takes tests and enters results
Analyzer device
3 Creates and writes result data in .csv file, the output file is saved
Chemist
4 Validates the result data before sending it to the process plant (Quality Control)
5 Result data file (.csv) is transferred to a specified folder and sent to the process plant via file transfer protocol (ftp)
File name is too long.
We are in hurry sending the files in time!
1
2
3
4
5
DEVELOPMENTS
* TBD: Acknowledging received samples with scanning the barcode
DEVELOPMENTS
* TBD: Providing access to the LIMS module UI
* Sample status update (reopening) would be possible
DEVELOPMENTS
* Files can be sent to a single folder, with recognizable file name

APPENDIX D. LIST OF FEATURES AND REQUIREMENTS OF THE LIMS'S

***Table 1** LIMS features and requirements: the first theme*

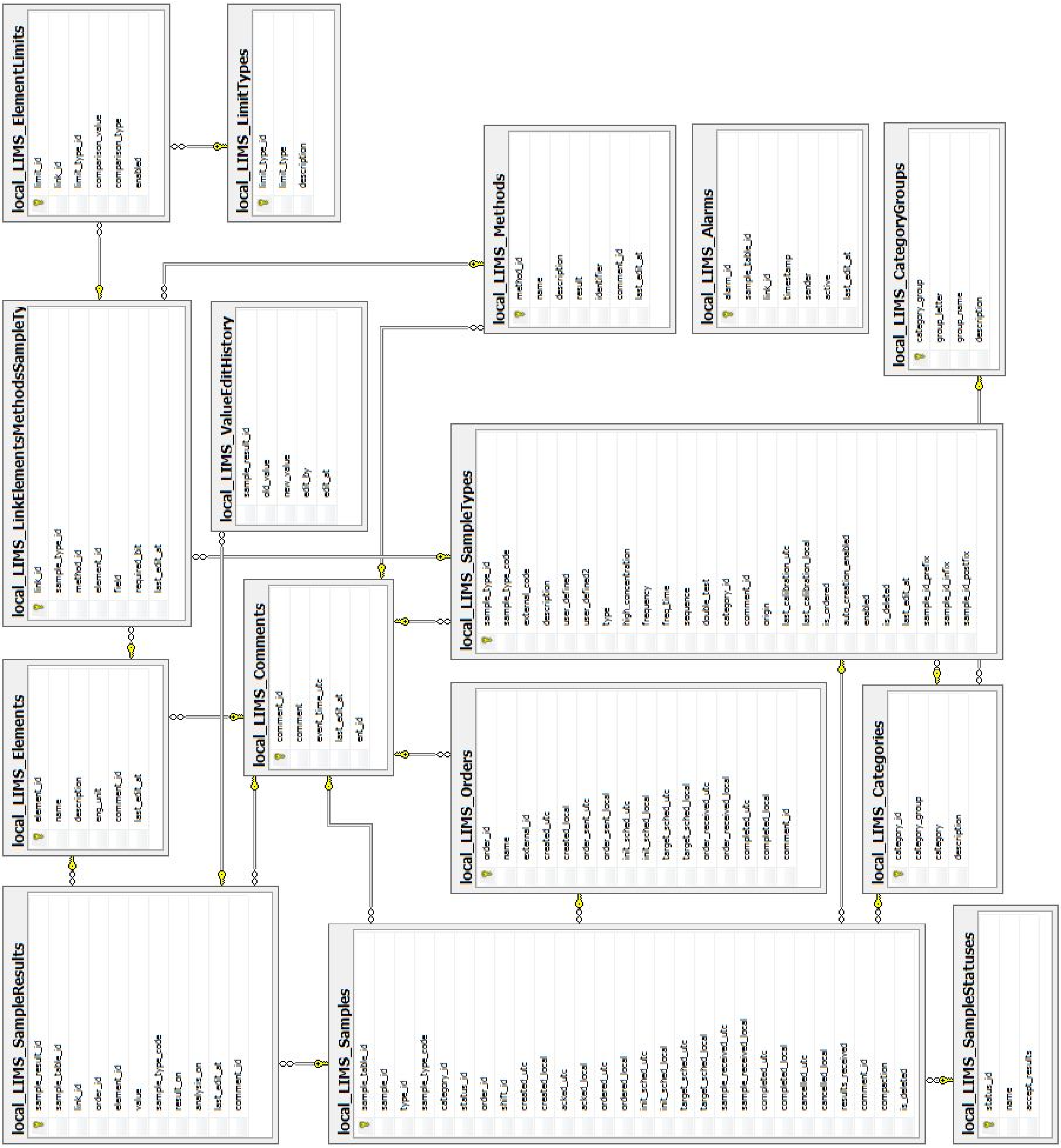
Feature	Must have	Nice to have	Out of scope
1 Data and sample management			
1.1 Task and event scheduling	x		
1.2 Sample registration and management	x		
1.3 Sample and result batching			x
1.4 Sample tracking		x	
1.5 Manual result entry	x		
1.6 Data viewing methods	x		
1.7 Trend and data analysis		x	
1.8 Data and equipment sharing	x		
1.9 Customizable fields and/or interface	x		
1.10 Query capability	x		
1.11 Import data		x	
1.12 Internal file or data linking	x		
1.13 External file or data linking			x
1.14 ELN support or integration			x
1.15 Export to spreadsheet	x		
1.16 Raw data management	x		
1.17 Data warehouse			x
1.18 Deadline control			x
1.19 Production control			x
1.20 Project and/or task management			x
1.21 Inventory management			x
1.22 Document creation and management			x
1.23 Case management			x
1.24 Workflow management			x
1.25 Specification management			x
1.26 Customer and supplier management			x
1.27 Billing management			x

***Table** 2 LIMS features and requirements: the second, the third and the fourth theme*

Feature	Must have	Nice to have	Out of scope
2 Security, quality and conformance			
2.1 QA/QC functions			x
2.2 Regulatory conformance			x
2.3 Performance evaluation			x
2.4 Audit trail	x		
2.5 Chain of custody			x
2.6 Configurable roles and security	x		
2.7 Data normalization	x		
2.8 Data validation		x	
2.9 Data encryption	x		
2.10 Version control			x
2.11 Automatic data backup	x		
2.12 Environmental monitoring			x
3 Reporting, printing and barcoding			
3.1 Printing reports	x		
3.2 Customized reporting		x	
3.3 Label support		x	
3.4 Barcode support		x	
3.5 PDF Export	x		
3.6 Word Export		x	
3.7 HTML/XML Export		x	
3.8 Fax integration			x
3.9 Email integration	x		
4 Basic functionality			
4.1 Modular	x		
4.2 Alarms and/or alerts	x		
4.3 Instrument interfacing and management	x		
4.4 Administrator management	x		
4.5 Mobile device integration		x	
4.6 Work-related time tracking			x
4.7 Voice recognition system			x
4.8 External monitoring		x	
4.9 Messaging			x
4.10 Multilingual	x		
4.11 Network-capable	x		
4.12 Web client or portal	x		
4.13 Online or integrated help		x	
4.14 Software as a service delivery model		x	
4.15 Usage-based cost		x	

APPENDIX E. COMPLETE DATABASE DIAGRAM

This appendix contains the complete database diagram, so that the diagram can be presented in the largest form possible. The diagram is located sideways due to that same reason.


local_LIMS_ElementLimits
limit_id
link_id
limit_type_id
comparison_value
comparison_type
enabled
local_LIMS_LimitTypes
limit_type_id
limit_type
description
local_LIMS_Methods
method_id
name
description
result
identifier
comment_id
last_edit_at
local_LIMS_Alarms
alarm_id
sample_table_id
link_id
timestamp
sender
active
last_edit_at
local_LIMS_CategoryGroups
category_group
group_letter
group_name
description
local_LIMS_LinkElementsMethodsSampleTy
link_id
sample_type_id
method_id
element_id
field
required_bit
last_edit_at
local_LIMS_ValueEditHistory
sample_result_id
old_value
new_value
edit_by
edit_at
local_LIMS_SampleTypes
sample_type_id
sample_type_code
external_code
description
user_defined
user_defined2
type
high_concentration
frequency
freq_time
sequence
double_test
category_id
comment_id
origin
last_calibration_utc
last_calibration_local
is_ordered
auto_creation_enabled
enabled
is_deleted
last_edit_at
sample_id_prefix
sample_id_infix
sample_id_postfix
local_LIMS_Elements
element_id
name
description
eng_unit
comment_id
last_edit_at
local_LIMS_Comments
comment_id
comment
event_time_utc
last_edit_at
ent_id
local_LIMS_Orders
order_id
name
external_id
created_utc
created_local
order_sent_utc
order_sent_local
init_sched_utc
init_sched_local
target_sched_utc
target_sched_local
order_received_utc
order_received_local
completed_utc
completed_local
comment_id
local_LIMS_Categories
category_id
category_group
category
description
local_LIMS_SampleResults
sample_result_id
sample_table_id
link_id
order_id
element_id
value
sample_type_code
result_on
analysis_on
last_edit_at
comment_id
local_LIMS_Samples
sample_table_id
sample_id
type_id
sample_type_code
category_id
status_id
order_id
shift_id
created_utc
created_local
acked_utc
acked_local
ordered_utc
ordered_local
init_sched_utc
init_sched_local
target_sched_utc
target_sched_local
sample_received_utc
sample_received_local
completed_utc
completed_local
cancelled_utc
cancelled_local
results_received
comment_id
compaction
is_deleted
local_LIMS_SampleStatuses
status_id
name
accept_results